PCPs for Arthur-Merlin Games and Communication Protocols

by

Andrew Donald Drucker

B.A., Swarthmore College (2006)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science in Computer Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2010

Author . . . .
Department of Electrical Engineering and Computer Science
May 21, 2010

Certified by . . . .
Scott Aaronson
Associate Professor
Thesis Supervisor

Accepted by . . . .
Terry P. Orlando
Chairman, Department Committee on Graduate Students

PCPs for Arthur-Merlin Games and Communication Protocols

by

Andrew Donald Drucker

Submitted to the Department of Electrical Engineering and Computer Science
on May 21, 2010, in partial fulfillment of the
requirements for the degree of
Master of Science in Computer Science and Engineering

Abstract

Probabilistically Checkable Proofs (PCPs) are an important class of proof systems that have played a key role in computational complexity theory. In this thesis we study the power of PCPs in two new settings: Arthur-Merlin games and communication protocols.

In the first part of the thesis, we give a 'PCP characterization' of AM analogous to the PCP Theorem for NP. Similar characterizations have been given for higher levels of the Polynomial Hierarchy, and for PSPACE; however, we suggest that the result for AM might be of particular significance for attempts to derandomize this class. To test this notion, we pose some 'Randomized Optimization Hypotheses' related to our stochastic CSPs that (in light of our result) would imply collapse results for AM. Unfortunately, the hypotheses appear over-strong, and we present evidence against them. In the process we show that, if some language in NP is hard-on-average against circuits of size $2^{\Omega(n)}$, then there exist hard-on-average optimization problems of a particularly elegant form.

In the second part of the thesis, we study PCPs in the setting of communication protocols. Using techniques inspired by Dinur's proof of the PCP Theorem, we show that functions $f(x, y)$ with nondeterministic circuits of size m have 'distributed PCP protocols' of proof length $O(\text{poly}(m))$ in which each verifier looks at a constant number of proof positions. We show a complementary negative result: a distributed PCP protocol using a proof of length ℓ, in which Alice and Bob look at k bits of the proof while exchanging t bits of communication, can be converted into a PCP-free randomized protocol with communication bounded by $\ell^{O(k)} \cdot t$.

In both parts of the thesis, our proofs make use of a powerful form of PCPs known as Probabilistically Checkable Proofs of Proximity, and demonstrate their versatility. In our work on Arthur-Merlin games, we also use known results on randomness-efficient soundness- and hardness-amplification. In particular, we make essential use of the Impagliazzo-Wigderson generator; our analysis relies on a recent Chernoff-type theorem for expander walks.

Thesis Supervisor: Scott Aaronson
Title: Associate Professor

Acknowledgments

I am deeply grateful to Scott Aaronson, my excellent advisor; to Anna Torres, my partner and dear friend; and to my parents, Erica Buhrmann and Ronald Drucker, for all their love and support over the years.

I also thank Madhu Sudan and Avi Wigderson for helpful comments and suggestions at different stages of this work.

Contents

Chapter 1

Introduction

1.1 Probabilistically Checkable Proofs

This thesis studies the power of *probabilistically checkable proofs (PCPs)* in computational complexity theory. PCPs are protocols in which a *prover* provides a computationally-bounded *verifier* with a 'proof' string (which need not be a proof in any standard logical language). With this proof string, the prover tries to convince the verifier that an input string x, known to prover and verifier alike, lies in some set (or 'language') $L \subseteq \{0,1\}^*$. The distinctive feature of PCP protocols is that, rather than reading the entire proof string, the verifier instead probabilistically chooses a small subset of the proof symbols to inspect, and accepts or rejects the presented proof based on the symbols he sees.

The PCP protocol is expected to have two important properties: first, for every $x \in L$, there should be a proof for x which causes the verifier to accept with probability 1. (In some cases we may only require an acceptance probability c close to 1.) Second, for every $x \notin L$, every purported proof that $x \in L$ will be accepted with at most some probability $s < c$. This reflects the interests of a skeptical verifier who does not wish to be deceived.

Important parameters of a PCP protocol include:

- the values c and s as above, called the *completeness* and *soundness* parameters;

- q, the maximum number of queries made by the verifier on any proof;

- the alphabet size $|\Sigma|$, where the proof consists of a sequence of some $\ell \geq 0$ symbols from Σ; and
- the proof length ℓ.

Often some or all of the parameters above are considered to depend on the length n of the mathematical statement being proved.

PCPs turn out to be amazingly powerful, as shown by the landmark *PCP Theorem* of Arora et al. [ALM$^+$98]. First, recall that the complexity class NP consists of the languages L for which there exists a polynomial-time verifier algorithm $V(x, w)$ receiving an input x and a 'proof' string w, such that:

(i) if $x \in L$, there exists a y of length $O(\text{poly}(|x|)$, and such that $V(x, w)$ accepts;

(ii) if $x \notin L$, then for all y, $V(x, w)$ rejects.

The PCP Theorem states that for every $L \in \mathsf{NP}$, there exists a PCP protocol for L, with a polynomial-time-bounded verifier, in which:

- $c = 1$ and $s \leq 1 - \Omega(1)$;
- $q = O(1)$;
- $\Sigma = \{0, 1\}$;
- $\ell = O(\text{poly}(|x|))$.

That is, one can transform any NP verifier into a an polynomial-time verifier for the same language in which the new verifier looks at only a *constant* number of input bits, independent of the input length n, yet still can distinguish with noticeable advantage between valid proofs of true claims and purported proofs of false claims!

The PCP Theorem has had a host of consequences within complexity theory, and in particular is of key importance in showing the NP-hardness of many approximation problems (see the survey [Tre04]). Due to its great interest, researchers have explored numerous variations and extensions of this result. One type of PCP research is aimed at improving the efficiency of PCP verifiers and in exploring the achievable tradeoffs between the various parameters in PCPs for NP languages (see, for example, [Tre04, BSGH$^+$06, Din07]).

A second type of PCP research explores the power of probabilistically checkable proofs and 'PCP-like' protocols, in settings different from that of the original PCP Theorem [KL94, CFLS95, CFLS97, HRTS07]. Our work is of this second type. We explore the power of PCPs in the following two scenarios:

1. **Arthur-Merlin games:** *Arthur-Merlin games* (or *Arthur-Merlin protocols*) are another kind of probabilistic proof system, first defined by Babai [Bab85] and predating the concept of PCPs. An Arthur-Merlin protocol is defined by a polynomial-time verifier algorithm $A(x, r, w)$. As with NP protocols, the intuition is that a prover ('Merlin') wants to convince a skeptical polynomial-time verifier ('Arthur') that input x is in some language L. To do so, Merlin provides a proof string w, in response to a random 'challenge' string r generated by Arthur; here, r and w are of length polynomial in $|x|$. We say that A is an Arthur-Merlin protocol for L if:

   (i) If $x \in L$, then for all r there exists w for which $A(x, r, w)$ accepts;

   (ii) if $x \notin L$, then with probability $2/3$ over a uniformly chosen r, $A(x, r, w)$ rejects for all w.

   The complexity class AM [Bab85] consists of all languages which have an Arthur-Merlin protocol as above. AM protocols generalize NP protocols, so $\mathsf{NP} \subseteq \mathsf{AM}$; it is widely believed (see, e.g., [SU07]) that the two classes are actually equal.

   In Chapter 2 of this thesis, we ask which AM languages can be defined by Arthur-Merlin protocols in which the verifier is allowed only to inspect a constant number of symbols of the proof string. The most interesting case, which we focus on, is when the verifier's test is allowed to depend only in a very limited way on the random string r. Even in this restricted case, we will show that PCP protocols are quite strong—strong enough to express all of AM. We will then explore the possible significance of our result to the goal of proving new upper bounds on the class AM itself.

2. **Communication protocols:** Two verifiers, Alice and Bob, hold strings x and y respectively, and hope to determine $f(x, y)$ for some Boolean function f. They wish to do so with minimal communication. A prover tries to convince them that $f(x, y) = 1$, with a proof string that Alice and Bob each inspect in a small number of positions. In Chapter 3, we will give very general conditions on f under which such a PCP protocol is possible. As a complementary negative result, we will also show *limits* on the efficiency of PCPs in this setting.

We describe these two settings and our results in greater detail in the next two sections. In giving background for our results on AM, we will also review earlier work exploring variants of the PCP Theorem for complexity classes other than NP.

1.2 PCPs for Arthur-Merlin Games

1.2.1 CSPs, PCPs, and complexity classes

A *Constraint Satisfaction Problem (CSP)* is a collection $\psi(x)$ of Boolean-valued constraints over variables on a bounded-size alphabet Σ. A CSP in which each constraint depends on at most k variables is called a *k-CSP*. A natural computational task is to determine the maximum fraction of constraints that can be satisfied by any assignment. The Cook-Levin Theorem [Coo71] states that this problem is NP-complete. The PCP Theorem of [ALM+98] can be equivalently stated as the assertion that, for a sufficiently small constant $\varepsilon > 0$, it is NP-hard even to output an estimate that is within ε of this maximum fraction. Specifically, it is NP-hard to distinguish between CSPs in which one can satisfy all constraints, and CSPs in which every assignment violates some ε fraction of the constraints. In both the Cook-Levin Theorem and the PCP Theorem, NP-hardness holds under the settings $k = 3, \Sigma = \{0, 1\}$.

Given the importance of the PCP Theorem for complexity theory, researchers have looked for analogues of the result for complexity classes other than NP. The 'CSP viewpoint' on the PCP Theorem is a fruitful perspective to look for such analogues. The PCP Theorem can be seen as stating that it is NP-hard to determine within ε the value of a 1-player 'solitaire' game defined by a 3-CSP; it is equally possible to study games played on a k-CSP in which 2 players alternate in setting values to designated blocks of variables, with one player trying to maximize the fraction of satisfied clauses and the other trying to minimize this fraction.

These games were explored in several works. Ko and Lin [KL94] showed that approximating the value of such a game is hard for the j-th level of the Polynomial Hierarchy, if the game lasts for j moves. In more recent work of Haviv, Regev, and Ta-Shma [HRTS07] this result was shown to hold even if each variable is allowed to appear in at most a constant number of constraints. If the game is allowed to last polynomially many rounds, the approximation problem becomes PSPACE-hard as shown by Condon et al. [CFLS95]. The same authors showed the approximation problem for poly(n) rounds is also PSPACE-hard if a maximizing player plays against a *random* player [CFLS97] (where the game's value now is the *expected* number of satisfied clauses under optimum play by the maximizer). Moreover, all of the hardness-of-approximation results mentioned so far are in fact completeness results for the corresponding promise classes, so they can be viewed as giving 'PCP characterizations' of NP, PSPACE, and the Polynomial Hierarchy.

One class that did not receive a PCP characterization based on CSP games was the 'Arthur-Merlin' class AM. In fact, there are few known natural complete problems for AM (technically, for its promise version, prAM; it is unknown if AM, a semantic class, has *any* complete problems. See Sec. 2.1.3 for the definition of prAM.). To the author's knowledge there is only one *approximation* problem previously known to be prAM-complete: Mossel and Umans [MU02] give a prAM-completeness result for approximating the VC dimension of set systems. This striking result does not fall within the framework of CSP games given above.

We note that there is a sense in which the PCP Theorem for NP immediately implies a 'PCP Theorem for AM'. Namely, once Arthur chooses a random challenge string r, Merlin's task to produce a proof which will satisfy Merlin can be reduced to the task of satisfying a certain 2-CSP (depending on r), and satisfiability of this 2-CSP can be proved using a PCP. However, this result seems uninteresting, and unlikely to provide new insights into AM. A much more natural and interesting question is whether a prAM-complete problem can be given within the CSP-games framework described above, in which the CSP in the PCP reduction depends only on the input, and *not* on the random challenge variables r. This is the approach we follow.

1.2.2 Our results

Our main result on Arthur-Merlin protocols is a new PCP characterization of prAM. Following [CFLS97], we consider 'stochastic' 2-CSPs $\psi(r, z)$, where r is a collection of Boolean variables and z a collection of variables over an alphabet Σ. Let $\mathrm{Val}_\psi(r, z)$ be the fraction of constraints of ψ satisfied by (r, z). In Section 2.3 we prove:

Theorem 1. *There is a finite alphabet Σ and a constant $\varepsilon > 0$, such that it is prAM-complete to distinguish between the following two sets of 2-CSPs:*

$$\Pi_{YES} = \{\psi : \textit{ for all } r \textit{ there exists } z \textit{ such that } \mathrm{Val}_\psi(r, z) = 1\};$$

$$\Pi_{NO} = \{\psi : \textit{ with probability } 1 - \exp(-\Omega(|r|)) \textit{ over random } r,\ \ \mathrm{Max}_z[\mathrm{Val}_\psi(r, z)] < 1 - \varepsilon\}.$$

In particular, this implies that $\varepsilon/2$-approximating the value of the 2-round game associated with $\psi(r, z)$ (where the first player plays randomly) is AM-hard.

AM is a class for which we feel such a PCP characterization might be especially important. There is compelling evidence that $\mathsf{AM} = \mathsf{NP}$, or at least that significant derandomization of AM is possible (see [SU07] for an overview of this line of research). One approach to to try and derandomize AM is to directly attack the 'easiest' AM-hard problems, and a problem like the one provided by Theorem 1 seems like a plausible candidate.

How might such an attack proceed? We make a concrete suggestion in the form of two 'Randomized Oracle Hypotheses'. In what follows $\psi(r, z)$ is a 2-CSP over ℓ Boolean variables

(r) and m variables (z) over a finite alphabet Σ.

Hypothesis A (Randomized Optimization Hypothesis for $\mathsf{P/poly}$). *Fix any $\delta > 0$. For every 2-CSP $\psi(r, z)$, there exists a circuit $C_\psi(r) : \{0,1\}^\ell \to \Sigma^m$ of size $O(\mathrm{poly}(|\psi|))$, such that with probability at least $1/\mathrm{poly}(\ell)$ over a random $r \in \{0,1\}^\ell$, we have*

$$\mathrm{Val}_\psi(r, C_\psi(r)) \geq \mathrm{Max}_z[\mathrm{Val}_\psi(r, z)] - \delta.$$

In a nutshell, this hypothesis suggests that it is easy to approximately-optimize over z for a random choice of r, if we allow our optimizer to depend nonuniformly on the 2-CSP ψ. (Such nonuniformity is necessary provided $\mathsf{P} \neq \mathsf{NP}$, in light of the PCP Theorem for NP.) This hypothesis, if true, would yield a collapse result for AM. In Section 2.4 we prove the following claim by a straightforward application of Theorem 1:

Claim 2. *Hypothesis A implies $\mathsf{AM} = \mathsf{MA}$.*

A strengthened hypothesis could have the stronger implication that $\mathsf{AM} = \mathsf{NP}$. Consider the following:

Hypothesis B (Randomized Optimization Hypothesis for NC^0). *For any $\delta > 0$, there is an integer $t = t(\delta) > 0$ such that the following holds. For every 2-CSP $\psi(r, z)$, there exists a function $F_\psi(r) : \{0,1\}^\ell \to \Sigma^m$, where each output coordinate of F_ψ depends on at most t bits of r, and such that with probability at least $1 - \delta$ over r,*

$$\mathrm{Val}_\psi(r, F_\psi(r)) \geq \mathrm{Max}_z[\mathrm{Val}_\psi(r, z)] - \delta.$$

In Section 2.4 we prove:

Claim 3. *Hypothesis B implies $\mathsf{AM} = \mathsf{NP}$.*

Given the potential consequences of these hypotheses, what chance do they have of being true? Unfortunately, it seems that each is unlikely. In Section 2.5 we prove two results, each to the effect that, if NP decision problems are hard on average for exponential-size circuits, then both hypotheses fail in a strong way. We state these results next. A language L is called *$p(n)$-hard for size* $s(n)$ if for every circuit C of size $s(n)$, $\Pr_{x\in\{0,1\}^n}[C(x)=L(x)]\le p(n)$.

Theorem 4. *Suppose there exists a $\gamma_1>0$ and an $L\in\mathsf{NP}\cap\mathsf{coNP}$ that is $(1-1/\operatorname{poly}(n))$-hard for size $2^{\gamma_1 n}$. Then there exists $c,\gamma_2,\theta>0$ and a polynomial-time constructible family $\{\psi_n(r,w)\}_{n>0}$ of 2-CSPs (with $|r|=cn, |w|=d(n)=O(\operatorname{poly}(n))$), such that:*

(4.i) for all r, there exists a w such that $\mathrm{Val}_{\psi_n}(r,w)=1$;

(4.ii) for all n, if $C:\{0,1\}^{cn}\to\{0,1\}^{d(n)}$ is a circuit of size at most $2^{\gamma_2 n}$, then

$$\Pr_r[\mathrm{Val}_{\psi_n}(r,C(r))>1-\theta]\le\exp\{-\Omega(n)\}.$$

Theorem 5. *There is an $\varepsilon_0>0$ such that the following holds. Suppose there exists a $\gamma_1>0$ and an $L\in\mathsf{NP}$ that is $(1/2+\varepsilon_0)$-hard for size $2^{\gamma_1 n}$. Then there exists a $c>0$, a polynomial-time constructible family $\{\psi_n(r,w)\}_{n>0}$ of 2-CSPs (with $|r|=cn, |w|=d(n)=O(\operatorname{poly}(n))$), and $\gamma_2,\theta>0$, such that:*

(5.i) With probability $\ge 1-\exp\{-\Omega(n)\}$ over r, there exists w with $\mathrm{Val}_{\psi_n}(r,w)=1$;

(5.ii) If $C:\{0,1\}^{cn}\to\{0,1\}^{d(n)}$ is any circuit of size at most $2^{\gamma_2 n}$, then

$$\Pr_r[\mathrm{Val}_{\psi_n}(r,C(r))>1-\theta]\le\exp\{-\Omega(n)\}.$$

We note that the hypothesis in Theorem 5 is implied by the hypothesis that there exists a *balanced* function $L\in\mathsf{NP}$ that is $(1-1/\operatorname{poly}(n))$-hard for some size $s(n)=2^{\Omega(n)}$; this follows from a result of O'Donnell [O'D02] (see also Healy et al. [HVV06], where the needed form of O'Donnell's result is made explicit and proved in a stronger form).

Theorems 4 and 5 both say that if NP (or $\mathsf{NP}\cap\mathsf{coNP}$) has sufficiently hard problems, then this hardness can be 'concentrated' into a kind of 'inapproximability-on-average' result for an optimization problem associated with a single, uniform family of (stochastic) CSPs. Note that the two results offer a tradeoff: Theorem 5 gives a slightly weaker conclusion from a presumably likelier hardness assumption. The assumptions in the above results are strong but, we feel, plausible. But at the very least, these results suggest that the approach we suggested to showing new upper-bounds on the power of AM must be modified to have a reasonable chance of succeeding.

We feel, however, that the Random Optimization Hypotheses are worthy of study in their own right, even if they turn out to be false; we pose some concrete questions about them at the end of Chapter 2. We also feel that Theorems 4 and 5 are interesting for the study of average-case hardness in NP, and that the CSP families they produce might have further applications in complexity theory.

1.2.3 Our methods

All of our three main results in Chapter 2—Theorems 1, 4, and 5—are essentially hardness results for computational tasks associated with 2-CSPs. In each case the reduction with which we prove our result uses a powerful type of PCP known as *Probabilistically Checkable Proofs of Proximity (PCPPs).* PCPPs were introduced independently by Ben-Sasson et al. [BSGH$^+$06] and by Dinur and Reingold [DR06], and the PCPPs we use were developed by Dinur [Din07] (in [DR06, Din07] PCPPs are referred to as *Assignment Testers*). We will define PCPPs in Section 2.1.2, and in Section 2.2 we derive a variant form of PCPPs (Lemma 9) that is more useful for our purposes.

Lemma 9 gives a general reduction (similar to past uses of PCPPs, e.g., in [Din07]) in which we start with a two-argument circuit $Q(r, w)$ and efficiently produce a 2-CSP $\psi(r, z)$. The basic hope for our reduction is as follows: first, for any r, if the restricted circuit $Q(r, \cdot)$ is satisfiable (i.e., there exists w such that $Q(r, w) = 1$), then the restricted 2-CSP $\psi(r, \cdot)$ should be satisfiable as well. Second, if $Q(r, \cdot)$ is unsatisfiable, then any assignment to $\psi(r, \cdot)$ should violate an $\Omega(1)$-fraction of the constraints in ψ. Unfortunately, this second requirement is too strong and cannot be met. What we *can* guarantee is that if r is 'far' in Hamming distance from any r' for which $Q(r', \cdot)$ is satisfiable, then for any z, (r, z) violates an $\Omega(1)$-fraction of constraints of ψ.

How does this reduction help us prove the prAM-hardness result in Theorem 1? Any instance x of a promise problem $\Pi = (\Pi_{YES}, \Pi_{NO})$ defines a predicate $Q(r, w)$ computed by a poly-size circuit. If $x \in \Pi_{YES}$ then for all r, $Q(r, \cdot)$ is satisfiable; while if $x \in \Pi_{NO}$ then for a $2/3$ fraction of r, $Q(r, \cdot)$ is unsatisfiable. In order to apply our reduction, we need a stronger condition in the second case: a random choice of r should be far from any r' for which $Q(r', \cdot)$ is satisfiable. In other words, we need an extremely low error probability in our underlying Arthur-Merlin protocol. This cannot be achieved by straightforward parallel repetition, but it is provided by a theorem of Bellare et al. [BGG93] which gives a randomness-efficient soundness-amplification for AM. Interestingly, Mossel and Umans [MU02] also used such amplification for their AM-hardness-of-approximation result on VC dimension, but for rather different reasons (unrelated to PCPs).

Our prAM-hardness proof is, we feel, more straightforward than the existing proofs of the analogous results for PSPACE and the Polynomial Hierarchy, modulo our use of sophisticated tools (PCPPs and efficient soundness-amplification) which we apply in a 'black-box' fashion. Of course we do not rule out that our result could be also proved more directly by adapting ideas from the earlier papers (which use some of the same property-testing ideas that have gone into constructions of PCPPs). But we feel that PCPPs in particular, which have already found applications in PCP construction, coding theory, and property testing (see [BSHLM08] for an overview), are a versatile tool which could be of more widespread use in complexity theory. In a very recent example of their utility, Williams [Wil] applied PCPPs to the study of circuit lower bounds.

Next we discuss our methods in Theorems 4 and 5. Our transformation in Lemma 9 from the circuit Q to the 2-CSP ψ has a further useful property: we can reduce the problem of *finding* satisfying assignments to $Q(r, \cdot)$, to the problem of finding nearly-optimal assignments to $\psi(r, \cdot)$. Roughly speaking, we show the following. Suppose there is an algorithm $P(r)$ producing an assignment z, such that with some probability p over r, $(r, P(r))$ satisfies 'almost all' of the constraints of ψ; then there is a second algorithm $\tilde{P}(r)$ such that $Q(r, \tilde{P}(r)) = 1$ with probability $p' \geq 2^{-\varepsilon|r|}p$ (where $\varepsilon > 0$ can be chosen arbitrarily small). This property of the reduction is somewhat more novel, although the techniques we use (involving error-correcting codes) still follow previous works.

To apply our reduction, we use the hardness assumptions in Theorems 4 and 5 to produce predicates $Q(r, w)$ such that $Q(r, \cdot)$ is satisfiable with high probability, while any 'small' witness-producing circuit C fails to solve the search problem associated with Q: that is, $Q(r, C(r)) = 0$ with high probability. Because of the exponential loss factor $2^{-\varepsilon|r|}$ in our reduction, we need the search problem associated with Q to be *extremely* hard: we need every 'small' circuit C to succeed with probability at most $\exp\{-\Omega(|r|)\}$ over r in achieving $Q(r, C(r)) = 1$.

To produce such extremely hard search problems from a more 'mild' hardness assumption, we use existing hardness-amplification techniques. In particular, we use the well-known Impagliazzo-Wigderson generator [IW97]. This generator, on input parameter n, takes a seed r of length $O(n)$, and produces n 'pseudorandom' outputs $g_1, \ldots, g_n$ each of length n. The generator has the property that if language L is mildly hard for sufficiently small (but exponential-size) circuits, then any sufficiently smaller circuit has success probability $\leq \exp\{-\Omega(n)\}$ in correctly guessing the n-bit string $(L(g_1), \ldots, L(g_n))$. Then, if our hard language L is in $\mathsf{NP} \cap \mathsf{coNP}$ (as in Theorem 4), defining our predicate Q is straightforward: we let $Q(r, w) = 1$ iff w contains 'proofs' for the n values $(L(g_1), \ldots, L(g_n))$.

If our hard language is merely in NP (as in Theorem 5), we need to work harder. In this case, we let $Q(r, w) = 1$ iff w contains proofs that $L(g_i) = 1$, for a 'sufficient number' of the strings g_i. The idea is that if a small circuit $C(r)$ could with some noticeable probability guess such proofs for 'almost all' the indices i for which $L(g_i) = 1$, then C could be modified to correctly guess $(L(g_1), \ldots, L(g_n))$ with noticeable probability, contrary to the properties of the generator. Making this idea work involves showing that the set size $|\{i \in [n] : L(g_i) = 1\}|$ is highly concentrated around its expectation. For this we rely on a recently proved concentration result called the 'strong Chernoff bound for expander walks' [WX05, WX08, Hea08]. This result is perfectly suited to analyze the Impagliazzo-Wigderson generator (which is partly defined in terms of walks on expander graphs).

The precise form of our assumptions in Theorems 4 and 5 are dictated by the hardness-amplification tools currently available. In particular, sufficiently strong hardness-amplification is only available if we make a hardness assumption against nonuniform, exponential-sized circuits. We believe versions of Theorems 4 and 5 should be possible for a *uniform* hardness assumption; recently Impagliazzo et al. [IJKW08] made partial progress towards the hardness-amplification tools needed.

1.3 PCPs for Communication Problems

1.3.1 Our results

Suppose Alice and Bob hold bitstrings $x, y \in \{0,1\}^n$ and want to compute some Boolean function $f(x, y)$, without exchanging too many bits of communication.[1] For some functions f, it is known that an all-knowing but untrustworthy Prover can convince the two parties that $f(x, y) = 1$ (if this is the case), by a proof consisting of far fewer bits than would be necessary in the worst case were Alice and Bob to converse alone. A well-known example is the Set Nondisjointness problem [KN96], where $f(x, y) = 1$ iff there exists an i such that $x_i = y_i = 1$. Computing f requires $\Omega(n)$ bits of communication even for randomized protocols [KN96], but a proof of length $\lceil \log n \rceil$ will convince both players that $f(x, y) = 1$. Namely, the prover provides any index i on which both strings are 1, and Alice and Bob individually check the proof's consistency which their respective strings.

Following the PCP Theorem, it is natural to ask about the power of 'distributed PCP protocols' in which Alice and Bob each look at only a very small number of randomly chosen positions of a proof provided to both of them. We provide a general positive result about the existence of distributed PCP protocols for communication problems:

Theorem 16 (informal, special case). *There exists a constant-sized alphabet Σ_0 such that the following holds. Suppose $f(x, y) : \{0,1\}^{2n} \rightarrow \{0,1\}$ has a nondeterministic circuit $C(x, y, z)$ of size $m = m(n) \geq n$. Then f has a distributed PCP protocol with a proof of*

[1] Our switch from languages to functions in this section simply follows the convention in communication complexity.

length poly(m) *over* Σ_0*, in which Alice and Bob each make 2 nonadaptive queries to the proof and independently accept or reject (without any communication). If* $f(x,y)=1$*, there exists a proof causing Alice and Bob to accept with probability 1. If* $f(x,y)=0$*, then for any proof provided, with probability* $\Omega(1)$ *at least one of Alice and Bob rejects.*

Moreover, the strategies of Alice and Bob can be implemented in polynomial time, given a description of C*.*

Thus Set Nondisjointness *does* have a constant-query distributed PCP protocol, if we settle for proofs of size poly(n) (rather than $\lceil \log n \rceil$ as in the original protocol).

The version of Theorem 16 we prove is more general in two ways: first, we allow more than two verifiers, generalizing to a situation where $r(n) \leq n$ verifiers hold n input bits in some arbitrary partition. Second, if $f(x)$ has a $\mathbf{\Sigma}^j$ circuit of size m (i.e. a circuit with j rounds of nondeterministic alternation; see Chapter 3, Sec. 3.1.1), then $f(x)$ has a PCP protocol as above where the proof becomes a j-round 'Probabilistically Checkable Debate' [CFLS95, CFLS97].

Theorem 16 gives some indication of the power of PCPs for communication, but leaves unanswered questions about the achievable PCP parameters. For example, could Set Nondisjointness have a protocol in which the Prover provides a proof of length $(\log n)^{O(1)}$, and Alice and Bob read $O(1)$ bits of the proof while exchanging $(\log n)^{O(1)}$ bits of communication?

We show a general result, Theorem 17 below, implying that the answer to the above question is No.[2]

Theorem 17 (informal, special case). *Suppose* $f(x,y)$ *has a distributed PCP protocol with (two-sided) bounded error, with proof length* ℓ*, in which Alice and Bob each look at* k *proof bits.*

Then f *has a (PCP-free) randomized protocol with bounded-error, with communication bounded by* $\ell^{O(k)}t$*. Moreover, if the original PCP protocol has perfect completeness, so does the PCP-free protocol.*

[2] I am grateful to Avi Wigderson for suggestions that improved the parameters in Theorem 17, and for the suggestion to generalize Theorem 16 to more than 2 verifiers.

Combining the above Theorem with the known $\Omega(n)$ lower bound on the randomized communication complexity of Set Nondisjointness, we can conclude that, for distributed PCP protocols in which Alice and Bob make $k = O(1)$ queries to the PCP and exchange $t = (\log n)^{O(1)}$ bits of communication, proofs of length $\ell = n^{\Omega(1)}$ are necessary to solve the Nondisjointness problem.

The version of the above result we prove provides the same conclusion even if the distributed PCP is generated by a probabilistically checkable debate between competing provers, which is inspected in k positions by Alice and Bob. Thus in the communication setting, PCPs are sharply limited in their power if we restrict to short proof lengths.

1.3.2 Our methods

Our first result on PCPs for communication protocols, Theorem 16, is inspired by Dinur's use of Probabilistically Checkable Proofs of Proximity for alphabet reduction in her recent alternative proof of the PCP Theorem [Din07]. The special case of our result, in which we convert small nondeterministic circuits into a distributed PCP, is quite similar to her technique. Given two sets of variables x, y (or more than two, in our proof), we build a proof which 'robustly' encodes x and y, and a set of constraints over the proof symbols that check the following conditions:

- consistency of the proof with x (checked by Alice);
- consistency with y (checked by Bob);
- justification for the claim that $f(x, y) = 1$ (checked by both parties).

The soundness of our protocol derives from the soundness properties of PCPPs as well as our use of error-correcting codes (again following [Din07]). The extension to more than 2 verifiers, and to multiround debates rather than proofs, follows a careful but straightforward generalization of the original technique. The one key difference in our application is that we use the *size-efficient* PCPPs that Dinur provides as one of the final products of her technique, rather than the inefficient Long-Code-based PCPPs that she uses in her subroutine for alphabet reduction.

Our final result (Theorem 17), establishing limits on the power of PCPs for 2-verifier communication problems, is our only result that does not make use of PCPPs, relying instead on ideas of efficient sampling. In this result Alice and Bob simulate a given PCP protocol a number of times. Rather than explicitly estimating the 'goodness' of every possible proof they could be given by the absent prover (i.e. its probability of making them accept, given their current inputs x, y), they estimate the 'goodness' of every pair of k-tuples of proof bits that could be viewed by Alice and Bob, and every possible setting to that k-tuple. In this way the number of values to be estimated is $\ell^{O(k)}$, rather than 2^{ℓ}. From these estimates, Alice can derive an accurate estimate for the acceptance probability of the PCP protocol on every proof string z, which by considering all z allows her to estimate the *maximum* acceptance probability of the PCP protocol. (Note that we do not claim this simulation is computationally efficient!) We feel that the sampling technique in this proof is interesting and could find other applications.

Chapter 2

A PCP Characterization of AM

2.1 Preliminaries

2.1.1 Basic definitions

We assume familiarity with the notion of polynomial-time algorithms, and with the complexity class P consisting of languages $L \subseteq \{0,1\}^*$ whose membership problem is decidable in polynomial time. We also assume basic familiarity with the classes NP and AM, which we reviewed in the Introduction, and with the Boolean circuit model of computation (which we'll review shortly). We define promise classes and the promise class prAM in Section 2.1.3. For more background on complexity theory see, e.g., [Pap94].

For a language $L \subseteq \{0,1\}^*$, we use $L(x)$ to denote the characteristic function of L. We use $|x|$ to denote the length of a string x over some (possibly non-Boolean) alphabet Σ. $d(x, y)$ denotes the Hamming distance between strings $x, y \in \Sigma^n$, and $d(x, S)$ is the Hamming distance between $x \in \Sigma^n$ and a set $S \subseteq \Sigma^n$. If $d(x, S) \leq c$ we say x is *c-close to S*, otherwise x is *c-far from S*. Similarly, for $\alpha \in [0, 1]$, if $d(x, S) \leq \alpha n$ we say x is *α-close in relative distance to S*, otherwise x is *α-far in relative distance from S*.

$H(t) : [0, 1] \to [0, 1]$ denotes the binary entropy function, $H(t) = -t \log t - (1-t) \log(1-t)$ for $t \in (0, 1)$ and $H(0) = H(1) = 0$. We let $V_{n,k}$ denote the discrete volume of the Hamming sphere of radius k in $\{0,1\}^n$; that is,

$$V_{n,k} := \sum_{0 \leq i \leq k} \binom{n}{i},$$

and we use the known bound $V_{n,\alpha n} \leq 2^{H(\alpha)n}$ (valid for $\alpha \in [0,1/2]$).

When we speak of circuits, unless otherwise mentioned we mean deterministic Boolean circuits of fanin-two, and we measure circuit size (denoted $|C|$ for circuit C) as the number of gates (including inputs). For functions $p(n) \in [0,1]$, $s(n) \geq 0$ we say that a language L is *$p(n)$-hard for size $s(n)$* if for every Boolean circuit C of size $\leq s(n)$, $\Pr_{x \in \{0,1\}^n}[C(x) = L(x)] \leq p(n)$. We extend this definition to general functions: we say that a function $F : \{0,1\}^n \to \{0,1\}^m$ is $p(n)$-hard for size $s(n)$ if for every m-output Boolean circuit C of size $\leq s(n)$, $\Pr_{x \in \{0,1\}^n}[C(x) = F(x)] \leq p(n)$.

2.1.2 CSPs, PCPPs, and codes

Fix an integer $k \geq 1$. A *k-local Constraint Satisfaction Problem*, or *k-CSP*, over finite alphabet Σ is a collection $\psi(x) = \psi_1(x), \ldots \psi_m(x)$ of Boolean-valued functions on the input $x = (x_1, \ldots x_n) \in \Sigma^n$, where each ψ_j depends only on some k variables of x and is specified by a k-tuple $I_j \subseteq [n]$ and a truth-table on these k variables. Define $\mathrm{Val}_\psi(x)$, the *value of ψ on x*, as the fraction of constraints ψ_j that are satisfied by x (i.e. such that $\psi_j(x) = 1$).

Next we define PCPPs. Fix a circuit $C(x)$ on n Boolean input variables, a finite alphabet Σ, and a parameter $\beta > 0$. We say that a k-CSP ψ is a *PCPP for C over Σ with security β* if:

1. ψ is defined on variable set (x, z), where x are the Boolean input variables to C and z are auxiliary 'proof' variables taking values in Σ;

2. For any $x \in \{0,1\}^n$, if $C(x) = 1$ then there exists a setting of z such that $Val_\psi(x, z) = 1$;

3. For all $x \in \{0,1\}^n$ and z, $\mathrm{Val}_\psi(x, z) \leq 1 - \beta \cdot \frac{d(x, C^{-1}(1))}{n}$.

The *proof size* of ψ is the number of variables in z.

The following positive result on PCPPs is due to Dinur.

Theorem 6. *[Din07, Cor. 9.3] There is a constant-size alphabet Σ_0, a constant $\beta > 0$, and a polynomial-time algorithm that, given a circuit $Q(x)$ of size t, produces a 2-CSP $\psi_Q(x, z)$ that is a PCPP for Q over Σ_0 with security β. Moreover, the proof size of ψ is $O(\mathrm{poly}(t))$.*

Following the techniques of earlier papers working with PCPPs, we will use PCPPs in conjunction with efficient error-correcting codes. A (binary) *code* is an injective map $E : \{0,1\}^N \to \{0,1\}^{N'}$ where $N' \geq N$. We also use E to denote the image of the map, i.e., we consider $E \subseteq \{0,1\}^{N'}$. The *minimum distance* of the code is the minimum over distinct $u, v \in E$ of $d(u, v)$. An algorithm D *decodes E from an η fraction of errors* if, given any string u at relative distance at most η from some $u' \in E$, $A(u)$ outputs u'. Note that for such decoding to be possible, the minimum distance must be greater than $2\eta N'$.

We will use the following well-known fact:

Theorem 7. *There is a polynomial-time computable code E for all input lengths N with output length $N' = O(N)$, and an $\eta > 0$, such that E can be polynomial-time decoded from an η fraction of errors.*

Many such constructions are known; recently Goldwasser et al. [GGH$^+$08] gave a construction in which the decoder algorithm can be implemented in AC^0, i.e., with constant-depth, polynomial-size Boolean circuits.

2.1.3 Promise problems and prAM

A *promise problem* is a pair $\Pi = (\Pi_{YES}, \Pi_{NO})$ of disjoint subsets of $\{0,1\}^*$ (the 'yes' and 'no' instances, respectively). For a function $s(n) \in [0,1]$, we say that $(\Pi_{YES}, \Pi_{NO}) \in \mathsf{prAM}_{1,s(n)}$ if there exists a polynomial-time randomized algorithm $M(x, r, w)$, with $|r|, |w| = O(\mathrm{poly}(n))$ such that:

1. (Completeness) If $x \in \Pi_{YES}$, then with probability 1 over the random string r, there exists a $w = w(r)$ such that $M(x, r, w) = 1$;

2. (Soundness) If $x \in \Pi_{NO}$ and $|x| = n$, then the probability over the random string r that there exists a w such that $M(x, r, w) = 1$, is at most $s(n)$.

The algorithm M defines an 'Arthur-Merlin protocol': we consider that a polynomially bounded verifier Arthur chooses a random 'challenge' r for the computationally unbounded Merlin, who sees r and gives a response w which Arthur accepts or rejects..

We define $\mathsf{prAM} = \mathsf{prAM}_{1,1/3}$. A promise problem $\Pi_1 = (\Pi_{YES}, \Pi_{NO})$ is prAM-*hard* if for all $\Pi' = (\Pi'_{YES}, \Pi'_{NO})$ in prAM, there exists a polynomial-time computable reduction $R(x)$, such that, if $x \in \Pi'_{YES}$, then $R(x) \in \Pi_{YES}$, while if $x \in \Pi'_{NO}$, then $R(x) \in \Pi_{NO}$. We say that Π is prAM-*complete* if Π is in prAM and is prAM-hard.

It is not hard to see that for any $\Pi \in \mathsf{prAM}$, the soundness parameter $1/3$ in the protocol can be made exponentially small in n, by parallel repetition of the original protocol. However, we require soundness-amplification that is more efficient in its use of randomness. This is provided by a result of Bellare et al. [BGG93]. They state their theorem for AM, not for prAM, but the proof carries over without changes to the promise setting and we state it for this setting.

Theorem 8. *[BGG93] Let $\Pi = (\Pi_{YES}, \Pi_{NO}) \in \mathsf{prAM}$, where $M(x, r, w)$ is a polynomial-time predicate defining an Arthur-Merlin protocol for Π. Let $n = |x|$, and fix a polynomial $m(n)$. Then there exists an Arthur-Merlin protocol for Π defined by a polynomial-time predicate $M'(x, r', w')$, with $|w'| \le O(\mathrm{poly}(n)), |r| \le |r'| \le O(|r| + m(n))$, and with soundness $2^{-m(n)}$.*

The randomness-efficiency in the above result has been improved in more recent work (see [MU02] for a discussion), but we do not need or use these improvements.

2.1.4 AM-k-CSPs

By an *AM-k-CSP* we mean a k-CSP $\psi(r, z)$, where r are Boolean and z may be non-Boolean. We call r the 'Arthur-variables' and z the 'Merlin-variables'. Informally speaking, we are interested in the game in which the r are first set uniformly by Arthur, and then Merlin sets z to try to maximize the fraction of constraints of ψ satisfied by (r, z).

For any fixed $k \geq 1$, soundness parameter $s = s(|r|) \in [0, 1]$, alphabet Σ, and fixed $\varepsilon \in (0, 1]$, we define the promise problem $\mathsf{Gap} - \mathsf{AM} - \mathsf{k} - \mathsf{CSP}_{1,1-\varepsilon,s(|r|)} = (\Pi_{AM-CSP,YES}, \Pi_{AM-CSP,NO})$ as follows. Both 'yes' and 'no' instances are AM-k-CSPs over Σ. If $\psi(r, z) \in \Pi_{AM-CSP,YES}$, we are promised that for all choices of r, there exists a z such that $\mathrm{Val}_\psi(r, z) = 1$. If $\psi(r, z) \in \Pi_{AM-CSP,NO}$, we are promised that only for at most an $s(|r|)$ fraction of strings r does there exist a z with $\mathrm{Val}_\psi(r, z) > 1 - \varepsilon$.

2.2 An Augmented PCPP

As a tool for proving Theorems 1, 4, and 5, we prove the following 'augmented' version of the PCPP Theorem (Theorem 6), which we derive from Theorem 6. We remark that the proof of our prAM-completeness result (Theorem 1) uses only condition (9.i) of the Lemma below; this first part of the Lemma is quite similar to previous uses of PCPPs. Also, our use of error-correcting codes will only be important for establishing condition (9.ii).

Lemma 9. *There is a finite alphabet Σ_0 such that the following holds. For any $\varepsilon > 0$ there is a $\nu > 0$ and a polynomial-time algorithm A that takes as input a Boolean circuit $C = C(r, w)$. A outputs a 2-CSP $\psi(r, z)$, where $|z| = O(\mathrm{poly}(|C|))$ and the variables of z are over Σ_0. Letting $\ell = |r|$, ψ has the following properties:*

(9.i) For all r, if there is a w such that $C(r, w) = 1$, then there is a z such that $\mathrm{Val}_\psi(r, z) = 1$. On the other hand, if r is $\alpha\ell$-far from any r' for which $C(r', \cdot)$ is satisfiable, then for all z, $\mathrm{Val}_\psi(r, z) < 1 - \Omega(\alpha)$.

(9.ii) Suppose $P(r)$ is any (possibly randomized) procedure such that with probability at least $p = p(\ell)$ over a uniform $r \in \{0, 1\}^\ell$ and any randomness in P, $P(r)$ outputs a z such that $\mathrm{Val}_\psi(r, z) > 1 - \nu$.

Then there exists a deterministic procedure $\tilde{P}(r)$, such that with probability at least $p(\ell) \cdot 2^{-\varepsilon\ell}$ over uniform r, $\tilde{P}(r)$ outputs a w such that $C(r, w) = 1$. Moreover, $\tilde{P}(r)$ is computable by a nonuniform, $\mathrm{poly}(|C|)$-sized circuit that makes a single oracle call to P on the same input length (with P's randomness fixed nonuniformly).

Proof. Let $N := |w|$. We can assume, by padding w if necessary, that $N \geq \ell$. Let $E : \{0,1\}^N \to \{0,1\}^{N'}$ be the error-correcting code given by Theorem 7 (with $N' = O(N)$), applied to inputs of length N.

Let $b = b(\ell) := \lceil \frac{N'}{\ell} \rceil$. Define a predicate $Q(r_1, r_2, \ldots, r_b, u)$, with $|r_i| = |r| = \ell$ for $i \leq b$ and $|u| = N'$, by the following rule: $Q(r_1, r_2, \ldots r_b, u) = 1$ iff $r_1 = r_2 = \ldots, = r_b$, $u = E(w)$ for some w, and $C(r_1, w) = 1$. Clearly we can efficiently construct a circuit of size $O(\text{poly}(|C|))$ computing Q. Note that by our setting of b, there are more variables in the blocks r_j than in u.

Let $\psi_0 = \psi_Q((r_1, \ldots, r_b, u), Z)$ be the PCPP 2-CSP for Q over alphabet Σ_0 given by Theorem 6, efficiently constructible and of size $O(\text{poly}(|C|))$. We take ψ_0 and make two changes. First, we substitute the variables of r for the corresponding variables of each vector r_i. Second, we allow the variables of u to range over all of Σ_0 (we may assume $\{0,1\} \subseteq \Sigma_0$), and modify each constraint to reject in the case where one or more of its u-variables are set to a non-Boolean value.

We denote the resulting 2-CSP by $\psi(r, u, Z)$. We claim that this efficiently constructible 2-CSP satisfies the conditions of Lemma 9's statement, with $z := (u, Z)$ and Σ_0 as in Theorem 6. Our setting of $\nu > 0$ will be determined later.

First, we show that condition (9.i) is satisfied. Consider any $r \in \{0,1\}^\ell$. Suppose that there exists $w \in \{0,1\}^N$ such that $C(r, w) = 1$. Then $Q(r, r, \ldots, r, E(w)) = 1$. Using the completeness property of PCPPs, there exists a Z such that $\text{Val}_\psi((r, E(w)), Z) = 1$. On the other hand, say r is $\alpha\ell$-far from any r' for which $C(r', \cdot)$ is satisfiable. Given any $u \in \Sigma_0^{N'}$, let us choose some string $u' \in \{0,1\}^{N'}$ which agrees with u on any variable where u is Boolean. We observe that $(r, r, \ldots, r, u')$ is $\alpha/2$-far in relative distance from $Q^{-1}(1)$. By the soundness property of PCPPs, for any choice of Z, $\text{Val}_\psi((r, u'), Z) < 1 - \frac{\alpha\beta}{2}$, where $\beta > 0$ is the constant from Theorem 6. Also, by the way we defined ψ, $\text{Val}_\psi((r, u), Z) \leq \text{Val}_\psi((r, u'), Z)$. We have verified condition (9.i).

Now we turn to condition (9.ii). Let $P(r)$ be as described in (9.ii). Note that by averaging, we may fix (nonuniformly) some value of the randomness used by P while preserving the lower-bound $p(\ell)$ on its success probability over the choice of r; we do so and consider P a deterministic algorithm from now on. We set $\nu := \eta\beta\gamma/4$, where η is the constant in Theorem 7, β is the constant in Theorem 6, and $\gamma \in (0,1)$ is a small constant to be announced.

Let $P'(r)$ be the procedure that, on input r, computes $z = P(r) = (u, Z)$ and runs the polynomial-time decoder for E on u, yielding a string $w \in \{0,1\}^N$. Let P' output w.

We analyze the behavior of P'. Let $z = (u, Z)$ be any output of $P(r)$ such that $\mathrm{Val}_\psi(r, z) > 1 - \nu$. By the soundness property of PCPPs, the string $(r, r, \ldots, r, u)$ must be $\frac{\nu}{\beta} = \frac{\eta\gamma}{4}$-close in relative distance to some string $(r_1, \ldots, r_b, u')$ for which $Q(r_1, \ldots, r_b, u') = 1$ (and thus $r_1 = \ldots = r_b$ and $u' \in E$). Since $|u'| = N' \leq b(\ell) \cdot |r| \leq 2N'$, we find that $d(r, r_1) < \gamma\ell$ and $d(u, u') < \eta N'$. The latter inequality implies that when P' applies the polynomial-time decoder to u, it correctly recovers $w = E^{-1}(u')$. Since $Q(r_1, \ldots, r_b, u') = 1$, we have $C(r_1, w) = 1$.

To analyze $\tilde{P}$, say that a string $r \in \{0,1\}^n$ is *good* if $P'(r)$ outputs a w such that there exists an r' at distance at most $\gamma\ell$ from r, such that $C(r', w) = 1$. Our analysis of P', combined with our original assumption about the success probability of P, shows that at least a $p(\ell)$ fraction of strings r are good.

Now we define the procedure $\tilde{P}(r)$: $\tilde{P}(r)$ first chooses a vector $v \in \{0,1\}^l$ uniformly from the set of all strings of Hamming weight at most γl, then outputs $P'(r + v)$. Note that, if r is selected uniformly, $r + v$ is also uniform and, after conditioning on its value, r is uniformly distributed over all strings at distance at most $\gamma\ell$ from $r + v$. Thus, conditioning on $r + v$ being good, we have at least a $1/V_{\ell,\gamma\ell} \geq 2^{-H(\gamma)\ell}$ chance that $C(r, P'(r + v)) = 1$. So the overall success probability of $\tilde{P}(r)$ is at least $p(\ell) \cdot 2^{-H(\gamma)\ell}$. Since $H(\gamma) \to 0$ as $\gamma \to 0$, we may choose $\gamma > 0$ so that the success probability is at least $p(\ell) \cdot 2^{-\varepsilon\ell}$.

P' is clearly a polynomial-time algorithm making one call to P, while $\tilde{P}$ simply makes one call to P' after its random sampling and bitwise addition mod 2. The choice of v can be nonuniformly fixed in a way that does not decrease the success probability, so $\tilde{P}$ can be implemented with the resources claimed. Thus we have verified condition (9.ii), completing the proof of the Lemma. □

2.3 Proof of Theorem 1

We restate Theorem 1 in the terminology of Section 2.1.4:

Theorem 1 (restated). *There is a finite alphabet* Σ *and a constant* $\varepsilon > 0$*, such that* $\mathsf{Gap-AM-2-CSP}_{1,1-\varepsilon,\exp\{-\Omega(|r|)\}}$ *is* prAM*-complete.*

Proof. First, we claim that for any $s(|r|) = o(1)$ and $\varepsilon > 0$, $\mathsf{Gap-AM-2-CSP}_{1,1-\varepsilon,s(|r|)} = (\Pi_{AM-CSP,YES}, \Pi_{AM-CSP,NO})$ is in prAM. The protocol is as follows: given a 2-CSP $\psi(r,z)$, Arthur picks r uniformly and Merlin responds with a setting of z. Arthur accepts iff $\mathrm{Val}_\psi(r,z) = 1$. If $\psi \in \Pi_{AM-CSP,YES}$, then clearly Arthur accepts with probability 1 when Merlin responds optimally. If $\psi \in \Pi_{AM-CSP,NO}$, then Arthur accepts with probability at most $s(|r|)$, which is greater than $2/3$ for large enough $|r|$. (For instances with $|r|$ below this threshold, Arthur can simply request certificates $z(r)$ for every setting of r and verify that each satisfies $\mathrm{Val}_\psi(r, z(r)) = 1$.)

Thus our main task is to show that the promise problem is prAM-hard, for appropriate choice of parameters. Let $\Pi = (\Pi_{YES}, \Pi_{NO}) \in \mathsf{prAM}$, and let $M_1(x, r_1, w_1)$ be a polynomial-time-computable predicate defining an Arthur-Merlin protocol for Π. We use parameters $n = |x|, \ell_1(n) = |r|$; by definition of prAM we have $\ell_1(n) = O(\mathrm{poly}(n))$ and $|w_1| = O(\mathrm{poly}(n))$. By padding r_1 if necessary we may assume $\ell_1(n) \geq n$. Apply Theorem 8 to M_1, with the setting $m(n) := \ell_1(n)$. Thus we get a new Arthur-Merlin protocol $M_2(x, r_2, w_2)$ for Π, with $|r_2| = \ell_2(n) \in [n, \ldots, D \cdot \ell_1(n)]$ (for some fixed $D > 0$), $|w_2| = O(\mathrm{poly}(n))$, and with soundness $2^{-\ell_1(n)}$.

Given an input $x \in \Pi_{YES} \cup \Pi_{NO}$, we construct a poly($n$)-sized circuit $C(r_2, w_2) = C_x(r_2, w_2)$ that accepts iff $M_2(x, r_2, w_2) = 1$. To this circuit we apply the algorithm A of Lemma 9 (with a setting of $\varepsilon > 0$ to be announced), yielding a 2-CSP $\psi = \psi(r_2, z)$ which we make the output of our reduction.

We show the correctness of the reduction. First, suppose that $x \in \Pi_{YES}$. Then for each choice of r_2, there exists a w_2 such that $M_2(x, r_2, w_2) = 1$. By condition (9.i) of Lemma 9, there exists z such that $\mathrm{Val}_\psi(r_2, z) = 1$. Thus $\psi \in \Pi_{AM-CSP,YES}$.

Now suppose that $x \in \Pi_{NO}$. Then by the soundness property of M_2, the number of strings r_2 for which $M_2(x, r_2, \cdot)$ is satisfiable is at most $2^{-\ell_1(n)} \cdot 2^{\ell_2(n)} \leq 2^{(1-\frac{1}{D})\ell_2(n)}$. Thus the number of r_2 for which there exists an r' at distance $\leq \alpha\ell_2(n)$ from r_2, such that $M_2(x, r', \cdot)$ is satisfiable, is at most

$$V_{\ell_2(n),\alpha\ell_2(n)} \cdot 2^{(1-\frac{1}{D})\ell_2(n)} \leq 2^{(H(\alpha)+1-\frac{1}{D})\ell_2(n)}.$$

Choosing $\alpha > 0$ such that $H(\alpha) < \frac{1}{D}$, we find that with probability $\geq 1 - \exp\{-\Omega(\ell_2(n))\}$ over a uniform choice of r_2, r_2 is $\alpha\ell_2(n)$-far from any r' such that $C(r', \cdot)$ is satisfiable. For such r_2 and for any z, condition (9.i) of Lemma 9 tells us that $\mathrm{Val}_\psi(r_2, z) < 1 - \Omega(\alpha)$.

Thus if we fix $\varepsilon > 0$ as an appropriately small constant and choose an appropriate $s(|r_2|) = \exp\{-\Omega(|r_2|)\}$, we have $\psi \in \Pi_{AM-CSP,NO}$. This completes the proof of correctness for our reduction. □

2.4 Randomized Optimization Hypotheses Imply Collapse of AM

What significance might Theorem 1, our 'PCP characterization of AM', have for the project of trying to prove new upper bounds on the power of this class? In the Introduction we gave two hypotheses inspired by Theorem 1. Each of these hypotheses, if true, would have major implications for the study of AM; this is the content of Claims 2 and 3 from the Introduction, which we prove next.

Proof of Claim 2. Let $L \in \mathsf{AM}$; then $(L, \overline{L}) \in \mathsf{prAM}$. Given an instance x, let Arthur run the reduction in Theorem 1 on input x, producing a 2-CSP $\psi(r, z)$. Let Merlin send Arthur a polynomial-sized circuit $C : \{0,1\}^\ell \to \{0,1\}^m$, with $\delta := \varepsilon$ (here ε is from Theorem 1). Then Arthur runs C on a sufficiently large ($O(\mathrm{poly}(n))$) number of random choices of r, accepting only if he finds an r such that $\mathrm{Val}_\psi(r, C(r)) \geq 1 - \varepsilon$.

First suppose $x \in L$; then by the completeness property of our reduction, for all r there exists a z for which $\mathrm{Val}_\psi(r, z) = 1$. If Merlin sends the circuit C_ψ assumed to exist by Hypothesis A, then with at least $1/\operatorname{poly}(|\ell|)$ probability over r, $\mathrm{Val}_\psi(r, C(r)) \geq 1 - \varepsilon$. So if Arthur samples a sufficiently large (polynomial) number of strings r, Arthur will accept with probability $> 2/3$.

Next suppose $x \notin L$; then our reduction guarantees that for all but an exponentially small fraction of strings r, for all z $\mathrm{Val}_\psi(r, z) < 1 - \varepsilon$. So Arthur's acceptance probability is negligible no matter what circuit Merlin sends. Thus we have an MA protocol for L. □

Proof of Claim 3. We apply Hypothesis B with $\delta := \varepsilon/3$, yielding a value $t = t(\delta)$. Let $L \in \mathsf{AM}$ be given, and let Arthur run the reduction from Theorem 1 on input x, yielding an instance $\psi(r, z)$. Let Merlin send a description of a function $F(r)$, where each output of F depends on at most t bits of r (note F can be described in polynomial size). Arthur performs explicit variable-substitutions $z = F(r)$ in ψ and uses linearity of expectation to exactly compute $\mathbb{E}_r[\mathrm{Val}_\psi(r, F(r))]$.

If $x \in L$ and Merlin sends F_ψ as given by Hypothesis B, this expectation is at least $(1 - \delta)^2 > 1 - 2\varepsilon/3$. On the other hand, if $x \notin L$ then, regardless of the function sent, this expectation is at most $(1 - \varepsilon) + \exp\{-\Omega(|r|)\}$. Thus for $|r|$ large enough we can distinguish the two cases. (If $|r|$ is below a fixed threshold, Arthur can instead request that Merlin send optimal values $z(r)$ for each r.) Arthur's computations are deterministic and polynomial-time, so the above defines an NP protocol for L. □

Note that Claim 3 would hold even if we weakened Hypothesis B, allowing each coordinate of $F(r)$ to depend on $t(\delta, n) = O_\delta(\log n)$ coordinates. We state Hypothesis B in a stronger form because, although we believe it is false, we don't know how to disprove it unconditionally even in the form given.

2.5 Evidence Against the Randomized Optimization Hypotheses

Next we use Lemma 9, in conjunction with known results about amplification of hardness, to prove Theorems 4 and 5. That is, under various complexity-theoretic assumptions, we exhibit families of 2-CSPs $\psi(r, z)$ for which it is hard on average to approximately optimize over z, for randomly chosen r. As mentioned earlier, the conclusions of both Theorems are easily seen to falsify both of our Randomized Optimization Hypotheses, and we consider this evidence that these hypotheses are probably false.

First, amplification of hardness in $\mathsf{NP} \cap \mathsf{coNP}$ from $(1 - 1/\operatorname{poly}(n))$-hardness to $2/3$-hardness is made possible by the following result of Impagliazzo [Imp95, essentially Thm. 2]:

Theorem 10. *[Imp95] Suppose that there exists a language L, a function $s(n)$, and a $c > 0$, such that L is $(1 - \frac{1}{n^c})$-hard for size $s(n)$. Then for any $c' > 0$, there exists another language L' such that L' is $(\frac{1}{2} + O(\frac{1}{n^{c'}}))$-hard for size $\frac{s(n)}{n^{O(1)}}$. Moreover, L' is polynomial-time truth-table reducible to L.*

Lemma 11. *Suppose that there exists a language $L \in \mathsf{NP} \cap \mathsf{coNP}$ and $\gamma, c > 0$ such that L is $(1 - \frac{1}{n^c})$-hard for size $2^{\gamma n}$. Then there exists another language $L' \in \mathsf{NP} \cap \mathsf{coNP}$ and a $\gamma' > 0$ such that L' is $2/3$-hard for size $2^{\gamma' n}$ (for sufficiently large n).*

Proof. Apply Theorem 10, with $s(n) := 2^{\gamma n}$ and with any $c' > 0$ and $\gamma' \in (0, \gamma)$, and use the fact that $\mathsf{NP} \cap \mathsf{coNP}$ is closed under polynomial-time reducibilities, i.e., $\mathsf{P}^{\mathsf{NP} \cap \mathsf{coNP}} = \mathsf{NP} \cap \mathsf{coNP}$. □

Next, the Impagliazzo-Wigderson pseudorandom generator [IW97] allows us to amplify 'moderate' hardness of the type produced by Lemma 11 into 'extreme' hardness, albeit of a function problem rather than a decision problem (in [IW97] additional techniques are used to produce extremely hard decision problems, but we do not follow this path). The next definition follows [IW97] (and previous works). Given a language L, an integer $c \geq 1$, a parameter $k = k(n)$, and a function $G(r) : \{0,1\}^{cn} \to \{0,1\}^{k \times n}$ (called a 'generator' function), define $L^k \circ G : \{0,1\}^{cn} \to \{0,1\}^k$ by

$$(L^k \circ G)(r) := (L(G_1(r)), L(G_2(r)), \ldots, L(G_k(r)))\,,$$

where the string $G(r)$ is divided into k blocks $G_1(r), \ldots, G_k(r)$, each of length n. The basic idea is that if $G(r)$ is appropriately 'pseudorandom', then the collection $G_1(r), \ldots, G_k(r)$ should behave in important respects like a truly independent collection of random strings. In particular, if it is somewhat hard to compute $L(x)$ for a random x, it should be very hard to compute $(L^k \circ G)(r)$ correctly when k is large.

The following result (a restatement of [IW97, Thm. 2.12]) gives the main hardness-amplification property of the generator defined in that paper, which we denote G_{IW}.

Theorem 12. *[IW97] For any $\gamma > 0$, there are $\gamma', c > 0$, and a polynomial-time computable $G_{IW} : \{0,1\}^{cn} \to \{0,1\}^{n \times n}$, such that: if L is $2/3$-hard for size $2^{\gamma n}$, then $(L^n \circ G_{IW})(r)$ is $2^{-\gamma' n}$-hard for size $2^{\gamma' n}$.*

(Recall our definition of average-case hardness for general functions from Section 2.1.1.)

Proof of Theorem 4. We begin by applying Lemma 11 to our language $L \in \mathsf{NP} \cap \mathsf{coNP}$, yielding a language $L' \in \mathsf{NP} \cap \mathsf{coNP}$ that is $2/3$-hard for circuits of size $2^{\gamma_0 n}$ for some $\gamma_0 > 0$. Then we apply Theorem 12 to L'; we derive a $\gamma' > 0$, such that $((L')^n \circ G_{IW})(r)$ is $2^{-\gamma' n}$-hard for size $2^{\gamma' n}$.

Since $L' \in \mathsf{NP} \cap \mathsf{coNP}$, there exists a polynomial-time witness predicate $M(x, w)$, producing outputs from $\{0, 1, ?\}$, satisfying:

1. for all $(x, w), M(x, w) \in \{L'(x), ?\}$;

2. for all x, there exists a w such that $M(x, w) = L'(x)$;

3. $|w| = O(\text{poly}(n))$.

Let $t(n) = |w|$. We reformat M if necessary to ensure that the first bit of w consists of a 'claim' bit, call it w_{cl}, such that for any (x, w) with $M(x, w) = L'(x)$, we have $w_{cl} = L'(x)$.

Next we define $M'(x, w)$, which outputs 1 if $M(x, w) \in \{0, 1\}$, 0 otherwise. M' is also polynomial-time computable.

Define a predicate $Q(r, w_1, \ldots, w_n) : \{0, 1\}^{cn} \times \{0, 1\}^{n \times t(n)} \to \{0, 1\}$ as follows: $Q(r, w_1, \ldots, w_n) = 1$ iff for all $i \in [n]$, $M'(G_{IW,i}(r), w_i) = 1$. Q is polynomial-time computable since G_{IW} and M' are, so let Q_n be a $O(\text{poly}(n))$-sized circuit for Q on input parameter n. Clearly Q_n is efficiently constructible.

We claim that Q defines a hard-on-average search problem. To see this, suppose $C(r) : \{0, 1\}^{cn} \to \{0, 1\}^{n \times t(n)}$ is any circuit of size at most $2^{\gamma' n}$ which has some $p(n)$ probability over r of outputting a collection $w_1, \ldots, w_n$ for which $Q(r, w_1, \ldots, w_n) = 1$. Then we may construct a circuit $C'(r)\{0, 1\}^{cn} \to \{0, 1\}^n$ that simply restricts the output of $C(r)$ to the 'claim' bits of the strings $w_1, \ldots, w_n$ that C produces. Observe that $C'(r)$ has a $p(n)$ chance (over r) of correctly outputting $((L')^n \circ G_{IW})(r)$. Moreover, $C'(r)$ also has size bounded by $2^{\gamma' n}$. We conclude $p(n) \leq 2^{-\gamma' n}$.

We invoke Lemma 9 with $\varepsilon := \gamma'/(2c)$, yielding a poly-time algorithm A (and an associated $\nu > 0$). We apply this A to Q_n, yielding a 2-CSP $\psi_n(r, z)$ (here $|z| = d(n) = O(\text{poly}(n))$). We claim that the 2-CSP family $\{\psi_n(r, z)\}_{n>0}$ satisfies the conditions of Theorem 4.

To see this, first note that for all r, $M'(r, \cdot)$ is satisfiable; so, there exists $w_1, \ldots, w_n$ such that $Q_n(r, w_1, \ldots, w_n) = 1$. Thus by condition (9.i) of Lemma 9, there exists z such that $\text{Val}_{\psi_n}(r, z) = 1$. So condition (4.i) is satisfied.

To establish condition (4.ii), let $\gamma_2 := \varepsilon$ and $\theta := \nu$. Suppose $C(r') : \{0, 1\}^{cn} \to \{0, 1\}^{|w'|}$ is a circuit of size at most $2^{\gamma_2 n}$, such that with some probability $q(n)$, $\text{Val}_{\psi_n}(r', C(r')) > 1 - \theta$. By condition (9.ii) of Lemma 9, there exists a circuit $\tilde{C}(r) : \{0, 1\}^{cn} \to \{0, 1\}^{n \times t(n)}$, such that with probability at least $q(n) \cdot 2^{-\varepsilon(cn)}$ over r, $Q_n(r, \tilde{C}(r)) = 1$. Moreover, $\tilde{C}$ is of size at most $|C| + O(\text{poly}(n))$, which for large enough n is less than $2^{\gamma' n}$. By our previous analysis we find that $q(n) \cdot 2^{-\varepsilon(cn)} \leq 2^{-\gamma' n}$, i.e., $q(n) \leq 2^{(\varepsilon c - \gamma')n} = 2^{-\gamma' n/2} = \exp\{-\Omega(n)\}$. We have proved condition (4.ii), for our settings of γ_2, θ. □

Next we turn to prove Theorem 5. For this, we need a more detailed analysis of the generator G_{IW}. Besides the hardness-amplification property summarized in Theorem 12, G_{IW} has another useful property: with very high probability over r, the fraction of the strings $G_{IW,1}(r), \ldots, G_{IW,n}(r)$ which are in L is close to $|L \cap \{0,1\}^n|/2^n$, that is, close to the fraction we'd expect if these strings were drawn independently and uniformly. To prove this fact (not proved or used in [IW97]), we first describe the generator in more detail.

The input r to G_{IW} consists of two parts, $r = (r_a, r_b)$. $G_{IW}(r_a, r_b)$ is defined blockwise for $i \in [n]$ as

$$G_{IW,i}(r_a, r_b) = K_i(r_a) + K'_i(r_b),$$

with $K_i : \{0,1\}^{|r_a|} \to \{0,1\}^n$, $K'_i : \{0,1\}^{|r_b|} \to \{0,1\}^n$, and with $+$ denoting bitwise addition mod 2. The definition of K'_i is not important to us; let us describe the functions K_i. The string r_a defines a random walk of length n (counting the starting vertex) on an explicit expander graph $\mathcal{G}_n$ with vertex set $\{0,1\}^n$. $\mathcal{G}_n$ is 16-regular with normalized second eigenvalue λ_n at most some fixed $\lambda < 1$. $v_i = K_i(r_a)$ represents the i-th vertex visited in this walk. v_1 is a uniform element, and each subsequent step v_{i+1} is a uniform choice from among the neighbors of v_i. (Note that this can be achieved with $|r_a| = O(n)$ random bits as claimed.)

We will use the following powerful result, called the 'strong Chernoff bound for expander walks', as proved by Healy [Hea08].

Theorem 13. *[Hea08] Let $G = (V, E)$ be a d-regular graph with second eigenvalue λ, let $m > 0$, and let $f_1, \ldots, f_m : V \to [0,1]$ have expectations $\mu_1, \ldots, \mu_m$ (over a uniform choice of $v \in V$). Taking a random walk $v_1, \ldots, v_m$ on G with uniform starting-point, we have for all $\varepsilon > 0$,*

$$\Pr\left[\left|\sum_{i \le m} f_i(v_i) - \sum_{i \le m} \mu_i\right| \ge \varepsilon m\right] \le 2e^{-\frac{\varepsilon^2(1-\lambda)m}{4}}.$$

A more general claim was made earlier by Wigderson and Xiao [WX05], but the proof contained an error, as pointed out in [WX08]. (A valid proof of the Theorem above, with different constants, can still be extracted from [WX05].)

For any $r \in \{0,1\}^{cn}$, let $\sharp(r) := |\{i \in [n] : L(G_{IW,i}(r)) = 1\}|$. Theorem 13 implies the following concentration bound for the generator G_{IW}:

Lemma 14. *Let L be an arbitrary language. Let $c_n = |L \cap \{0,1\}^n|/2^n$. Then for any fixed $\delta > 0$,*

$$\Pr_r[|\sharp(r) - c_n \cdot n| \geq \delta n] \leq \exp\{-\Omega(n)\}.$$

Proof. Recall that $r = (r_a, r_b)$. We show that the above inequality is true after conditioning on any value of r_b; this will prove the Lemma. Let $(v'_1, \ldots, v'_n) = (K'_1(r_b), \ldots, K'_n(r_b))$. Then for $i \in [n]$, $G_{IW,i}(r) \in L$ iff $K_i(r_a) + v'_i \in L$, or equivalently $K_i(r_a) \in L_n + v'_i$ (where $L_n := L \cap \{0,1\}$ and $L_n + v'_i = \{x + v_i : x \in L_n\}$).

Define $f_i : \{0,1\}^n \to \{0,1\}$ to be the characteristic function of $L_n + v'_i$. Clearly $\mu_i = c_n$ for all i. The result now follows by a direct application of Theorem 13, using the fact that $\mathcal{G}_n$ has second eigenvalue bounded away from 1. □

Proof of Theorem 5. Our choice of ε_0, determined later, will be no larger than 1/6, so by Theorem 12, our hypothesis implies there is a $\gamma' > 0$, a $c > 0$, and a polynomial-time $G_{IW} : \{0,1\}^{cn} \to \{0,1\}^{n \times n}$, such that: for any circuit $C : \{0,1\}^n \to \{0,1\}^n$ of size at most $2^{\gamma' n}$,

$$\Pr_r[C(r) = (L^n \circ G_{IW})(r)] \leq 2^{-\gamma' n}.$$

Let $M(x, w)$ be a polynomial-time verifier for L: $x \in L$ iff there exists w such that $M(x, w) = 1$. Let $t(n) = |w| = O(\text{poly}(n))$.

Let $\sharp(r)$ be as defined after Theorem 13. If $C : \{0,1\}^{cn} \to \{0,1\}^{t(n) \times n}$ is a circuit producing n strings $w_1, \ldots, w_n$, each of length $t(n)$, define

$$\sharp_C(r) := |\{i \in [n] : M(G_{IW,i}(r), w_i) = 1\}|.$$

Claim 15. *There exists $\gamma'' > 0$ such that the following holds. If $C(r) : \{0,1\}^{cn} \to \{0,1\}^{t(n) \times n}$ is a circuit of size at most $2^{\gamma'' n}$, then*

$$\Pr_r[\sharp(r) < \sharp_C(r) + \gamma'' n] < 2^{-\gamma'' n}.$$

Proof (of Claim 15). Let $\alpha > 0$. Say $C(r) : \{0,1\}^{cn} \to \{0,1\}^{t(n) \times n}$ is a circuit of size at most

$2^{\alpha n}$, such that $\Pr_r[\sharp(r) < \sharp_C(r) + \alpha n] \geq 2^{-\alpha n}$. Consider the following randomized procedure that attempts to compute $(L^n \circ G_{IW})(r)$:

- Let $C(r) = (w_1, \ldots, w_n) \in \{0,1\}^{t(n) \times n}$. Let $I \subseteq [n]$ be the indices i for which $M(G_{IW,i}(r), w_i) = 1$. Pick a random subset J of $[n]$, uniformly from the set of all subsets of size less than αn (including the empty set). Output the characteristic vector of $I \cup J$.

We analyze this procedure. Suppose r is any input for which $\sharp(r) < \sharp_C(r) + \alpha n$. Note that by definition of M, we always have $I \subseteq \{i \in [n] : L(G_{IW,i}(r)) = 1\}$. Then there exists a $J \subseteq [n] \setminus S$, of size less than αn, such that $I \cup J = \{i \in [n] : L(G_{IW,i}(r)) = 1\}$. Thus conditioned on this event, the procedure succeeds with probability at least $\frac{1}{V_{n,\alpha n}} \geq 2^{-H(\alpha)n}$. So the overall success probability is at least $2^{-\alpha n} \cdot 2^{-H(\alpha)n} = 2^{-(\alpha + H(\alpha))n}$.

Let us nonuniformly fix a setting J that maximizes the procedure's success probability, and use this choice to run the procedure. The result is a (nonuniform) circuit of size $2^{\alpha n} + O(\text{poly}(n))$, with success probability $\geq 2^{-(\alpha+H(\alpha))n}$. For α sufficiently small this contradicts the hardness of $(L^n \circ G_{IW})$, proving the claim. □

Now we set $\varepsilon_0 := \min(1/6, \gamma''/4)$. Fix any circuit $C : \{0,1\}^{cn} \to \{0,1\}^{t(n) \times n}$ of size at most $2^{\gamma'' n}$. We use Lemma 14 applied to $\delta := \varepsilon_0$, and the previous Claim, to find that, with probability $\geq 1 - \exp\{-\Omega(n)\}$ over r, we have the simultaneous inequalities $(c_n + \varepsilon_0)n > \sharp(r) > (c_n - \varepsilon_0)n$ and $\sharp(r) \geq \sharp_C(r) + \gamma'' n$. Call a string r with this property *C-typical.*

What is c_n? We claim it must lie in $[1/2 - \varepsilon_0, 1/2 + \varepsilon_0]$. For otherwise, a size-1 circuit could guess $L(x)$ with probability greater than $1/2 + \varepsilon_0$ by guessing the majority value on length n, contrary to our hardness assumption about L. Thus for a C-typical r, $\sharp_C(r) \leq (1/2 + \varepsilon_0)n - \gamma'' n < (1/2 - 3\gamma''/4)n$ and also $\sharp(r) \geq (1/2 - \varepsilon_0)n - \varepsilon_0 n \geq (1/2 - \gamma''/2)n$.

Defining η as some rational number in the interval $(1/2 - 3\gamma''/4, 1/2 - \gamma''/2)$, define a predicate $Q(r, w_1, \ldots, w_n) : \{0,1\}^{cn} \times \{0,1\}^{n \times t(n)} \to \{0,1\}$ as follows: $Q(r, w_1, \ldots, w_n) = 1$ iff for at least an η fraction of indices i we have $M(G_{IW,i}(r), w_i) = 1$. Q is itself polynomial-time computable, computed by some uniform family $\{Q_n\}_{n>0}$ of poly-size circuits. We have the key property that for a C-typical r, there exist $w_1, \ldots, w_n$ such that $Q(r, w_1, \ldots, w_n) = 1$, yet $Q(r, C(r)) = 0$.

Invoke Lemma 9 with $\varepsilon := \gamma''/(2c)$, yielding an algorithm A (and an associated $\nu > 0$). Then we claim $\{A(Q_n)\}_{n>0} = \{\psi_n(r,z)\}_{n>0}$ is the desired family of 2-CSPs (here $|r| = cn, |z| = d(n) = O(\text{poly}(n))$). First we verify condition (5.i). Consider any r for which there exists a $w_1, \ldots, w_n$ such that $Q_n(r, w_1, \ldots, w_n) = 1$. By condition (9.i) of Lemma 9, we find that in this case there exists z such that $\text{Val}_{\psi_n}(r, z) = 1$. Since all but an $\exp\{-\Omega(n)\}$ fraction of r have this property, condition (5.i) is satisfied.

To establish condition (5.ii), fix γ_2 as any value in $(0, \gamma'')$ and let $\theta := \nu$. Suppose $C(r) : \{0,1\}^{cn} \to \{0,1\}^{d(n)}$ is a circuit of size at most $2^{\gamma_2 n}$, such that with some probability $q(n)$, $\text{Val}_{\psi_n}(r, C(r)) > 1-\theta$. By condition (9.ii) of Lemma 9, there exists a circuit $\tilde{C}(r) : \{0,1\}^{cn} \to \{0,1\}^{n \times t(n)}$, such that with probability at least $q(n) \cdot 2^{-\varepsilon(cn)}$ over r, $Q_n(r, \tilde{C}(r)) = 1$. Note that such an r fails to be $\tilde{C}$-typical. Moreover, $\tilde{C}$ is of size at most $|C| + O(\text{poly}(n))$, which for large enough n is less than $2^{\gamma'' n}$.

So, by our previous analysis we find that $q(n) \cdot 2^{-\varepsilon cn} \leq 2^{-\gamma'' n}$, i.e., $q(n) \leq 2^{(\varepsilon c - \gamma'')n} = 2^{-\gamma'' n/2} = \exp\{-\Omega(n)\}$. We have proved condition (5.ii). This completes the proof of Theorem 5. □

Finally, we note that versions of Theorems 4 and 5 can be proved, in which both the hypotheses and conclusions apply, not to general circuits, but to the class of TC^0 circuits (i.e., constant-depth Boolean circuits with majority gates), or any circuit class containing TC^0. This is because all the reductions involved can be carried out in TC^0. (For a discussion of why the Impagliazzo and Impagliazzo-Wigderson constructions amplify hardness in TC^0, see Agrawal [Agr01]; the difficulties in amplifying hardness in lower classes like AC^0 were explored by Shaltiel and Viola [SV08].)

2.6 Questions for Further Research

- Does our approximation problem remain prAM-complete if each variable in the CSP $\psi(r, z)$ is restricted to appear in only a constant number of constraints? (The 'expander-replacement' technique [PY91, Pap94] allows us to restrict the occurrences of z-variables in our prAM-completeness proof; it is the 'stochastic' r-variables which pose a challenge.) Alternatively, can one perhaps show that under this restriction the problem lies in NP?

- Can we unconditionally disprove Hypothesis B? Given the sharp limitations of NC^0 circuits this might be possible.

- Can PCP ideas be used to give new upper bounds on the class AM?

- Find more applications of PCPPs in complexity theory.

Chapter 3

The Power of PCPs for Communication

We now turn to the second setting in which we investigate the power of PCPs, the communication model. We study both 2-party and multiparty communication in the presence of PCPs.

3.1 Preliminaries

In this chapter we will make use of the definitions in Sections 2.1.1 and 2.1.2, and in particular will use the results desribed there on error-correcting codes and PCPPs.

3.1.1 Σ^j circuits

We need the standard notion of Σ^j circuits (a generalization of nondetermistic circuits), which we define next. Consider a circuit $C(x, w_1, \ldots, w_j)$ receiving an input string x and 'debate strings' $w_1, \ldots, w_j$ of fixed length. Consider the game in which two Provers each learn the input x and alternate setting values to the strings $w_1, \ldots, w_j$ (with full knowledge of each other's plays so far). Prover 1 plays first, so that on the i-th move, if i is odd, Prover 1 sets the value of w_i, while if i is even, Prover 2 sets the value of w_i.

The circuit C is called a Σ^j*-circuit* for Boolean function $f(x)$ if the following holds:

- If $f(x) = 1$, there exists a Prover 1 strategy for setting $w_1, w_3, \ldots$ such that, regardless of Prover 2's strategy, $C(x, w_1, \ldots, w_j) = 1$;
- if $f(x) = 0$, there exists a Prover 2 strategy such that, regardless of Prover 1's strategy, $C(x, w_1, \ldots, w_j) = 0$.

Note that a nondeterministic circuit for f (in the usual sense) is precisely a Σ^1 circuit for f. In measuring the size of Σ^j circuits, we as usual count the number of gates, and input gates of each type are included in this count.

3.1.2 Two-party communication complexity

We now review basic notions of communication complexity; for a more detailed treatment see [KN96].

In the standard model of 2-party communication protocols, Alice and Bob hold two strings x, y, each usually assumed to have the same length $n > 0$. They wish to compute some Boolean function $f(x, y)$, using as little communication as possible. Communication proceeds in turns; Alice and Bob alternate sending 1 bit at a time (beginning with Alice). The protocol terminates when Alice chooses an output bit.

At each step, Alice and Bob may choose what bit to communicate next based on their own input string and on the communication they have received so far. We consider the setting in which Alice and Bob may each follow a randomized communication strategy. Say that a joint strategy (i.e., a pair of strategies for Alice and Bob) computes $f(x, y)$ with *completeness* c and *soundness* s if for any x, y, $f(x, y) = 1$ implies Pr[Alice outputs 1] $\geq c$, while $f(x, y) = 0$ implies Pr[Alice outputs 1] $\leq s$.

A strategy is $t(n)$*-communication-bounded* if on every execution on strings of length n, Alice and Bob communicate at most $t(n)$ bits total. For a Boolean function $f(x, y)$, we let $R_{c,s}(f)$, the (c, s)*-randomized communication complexity*, denote the minimum value $t = t(n)$ such that some t-communication-bounded protocols computes f with completeness

c and soundness s. Note, this measure places no computational restrictions on the strategies of Alice and Bob.

3.1.3 PCPs for 2-party communication

We give two models of PCP protocols for communication: a two-party model, in which Alice and Bob may communicate while querying the proof string, and a multiparty model, in which no communication occurs between the verifiers. In either case we consider a generalization of the usual situation in which a single prover presents a proof string, to the notion of a *probabilistically checkable debate system* [CFLS95, CFLS97], in which two competing Provers try to convince the Verifiers to output 1 or 0, respectively.

2-verifier PCP protocols with communication. In a *2-verifier distributed PCP protocol with communication*, Alice and Bob alternate sending bits to each other as in a standard communication protocol. However, Alice and Bob are also provided with a binary *proof string* $w \in \{0,1\}^{\ell}$. At each step of the protocol, Alice and Bob may choose to look at one or more bits of w. At the end of the protocol, Alice produces a Boolean output.

The PCP protocol P is called an (t, k, ℓ)*-2-party protocol* if the proof w is of length ℓ and, for any settings to the strings (x, y, w), at most t bits are exchanged, and Alice and Bob each inspect at most k bits of w.

Now fix $j \geq 1$, an $f : \{0,1\}^n \rightarrow \{0,1\}$, and values $s < c$ both in $[0,1]$. Suppose the proof w is broken into j disjoint blocks as $w = w_1, \dots w_j$. Consider the j-round, perfect-information game where two Provers (distinct from Alice and Bob) alternate fixing the values of the blocks, starting with Prover 1. (Informally, Prover 1 wants the Verifiers to believe $f(x) = 1$, while Prover 2 wants them to believe $f(x) = 0$.)

P is a Σ^j*-PCP protocol for* f *with completeness* c *and soundness* s if the following holds:

- If $f(x) = 1$, there exists a Prover 1 strategy such that, regardless of Prover 2's strategy, Alice outputs 1 with probability at least c;
- if $f(x) = 0$ there exists a Prover 2 strategy such that, regardless of Prover 1's strategy, Alice outputs 1 with probability at most s.

3.1.4 Multiparty PCP protocols

In the multiparty setting, we consider an input $x \in \{0,1\}^n$ to be distributed among $r(n) \geq 1$ agents called 'Verifiers', according to a partition $\mathbf{A} = \{A_1, \ldots A_{r(n)}\}$ of $[n]$, so that Verifier i receives the values x_{A_i} of x restricted to the indices in A_i. (We remark that our partition model is the so-called 'Number-in-Hand' model, which differs from the widely studied 'Number on Forehead' model [KN96].) The Verifiers are interested in computing the value $f(x)$ for some Boolean f.

In a *distributed PCP protocol*, in addition to their shares of the input, the Verifiers have joint access to a *proof string* w with variables over some finite 'proof alphabet' Σ (unlike in the 2-party setting, we consider non-Boolean alphabets).

Communication-free PCP Protocols. In an $r(n)$-*Verifier distributed, communication-free* $(\mathbf{A}, \Sigma, k, \ell)$ *PCP protocol* P, each Verifier i looks at x_{A_i} and inspects some k symbols of w (where $w \in \Sigma^\ell$), then outputs a bit y_i without exchanging any communication with the other Verifiers. We say that P is a $\mathbf{\Sigma}^j$-*PCP protocol for* f *with security* $\varepsilon > 0$ if the following holds (note the difference from earlier definitions):

Case 1: j is odd. Then:

- If $f(x) = 1$, there exists a Prover 1 strategy such that, regardless of Prover 2's strategy, every Verifier outputs $y_i = 1$ with probability 1;
- if $f(x) = 0$ there exists a Prover 2 strategy such that, regardless of Prover 1's strategy, with probability at least ε some Verifier outputs $y_i = 0$.

Case 2: j is even. Then:

- If $f(x) = 1$, there exists a Prover 1 strategy such that, regardless of Prover 2's strategy, with probability at least ε some Verifier outputs $y_i = 1$;
- If $f(x) = 0$ there exists a Prover 2 strategy such that, regardless of Prover 1's strategy, with probability 1 every Verifier outputs $y_i = 0$.

3.2 Our results

In our first result on communication problems, we use PCPPs to construct distributed communication-free Σ^j PCP protocols for any function having small Σ^j circuits.

Theorem 16. *There exists a constant-sized alphabet Σ_0 such that the following holds. Suppose $f(x) : \{0,1\}^n \to \{0,1\}$ has a Σ^j circuit C of size $m = m(n) \geq n$. For any n, let $\mathbf{A} = A_1 \cup \ldots \cup A_{r(n)}$ be a partition of $[n]$. Then f has an $(\mathbf{A}, \Sigma_0, 2, \mathrm{poly}(m))$ communication-free Σ^j-PCP protocol with security $\Omega(1/j)$, which, moreover, is executable by all Verifiers in uniform $O(\mathrm{poly}(m))$ time, given a description of C and $\mathbf{A}$. The Verifiers make nonadaptive queries.*

Note that this does indeed generalize the form of the Theorem stated in the Introduction.

In our next theorem, we complement Theorem 16 by giving an upper bound on the power of distributed PCP protocols, even ones which involve some limited amount of communication between the Verifiers. (We remark that this will be the only result in this thesis that does not use PCPPs in some way.) We restrict ourselves to the 2-Verifier case to avoid having to specify a model of communication for more than 2 parties, but the ideas here can be extended to give similar results for 3 or more Verifiers. We also, for simplicity, restrict attention to the case where the PCP string is Boolean, although this too is inessential.

Theorem 17. *Suppose $f(x,y)$ has a 2-party, (t,k,ℓ), Σ^j-PCP protocol with completeness c and soundness s (which need not be a computationally bounded protocol).*

Then: (17.i) $R_{2/3,1/3}(f) \leq \frac{\ell^{O(k)}t}{(c-s)^3}$.

(17.ii) If the PCP protocol has perfect completeness $c = 1$, then the same upper bound holds for $R_{1,1/3}(f)$.

Remark. The PCP-free randomized communication protocols we give in Theorem 17 are computationally unbounded. It is not clear how to make them computationally efficient, even if we assume the existence of a computationally eficient PCP protocol for f.

3.3 Proofs of the Theorems

3.3.1 Theorem 16

Proof of Theorem 16. We will only treat the case where j is odd; the case where j is even is handled very similarly.

Let $C(x, y_1, \ldots y_j)$ be a $\mathbf{\Sigma}^j$-circuit of size m for $f(x)$. By padding the y_i's if necessary, we may assume all blocks y_i are of some uniform length n'. The circuit size is increased only by a polynomial factor in this modification.

Let $ENC_a(x) : \{0,1\}^n \to \{0,1\}^{2m}, ENC_b(y) : \{0,1\}^{n'} \to \{0,1\}^{2m}$ be efficiently computable error-correcting codes, efficiently recognizable and efficiently decodable from at least γm errors, for some $\gamma = \Omega(1)$. (This is possible since $m \geq n + n'$.) Let $DEC_b : \{0,1\}^{2m} \to \{0,1\}^{n'}$ be the efficient decoder algorithm for ENC_b that recovers messages from γm errors.

Fix any setting $x_{A_t} = v_t$ to the t-th block of variables in the partition $\mathbf{A}$. Define the predicate $Q_{t,v_t}(X, Y_1, \ldots Y_j) : \{0,1\}^{(2m)(j+1)} \to \{0,1\}$ which equals 1 iff the following all hold:

(i) $X = ENC_a(x')$ for some x' such that $x'_{A_t} = v_t$;

(ii) For each *odd* i, Y_i equals $ENC_b(y_i)$ for some y_i;

(iii) One of the following two conditions hold:

(iiia) for all even i, all Y_i are $\gamma m/2$-close to a codeword of ENC_b, and $C(x', DEC_b(Y_1), \ldots DEC_b(Y_j)) = 1$;

or,

(iiib) some Y_i (for i even) is $\gamma m/2$-far from any codeword of ENC_b.

Note that Q_{t,v_t} is computed by a poly(m)-sized circuit that is efficiently constructible given v_t. (If (iiib) holds for some even i, we can verify it by observing that either $ENC_b(DEC_b(Y_i))$ is $\gamma m/2$-far from Y_i, or $DEC_b(Y_i)$ fails to return a value.)

Thus, by Theorem 6, we may efficiently construct a PCPP 2CSP $\psi_{t,v_t}((X, Y_1, \ldots Y_j), Z_t)$ for Q_{t,v_t} with alphabet Σ_0 of constant size and security $\beta = \Omega(1)$. Without loss of generality, we may let the variable-set Z_t be independent of the value v_t.

Here, then, is our $\mathbf{\Sigma}^j$-PCP protocol for $f(x)$. Let each Verifier $t \in [r(n)]$, receiving some piece v_t of the input, derive the 2CSP $\psi_{t,v_t}((X, Y_1, \ldots Y_j), Z_t)$. Let the PCP string consist of the following j blocks, to be assigned in this order:

- block 1 consists of (X, Y_1);

- block $i \in [2, j-1]$ consists of Y_i;

- block j consists of $(Y_j, Z_1, \ldots Z_{r(n)})$, where the Z-blocks are mutually disjoint.

Note that the total number ℓ of PCP variables is $O(\text{poly}(m))$, as required.

The t-th Verifier selects a uniformly random clause of ψ_{t,v_t}, checks it on the PCP variables fixed by the two Provers, and outputs 1 iff the clause selected is satisfied. Thus each Verifier makes at most 2 nonadaptive queries.

Now we show correctness of the PCP protocol. First, suppose that $f(x) = 1$. Then there exists a (deterministic) strategy S_1 for Prover 1 on input x that forces $C(x, y_1, \ldots y_j) = 1$. Consider the following strategy S_1' for Prover 1 in the $\mathbf{\Sigma}^j$-PCP protocol:

- On the first move, set $X = ENC_1(x), Y_1 = ENC_2(y_1)$, where y_1 is the move dictated by S_1.

- On the i-th move ($1 < i \le j$), if every variable-set $Y_{i'}$ assigned so far ($i' < i$) is $\gamma m/2$-close to a (necessarily unique) codeword $y_{i'}$, assign $Y_i = ENC_b(y_i)$, where y_i is the move dictated by S_1 in response to $y_1, \ldots y_{i-1}$. Otherwise, set $Y_i = ENC_b(0^{n'})$;

- Additionally, on the final, j-th move, do the following for each t. If $Q_{t,x_t}(X, Y_1, \ldots Y_j) = 1$, set Z_t in such a way as to ensure $\text{Val}_{\psi_{t,x_t}}((X, Y_1, \ldots Y_j), Z_t) = 1$, as is possible by the completeness property of PCPPs. Otherwise set Z_t arbitrarily.

We claim that the strategy S_1' ensures that after Prover 1 sets Y_j, $Q_{t,x_t}(X, Y_1, \ldots Y_j) = 1$ for all $t \leq r(n)$. To see this, fix any strategy of Prover 2. First suppose that the interaction of these two strategies causes Prover 2 to fix a block Y_i in a way that is $\gamma m/2$-far from a codeword. Then condition (iiib) is met and $Q_{t,x_t} = 1$ for all t. So, suppose this does not happen. Then, as S_1 is a winning strategy and by definition of S_1', we get $C(x, DEC_b(Y_1), \ldots DEC_b(Y_j)) = 1$. Thus (iiia) is satisfied, again causing $Q_{t,x_t} = 1$ for all t. In either case, the final setting of the Z-variables causes all Verifiers to accept with probability 1. This completes the analysis of the case $f(x) = 1$.

Now suppose $f(x) = 0$. Let S_2 be a winning strategy for Prover 2 (i.e. that forces $C(x, y_1, \ldots y_j) = 0$); we use it to define a strategy S_2' for Prover 2 in the Σ^j-PCP protocol. At each stage i (i even), if Prover 1 has previously set some block $Y_{i'}$ ($i' < i$) to an assignment $\gamma m/2$-far from a codeword of ENC_b, let Prover 2 set Y_i arbitrarily. Otherwise, let Prover 2 set $Y_i = ENC_b(y_i)$, where y_i is the move dictated by S_2 in response to $DEC_b(Y_1), \ldots DEC_b(Y_{i-1})$.

Consider the result when S_2' plays against any Prover 1 strategy. First, suppose Prover 1 on any round sets X to some value $\gamma m/2$-far from any codeword of ENC_a. In this case, for each $t \leq r(n)$, whatever values are assigned to $Y_1, \ldots Y_j$, the string $(X, Y_1, \ldots Y_j)$ is $\gamma m/2$-far from any input for which $Q_{t,v_t} = 1$. Since $(X, Y_1, \ldots Y_j)$ is of length $j + 1$ we find that (by the soundness property of PCPPs) whatever values Prover 1 assigns to Z_t, the t-th Verifier outputs 0 with probability $\Omega(\frac{\gamma}{j})$.

Similarly, if on any i-th round (i odd) Prover 1 sets Y_i to a value $\gamma m/2$-far from any codeword of ENC_b, each t-th Verifier outputs 0 with probability $\Omega(\frac{\gamma}{j})$. Now suppose Prover 1 sets X to be $\gamma m/2$-close to $ENC_a(x')$ for some $x' \neq x$. Let $t \leq r(n)$ be an index for which $x'_{A_t} \neq x_{A_t}$. Then referring to condition (i) in the definition of the predicate $Q_{t,x_{A_t}}$, and using the fact that ENC_a has minimum distance at least 2γ, we see that $(X, Y_1, \ldots Y_j)$ is γm-far from any string accepted by $Q_{t,x_{A_t}}$. Thus with probability $\Omega(\frac{\gamma}{j})$, Verifier t outputs 0.

Thus, we may assume that Prover 1 sets X to be $\gamma m/2$-close to $ENC_a(x)$, and at each round i (i odd) sets Y_i to be $\gamma m/2$-close to $ENC_b(y_i)$, for some y_i. Let $(X, Y_1, \ldots Y_j)$ be the result, and, fixing any $t \leq r(n)$ let $(X', Y_1', \ldots Y_j')$ be any input accepted by $Q_{t,x_{A_t}}$; we show that $d((X, Y_1, \ldots Y_j), (X', Y_1', \ldots Y_j')) \geq 3\gamma m/2$ (and thus Verifier t outputs 0 with probability $\Omega(\frac{\gamma}{j})$, regardless of Prover 1's settings of the Z-variables).

This is certainly the case if $X' = ENC_a(x')$ for some $x' \neq x$, so suppose $X' = ENC_a(x)$. Now since S_2 is a winning strategy for Prover 2, $C(x, DEC_b(Y_1), \ldots DEC_b(Y_j)) = 1$. Thus for at least one i we must have $DEC_b(Y_i') \neq DEC_b(Y_i)$. Now Y_i' is a codeword of ENC_b, while Y_i is $\gamma m/2$-close to some distinct codeword, so $d(Y_i, Y_i') \geq 2\gamma m - \gamma m/2 = 3\gamma m/2$.

We have shown that if $f(x) = 0$, there exists a Prover 2 strategy such that, for any Prover 1 strategy, with probability $\Omega(\gamma/j) = \Omega(1/j)$ at least one Verifier outputs 0. Thus the protocol we have defined is an $r(n)$-Verifier distributed, communication-free $(\mathbf{A}, k, \ell)$ Σ^j-PCP protocol with security $\Omega(1/j)$, as required.

□

3.3.2 Theorem 17

Proof of Theorem 17. (17.i): Let P denote Alice and Bob's strategy for the PCP protocol. Without loss of generality, assume that Alice and Bob each query *exactly* k distinct bits of the proof string z on all tuples (x, y, z). We write $z = (z_1, \ldots z_j)$ to indicate the successive blocks of z fixed by Provers 1 and 2.

We do some preliminary analysis. Fix any input (x, y). For a proof string $z \in \{0,1\}^\ell$, for $S, T \subseteq [\ell]$ each of size exactly k, and for $u, v \in \{0,1\}^k$, let $\gamma_{S,T,u,v}(z)$ equal 1 if the restriction of z to S, which we denote z_S, equals u and $z_T = v$ (in these equalities, the bits of z_S and z_T are read in ascending index-order), and $\gamma_{S,T,u,v}(z) = 0$ otherwise. Let $p_{S,T,u,v}(= p_{S,T,u,v}(x, y))$ be the probability that Alice and Bob, given a proof string z' satisfying $z'_S = u$ and $z'_T = v$, query z' in the positions S, T respectively and subsequently both accept. Note that $p_{S,T,u,v}$ indeed depends only on (S, T, u, v), and not on the other bits of z'. Note also that the conditions $z'_S = u, z'_T = v$ may be inconsistent, in which case we take $p_{S,T,u,v} = 0$ as convention.

Let $p_z(= p_z(x, y))$ be the probability that Alice and Bob both accept (x, y), given proof string z. By partitioning the possible accepting computations according to the set of query positions S, T made by Alice and Bob, we can decompose p_z in the following useful way:

$$p_z = \sum_{S,T,u,v} \gamma_{S,T,u,v}(z) \cdot p_{S,T,u,v}.$$

In the randomized (PCP-free) protocol we will construct, Alice and Bob's goal is to cooperatively compute approximations $\tilde{p}_{S,T,u,v}$ to each term $p_{S,T,u,v}$ that are (with high probability) accurate enough to allow Alice alone to estimate the above sum to within an additive error of $(c-s)/3$, for *every* choice of z (recall that the $p_{S,T,u,v}$ depend only on x, y). The first stage of our protocol will give Alice such a collection of estimates with probability at least $2/3$; in this event say Alice *gets good estimates.*

Assume for now the existence of such a subroutine. Then after the first stage, we have Alice, using her unbounded computational power, compute an estimate $\tilde{p}_z$ for *every* $z \in \{0,1\}^\ell$. Writing z in its j blocks as $z = (z_1, \ldots z_j)$, Alice considers the j-round game $\tilde{G}$ where $z_1, \ldots z_j$ are alternately set by Prover 1 and Prover 2, as in the original PCP protocol, and the payoff to Prover 1 on result z is given by $\tilde{p}_z$. Alice computes the value κ of this game, and accepts iff $\kappa > (c-s)/2$.

We show this protocol is correct (again assuming the subroutine). The original PCP protocol defines a j-round game G between Prover 1 and Prover 2. If $f(x,y) = 0$, G has value at most s. If then Alice gets good estimates, it is easily seen that $\tilde{G}$ has value at most $s + (c-s)/3 < (c-s)/2$. Similarly, if $f(x,y) = 1$ and Alice gets good estimates, then $\tilde{G}$ has value at least $c - (c-s)/3 > (c-s)/2$. Since Alice gets good estimates with probability at least $2/3$ (by assumption), the procedure is a bounded-error protocol for f.

We now give the promised subroutine to compute estimates $\{\tilde{p}_z : z \in \{0,1\}^\ell\}$. Let $N := \binom{\ell}{k}^2 \cdot 2^{2k}$ denote the size of the collection of all tuples $(S,T,u,v) \in \binom{[\ell]}{k} \times \binom{[\ell]}{k} \times \{0,1\}^k \times \{0,1\}^k$.

Let $M := \left\lceil \frac{1296\binom{\ell}{k}^2 \ln(6N)}{(c-s)^3} + 1 \right\rceil$. Alice maintains a 'count' variable $ct(S,T,u,v)$, initialized to 0, for each tuple (S,T,u,v). Alice and Bob perform the following procedure ('round i'), for $i = 1, 2, \ldots M$:

- Alice and Bob each generate random strings $r_A, r_B (= r_{A,i}, r_{B,i})$ of the length used in the PCP protocol P for f. For every pair (w_A, w_B) of k-bit vectors, Alice and Bob simulate $P(x, y)$ on the fixed choice of randomness r_A, r_B, passing messages as they would in P. In this simulation Alice answers her j-th simulated proof-string query $j \leq k$ with the j-th bit of w_A, and Bob answers his j-th simulated proof-string query with the j-th bit of w_B.

- After each such simulated run (on a particular (w_A, w_B)), if Bob accepts on the simulated run he sends to Alice the k-tuple $T \subseteq [\ell]$ on which he made his queries, along with the value $v \in \{0,1\}^k$ Bob observed of the proof string on its restriction to T (note v is just a rearrangement of the bits of w_B). If Bob rejects on the simulated run he simply indicates this fact to Alice.

  If Alice also accepted on this simulated run, then Alice observes the k-subset $S \subseteq [\ell]$ of indices she queried and the value u of the simulated proof string's restriction to S (a rearrangement of w_A). If the conditions $z_S = u, z_T = v$ are consistent, then Alice increments the counter $ct(S, T, u, v)$.

After this process has been repeated M times, Alice sets $\tilde{p}_{S,T,u,v} := \frac{ct(S,T,u,v)}{M}$, and for each $z \in \{0,1\}^\ell$ computes

$$\tilde{p}_z := \sum_{S,T,u,v} \gamma_{S,T,u,v}(z) \cdot \tilde{p}_{S,T,u,v}.$$

We show that for our setting of M, with probability 2/3 the above estimates satisfy $|p_z - \tilde{p}_z| < (c - s)/3$ for all z. First we analyze a single round i. Given any fixed tuple (S, T, u, v), we claim that $ct(S, T, u, v)$ can be incremented at most once on the i-th round. To see this, note that r_A, r_B are fixed on each round, so distinct pairs $(w_A, w_B) \neq (w'_A, w'_B)$ induce simulated runs whose queried proof bits disagree on some index; but any two simulated runs causing $ct(S, T, u, v)$ to increment must agree on their queried proof bits, which are exactly specified by (S, T, u, v).

Now for any (S, T, u, v) such that the conditions $z_S = u, z_T = v$ are consistent, what is the probability that $ct(S, T, u, v)$ is incremented on at least one (and hence, on exactly one) simulated run (w_A, w_B) on the i-th round? We claim it is precisely $p_{S,T,u,v}$. To see this, fix any z' with $(z'_S, z'_T) = (u, v)$. Note that whatever the choice of r_A, r_B on the i-th round, *exactly one* choice of (w_A, w_B) gives the sequence of proof bits seen by Alice and Bob respectively if they were to run $P(x, y)$ with proof z' and randomness r_A, r_B. Moreover, in this simulated run they also query the same positions as they would in the corresponding run on z'.

Thus the probability $ct(S, T, u, v)$ increments on round i is the probability over r_A, r_B that in $P(x, y)$ on proof z', Alice and Bob view z' on S, T respectively and then both accept. This is $p_{S,T,u,v}$ by definition.

Fix any tuple (S, T, u, v). Let $ct_M(S, T, u, v)$ denote the final value of $ct(S, T, u, v)$ after M rounds. The rounds are independent, so that $ct_M(S, T, u, v)$ is distributed as a sum of M independent 0/1 Bernoulli trials with success probability $p_{S,T,u,v}$.

Define $\alpha := \frac{(c-s)}{12\binom{\ell}{k}^2}$. Suppose first that $p_{S,T,u,v} < \alpha$. By the multiplicative version of Chernoff's bound we have $\Pr\left[\frac{ct_M(S,T,u,v)}{M} \geq 2\alpha\right] \leq e^{\frac{-\alpha M}{3}}$.

Now suppose $p_{S,T,u,v} \geq \alpha$. In this case, applying a two-sided Chernoff's bound, we have

$$\Pr\left[\left|\frac{ct_M(S,T,u,v)}{M} - p_{S,T,u,v}\right| \geq \frac{(c-s)}{6} \cdot p_{S,T,u,v}\right] \leq 2e^{\frac{-(c-s)^2\alpha M}{3 \cdot 36}} = 2e^{\frac{-(c-s)^3 M}{1296\binom{\ell}{k}^2}},$$

which by our setting of M is less than $\frac{1}{3N}$. Similarly we conclude that in the previous case, where $p_{S,T,u,v} < \alpha$, we also have $\Pr\left[\frac{ct_M(S,T,u,v)}{M} \geq 2\alpha\right] < \frac{1}{3N}$.

By a union bound over *all* (S, T, u, v), with probability greater than $1 - N \cdot \frac{1}{3N} = 2/3$, the following holds: for every (S, T, u, v) with $p_{S,T,u,v} < \alpha$, we get $\frac{ct_M(S,T,u,v)}{M} < 2\alpha$, and for every (S, T, u, v) with $p_{S,T,u,v} \geq \alpha$, we get $\frac{ct_M(S,T,u,v)}{M} \in ((1 - \frac{(c-s)}{6})p_{S,T,u,v}, (1 + \frac{(c-s)}{6})p_{S,T,u,v})$. Now $\tilde{p}_{S,T,u,v} = \frac{ct_M(S,T,u,v)}{M}$, so in such an outcome we get, for every $b = (\ldots b_{S,T,u,v} \ldots) \in \{0,1\}^N$, the inequalities

$$\sum_{S,T,u,v} b_i \tilde{p}_{S,T,u,v} < (1 + (c-s)/6) \cdot \left(\sum_{S,T,u,v} b_i p_{S,T,u,v}\right) + 2\alpha ||b||$$

(where $||b||$ is the Hamming weight of b), and similarly

$$\sum_{S,T,u,v} b_i \tilde{p}_{S,T,u,v} > (1-(c-s)/6) \cdot \left(\sum_{S,T,u,v} b_i p_{S,T,u,v} \right) - 2\alpha ||b||.$$

Now if $b = (\ldots \gamma_{S,T,u,v}(z) \ldots)$ for some $z \in \{0,1\}^\ell$, we have $||b|| = \binom{\ell}{k}^2$, since for each S, T there's exactly one pair u, v such that $\gamma_{S,T,u,v}(z) = 1$. Thus $2\alpha||b|| = 2(c-s)\binom{\ell}{k}^2/(12\binom{\ell}{k}^2) = (c-s)/6$.

We now plug in our definition $\tilde{p}_z = \sum_{S,T,u,v} \gamma_{S,T,u,v}(z) \cdot \tilde{p}_{S,T,u,v}$ and our earlier observation that $p_z = \sum_{S,T,u,v} \gamma_{S,T,u,v}(z) \cdot p_{S,T,u,v}$ into these inequalities, getting

$$\begin{aligned} \tilde{p}_z &\in \left((1 - \frac{(c-s)}{6})p_z - (c-s)/6, (1 + \frac{(c-s)}{6})p_z + (c-s)/6 \right) \\ &\subseteq (p_z - (c-s)/3, p_z + (c-s)/3) \,. \end{aligned}$$

Thus we get estimates of the desired accuracy with probability greater than 2/3. Combined with our description and analysis of the second stage of the protocol, this proves its correctness.

To analyze the total communication used by Alice and Bob, note that in each of the M rounds, they simulate P, the PCP protocol, 2^{2k} times (using t bits of communication each time by assumption on P). After each run Bob additionally sends a description of length $\log \binom{\ell}{k} + k + O(1)$. Thus the total communication is bounded by

$$\begin{aligned} M \cdot 2^{2k} \cdot \left(t + \log \binom{\ell}{k} + k + O(1) \right) &= O\left(\left[\binom{\ell}{k}^2 \log(N)(c-s)^{-3} \right] \cdot 2^{2k} \cdot (t + k \log \ell) \right) \\ &= O\left(\binom{\ell}{k}^2 \cdot k \log(\ell) \cdot 2^{2k}(t + k \log \ell)(c-s)^{-3} \right) \\ &= \frac{\ell^{O(k)} t}{(c-s)^3}. \end{aligned}$$

This proves (17.i).

For (17.ii), we use the same randomized protocol as in (17.i). We claim that for any z, if

$p_z = 1$ then $\tilde{p}_z$ is always 1, so that the protocol has perfect completeness. Thus, if the game G defined by the protocol (as described earlier) has value 1, so does the game $\tilde{G}$ evaluated by Alice. This will prove (17.ii), once we've established our claim.

To prove our claim, note that if the original PCP protocol has completeness $c = 1$, then for all $(x, y) \in f^{-1}(1)$, the game G defined earlier has value 1. That is, there exists a Prover 1 strategy for setting blocks of the proof string z such that, for all Prover 2 strategies, the resulting string z has $p_z = 1$. Now fix any z with $p_z = 1$. Note that on each round i, exactly 1 count-variable of form $ct(S, T, z_S, z_T)$ is incremented (namely, the S, T describing Alice and Bob's queries on z when they use the randomness r_A, r_B chosen in the i-th round and subsequently accept). Since $\tilde{p}_{S,T,u,v}$ is set to $\frac{ct(S,T,u,v)}{M}$ and there are M rounds, $\sum_{S,T} \tilde{p}_{S,T,z_S,z_T} = 1$. Then $\tilde{p}_z = \sum_{S,T,z_S,z_T}(1) \cdot \tilde{p}_{S,T,z_S,z_T} = 1$. Thus $\tilde{G}$ has value 1 as well, which causes Alice to accept. This means the PCP-free protocol has perfect completeness as claimed, and 1/3 soundness by our previous analysis. □

3.4 Questions for Further Research

There is probably room for improvement in one or both of our theorems on PCPs for communication, either for specific functions f or in general. More-efficient constructions of PCPPs would lead to an immediate improvement in Theorem 16, for example. Understanding when the PCP-free protocol in Theorem 17 can be made computationally efficient is another possible area for further study.

Bibliography

[Agr01] Manindra Agrawal. Hard sets and pseudo-random generators for constant depth circuits. In *FSTTCS*, pages 58–69, 2001.

[ALM$^+$98] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *J. ACM*, 45(3):501–555, 1998.

[Bab85] Lászl ó Babai. Trading group theory for randomness. In *STOC*, pages 421–429. ACM, 1985.

[BGG93] Mihir Bellare, Oded Goldreich, and Shafi Goldwasser. Randomness in interactive proofs. *Computational Complexity*, 3:319–354, 1993.

[BSGH$^+$06] Eli Ben-Sasson, Oded Goldreich, Prahladh Harsha, Madhu Sudan, and Salil P. Vadhan. Robust PCPs of proximity, shorter PCPs, and applications to coding. *SIAM J. Comput.*, 36(4):889–974, 2006.

[BSHLM08] Eli Ben-Sasson, Prahladh Harsha, Oded Lachish, and Arie Matsliah. Sound 3-query PCPPs are long. In *ICALP (1)*, pages 686–697, 2008.

[CFLS95] Anne Condon, Joan Feigenbaum, Carsten Lund, and Peter W. Shor. Probabilistically checkable debate systems and nonapproximability of pspace-hard functions. *Chicago J. Theor. Comput. Sci.*, 1995, 1995.

[CFLS97] Anne Condon, Joan Feigenbaum, Carsten Lund, and Peter W. Shor. Random debaters and the hardness of approximating stochastic functions. *SIAM J. Comput.*, 26(2):369–400, 1997.

[Coo71] Stephen A. Cook. The complexity of theorem-proving procedures. In *STOC*, pages 151–158, 1971.

[Din07] Irit Dinur. The PCP theorem by gap amplification. *J. ACM*, 54(3):12, 2007.

[DR06] Irit Dinur and Omer Reingold. Assignment testers: Towards a combinatorial proof of the PCP theorem. *SIAM J. Comput.*, 36(4):975–1024, 2006.

[GGH$^+$08] Shafi Goldwasser, Dan Gutfreund, Alexander Healy, Tali Kaufman, and Guy N. Rothblum. A (de)constructive approach to program checking. In *STOC*, pages 143–152, 2008.

[Hea08] Alexander Healy. Randomness-efficient sampling within NC^1. *Computational Complexity*, 17(1):3–37, 2008.

[HRTS07] Ishay Haviv, Oded Regev, and Amnon Ta-Shma. On the hardness of satisfiability with bounded occurrences in the polynomial-time hierarchy. *Theory of Computing*, 3(1):45–60, 2007.

[HVV06] Alexander Healy, Salil P. Vadhan, and Emanuele Viola. Using nondeterminism to amplify hardness. *SIAM J. Comput.*, 35(4):903–931, 2006.

[IJKW08] Russell Impagliazzo, Ragesh Jaiswal, Valentine Kabanets, and Avi Wigderson. Uniform direct product theorems: simplified, optimized, and derandomized. In *STOC*, pages 579–588, 2008.

[Imp95] Russell Impagliazzo. Hard-core distributions for somewhat hard problems. In *FOCS*, pages 538–545, 1995.

[IW97] Russell Impagliazzo and Avi Wigderson. $P = BPP$ if E requires exponential circuits: Derandomizing the XOR lemma. In *STOC*, pages 220–229, 1997.

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

[KN96] Eyal Kushilevitz and Noam Nisan. *Communication Complexity.* Cambridge University Press, 1996.

[MU02] Elchanan Mossel and Christopher Umans. On the complexity of approximating the VC dimension. *J. Comput. Syst. Sci.*, 65(4):660–671, 2002.

[O'D02] Ryan O'Donnell. Hardness amplification within NP. In *STOC*, pages 751–760, 2002.

[Pap94] Christos H. Papadimitriou. *Computational Complexity.* Addison Wesley, 1994.

[PY91] Christos H. Papadimitriou and Mihalis Yannakakis. Optimization, approximation, and complexity classes. *J. Comput. Syst. Sci.*, 43(3):425–440, 1991.

[SU07] Ronen Shaltiel and Christopher Umans. Low-end uniform hardness vs. randomness tradeoffs for AM. In *STOC*, pages 430–439, 2007.

[SV08] Ronen Shaltiel and Emanuele Viola. Hardness amplification proofs require majority. In *STOC*, pages 589–598, 2008.

[Tre04] Luca Trevisan. Inapproximability of combinatorial optimization problems. *CoRR*, cs.CC/0409043, 2004.

[Wil] Ryan Williams. Improving exhaustive search implies superpolynomial lower bounds. To appear in STOC 2010.

[WX05] Avi Wigderson and David Xiao. A randomness-efficient sampler for matrix-valued functions and applications. In *FOCS*, pages 397–406, 2005.

[WX08] Avi Wigderson and David Xiao. Derandomizing the Ahlswede-Winter matrix-valued Chernoff bound using pessimistic estimators, and applications. *Theory of Computing*, 4(1):53–76, 2008.