Interactive Proofs with Space Bounded Provers

Joe Kilian *
NEC Research Institute
Princeton, NJ 08540

Ronitt Rubinfeld †
Department of Computer Science
Princeton University
Princeton, NJ 08544

Abstract

Recent results in interactive proof systems [12][13] [1] seem to indicate that it is easier for a prover in a single prover interactive proof system to cheat the verifier than it is for a prover in a multiple prover interactive proof system. We show that this is not the case for a single prover in which all but a fixed polynomial of the prover's space is erased between each round. One consequence of this is that any multiple prover interactive protocol in which the provers need only a polynomial amount of space can be easily transformed into a single prover interactive protocol where the prover has only a fixed polynomial amount of space. This result also shows that one can easily transform checkers [5] into adaptive checkers [7] under the assumption that the program being checked has space bounded by a fixed polynomial.

1 Introduction

Recent results in complexity theory have shown that IP=PSPACE [12][13] and that MIP=NEXPTIME [1]. This gives reason to believe that there is a significant difference in the power of a single prover interactive proof system versus the power of multiple prover interactive proof systems, i.e. that since the multiple provers are constrained to be consistent with each other, they cannot cheat the verifier as easily, and thus more difficult languages can have such proof systems. It has been shown that the same set of

*Supported by an NSF fellowship while at MIT.
†Supported by DIMACS, NSF-STC88-09648.

J. Feigenbaum (Ed.): Advances in Cryptology - CRYPTO '91, LNCS 576, pp. 225-231, 1992.

languages is accepted by the following three types of interactive proof systems: multiple prover systems, single prover systems where the prover is constrained to answer according to functions that are fixed in advance, and single prover systems in which the memory of the prover gets wiped out between each question (i.e. the prover has no memory of the conversation) [9][3].

One might conjecture that allowing the prover in a single prover system to have partial memory of the conversation could increase his ability to cheat substantially, thus decreasing the power of the system. We show that this is *not* the case: that if there is an interactive protocol against a prover that does not remember anything between questions, then for any s it can be modified into an interactive protocol that works against a prover that remembers s bits between questions. The running time of the new protocol is polynomial in the running time of the old protocol and s. Note that the IP prover need only remember the history of the conversation between rounds, which is a polynomial number of bits (however, here the polynomial is chosen after the protocol is decided upon rather than before).

This result has the following application to cryptography: it shows that two prover protocols for identification implemented by two credit cards can be implemented by a single credit card, as long as the credit card is guaranteed to have a limited amount of memory.

The results in this paper apply to program result checking as well [5] [7]. They show how to transform a checker that works assuming that the program is a *fixed* function, into an adaptive checker which instead assumes only that the program has polynomial space at its disposal. This is interesting because it allows one to assume that the checker works even if hardware faults evolve over time, or in the case that the software is written such that running the program on certain inputs may have unintended side effects on the program's future behavior.

A somewhat related result in [10] shows how to self-correct [6] some functions from space bounded tested programs. This result applies only to functions which are polynomials, and does not show how to give program result checkers for those functions.

2 Definitions

We informally describe the following modifications of definitions of interactive proof systems and IP given in [11]:

DEFINITION 2.1 *A* s-space, t-round Interactive Protocol (A, B) *is a pair of turing machines (TM)* (A, B) *which share an input tape (read only). Both have a private read/write work tape and read-only random tape. There are two communication tapes: one which* B *has write-only access to, and* A *has read-only access to, and one which* A *has write-only access to, and* B *has read-only access to. We think of the first tape as containing messages sent to, or "questions" asked of* A*, and the second as messages sent to, or "answers" to* B*. The machines take* t *turns being active with* B *going first. Before each message to* A*, all but the first* s *bits of* A*'s private work tape are erased.* A *is computationally unbounded and* B *is polynomial time bounded.*

DEFINITION 2.2 *Let* $L \subset \{0,1\}^*$*. We say that* L *has a* s-space t-round interactive proof system (IPS) *if there exists a TM* V *such that*

1. *There is a TM* P *s.t.* (P, V) *is a* s*-space* t*-round interactive protocol and for all* $x \in L$ *s.t.* $|x|$ *is sufficiently large,* $Pr[V accepts] > 2/3$ *(when probabilities are over coin tosses of* P *and* V*).*
2. *For all TM's* P' *s.t.* (P', V) *is a* s*-space* t*-round interactive protocol and for all* $x \notin L$ *s.t.* $|x|$ *is sufficiently large,* $Pr[V accepts] < 1/3$ *(when probabilities are over coin tosses of* P' *and* V*).*

We say that (P, V) *is a* s-space t-round interactive proof system for L.

Define $IP(s,t) = \{L | L$ *has an* s*-space* t*-round interactive proof system*$\}$

We may think of P as deterministic, giving optimal answers to maximize the probability that V accepts.

3 Main Theorem

Theorem 1 *If* $L \in IP(0,t)$*, then for all* s*,* $L \in IP(s, O(st))$.

Proof: We show that if there is a 0-space, t-round interactive protocol for L then we can construct an s-space $O(st)$-round interactive protocol for L.

The 0-space, t-round protocol for L can be run $O(s)$ times in order to reduce the error probability from $\frac{1}{3}$ to $\frac{1}{6}2^{-s}$. Call the resulting low error protocol C_P^V, and let

$C_P^V(w,r) = (x_1, y_1, \ldots, x_m, y_m)$ denote the m-round conversation between verifier V and prover P, where w is the input, r is the random string used by the verifier, the x_i's are the "questions" sent by the verifier to the prover, and the y_i's are the "answers" sent by the prover to the verifier (note that m is $O(st)$).

Let $\Phi(w, r, C_P^V(w,r))$ be the function which the verifier evaluates after the conversation in order to decide whether to accept or reject w.

Let $\tilde{P}$ denote any prover that can remember s bits between questions.

We are now ready to present the protocol:

s-space $O(st)$-round interactive protocol:
On input w:

1. Run protocol $C_{\tilde{P}}^V(w,r) = (x_1, y_1, \ldots, x_m, y_m)$. If $\Phi(w, r, C_{\tilde{P}}^V(w,r)) =$ "REJECT" then reject and halt.
2. Do $3m$ times:
   Pick $i \in_R [1, m]$
   Verifier asks question x_i and receives answer $\hat{y}_i$.
   If $y_i \neq \hat{y}_i$ then reject and halt.
3. Accept w.

Proof of Correctness of s-space $O(st)$-round protocol: If $w \in L$, then it is obvious from our assumption that $L \in IP(0,t)$ that there is a prover P which can only remember s bits after every question such that $Pr_r[\Phi(w, r, C_P^V(w,r)) = \text{"ACCEPT"}] \geq 2/3$.

To prove the theorem for the case when w is not in L, we need to show that no prover that can remember s bits is likely to fool the verifier into accepting L.

We first note that any space s prover $\tilde{P}$ can be viewed as a collection of 2^s functions in the following manner: consider a deterministic finite state automaton with 2^s states, where each state i is labeled with function P^i. The transitions between functions are labeled by all the possible questions that a verifier could ask of the prover. Then the prover is at one of the 2^s states, and whenever the verifier asks a question, the prover answers the question according to the function which labels the state, and goes to a new state according to the transition function applied to the current state and the question just asked by the verifier.

For fixed i, we say that r is *P^i-bad for w* if $\Phi(w, r, C_{P^i}^V(w,r)) =$ "ACCEPT" (where the prover P^i is the prover which answers according to function P^i). Since $C_{P^i}^V$ is a

0-space interactive protocol for L with error $\leq \frac{1}{6}2^{-s}$, we know that r is P^i-bad with the same probability. We say that r is $\tilde{P}$-*bad for* w if there is an i such that r is P^i-bad for w. Since there are only 2^s P^i's, $Pr[r \text{ is } \tilde{P} - \text{bad for } w] \leq 2^s \cdot \frac{1}{6}2^{-s} = \frac{1}{6}$.

For each r, one of the following three cases must hold

1. $\Phi(w, r, C_{\tilde{P}}^V(w, r)) =$ "REJECT", in which case the verifier rejects.
2. r is $\tilde{P}$-bad for w. By the above reasoning, this case happens with probability $\leq 1/6$.
3. $\Phi(w, r, C_{\tilde{P}}^V(w, r)) =$ "ACCEPT", but r is not $\tilde{P}$-bad for w. Thus for all i,

$$\Phi(w, r, C_{P_i}^V(w, r) = (a_1, b_1), \ldots, (a_m, b_m)) = \text{"REJECT"}.$$

Then for all i, there is a j such that $x_j = a_j$ but $y_j \neq b_j$ (since the conversations cannot be the same, and the verifier follows the same algorithm, the first difference must come from the prover). Therefore, no matter which state the prover is in during a loop of Step 2 of the protocol, the probability that a question is asked for which the answer $y_j \neq \hat{y}_j$ is at least $1/m$. After $3m$ times, the probability that the verifier will not reject is at most e^{-3}.

Thus, if w is not in L, the verifier will reject with probability at least $1 - \frac{1}{6} - e^{-3} \geq 2/3$.

■

4 Bounding the Power of an s-space Prover.

The transformation used in the proof of Theorem 1 works only for deterministic provers, since a legitimate probabilistic prover might cause the verifier to reject in Step 2 by giving inconsistent answers to its questions. One can replace any probabilistic prover with an "optimal" deterministic prover, but this new prover may have much greater computational requirements. Hence, such a simple fix will not allow us to carry over our results to program result checking.

However, we can use the general idea used in the proof of Theorem 1 to directly bound the advantage of an s-space bounded prover over a 0-space bounded prover. Our theorem is as follows:[1]

[1] This result was independently discovered by Lund (personal communication).

Theorem 2 *Suppose that in an interactive protocol (P, V) the prover's memory is partially erased at most t times. Let P_s be a prover that is allowed to remember s bits between partial erasures, and let P_0 be an optimal prover that is not allowed to remember any bits between erasures. Then, for all x,*

$$\frac{Pr[(P_s, V) \text{ accepts } x]}{Pr[(P_0, V) \text{ accepts } x]} \leq 2^{st}.$$

Proof: We assume without loss of generality that P_s is deterministic, and denote by p the probability that (P_s, V) accepts x. We now construct a 0-space bounded prover P_0 such that (P_0, V) will accept x with probability at least $2^{-st}p$. P_0 works exactly as does P_s, except that whenever its memory is (totally) erased it sets the first s bits of its memory to a random s-bit string.

Suppose that (P_s, V) accepts when V uses r as its random input. It suffices to show that (P_0, V) will accept with probability 2^{-st} when V uses r as its random input. Let S_i^r denote the contents of the first s bits of P_s's memory after the ith (partial) memory erasure. With probability 2^{-st}, it will be the case that for each i, $1 \leq i \leq t$, P_0 will fill its memory with S_i^r after the ith memory erasure. Whenever this happens, the behavior of P_0 will be identical to that of P_s, and V will accept. Thus, (P_0, V) will accept with probability at least 2^{-st} whenever (P_s, V) accepts, and the theorem follows. ■

Thus, any protocol which achieves a sufficiently low probability of error, using sufficiently few memory erasures, is automatically robust against an s-space bounded prover, without modification. Therefore, given an interactive proof system robust against 0-space bounded provers, using only t memory erasures, one needs only to reduce the error probability to less than $\frac{1}{3} \cdot 2^{-st}$, while preserving the total number of memory erasures.

5 Acknowledgments

We would like to thank Dick Lipton for suggesting the problem. We would also like to thank Uri Feige, Shafi Goldwasser, Diane Hernek, Hugo Krawczyk and Mike Luby for very extensive and helpful discussions on this subject, and for helpful comments on the writeup.

References

[1] Babai, L., Fortnow, L., Lund, C., "Non-Deterministic Exponential Time has Two-

Prover Interactive Protocols", Technical Report 90-03, University of Chicago, Dept. of Computer Science. Also in *Proceedings of the 31st Annual Symposium on Foundations of Computer Science,*, 1990.

[2] Ben-Or, M., Goldwasser, S., Kilian, J., and Wigderson, A., "Efficient Identification Schemes Using Two Prover Interactive Proofs", *Advances in Cryptology - CRYPTO '89*, Springer-Verlag.

[3] Ben-Or, M., Goldwasser, S., Kilian, J., and Wigderson, A., "Multi-Prover Interactive Proofs: How to Remove Intractability Assumptions", *Proc. 20th ACM Symposium on Theory of Computing*, 1988, pp. 113-131.

[4] Blum, M., "Designing programs to check their work", Submitted to *CACM*.

[5] Blum, M., Kannan, S., "Program correctness checking ... and the design of programs that check their work", *Proc. 21st ACM Symposium on Theory of Computing*, 1989.

[6] Blum, M., Luby, M., Rubinfeld, R., "Self-Testing/Correcting Programs with Applications to Numerical Problems", *Proc. 22nd ACM Symposium on Theory of Computing*, 1990.

[7] Blum, M., Luby, M., Rubinfeld, R., "Program Result Checking against Adaptive Programs and in Cryptographic Settings", *Distributed Computing and Cryptography*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol. 2, 1991.

[8] Feige, U., "NEXPTIME has Two-Provers One-Round Proof Systems With Exponentially Small Error Probability", Manuscript.

[9] Fortnow, L., Rompel, J., Sipser, M., "On the Power of Multi-Prover Interactive Protocols", Proc. 3^{rd} Structure in Complexity Theory Conference, 1988, pp. 156-161.

[10] Gemmell, P., Lipton, R., Rubinfeld, R., Wigderson, A., "Self-Testing/Correcting for Polynomials and for Approximate Functions", *Proceedings of 23rd ACM STOC*, 1991.

[11] Goldwasser, S., Micali, S., Rackoff, C., The Knowledge Complexity of Interactive Proof Systems", *SIAM J. Comput.*, 18(1),1989, pp. 186-208.

[12] Lund, C., Fortnow, L., Karloff, H., Nisan, N., "Algebraic Methods for Interactive Proof Systems", *Proceedings of the 31st Annual Symposium on Foundations of Computer Science,*, 1990.

[13] Shamir, Adi, "IP=PSPACE", *Proceedings of the 31st Annual Symposium on Foundations of Computer Science, 1990.*