Private Coins versus Public Coins in Interactive Proof Systems

Shafi Goldwasser*

Computer Science Department
MIT

Michael Sipser**

Computer Science Department
University of California at Berkeley
and
Mathematics Department
MIT

Abstract

An interactive proof system is a method by which one party of unlimited resources, called the *prover,* can convince a party of limited resources, call the *verifier,* of the truth of a proposition. The verifier may toss coins, ask repeated questions of the prover, and run efficient tests upon the prover's responses before deciding whether to be convinced. This extends the familiar proof system implicit in the notion of NP in that there the verifier may not toss coins or speak, but only listen and verify. Interactive proof systems may not yield proof in the strict mathematical sense: the "proofs" are probabilistic with an exponentially small, though non-zero chance of error.

We consider two notions of interactive proof system. One, defined by Goldwasser, Micali, and Rackoff [GMR] permits the verifier a coin that can be tossed in *private,* i.e., a secret source of randomness. The second, due to Babai, [B] requires that the outcome of the verifier's coin tosses be *public* and thus accessible to the prover.

Our main result is that these two systems are equivalent in power with respect to language recognition.

The notion of interactive proof system may be seen to yield a probabilistic analog to NP much as BPP is the probabilistic analog to P. We define the *probabilistic, nondeterministic, polynomial time Turing machine* and show that it is also equivalent in power to these systems.

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

© 1986 ACM 0-89791-193-8/86/0500/0059 $00.75

1. Introduction

In this century, the notions of proof and computation have been formalized and understood. With the arrival of complexity theory, the notion of what is efficiently provable became of interest. The class *NP* captured this notion, containing those languages for which proofs of membership can be verified by a deterministic polynomial time Turing machine. We can view NP as a proof-system consisting of two communicating Turing machines: the *prover* who guesses the proof

* Research supported in part by NSF Grant 8509905 DCR.

** Research supported in part by NSF Grant MCS-8304769 and Air Force Grant AFOSR-82-0326.

and the polynomial time deterministic *verifier*, who checks the correctness of the proof.

Randomization has been recognized to be a fundamental ingredient in defining what is efficiently computable (e.g RP, BPP, RNC). In this paper, we seek to understand how randomization affects the definition of what is efficiently provable.

A conventional deterministic NP verifier does not accept statistical evidence as a convincing argument, regardless of how overwhelming it may be. As a consequence, the kind of languages contained in NP are precisely those whose proofs of membership can be fully put down in writing and shown to others. The verifier does not actively participate in the proof process or interact with the prover in any way. It suffices for the prover to speak and the verifier to listen.

Randomization and interaction are essential ingredients of two recent formalizations of the concept of an efficient proof system. One formalization is due to Babai [B] and the other to Goldwasser, Micali and Rackoff [GMR]. Both definitions would collapse to NP if no coins were flipped.

Interactive Proof Systems

In defining what they called *interactive proof systems*, Goldwasser, Micali and Rackoff's intent was to make as general a definition as possible of what is provable to a probabilistic verifier willing to accept statistical evidence. Their broader goal was to define the concept of the "knowledge" communicated during a proof.

An *interactive proof system* consists of a prover with unlimited computation power and a probabilistic polynomial time verifier who receive a common input x. The prover and the verifier can exchange messages back and forth for a polynomial in the length of x number of times. There are no restrictions on how the verifier may use his coin tosses: he can toss coins, perform any polynomial time computations on them and send the outcome of the computation to the prover. In particular, he need not show the outcome of the coins to the prover.

The secrecy of the verifier's coin tosses seemed essential to certain examples of interactive proof systems. The most notable is a recent result of Goldreich, Micali and Wigderson [GMW] showing an interactive proof-system for *the graph non-isomorphism problem*. This is somewhat remarkable in light of the fact that graph non-isomorphism is not known to be in NP. We sketch this example in section 2.1.

The interactive proof system (IP) defines a hierarchy of languages. Namely, L is in $IP[k]$ if there exists a k-move (k alternations of message exchanges between prover and verifier with the verifier sending the first message) interactive proof system such that: for every input $x \in L$, the probability that the verifier accepts is greater than 2/3, and for every input x not in L, even against an optimal prover, the probability that the verifier accepts is less than 1/3.

Arthur-Merlin Games: An Interactive Proof System with a Public Coin

Babai's formalization of efficient proof system attempts to capture the smallest class of languages extending NP, for which statistical proofs of membership exist. The primary motivation was to place the *matrix group non-membership* and *matrix group order* problems in a complexity class "just above NP". His proof-system, presented as a game, consists of a powerful prover (capable of optimal moves) called *Merlin*, and a probabilistic polynomial time verifier called *Arthur* which receive a common input x. Merlin wins the game if he can make Arthur accept x. Arthur and Merlin alternate exchanging messages back and forth for at most a polynomial in the length of x times. At the end of the interaction, Arthur decides whether to accept or reject (i.e., whether Merlin won or lost).

The difference between the Arthur-Merlin proof system and the GMR proof system is in the restricted way that Arthur is allowed to use his coin tosses during the game. Arthur's moves consist merely of tossing coins and sending their outcomes to Merlin. Thus the Arthur-Merlin game is a special case of an interactive proof system.

The Arthur-Merlin games define a hierarchy of complexity classes, in a manner similar to IP. We say L is in $AM[k]$ if there exists an Arthur-Merlin k-move game (i.e., k alternating message exchanges between Arthur and Merlin, Arthur sending first) such that for every input $x \in L$, the probability that Arthur accepts x is greater than 2/3; and for every input x not in L, the probability that an optimal Merlin wins is less than 1/3.

The elegant simplicity of the definition of the Arthur-Merlin game facilitates additional results. Babai showed that for every constant k, AM[k] collapses to AM[2]. This in turn is a subset of both Π_2^P and nonuniform-NP. The relative power of proof systems with a bounded and an unbounded number of exchanged messages remains an interesting open question.

In this paper we prove the equivalence of these two types of interactive proofs with respect to language recognition. As a consequence the above results extend to IP.

Our Result

Let Q denote a polynomial. Let $IP[Q]$ (and $AM[Q]$) denote those languages L for which there exists a Q-move interactive proof system (and Q-move Arthur-Merlin proof-system respectively). Namely, $Q(|x|)$ message exchanges between the prover and the verifier are allowed on input x.

In this paper, we show that for any polynomial Q,

$$IP[Q] \subseteq AM[Q+2]$$

i.e., the GMR proof system is as powerful as the Babai proof system.

2. Examples and Related Work

2.1. An Example of An Interactive Proof System

Goldreich, Micali and Wigderson [GMW] have recently demonstrated the following interactive proof system for the graph non-isomorphism problem.

Let $NONISO = \{(G_0, G_1)$ such that the graph G_1 is not isomorphic to the graph $G_0\}$.

Theorem (GMW): $NONISO \in$ IP.

proof: Let the prover and verifier receive as input two n node graphs G_0 and G_1 on vertices V. The following steps 1 and 2 get executed n times in parallel.

> **step 1**: The verifier flips a fair coin to choose $c \in \{0,1\}$ and a random permutation π of V. The verifier then computes $R = \pi(G_c)$ and sends R to the prover.
>
> **step 2**: The prover tells the verifier whether c=0 or 1.
>
> **final step**: if the prover makes a mistake in step 2 in guessing what c is, the verifier rejects, otherwise he accepts.

If the two input graphs G_1 and G_2 are not isomorphic to each other, then there exists a prover who can distinguish the case that R is isomorphic to G_0 from the case that R is isomorphic to G_1, and thus can always tell correctly in step 2 of the protocol whether $c=0$ or $c=1$, and make the verifier accept.

On the other hand, if G_0 is isomorphic to G_1, then by the randomness of the permutation π, R is as likely to be $\pi(G_0)$ as it is to be $\pi(G_1)$. The prover who does not know c will err in step 2 of the protocol with probability 1/2. QED

Clearly, the secrecy of the coin is essential to this protocol.

One consequence of this result, combined with Babai's and ours, is that the graph non-

isomorphism problem has polynomial-size, non-deterministic circuits.

Several other interactive proof systems for number theoretic problems and the matrix group membership problem appear in [GMR,B].

2.2. Related Work

The difference between IP and AM games is analogous to that between alternation [CKS] and alternation with partial information [R]. In the case of alternation both players play optimally subject to their knowledge and only the referee who determines the outcome is required to be polynomial time bounded. Condon and Ladner [CL] describe this connection in a general setting.

Other relevant results concerning interactive proof systems appear in [H] and [F]. Prior to our result, Johan Hastad [H] showed that the union $\bigcup_{k>0} IP[k]$ is contained in Σ_3. Paul Feldman [F] showed that the prover in an interactive proof system with a polynomial number of interactions need not be more powerful than a PSPACE machine. Boppana and Hastad [BH] showed that if co-NP $\subseteq$ AM then for any i, $\Sigma_i^P \subseteq AM$. This and our result show that the polynomial hierarchy collapses to Σ_2^P if graph isomorphism is NP-complete.

Fortnow and Sipser [FS] have shown that there is an oracle F such that co-$NP^F \not\subseteq IP^F$.

Other works related to the study of randomized proof systems appear in [Pa] and [ZF]. In Papadimitriou's "Games Against Nature", the verifier is also a probabilistic polynomial time machine which flips coins and presents them to an the prover which is a capable of optimal moves. The difference is that the probability of convincing the verifier need not be bounded away from 1/2. This apparently affects the strength of the system as Papadimitriou's games are as powerful as PSPACE.

Furer and Zachos [ZF], in a work investigating the robustness of probabilistic complexity classes, introduce a framework of probabilistic existential and universal quantifiers and prove several combinatorial lemmas about them. The AM complexity classes can be formulated in terms of these special quantifiers.

2.3. Connections with Cryptography

An interactive proof system can be viewed as a model for proving the correctness of two party cryptographic protocols [GMR]. The prover and verifier in an interactive proof system model the two participants in a cryptographic protocol with one exception: the cryptographic prover is not all powerful, but a probabilistic polynomial time machine with a secret unknown to the verifier.

A key property of cryptographic protocols is the amount of "knowledge" released to the verifier during the execution of the protocol. Very informally, we say that a prover in an interactive proof system releases "zero-knowledge" if even a devious verifier can learn no more than the validity of the assertion being proved. In a "secure" protocol the prover has this property. (see [GMR] for precise definitions). In [GMW] it has been shown, that if one-way functions exist then every language in NP has a zero-knowledge proof protocol.

Using this and our transformation from private-coin to public-coin protocols, Ben-or [Be] has very recently shown that all languages in IP have zero-knowledge, public-coin protocols, given the existence of one-way functions. Implications of this are not yet fully clear, but it may be that public coins are adequate for secure cryptographic protocols.

3. Definitions

We represent the *verifier* and the *prover* of an interactive proof system as two functions V and P.

Definition: An *Interactive proof protocol* is given by two functions:

$$V: \Sigma^* \times \Sigma^* \times \Sigma^* \rightarrow \Sigma^* \cup \{\textbf{accept}, \textbf{reject}\}$$

$$P: \Sigma^* \rightarrow \Sigma^*$$

Let s_i denote the concatenation of i pairs of messages, $s_i = \#x_1\#y_1\# \cdots \#x_i\#y_i$. We write $V(w,r,s_i)=x_{i+1}$ to mean that V on input w, with random sequence r, and current message stream s produces next message x_{i+1}. We say $P(s_i\#x_{i+1})=y_{i+1}$ to mean that P produces next message y_{i+1} given current message stream $s_i\#x_{i+1}$. The exchange of a single pair of messages is called a *round.*

For a given input w and random sequence r we say

$$(V^*P)(w,r) \text{ accepts}$$

if there exists a message stream $s = \#x_1\#y_1\# \cdots \#x_l\#y_l$ such that $V(w,r,s)=$**accept**, and for each $i<l$, $V(w,r,s_i)=x_{i+1}$ and $P(s_i\#x_{i+1})=y_{i+1}$.

Let us assume for simplicity that there is a function l such that for inputs w of length n, V will only accept if the length of r is $l(n)$. Then we write

$$\Pr[(V^*P)(w) \text{ accepts}]$$

to mean $\Pr[(V^*P)(w,r)$ accepts$]$ for r chosen randomly from $\Sigma^{l(|x|)}$.

Further we let

$$\Pr[V(w) \text{ accepts}]$$

denote $\max_P \Pr[(V^*P)(w)$ accepts$]$.

Let the language of the verifier, $L(V)=$

$$\{w: \Pr[V(w) \text{ accepts}] > 1/2\}$$

Say V has error probability e if for all $w \in \Sigma^*$:

1) if $w \in L(V)$, $\Pr[V(w)$ accepts$] \geq 1-e$
2) if $w \notin L(V)$, $\Pr[V(w)$ accepts$] \leq e$

For $W \subseteq \Sigma^*$, we say $W \in$ IP if there is a polynomial time verifier V with error probability 1/3 accepting W. As we shall see later, the class IP is unaffected if we substitute e for 1/3, where $2^{-\text{poly}(n)} \leq e \leq 1/2 - 2^{-\text{poly}(n)}$.

Definition: An *Interactive proof protocol with public coin* is defined as above with the following difference. The random input r is considered to be the concatenation of l strings $r = r_1 r_2 \cdots r_l$ where l is the number of rounds and V is restricted to produce r_i as it's i^{th} message, i.e., for $i \leq l$, $V(w,r,s_i)=r_i$ or **accept** or **reject**.

This notion is essentially identical to that of the Arthur-Merlin game defined by Babai in [B]. Following his terminology we say that for $W \subseteq \Sigma^*$, $W \in$ AM(poly) if $W \in$ IP as above and the interactive proof protocol uses a public coin. We refer to an Arthur-Merlin game as an *A-M* protocol.

For polynomial Q, say $W \in IP[Q(n)]$ if $W \in IP$ with a verifier which never sends more than $Q(n)$ messages for inputs of length n. Similarly define $AM[Q(n)]$.

4. The equivalence of public vs. private coins

4.1. Approximate lower bound lemma

This lemma, an application of Carter-Wegman universal hashing [CW], due to Sipser [Si], plays a key role in our proof of equivalence. Its application to approximate lower bounds was first given by Stockmeyer [St]. Its application in Arthur-Merlin protocols first appears in Babai [B].

Definition: Let D be a $k \times b$ Boolean matrix. The *linear function* $h_D: \Sigma^k \rightarrow \Sigma^b$ is given by $h_D(x) = x \cdot D$ using ordinary matrix multiplication modulo 2. A *random linear function* is obtained by selecting the matrix D at random. If $H = \{h_1, \ldots, h_l\}$ is a collection of functions, $C \subseteq \Sigma^k$, and $D \subseteq \Sigma^b$ then $H(C)$ denotes $\bigcup h_i(C)$, and $H^{-1}(D)$ denotes $\bigcup h_i^{-1}(D)$. Let $|C|$ denote the cardinality of C.

Lemma: Given $b,k,l>0$, $l>\max(b,8)$, and $C\subseteq\Sigma^k$. Randomly select l linear functions $H=\{h_1,\ldots,h_l\}$, $h_i:\Sigma^k\to\Sigma^b$ and l^2 strings $Z=\{z_1,\ldots,z_{l^2}\}\subseteq\Sigma^b$. Then

1. If $b=2+\lceil\log|C|\rceil$ then
   a) $\Pr[|H(C)| \geq |C|/l] \geq 1-2^{-l}$
   b) $\Pr[H(C)\cap Z \neq \emptyset] \geq 1-2^{-l/8}$
2.
   a) $|H(C)| \leq l|C|$
   b) If for $d>0$, $|C| \leq 2^b/d$ then: $\Pr[H(C)\cap Z\neq\emptyset] \leq l^3/d$

Proof 1a: Since $2^b\geq 4|C|$ the following chain of statements are easily verified. Let $(h_i(x))^j$ denote the j^{th} bit of the string $h_i(x)$. Fix $x,y\in\Sigma^k$, $x\neq y$, $i,j>0$, except where quantified.

$\Pr[(h_i(x))^j=(h_i(y))^j]=1/2$

$\Pr[h_i(x)=h_i(y)]=2^{-b}$

$\Pr[\exists y\in C\ (x\neq y \& h_i(x)=h_i(y))]\leq |C|\cdot 2^{-b}\leq 1/4$

$\Pr[\forall i\leq l\ \exists y\in C\ (x\neq y\& h_i(x)=h_i(y))]\leq 4^{-l}$

$\Pr[\exists x\in C\ \forall i\leq l\ \exists y\in C\ (x\neq y\& h_i(x)=h_i(y))]$
$\leq |C|\cdot 4^{-l}\leq 2^{-l}$

Therefore $\Pr[|H(C)|\geq|C|/l]\geq 1-2^{-l}$

Proof 1b: Since $|C|\geq 2^b/8$, if $|H(C)|\geq|C|/l$ then

$$\frac{|H(C)|}{|\Sigma^b|}\geq\frac{1}{8l}$$

Thus it is likely that one of the l^2 strings in Z will be in $H(C)$.

$\Pr[H(C)\cap Z=\emptyset]\leq(1-1/8l)^{l^2}+2^{-l}<2^{-l/8}$

Proof 2a: Obvious.

Proof 2b: Since

$$\frac{|H(C)|}{\Sigma^b}\leq\frac{l|C|}{d|C|}=\frac{l}{d}$$

The probability that each z_i is in $H(C)$ is at most l/d. Thus the probability that any of the l^2 strings in Z is in $H(C)$ is at most l^3/d. ■

We use this lemma to obtain Arthur-Merlin protocols for showing an approximate lower bound on the size of sets. Let C be a set in which Arthur can verify membership, possibly with Merlin's help. Then let Arthur picks random H and Z and Merlin attempt to respond with $x\in C$ such that some $x\in H^{-1}(z)$. If C is large then he will likely succeed and if C is small he will likely fail.

4.2. Main Theorem

Theorem: $\mathrm{IP}[Q(n)]=\mathrm{AM}[Q(n)+2]$ for any polynomial $Q(n)$

An informal proof sketch: Let's focus on 1-round protocols. Assume V has an exponentially small error probability e, sends only messages of length m, and uses random sequences of length l. For each $x\in\Sigma^m$ let $\beta_x=\{r: V(r,w,\#)=x\}$. For every $y\in\Sigma^m$ let $\alpha_{xy}=\{r: r\in\beta_x \ \&\ V(r,w,\#x\#y)=\textbf{accept}\}$. Clearly, for each x, the optimal prover will select a y_x maximizing $|\alpha_{xy}|$. Let $\alpha_x=\alpha_{xy_x}$. Let $\alpha_0=\bigcup_x\alpha_x$. Then $\Pr[V(w)\text{ accepts}] = |\alpha_0|/2^l$.

We next present the protocol by which A and M simulate V and P. M tries to convince A that $|\alpha_0| > e\cdot 2^l$ because this implies that $\Pr[V(w)\text{ accepts}] > e$ and hence ≈ 1. He does this by showing that there are "many" α_x's which are "large", where "many"$\times$"large"$>e\cdot 2^l$. The tradeoff between "many" and "large" is governed by a parameter b sent by M to A.

More precisely, M first sends b to A. Then two approximate lower bound protocols ensue. The first convinces A that $|\{x: |\alpha_x|\geq 2^b/(e\cdot 2^l)\}| \geq 2^b$. M produces an x in that set as per the approximate lower bound lemma. The second convinces A that x really is in that set as claimed, i.e., that $|\alpha_x| \geq 2^b/(e\cdot 2^l)$.

For g-round protocols iterate the first approximate lower bound protocol to obtain $\alpha_0\supseteq\alpha_1\supseteq\cdots\supseteq\alpha_g$ where there are "many$_i$" ways to extend α_{i-1} to α_i and α_g is "large". Require that $(\Pi\text{"many}_i\text{"})\text{"}\times\text{"large"} \geq e\cdot 2^l$.

Full proof: Let $W \in \mathrm{IP}[Q(n)]$. We may assume, without loss of generality, that on inputs w of length n there are exactly $g(n)=Q(n)/2$ pairs of messages sent between V and P, these messages are exactly $m(n)$ long and the random input r to V is $l(n)$ long. Let $e(n)$ bound the error probability.

Amplification Lemma: Let $p(n)$ be a polynomial. Let V be a verifier which on inputs of length n a total of at most $g(n)$ messages, each of length $m(n)$, using $l(n)$ random bits, and with error probability at most 1/3. Then there is a V' such that $L(V)=L(V')$, with a total of at most $g(n)$ messages, each of length $O(p(n)m(n))$, using $O(p(n)l(n))$ random bits and with an error probability of at most $2^{-p(n)}$.

proof: V' performs $O(p(n))$ independent parallel simulations of V and takes the majority vote of the outcomes. Details left to the reader. ■

By this lemma we may assume:

$$e(n) \leq l(n)^{-12g^2(n)}$$

Further we may assume that $l(n) > \max(g(n), m(n), 80)$. We write g, m, e, l for $g(n)$, $m(n)$, $e(n)$ and $l(n)$ where n is understood.

We now describe the functions A and M, simulating V and P, informally as two parties exchanging messages. The variables x_i and y_i represent messages sent by V and P respectively. In essence, the idea is for A to use the random hash functions to force M to produce a generic run of the V,P protocol and then finally to prove that this run would likely cause V to accept. The numbers b_i that M produces roughly correspond to the log of the number of possible generic messages that V can make at round i.

Arthur's protocol

Round 0:

A initially makes a null move and receives number b_1 from M. Go to round 1.

Round i ($1 \leq i \leq g$):

So far A has received $b_1, \ldots, b_i$, and strings $x_1, \ldots, x_{i-1}, y_1, \ldots, y_{i-1}$ from M. Now A randomly selects l linear functions $H=\{h_1, \ldots, h_l\}$, $h_i: \Sigma^m \to \Sigma^{b+1}$ and l^2 strings $Z=\{z_1, \ldots, z_{l^2}\} \subseteq \Sigma^{b+1}$ and them sends to M. A then expects to receive strings x_i and y_i and number b_{i+1} from M. A checks that $x_i \in H^{-1}(Z)$. If not then A immediately rejects. Then A performs round $i+1$.

Final round $g+1$:

Let $s_i = x_1 \# y_1 \# \cdots \# x_i \# y_i$. A randomly selects l linear functions $H=\{h_1, \ldots, h_l\}$, $h_i: \Sigma^l \to \Sigma^{b_{g+1}}$ and l^2 strings $Z \subseteq \Sigma^{b_{g+1}}$. It then expects to receive a string $r \in \Sigma^l$ from M and checks that $r \in H^{-1}(Z)$. A accepts if for each $i \leq g$ $V(w,r,s_i)=x_{i+1}$, $V(w,r,s_g)=$**accept** and $\sum b_i \geq l - g \log l$.

Can Merlin convince Arthur?

Now we show that $\Pr[V(w) \text{ accepts}] > e(n)$ iff $\Pr[A(w) \text{ accepts}] \geq 2/3$.

($\rightarrow$) Merlin's protocol when $w \in W$

First some notation. For $r \in \Sigma^l$ and $s = v_1 \# v_2 \# \cdots \# v_k$ a stream of messages we say

$$(V^*P)(w,r) \text{ accepts via } s$$

if the first k messages sent by V and P agree with s and $(V^*P)(w,r)$ accepts.

Suppose $\Pr[V(w) \text{ accepts}] \geq 2/3$. Fix any P such that $\Pr[(V^*P)(w) \text{ accepts}] \geq 2/3$. We now exhibit a protocol for M such that $\Pr[(A^*M)(w) \text{ accepts}] \geq 2/3$.

Round 0:

Let $i=1$. Proceed with "obtain b_1".

Obtain b_i ($i \leq g$): Let $s_{i-1} = \# x_1 \# y_1 \# \cdots \# x_{i-1} \# y_{i-1}$ be the message stream for the V-P protocol produced so far. For each $x \in \Sigma^m$ let $\alpha_x = \{r: (V^*P)(w,r) \text{ accepts via } s_{i-1} \# x\}$. Group these α's into l classes $\gamma_1, \ldots, \gamma_l$ where γ_d contains α's of size $> 2^{d-1}$ and $\leq 2^d$. Choose the class $\gamma_{\max}$

whose union $\bigcup\gamma_{max} = \bigcup\{\alpha_x: \alpha_x \in \gamma_{max}\}$ is largest. Send $b_i = 2 + \lceil \log|\gamma_{max}| \rceil$.

Round i:

M receives $h_1, \ldots, h_l$ from A and strings $z_1, \ldots, z_{l^2}$. If there is an $x \in H^{-1}(Z)$ such that $\alpha_x \in \gamma_{max}$, call it x_i. Then, M responds with the pair x_i, y_i where $y_i = P(s_{i-1} \# x_i)$. Otherwise M responds with "failure". In the later analysis we refer to the set α_{x_i} as α_i. Set $i \leftarrow i+1$. Goto "obtain b_i".

Obtain b_{g+1}: M produces the value b_{g+1} as follows: Let $s_g = s_{g-1} \# x_g \# y_g$ be the message stream that has been selected. So $\alpha_g = \{r: (V^*P)(w,r)$ accepts via $s_g\}$. Send $b_{g+1} = 2 + \lceil \log|\alpha_g| \rceil$.

Round $g+1$:

M receives $h_1, \ldots, h_l$ and strings $z_1, \ldots, z_{l^2} \in \Sigma^{b_{g+1}}$. If there is an $r \in \alpha_g \cap H^{-1}(Z)$, then M responds with r. Otherwise M responds with "failure". (Note that $r \in \alpha_g$ implies that $V(w,r,s_g) =$ **accept**)

End of Protocol.

We now show that $\Pr[(A^*M)(w)$ accepts$] \geq 2/3$. Let $\alpha_0 = \{r: (V^*P)(w,r) =$ **accept**$\}$. Since $\Pr[V$ accepts $w]$ is high, $|\alpha_0| \geq (2/3)2^l$. By the definition of M, A will accept provided M never responds "failure" and $\sum b_i \geq l - g \log l$. By the approximate lower bound lemma the probability that M responds failure at any round is $\leq 2^{-l/8}$. Hence, the probability that M ever responds failure is $\leq g 2^{-l/8} << 1/3$.

The following two claims show that $\sum b_i \geq l - g \log l$.

Claim 1: For each $0 \leq i < g$

$$|\alpha_i| \geq \frac{|\alpha_{i-1}|}{l 2^{b_i}}$$

Proof: Consider round i and the sets α_x defined in "obtain b_i". By definition the α_x's partition α_{i-1} and hence $\bigcup_x \alpha_x = \alpha_{i-1}$. Hence

$$|\bigcup\gamma_{max}| \geq \frac{|\alpha_{i-1}|}{l}$$

Since all members of γ_{max} differ in size by at most a factor of 2 and since $\alpha_i \in \gamma_{max}$ we have

$$|\alpha_i| \geq \frac{|\bigcup\gamma_{max}|}{2|\gamma_{max}|}$$

and since $b_i = 2 + \lceil \log|\gamma_{max}| \rceil$ we have

$$2^{b_i+1} \geq 2|\gamma_{max}|$$

Thus

$$|\alpha_i| \geq \frac{|\bigcup\gamma_{max}|}{2^{b_i}} \geq \frac{|\alpha_{i-1}|}{l 2^{b_i}}$$

■

Claim 2: $\sum b_i \geq l - g \log l$

Proof: By Claim 1 we have:

$$|\alpha_g| \geq \frac{|\alpha_0|}{l^g \cdot \prod_{i \leq g} 2^{b_i}}$$

Since $|\alpha_0| \geq (2/3)2^l$ and taking logs

$$\log|\alpha_g| \geq (l-1) - (g \log l + \sum_{i \leq g} b_i)$$

Since $b_{g+1} > 1 + \log|\alpha_g|$

$$\sum_{i \leq g+1} b_i \geq l - g \log l$$

■

(←) Merlin's impotence when $w \notin W$

Show that if $\Pr[V(w)$ accepts$] \leq e$, then $\Pr[A(w)$ accepts$] \leq 1/3$.

For every $i > 0$ and $s_i = x_1 \# y_1 \# \cdots \# x_i \# y_i$ let $a(s_i) = \max_P \Pr[(V^*P)(w)$ accepts via $s_i]$. For each $x \in \Sigma^m$ let y_x be any $y \in \Sigma^m$ maximizing $a(s_i \# x \# y)$.

The following three claims show that $a(s_{i+1})$ is likely to be much smaller than $a(s_i)$.

Claim 3: $a(s_i) = \sum_x a(s_i \# x \# y_x)$ ■

Fix $0 \leq i < g$ and s_i. For every $c > 0$ let $X_c = \{x: a(s_i \# x \# y_x) \geq a(s_i)/c\}$

Claim 4: $|X_c| \leq c$ ■

Fix $b,d>0$. Choose l random linear functions $H=\{h_1, \ldots, h_l\}$, $h_i:\Sigma^m \rightarrow \Sigma^b$ and l^2 random strings $Z \subseteq \Sigma^b$. Pick any $x \in H^{-1}(Z)$ and any $y \in \Sigma^m$. Let $s_{i+1}=s_i\#x\#y$.

We now describe a collection of events corresponding to exceptional luck on Merlin's part.

Call the following event E_{i+1}:

$$a(s_{i+1}) \geq \frac{a(s_i)}{2^b/d}$$

Claim 5: $\Pr[E_i] \leq l^3/d$

Proof: Let $c=\lfloor d/2^b \rfloor$. Then $|X_c| \leq 2^b/d$ by claim 4. Since $a(s_i\#x\#y_x) \geq a(s_{i+1})$ by the definition of y_x, if $a(s_{i+1}) \geq a(s_i)/(2^b/d)$ then $x \in X_c$. Since $x \in H^{-1}(Z)$,

$$Pr\left[a(s_{i+1}) \geq \frac{a(s_i)}{2^b/d}\right]$$

$$= \Pr[x \in X_c \cap H^{-1}(Z)]$$

$$= \Pr[H(X_c) \cap Z \neq 0]$$

$$\leq l^3/d$$

by the approximate lower bound lemma part 2b. ■

Fix s_g. Choose l random linear functions $H=\{h_1, \ldots, h_l\}$, $h_i:\Sigma^l \rightarrow \Sigma^{b_{g+1}}$ and l^2 random strings $Z \subseteq \Sigma^{b_{g+1}}$. Pick any $r \in H^{-1}(Z)$. Call the following event E_{g+1}:

$2^l a(s_g) \leq 2^b/d$ and $(V^*P)(w,r)$ accepts via s_g

Claim 6:

$$\Pr[E_{g+1}] \leq l^3/d$$

Proof: By the approximate lower bound lemma part 2b, since $|\{r: (V^*P)(w,r) \text{ accepts via } s_g\}| = 2^l a(s_g)$. ■

In any run of A and M, event E_i may occur during round i, where $b=b_i$ for $i \leq g+1$. The probability that each occurs is at most l^3/d and therefore the probability that any occurs is at most $(g+1)l^3/d$. Choose

$$d=3(g+1)l^3.$$

Then $\Pr[\exists\, i E_i \text{ occurs}] \leq 1/3$.

Assume no E_i occurs. Then we show that A will reject, provided that $\Pr[V(w) \text{ accepts}] \leq e$.

Since $\forall i \leq g$, $\neg E_i$, we have:

$$\frac{a(s_0)}{\prod_{i \leq g}(2^{b_i}/d)} \geq a(s_g)$$

Since $\neg E_{g+1}$:

$$(V^*P)(w,r) \neq \textbf{accept}$$

or

$$2^l a(s_g) \geq 2^{b_{g+1}}/d$$

Thus if $(V^*P)(w,r)$ accepts, combining the above:

$$2^l a(s_0) \geq \prod_{1 \leq i \leq g+1} (2^{b_i}/d)$$

so, since $l \geq g+1$, taking logs:

$$\begin{aligned} l + \log a(s_0) &\geq \sum b_i - (g+1)\log d \\ &\geq \sum b_i - (g+1) \\ &\geq \sum b_i - 10g\log l \end{aligned}$$

but

$$a(s_0) = \Pr[V(w) \text{ accepts}] \leq e \leq l^{-12g}$$

so

$$l - 12g^2 \log l \geq \sum b_i - 10g\log l$$

Thus

$$\sum b_i \leq l - 2g\log l < l - g\log l$$

Recall that Arthur only accepts if $(V^*P)(w,r)$ accepts and $\sum b_i \geq l - g\log l$. Therefore if $\forall\, i \leq g+1$, E_i occurs and $\Pr[V(w) \text{ accepts}] \leq e$, then Arthur will reject. Hence $\Pr[A(w) \text{ accepts}] \leq 1/3$. ■

5. Probabilistic, nondeterministic Turing machines

We can define a new type of Turing machine which accepts precisely those languages in IP. This gives an automata theoretic characterization of this class.

Definition: A *Probabilistic, nondeterministic, Turing machine, N*, is defined conventionally except that it has two kinds of nondeterministic states: random states denoted  and guess states, denoted  .

Given a configuration, c of such a machine we assign it a probability $p(c)$ of accepting as follows: if c is an accept configuration then $p(c)=1$, if c is a reject configuration then $p(c)=0$, if c is a deterministic configuration then $p(c)=p(c')$ where c' is the successor of c, if c is a random configuration, then $p(c)$ is the average of $p(c')$, and if c is a guess state then $p(c)$ is the maximum of $p(c')$ for c' a successor. We say that $\Pr[N(w) \text{ accepts}] = p(c_{\text{start}})$ where c_{start} is the starting configuration for N on input w. One way to think of computations on these machines is that at every random state, a coin is flipped to determine the successor and at every guess state the successor with highest probability of eventually accepting is selected.

Definition: Say $W \in \text{BPNP}$ if there is a probabilistic, nondeterministic, polynomial time Turing machine N such that for all $w \in \Sigma^*$:

1) if $w \in W$ then $\Pr[N(w) \text{ accepts}] > 2/3$
2) if $w \notin W$ then $\Pr[N(w) \text{ accepts}] < 1/3$.

Theorem: $\text{IP} = \text{AM(poly)} = \text{BPNP}$

Proof: Immediate. These machines are just a reformulation of Arthur-Merlin games.

6. Open Questions

1. Is $\text{IP}[2] = \text{IP}$, or perhaps show an oracle F such that $\text{IP}^F \neq \text{IP}[2]^F$?

2. In [B] Babai states that $\text{AM}[2] = \{W: W \in NP^R, \text{ for almost all oracles } R\}$. However, an oversight in his argument leaves this equality an open question.

3. Can our transformation of interactive proof systems with public coins into Arthur-Merlin games be modified to preserve "zero-knowledge" without assuming the existence of one-way functions?

Acknowledgements

We are grateful to Oded Goldreich and Johan Hastad for pointing out that our proof works for a polynomial number of interactions. Silvio Micali's comments have been inspiring, as always. Thanks again to Oded for a careful reading of this paper and extensive suggestions. Discussions with Laszlo Babai, Anne Condon, Jack Edmonds, and Eva Tardos were very helpful. Ed Bein, Danny Soroker, and Jeannine St.Jacques provided much appreciated last minute assistance.

References

[B] L. Babai, *Trading group theory for randomness*, Proc. of 17th Symposium on the Theory of Computation, Providence, Rhode Island, 1985.

[Be] M. Ben-Or, personal communication.

[BH] R. Boppana, J. Hastad, *If co-NP Has Interactive Proof Systems with a Constant Number of Interactions, then the Polynomial Time Hierarchy Collapses*, In prep.

[CKS] Chandra, Kozen, Stockmeyer, *Alternation*, JACM 1981, p. 114.

[Co] S. Cook, *The Complexity of Theorem Proving Procedures*, 3rd STOC, 1971.

[CW] J.L. Carter, and M.N. Wegman, *Universal classes of hash functions*, JCSS 18, no. 2, 1979, 143-154.

[F] P. Feldman, *The Prover in IP Need Not be More Powerful than PSPACE*, personal communication.

[FS] L. Fortnow, M. Sipser, personal communication.

[H] J. Hastad, personal communication.

[GMR] S. Goldwasser, S. Micali, C. Rackoff, *The Knowledge complexity of interactive proofs*, Proc. of 17th Symposium on the Theory of Computation, Providence, Rhode Island, 1985.

[GMW] O. Goldreich, S. Micali, A. Wigderson, *Proofs that Yield Nothing but the Validity of their Assertion*, In preparation.

[P] C. Papadimitriou, *Games against nature*, 24th FOCS 1983, 446-450.

[R] J. Reif, *Games with imperfect information*, JCSS 29, 274-301, 1984.

[Si] M. Sipser, *A Complexity Theoretic Approach to Randomness*, 15th STOC, 1983, 330-335.

[St] L. Stockmeyer, *The complexity of approximate counting*, Proc. of Symposium on the Theory of Computation, 1984.

[ZF] S. Zachos, M. Furer, *Probabilistic quantifiers vs. distrustful adversaries*, to appear.