PSPACE Is Provable By Two Provers In One Round

Jin-yi Cai[*]
Department of Computer Science
Princeton University
Princeton, New Jersey 08544

Anne Condon [†]
Computer Science Department
University of Wisconsin-Madison
Madison, Wisconsin, 53706

Richard J. Lipton[‡]
Department of Computer Science
Princeton University
Princeton, New Jersey 08544

[*]Research supported by NSF grant CCR-8709818.
[†]Work supported by NSF grant number DCR-8402565
[‡]Research supported by DARPA and ONR contracts N00014-85-C-0456 and N00014-85-K-0465

Proposed running head (less than 35 characters):

PSPACE is Provable by Two Provers

Name and mailing address of author to whom proofs should be sent:

Anne Condon
Computer Science Department
University of Wisconsin at Madison
Madison, WI 53706.

Abstract

We show that every language in PSPACE, or equivalently every language accepted by an unbounded round interactive proof system, has a 1-round, 2-prover interactive proof system with exponentially small error probability. To obtain this result, we prove the correctness of a simple but powerful method for parallelizing 2-prover interactive proof systems to reduce their error.

1 Introduction

We describe a general methodology for parallelizing *unbounded round* interactive proof systems to obtain *1-round*, 2-prover interactive proof systems. We show that this methodology yields a 1-round, 2-prover interactive proof system for any language in PSPACE. Our interactive proof systems have *exponentially small* error probability.

The notion of a single-prover interactive proof system was introduced by Goldwasser, Micali and Rackoff [12] and by Babai [1] and was generalized to two and more provers by Ben-Or, Goldwasser, Kilian and Wigderson [5]. In a single-prover interactive proof system, a prover interacts with a probabilistic polynomial time verifier. Informally, the prover and the verifier share an input x, and the prover tries to convince the verifier that x is in some language. In a 2-prover interactive proof system, two provers interact with the verifier, but the provers cannot communicate with each other during the proof.

Recent breakthroughs have settled many questions on the power of interactive proof systems, culminating in the results of Lund, Fortnow, Karloff and Nisan [16] and Shamir [17] that any language in PSPACE has a single-prover interactive proof system, and Babai, Fortnow and Lund [3] that any language in nondeterministic exponential time has a two-prover interactive proof system. However until recently, the power of *bounded round*, 2-prover interactive proof systems remained a major open problem. In a *round* of an interactive proof system, the verifier sends a string to each prover in turn and receives a response from each prover. A bounded round interactive proof system is one in which the number of rounds is a constant independent of the input size.

In this paper we show how to simulate unbounded round single-prover interactive proof systems by 1-round, 2-prover interactive proof systems, proving the following theorem.

Theorem 1.1 *Every language in PSPACE has a 1-round, 2-prover interactive proof system with exponentially small error probability.*

Our result contrasts sharply with known results on bounded round, single-prover interactive proof systems. In the single prover case, Babai [1] and Goldwasser and Sipser [13] showed that any bounded round interactive proof system can be simulated by a 2-round interactive proof system. Thus the hierarchy of language classes defined by bounded round interactive proof systems collapses and is contained in the second level of the polynomial time hierarchy.

Roughly, our proof technique is as follows. In an unbounded round, single-prover interactive proof system, the verifier sends random bits to the prover in polynomially many rounds. To reduce the number of rounds, the verifier sends all the bits in one round to prover I of the 2-prover system. However, this prover can "cheat" by computing the responses to the verifier, based on all bits at once. To ensure that prover I does not cheat in this way, the verifier sends a *subsequence* of the random bits to prover II, and compares the answers of both provers. Polynomially many copies of this protocol are executed *in parallel*, to reduce the probability of error without increasing the number of rounds.

This idea is a natural one, but although it has been studied in a number of previous papers, this is the first to prove that it yields 1-round interactive proof systems that have exponentially small error probability. It was first proposed by Fortnow, Rompel and Sipser [11]. However, Fortnow [8] later observed that the polynomially many copies of the protocol are not independent, which complicates the analysis. Thus the proof in [11] is incorrect. In [6] and later in [7], the present authors studied the problem of parallelizing 2-prover interactive proof systems. Their analysis showed that any language accepted by an unbounded round, single-prover interactive proof system is accepted by a 1-round, 2-

prover interactive proof system with error probability at most a constant (7/8). This paper extends their technique and combines it with recent work of Lund et al. [16] and Shamir [17] to achieve exponentially small error probability.

The rest of the paper is organized as follows. In this section we give a precise description of the model of an interactive proof system and summarize related work on the model. In Section 2, we review Shamir's interactive proof system for the Quantified Boolean Formula (QBF) problem, a PSPACE-complete problem. We then describe a 1-round, 2-prover interactive proof system for the QBF problem, which is obtained by parallelizing the protocol of Shamir. In Section 3, we prove that this new, 2-prover interactive proof system has exponentially small error probability.

1.1 Background and Related Work

We now give a precise definition of a multi-prover interactive proof system, which was discussed informally in the previous section. Let k be a constant. A *k-prover interactive proof system* is a tuple $(P_1, P_2, \ldots, P_k, V)$, where V is a probabilistic Turing machine, with a read-only input tape, a read-write worktape and a source of random bits (a coin). In addition, the verifier has k special communication tapes, which allow the verifier to communicate with the provers.

The states of V are partitioned into two types, reading and communication states. A transition function describes the one-step transitions of the verifier. Whenever the verifier is in a reading state, the transition function of the verifier determines the next configuration of the verifier, based on the symbol under the tape heads, the state and the outcome of an unbiased coin toss. Whenever the verifier is in a communication state, the next configuration is determined as follows. For $1 \leq i \leq k$, the contents of the ith communication tape are replaced by a string written by the ith prover. Then, the next state of the verifier is determined just as when V is in a reading state.

Each prover P_i is specified by a prover transition function. This function determines what string is written by the prover, based on the input and the sequence of all past strings written by the verifier on the ith communication tape. Formally, if Σ is the tape alphabet of the verifier and provers, and $\mathcal{H}$ is the set of all finite sequences of strings over Σ^*, then P_i is a mapping $P_i : \Sigma^* \times \mathcal{H} \rightarrow \Sigma^*$. There is no restriction on the complexity of a prover transition function.

The states of V include special accepting and rejecting states, and the computation halts once one of these states is reached. If an accepting or rejecting state is reached on a particular computation, we say that $(P_1, \ldots, P_k, V)$ accepts or rejects, respectively, on that computation.

We say $(P_1, \ldots P_k, V)$ is a k-prover interactive proof system for language L with error probability $\epsilon < 1/2$ if there is some N such that for all strings w of length $\geq N$,

- if $w \in L$, the probability that $(P_1, \ldots P_k, V)$ accepts w is $> 1 - \epsilon$,
- if $w \notin L$, then for all provers $P_1^*, \ldots, P_k^*$, the probability that $(P_1^*, \ldots, P_k^*, V)$ rejects w is $> 1 - \epsilon$.

Informally, this definition states that a language is accepted by an interactive proof system if on all inputs in the language, the provers can convince the verifier to accept with high probability, whereas on inputs not in the language, the verifier rejects with high probability, on any strategy of the provers. We denote by IPS and 2IPS interactive proof systems with 1 and 2 provers respectively, and the class of languages they accept by IP and 2IP, respectively.

The *number of rounds* of a protocol is the number of times V enters a communication state. If at the jth round, V enters a communication state with a string x written on ith communication tape, we say that V *sends the string* x *to* P_i *at the* j*th round.* Similarly, if P_i writes the string y on the ith communication tape, we say that V *receives the string* y *from* P_i *at the* j*th round.*

There has recently been much progress in understanding the power of interactive proof systems. Lund, Fortnow, Karloff and Nisan [16] found an interactive proof system for the permanent function, which is hard for $\#P$ by a result of Valiant [19] and thus hard for the polynomial time hierarchy, PH, by a result of Toda [18]. Thus they showed that any language in PH is in IP. Their proof uses a result of Lipton [15] that the permanent of square matrices over a finite field is random self-reducible. Lipton's proof is based on Beaver and Feigenbaum's [4] construction of "instance hiding schemes" for arbitrary Boolean functions. Shamir [17] generalized their proof to show that all languages in PSPACE have interactive proof systems. In contrast, the bounded round IPS hierarchy collapses, and is contained in the second level of the polynomial hierarchy. This follows from results of Babai [1] and Goldwasser and Sipser [13]. Finally, Babai, Fortnow and Lund [3] showed that there are 2-prover interactive proof systems for any language in nondeterministic exponential time, that is, 2IP = NEXP.

Related work of Feige [9] shows that any language in nondeterministic exponential time has a 1-round 2IPS, with error probability $< 1/2$. A similar result is also attributed to Kilian [private communication in Feige [9]]. The main difference between Kilian's result and Feige's result is that in Kilian's result, the number of rounds is 2. Subsequent to the work described in this paper, Verbitsky [20] showed that there is a constant round, 2-prover protocol for any language in nondeterministic exponential time, based partly on our methods. Independently, Shamir and Lapidot [14] constructed a 1-round, multi-prover IPS with exponentially small error probability for any language in nondeterministic exponential time. Feige [10] has reduced the number of provers in their protocol to two, thus proving that the class of languages accepted by 1-round, 2-prover interactive proof systems is exactly nondeterministic exponential time.

2 A 1-round, 2-prover IPS for the QBF Problem

In this section we present our 1-round, 2-prover IPS for the Quantified Boolean Formula (QBF) problem. The set of quantified Boolean formulas is the closure of the set of Boolean variables x_i and their negations $\bar{x}_i$ under the operations $\vee$, (or) $\wedge$ (and), $\forall x_i$ and $\exists x_i$. The QBF problem is to determine whether a given QBF is true.

Shamir [17] showed that any QBF can be converted in polynomial time to a *simple* QBF which is true if and only if the original QBF is true. A simple QBF is one in which every occurrence of each variable is separated from its point of quantification by at most one universal quantifier. The conversion is done by first renaming all Boolean variables after each universal quantifier, so that the name of variable x_i when it appears between the jth and $(j+1)$st universal quantifier after its first appearance, is x_i^j. Then, each variable $x_i^j, j \geq 1$ is existentially quantified just after the jth universal quantifier and is equated with x_i^{j-1}. For example, the QBF

$$\exists x_1 \forall x_2 \forall x_3 (x_1 \vee (x_3 \wedge \bar{x}_2))$$

is converted into the simple QBF

$$\exists x_1^0 \, \forall x_2^0 \, \exists x_1^1 \, (x_1^1 = x_1^0) \wedge \forall x_3^0 \, \exists x_1^2 \, \exists x_2^1 \, (x_1^2 = x_1^1) \wedge (x_2^1 = x_2^0) \wedge (x_1^2 \vee (x_3^0 \wedge \bar{x}_2^1)).$$

Hence without loss of generality, we need only consider simple QBF's in this discussion.

Our 2-prover IPS is obtained by parallelizing the IPS of Shamir [17] for the QBF problem. We first present some useful background and notation, based on the work of Lund et al. [16] and Shamir [17]. A QBF Q can be transformed in polynomial time to a closed arithmetic form A so that Q is true if and only if $A \neq 0$. To do this, the symbols $\vee$ and $\wedge$ are replaced by $+$ and $\cdot$ respectively; similarly $\forall x_i$ and $\exists x_i$ are replaced by $\prod_{x_i \in \{0,1\}}$ and $\sum_{x_i \in \{0,1\}}$ respectively and finally, $\bar{x}_i$ is replaced by $(1 - x_i)$.

The heart of Shamir's IPS is to verify that such a closed arithmetic form A is not equal to 0. Since the value of A may be double exponential in the size of A, the verifier actually checks that $A = a \bmod p$, where $a \neq 0$ and p is a prime of length polynomial in the size of A. We assume that $p > 2^n$, where n denotes the number of quantifiers of A.

We introduce the following notation to describe this IPS precisely. For any closed arithmetic form A, let $A'(x_i)$ be the polynomial obtained from A by eliminating the leftmost $\sum_{x_i \in \{0,1\}}$ or $\prod_{x_i \in \{0,1\}}$, so that x_i is a free variable in the field $\mathbf{Z}/p\mathbf{Z}$. A key property of a simple QBF is that if A is its corresponding arithmetic form, then the degree of $A'(x_i)$ is polynomial in the size of A. (See Shamir [17], Theorem 5, for a proof). Let $r_1, \ldots, r_n$ be a sequence of values in $\mathbf{Z}/p\mathbf{Z}$. For $0 \leq i \leq n$ we define closed arithmetic forms A_i and the associated polynomials $A_i'(x_i)$ as follows: $A_0 = A$ and for $i > 0$, $A_i = A_{i-1}'(r_i)$. The number of quantifiers of the closed form A_i is $n - i$. We assume that the degree of $A_0'(x_1)$ (and hence the degree of all the $A_i'(x_i)$) is bounded by n^k where k is a constant and n is the number of quantifiers of A. Note that the closed formulas A_i may have constant coefficients from the field $\mathbf{Z}/p\mathbf{Z}$. For example, suppose $p = 11$, $r_1 = 6$, $r_2 = 8$ and

$$A = A_0 = \sum_{x_1 \in \{0,1\}} \prod_{x_2 \in \{0,1\}} \left((x_1 + x_2) + \sum_{x_3 \in \{0,1\}} x_2 x_3 \right).$$

Then

$$A_1 = \prod_{x_2 \in \{0,1\}} \left((6 + x_2) + \sum_{x_3 \in \{0,1\}} x_2 x_3 \right)$$

and

$$A_2 = 3 + \sum_{x_3 \in \{0,1\}} 8 x_3.$$

We now summarize the protocol of the verifier of Shamir's IPS to verify that $A = a \bmod p$.

1. Choose a sequence $r_1, \ldots, r_n$ of values in $\mathbf{Z}/p\mathbf{Z}$ independently and uniformly at random. Let $i = 1$.
2. Receive from the prover a polynomial $f_i(x_i) \in \mathbf{Z}/p\mathbf{Z}[x_i]$.
3. For each $i, 1 \leq i \leq n$,
   - if the leftmost quantifier of A_{i-1} is $\sum$, then if $i > 1$, check that $f_i(0) + f_i(1) = f_{i-1}(r_{i-1}) \bmod p$ and if $i = 1$, that $f_i(0) + f_i(1) = a \bmod p$.
   - if the leftmost quantifier of A_{i-1} is $\prod$, then if $i > 1$, check that $f_i(0) f_i(1) = f_{i-1}(r_{i-1}) \bmod p$ and if $i = 1$, that $f_i(0) f_i(1) = a \bmod p$.

   If the check fails, halt and reject. Otherwise if $i < n$, send r_i to the prover, set $i = i + 1$, and repeat the protocol from step 2. If $i = n$, check that $f_n(r_n) = A_n$ and if so, halt and accept.

The prover is defined so that $f_i(x_i) = A'_{i-1}(x_i) \bmod p$. Thus the verifier always accepts when $A = a \bmod p$.

For completeness, we describe informally, why this protocol is correct when $A \neq a \bmod p$. Suppose that all of the verifier's checks are valid, up to the last check that $f_n(r_n) = A_n$. We show that the last test must fail with high probability. The key to the proof is that for all $i \geq 1$, if the polynomial $f_i(x_i)$ sent by the prover to the verifier at the ith round is not equal to $A'_{i-1}(x_i)$, then with high probability, $f_i(r_i) \neq A'_{i-1}(r_i)$. To see this, note that if two polynomials of degree at most n^k are not equal, they agree at $\leq n^k$ points. Hence, they are very unlikely to agree at a point r_i chosen randomly and uniformly from a range exponential in n. From this, and the definition of $A'_{i-1}(x_i)$, it follows that if $f_i(x_i) \neq A'_{i-1}(x_i)$, then with high probability, $f_{i+1}(x_{i+1}) \neq A'_i(x_{i+1})$. But in order to pass the initial test when $i = 1$, it must be the case that $f_1(x_1) \neq A'_0(x_1)$. Thus for all i, $f_i(x_i) \neq A'_{i-1}(x_i)$ with high probability and in particular, this is true for $i = n$. Finally, if $f_n(x_n)$ is not equal to $A'_{n-1}(x_n)$, then the last check of the verifier, that $f_n(r_n) = A_n$, fails.

We now describe a 1-round 2IPS (P_1, P_2, V), to verify that $A = a \bmod p$. Roughly, the protocol "parallelizes" the above single-prover IPS and duplicates it d times (where d is polynomial in n). The verifier V sends d random sequences, $r_1^j, \ldots, r_n^j$, $1 \leq j \leq d$, to prover I and sends a random subsequence of each of the d sequences to prover II. As in the above protocol, each of the d random sequences specifies a sequence of arithmetic forms with a decreasing number of quantifiers, which we denote by $A_0^j, \ldots, A_n^j$.

1. Uniformly and independently, choose random sequences $\langle r_1^1, \ldots, r_n^1 \rangle, \ldots, \langle r_1^d, \ldots, r_n^d \rangle$ of values in $\mathbf{Z}/p\mathbf{Z}$, where $d = n^2$. Send all of them to prover I.

   Uniformly and independently, choose random values $1 \leq \ell_1, \ldots, \ell_d \leq n$ and send to prover II the sequences $\langle r_1^1, \ldots, r_{\ell_1 - 1}^1 \rangle, \ldots, \langle r_1^d, \ldots, r_{\ell_d - 1}^d \rangle$. If $\ell_i = 1$ then the ith sequence is empty.

2. Receive from provers I and II the sequences of polynomials

   $$\langle f_1^1(x_1), \ldots, f_n^1(x_n) \rangle, \ldots, \langle f_1^d(x_1), \ldots, f_n^d(x_n) \rangle \text{ and}$$
   $$\langle g_1^1(x_1), \ldots, g_{\ell_1}^1(x_{\ell_1}) \rangle, \ldots, \langle g_1^d(x_1), \ldots, g_{\ell_d}^d(x_{\ell_1}) \rangle,$$

   respectively, where each $f_i^j(x_i), g_i^j(x_i) \in \mathbf{Z}/p\mathbf{Z}[x_i]$.

3. For each i, j, $1 \leq i \leq n$, $1 \leq j \leq d$,

   - if the leftmost quantifier of A_{i-1}^j is $\sum$, then if $i > 1$, check that $f_i^j(0) + f_i^j(1) = f_{i-1}^j(r_{i-1}) \bmod p$ and if $i = 1$, that $f_i^j(0) + f_i^j(1) = a \bmod p$.
   - if the leftmost quantifier of A_{i-1}^j is $\prod$, then if $i > 1$, check that $f_i^j(0) f_i^j(1) = f_{i-1}^j(r_{i-1}) \bmod p$ and if $i = 1$, that $f_i^j(0) f_i^j(1) = a \bmod p$.

   Check that $f_n^j(r_n) = A_n^j$, $1 \leq j \leq d$.

4. Finally, check that $f_i^j(x_i) = g_i^j(x_i)$, for $1 \leq i \leq \ell_j$, $1 \leq j \leq d$.

   Accept if and only if all the conditions of steps 3 and 4 are satisfied.

The provers P_1 and P_2 are defined so that $f_i^j(x_i) = g_i^j(x_i) = (A_{i-1}^j)'(x_i), 1 \leq i \leq n$.

The proof of correctness of this protocol is quite different from that of the single prover protocol. In the 2-prover protocol, prover I receives all of the r_i's before sending any of the $f_i(x_i)$'s to the verifier. Hence on each of the d parallel copies, prover I can easily "cheat" by sending the verifier some $f_i(x_i)$

that is not equal to $A'_{i-1}(x_i)$, but agrees with $A'_{i-1}(x_i)$ at r_i *and* passes the verifier's check of step 4. However, such information is hidden from prover II, who, with high probability, may actually be given a string that extends just until the "sneak in" point, i. In this case, prover II's response is not consistent with prover I and the verifier rejects at step 4. However, the probability that the verifier rejects on any fixed copy is small. The main technical contribution of this paper is to show that the verifier rejects on *some* copy with high probability, even though the copies are not independent.

3 Proof of Theorem 1.1

We now give the formal proof that our IPS for the QBF problem is correct. Clearly if $A = a \bmod p$, then the two provers can convince the verifier that this is so, by simply letting $f_i^j(x_i) = g_i^j(x_i) = (A_{i-1}^j)'(x_i)$ for all i.

Suppose that $A \neq a \bmod p$. Fix provers P_1^* and P_2^*. We show that (P_1^*, P_2^*, V) accepts with exponentially small error probability. We first define the notion of a *reasonable* sequence that the verifier sends to P_1^*. Then we show in Lemma 3.1 that a random sequence is reasonable with overwhelming probability. Finally we show in Lemma 3.2 that if V sends a reasonable sequence to P_1^*, then (P_1^*, P_2^*, V) rejects with overwhelming probability.

Let the sequences of polynomials that the verifier receives from prover P_1^* be

$$f_1^j(x_1; \langle r_1^1, \ldots, r_n^1\rangle, \ldots, \langle r_1^d, \ldots, r_n^d\rangle), \ldots, f_n^j(x_n; \langle r_1^1, \ldots, r_n^1\rangle, \ldots, \langle r_1^d, \ldots, r_n^d\rangle), 1 \leq j \leq d,$$

when the verifier sends the sequences $\langle r_1^1, \ldots, r_n^1\rangle, \ldots, \langle r_1^d, \ldots, r_n^d\rangle$ to P_1^*. Similarly, let the sequences of polynomials that the verifier receives from prover P_2^* be

$$g_1^j(x_1; \langle r_1^1, \ldots, r_{\ell_1}^1\rangle, \ldots, \langle r_1^d, \ldots, r_{\ell_d}^d\rangle), \ldots, g_{\ell_j}^j(x_{\ell_j}; \langle r_1^1, \ldots, r_{\ell_1}^1\rangle, \ldots, \langle r_1^d, \ldots, r_{\ell_d}^d\rangle), 1 \leq j \leq d,$$

when the verifier sends the sequences $\langle r_1^1, \ldots, r_{\ell_1-1}^1\rangle, \ldots, \langle r_1^d, \ldots, r_{\ell_d-1}^d\rangle$ to P_2^*. We use this notation to emphasize that the polynomials of each prover may depend on *all* of the values that the prover receives from the verifier.

Fix any $d-1$ sequences $\langle r_1^2, \ldots, r_n^2\rangle, \ldots, \langle r_1^d, \ldots, r_n^d\rangle$. For any $1 \leq i \leq n$ and any $r_1, \ldots, r_{i-1}$, a polynomial $g_i(x_i) \neq (A_{i-1}^1)'(x_i)$ is called a 1-*popular* polynomial if the set of vectors $(i_2, \ldots, i_d), 0 \leq i_2, \ldots, i_d < n$ for which

$$g_i^1(x_i; \langle r_1, \ldots, r_{i-1}\rangle, \langle r_1^2, \ldots, r_{i_2}^2\rangle, \ldots, \langle r_1^d, \ldots, r_{i_d}^d\rangle) = g_i(x_i)$$

has cardinality $\geq n^{d-1-\sqrt{n}}$. For each i, $1 \leq i \leq n$, and $r_1, \ldots, r_{i-1}$, there can be at most $n^{\sqrt{n}}$ 1-popular $g_i(x_i)$. Since the degree of a 1-popular g_i is at most n^k, there can be at most n^k values $r_i \in \mathbf{Z}/p\mathbf{Z}$ for which $g_i(r_i) = (A_{i-1}^1)'(r_i)$. Such a value r_i is called a 1-*exceptional* value for the 1-popular polynomial g_i.

We say that a sequence $\langle r_1, \ldots, r_n\rangle$ is 1-*reasonable* with respect to $\langle r_1^2, \ldots, r_n^2\rangle, \ldots, \langle r_1^d, \ldots, r_n^d\rangle$, if for all i, $1 \leq i \leq n$, r_i is not a 1-exceptional value for any 1-popular g_i, with respect to the given $r_1, \ldots, r_{i-1}$ and $\langle r_1^2, \ldots, r_n^2\rangle, \ldots, \langle r_1^d, \ldots, r_n^d\rangle$. We can similarly define $\langle r_1, \ldots, r_n\rangle$ to be j-reasonable with respect to $\langle r_1^1, \ldots, r_n^1\rangle, \ldots, \langle r_1^{j-1}, \ldots, r_n^{j-1}\rangle, \langle r_1^{j+1}, \ldots, r_n^{j+1}\rangle, \ldots, \langle r_1^n, \ldots, r_n^n\rangle$, by using the appropriate definitions of j-popular and j-exceptional. Finally, we say that $\langle r_1^1, \ldots, r_n^1\rangle, \ldots, \langle r_1^d, \ldots, r_n^d\rangle$ is *reasonable* if for all j, $1 \leq j \leq d$, $\langle r_1^j, \ldots, r_n^j\rangle$ is j-reasonable with respect to the other sequences. We next show that a random sequence $\langle r_1^1, \ldots, r_n^1\rangle, \ldots, \langle r_1^d, \ldots, r_n^d\rangle$ is reasonable with overwhelming probability.

Lemma 3.1 *Suppose the sequence $\langle r_1^1, \ldots, r_n^1\rangle, \ldots, \langle r_1^d, \ldots, r_n^d\rangle$ is chosen randomly and uniformly. Then*

$$Prob[\langle r_1^1, \ldots, r_n^1\rangle, \ldots, \langle r_1^d, \ldots, r_n^d\rangle \text{ is reasonable }] \geq 1 - o(2^{-n/2}).$$

Proof: First, the probability that a random sequence $\langle r_1, \ldots, r_n\rangle$ is 1-reasonable with respect to any fixed sequences $\langle r_1^2, \ldots, r_n^2\rangle, \ldots, \langle r_1^d, \ldots, r_n^d\rangle$ is bounded below by

$$\left(1 - \frac{n^{\sqrt{n}}n^k}{p}\right)^n \geq \left(1 - \frac{n^{\sqrt{n}+k}}{2^n}\right)^n \geq 1 - \frac{n^{\sqrt{n}+k+1}}{2^n}.$$

Thus, a random sequence $\langle r_1^1, \ldots, r_n^1\rangle, \ldots, \langle r_1^d, \ldots, r_n^d\rangle$ is reasonable with probability at least

$$1 - \frac{dn^{\sqrt{n}+k+1}}{2^n} = 1 - o(2^{-n/2}),$$

since $d = n^2$. □

Suppose now that the verifier chooses a reasonable sequence $\langle r_1^1, \ldots, r_n^1\rangle, \ldots, \langle r_1^d, \ldots, r_n^d\rangle$. We show next that the verifier rejects with overwhelming probability.

Let Q^d be the d-dimensional cube of lattice points up to $n-1$ and let the *flats* along any dimension $j, 1 \leq j \leq d$ be the sets of the form $\{ (i_1, .., i_{j-1}, k, i_{j+1}, .., i_d) \mid 0 \leq i_1, \ldots, i_d < n \}$, for $0 \leq k < n$. There is a natural 1-1 correspondence between the vertices of the cube Q^n and the sequence the verifier sends to P_2^*: the vertex $(i_1, \ldots, i_d)$ corresponds to the sequence $\langle r_1^1, \ldots, r_{i_1}^1\rangle, \langle r_1^2, \ldots, r_{i_2}^2\rangle, \ldots, \langle r_1^d, \ldots, r_{i_d}^d\rangle$.

Lemma 3.2 *For each dimension $j, 1 \leq j \leq d$, there is a flat F_j along dimension j and a subset $H_j \subseteq F_j$ of cardinality $\geq n^{d-1}\left(1 - \frac{1}{n^{\sqrt{n}}}\right)$, such that, if the sequence the verifier sends to P_2^* corresponds to a point on H_j, then the verifier rejects.*

Proof: For notational convenience, we set $j = 1$. Clearly the verifier rejects if some condition of step 3 of the protocol is not satisfied, so suppose that all are satisfied. In the sequence $\langle f_1^1(x_1), \ldots, f_n^1(x_n)\rangle$ of P_1^*, there exists ℓ_1, $1 \leq \ell_1 \leq n$, such that

$$f_{\ell_1}^1(x_{\ell_1}) \neq (A_{\ell_1-1}^1)'(x_{\ell_1}) \quad \text{and} \quad f_{\ell_1}^1(r_{\ell_1}^1) = A_{\ell_1}^1. \tag{1}$$

This is because we are assuming that all conditions of step 3 of the protocol are satisfied. Consider the response of P_2^* on the flat $F_1 = \{ (\ell_1 - 1, i_2, \ldots, i_d) \mid 0 \leq i_2, \ldots, i_d < n \}$. Since $\langle r_1^1, \ldots, r_n^1\rangle$ is 1-reasonable with respect to the other sequences, $r_{\ell_1}^1$ is not a 1-exceptional value for any 1-popular polynomial. Hence by (1), $f_{\ell_1}^1$ is not a 1-popular polynomial on F_1.

Let H_1 be the subset of vectors $(\ell_1 - 1, i_2, \ldots, i_d)$ of F_1 such that

$$g_{\ell_1}^1(x_{\ell_1}; \langle r_1^1, \ldots, r_{\ell_1-1}^1\rangle, \ldots, \langle r_1^d, \ldots, r_{i_d}^d\rangle) \neq f_{\ell_1}^1(x_{\ell_1}).$$

Then H_1 must have cardinality $\geq n^{d-1} - n^{d-1-\sqrt{n}} = n^{d-1}\left(1 - \frac{1}{n^{\sqrt{n}}}\right)$. Since P_2^* disagrees with P_1^* on all sequences of values corresponding to a point in H_1, the verifier rejects at step 4 on all such sequences. □

Thus, on the reasonable sequence $\langle r_1^1, \ldots, r_n^1\rangle, \ldots, \langle r_1^d, \ldots, r_n^d\rangle$, the probability that (P_1^*, P_2^*, V) rejects is $|\bigcup_{j=1}^d H_j|/n^d$.

Now

$$\begin{aligned} |\bigcup_{j=1}^{d} H_j| &\geq |\bigcup_{j=1}^{d} F_j| - \sum_{j=1}^{d} |F_j - H_j| \\ &\geq n^d \left(1 - \left(1 - \frac{1}{n}\right)^d\right) - dn^{d-1-\sqrt{n}} \\ &= n^d(1 - o(2^{-\sqrt{n}})), \end{aligned}$$

since $d = n^2$. Hence, (P_1^*, P_2^*, V) accepts with probability at most 1 if V initially chooses a sequence that is not reasonable and with probability $o(2^{-\sqrt{n}})$ otherwise. Moreover, the probability of choosing a sequence that is not reasonable is at most $o(2^{-n/2})$. Hence the probability that (P_1^*, P_2^*, V) accepts is $o(2^{-\sqrt{n}})$. This completes the proof of Theorem 1.1.

4 Conclusions and Open Problems

The main result of this paper is that for any constant ϵ, any language accepted by an unbounded round IPS has a bounded round, 2-prover 2IPS that has error probability ϵ. The following question is still open, however. If a 2IPS accepts an input w with probability p, what is the probability that the d-product of the 2IPS accepts w? In [6], Cai, Condon and Lipton provide bounds on this probability for restricted types of 2IPS's, showing for example that as $d \to \infty$, the probability that the d-product 2IPS accepts $w \to 0$. However, even for these restricted types of 2IPS's, it is not known if this probability is strictly decreasing as d increases. Feige and Lovász [10] use techniques based on quadratic programming to bound the probability that the d-product accepts w for other special cases, but the general problem is still open.

Acknowledgement

The first author thanks Lance Fortnow and Uriel Feige for pointing out that the error bound in Section 3 can be made exponentially small. We also thank the anonymous referees for many valuable comments on the presentation.

References

[1] L. Babai, Trading group theory for randomness, Proceedings of the 17th Annual Symposium on the Theory of Computing, May 1985, 421-429.

[2] L. Babai and S. Moran, Arthur-Merlin games: A randomized proof system, and a hierarchy of complexity classes. *Journal of Computer and System Sciences*, **36** (1988), 254-276.

[3] L. Babai, L. Fortnow and C. Lund, Nondeterministic exponential time has two-prover interactive protocols, *Computational Complexity* **1** (1990), 3-40.

[4] D. Beaver and J. Feigenbaum, Hiding Instances in Multioracle Queries, Proceedings of the 7th Symposium on Theoretical Aspects of Computer Science, Lectures Notes in Computer Science 415 February, 1990, 37-48.

[5] M. Ben-Or, S. Goldwasser, J. Kilian and A. Wigderson, Multi-prover interactive proofs: how to remove intractability, Proceedings of the 20th Annual Symposium on the Theory of Computing, May, 1988, 113-121.

[6] J. Cai, A. Condon and R.J. Lipton, Playing games of incomplete information, Proceedings of the Symposium on Theoretical Aspects of Computer Science, Lecture Notes in Computer Science, February 1990. To appear in *Theoretical Computer Science*, 1992.

[7] J. Cai, A. Condon and R.J. Lipton, On bounded round multi-prover interactive proof systems, Proceedings of Fifth Annual Conference on Structure in Complexity Theory, June 1990, 45-54.

[8] L. Fortnow, Complexity-Theoretic Aspects of Interactive Proof Systems, Ph. D. Thesis, Tech Report #MIT/LCS/TR-447, MIT.

[9] U. Feige, On the success probability of two provers in one-round proof systems, Proceedings of Sixth Annual Conference on Structure in Complexity Theory, July 1991, 116-123.

[10] U. Feige and L. Lovász, Two-prover one-round proof systems: their power and their problems, To appear in the Proceedings of the 24th Annual Symposium on the Theory of Computing, May 1992.

[11] L. Fortnow, J. Rompel and M. Sipser, On the power of multi-prover interactive protocols, Proceedings of the Third Annual Conference on Structure in Complexity Theory, June 1988, 156-161.

[12] S. Goldwasser, S. Micali and C. Rackoff, The knowledge complexity of interactive proof systems, *SIAM J. Comput.* **18** (1989) 186-208.

[13] S. Goldwasser and M. Sipser, Public coins vs. private coins in interactive proof systems, *Randomness and Computation*, Volume 5 of JAI Press, Greenwich, 73-90, 1989.

[14] D. Lapidot and A. Shamir, Fully Parallelized Multi Prover Protocols for NEXP-time, Proceedings of the 32nd Annual IEEE Symposium on the Foundations of Computer Science, 1991, 13-18.

[15] R.J. Lipton, New directions in testing, *Distributed Computing and Cryptography, DIMACS Series on Discrete Mathematics and Theoretical Computer Science*, 2 (1991), American Mathematical Society, 191-202.

[16] C. Lund, L. Fortnow, H. Karloff and N. Nisan, Algebraic Methods for Interactive Proof Systems, Proceedings of the 30th IEEE Symposium on the Foundations of Computer Science, October 1990, 2-10.

[17] A. Shamir, IP=PSPACE, Proceedings of the 30th IEEE Symposium on the Foundations of Computer Science, October 1990, 11-15.

[18] S. Toda, PP is as hard as the polynomial time hierarchy, *SIAM Journal on Computing*, **20(5)**:865-877, 1991.

[19] L. Valiant, The complexity of computing the permanent, *Theoretical Computer Science*, **8**, (1979), 189-201.

[20] Oleg Verbitsky, MIP is recognizable in constantly many rounds, Manuscript, Moscow State University, Moscow, Russia 117899, 1991.