COMBINATORICA
Akadémiai Kiadó — Springer-Verlag

COMBINATORICA 10 (1) (1990) 3—25

ON THE POWER OF INTERACTION

W. AIELLO[1], S. GOLDWASSER[2] and J. HASTAD[3]

Received February 2, 1988
Revised August 24, 1989

Let $IP[f(n)]$ be the class of languages recognized by interactive proofs with $f(|x|)$ interactions. Babai [2] showed that all languages recognized by interactive proofs with a bounded number of interactions can be recognized by interactive proofs with only two interactions; i.e., for every constant k, $IP[k]$ collapses to $IP[2]$.

In this paper, we give evidence that interactive proofs with an unbounded number of interactions may be more powerful than interactive proofs with a bounded number of interactions. We show that for any polynomially bounded polynomial time computable function $f(n)$ and any $g(n)=o(f(n))$ there exists an oracle B such that $IP^B[f(n)] \not\subseteq IP^B[g(n)]$.

The techniques employed are extensions of the techniques for proving lower bounds on small depth circuits used in [6], [14] and [10].

1. Introduction

The class *NP* has traditionally been recognized to capture the notion of efficient provability, containing those languages for which there exist short proofs of membership which can be verified efficiently. The *NP* proof-system consists of a powerful prover which guesses the short proof, and a polynomial time verifier which checks the correctness of the proof. The interaction between the prover and the verifier consists of the prover sending a single string (the proof) to the verifier.

Recently, Goldwasser, Micali, and Rackoff [8], and Babai [2] each extended the familiar *NP* proof-system to incorporate randomness and more complex interaction. In both cases the verifier is a randomized polynomial time machine which exchanges messages with the prover before deciding whether to be convinced by the "proof". Two new complexity hierarchies arise, corresponding to the number of messages exchanged between prover and verifier, both of which would collapse to *NP* if the verifier tosses no coins.

Goldwasser, Micali, and Rackoff [8] define their hierarchy through the notion of an "interactive proof system". An interactive proof system consists of a prover of unlimited computational power and a probabilistic polynomial time verifier. Both receive a common input x, and exchange up to polynomial in $|x|$ number of messages, each being of length at most a polynomial in $|x|$, with the verifier

[1] Research done while in the Department of Mathematics at M. I. T. and supported by an ONR graduate fellowship.
[2] Supported in part by NSF Grant DCR MCS8509905.
[3] Research done while at the Laboratory for Computer Science at M. I. T. and Supported by an IBM fellowship.
AMS subject classification (1980): 68Q15.

sending the first message. The verifier's ith response is the result of a random polynomial time computation on input x, and all messages sent so far. At the end of the interaction the verifier makes a polynomial time computation to determine whether to accept or reject x.

We say that an interactive proof system recognizes a language L if the probability that the prover can make the verifier accept is $\geq \frac{2}{3}$ when $x \in L$ and $\leq \frac{1}{3}$ when $x \notin L$. The interactive proof (IP) hierarchy is now defined as follows. Language $L \in IP[f(n)]$ if there exists an $f(n)$-move interactive proof system which recognizes L.

Babai's [2] proof system is defined via a combinatorial game played by a random player, Arthur (in the role of the verifier), and an optimal player Merlin (in the role of the prover). As with the system of Goldwasser, Micali, and Rackoff, on input x, Arthur and Merlin alternate exchanging messages, with Arthur moving first, until Arthur accepts or rejects x (Merlin wins or loses). However, Arthur is restricted to flipping a prescribed number of coins and sending their outcome to Merlin. The class $AM[f(n)]$ is defined in the same manner as $IP[f(n)]$.

Goldwasser and Sipser [9] showed that the more powerful verifier of IP does not increase the power of the model with respect to language recognition. Namely, for any polynomially bounded function $f(n)$, $IP[f(n)] = AM[f(n)]$.

Due to its simple combinatorial formulation, the Arthur—Merlin model is easier to work with in the context of this paper. We will thus prove our results using the Arthur—Merlin game and AM hierarchy terminology. Clearly, all results in this paper concerning the AM hierarchy extend to the IP hierarchy.

One note on notation before we start. All functions $f(n)$ and $g(n)$ we consider are integer valued.

1.1. The finite level hierarchy collapses

Babai [2] showed that the finite levels of the Arthur—Merlin hierarchy collapse to the second level: $AM[k] = AM[2]$ for all integers $k > 2$. Moreover, Babai [2] showed that $AM[2]$ is contained in Π_2. These proofs relativize, that is for all B, $AM^B[k] \subset \Pi_2^B$. Babai conjectured that $AM[Poly] = \bigcup_c AM[n^c]$ is contained in Σ_k for some k. In addition, he made the stronger conjecture that $AM[Poly] = AM[2]$.

One of the most fundamental complexity issues concerning interactive proofs is to resolve whether more rounds add language recognition power. On the positive side, Babai and Moran [3] subsequently showed that $AM[f(n)] = AM[cf(n)]$ for all constants $c > 0$; this proof holds for all oracles.

On the negative side we show that this is the best possible collapse theorem which relativizes.

1.2. Our results

We prove that for polynomially bounded polynomial time computable function $f(n)$ and any $g(n)=o(f(n))$, there is an oracle B such that $AM^B[f(n)]$ is not contained in $\Sigma^B_{g(n)}$ where $\Sigma^B_{g(n)}$ is the class of languages recognized by a polynomial time alternating Turing machines with $g(n)$ alternations and access to oracle B. Further, we extend Babai's [2] proof that constant round AM is contained in Π_2 to show that for all $g(n)$, $AM^B[g(n)]\subseteq\Sigma^B_{g(n)+1}$ for all B. Hence, there is an oracle B such that $AM^B[f(n)]$ strictly contains $AM^B[g(n)]$.

These results indicate that proving the collapse of the entire AM hierarchy, if it does indeed collapse, will require proof techniques which do not relativize.

In Babai's proof that a k round AM game can be simulated by a two round AM game, the length of the messages of the two-round simulating AM game is a polynomial factor greater than for the k round game. We show that in a relativized setting the increase in complexity is inherent. Let $AM[k, r_1, r_2]$ be the class of languages recognized by $AM[k]$ games where the length of each message is bounded by n^{r_1} and Arthur's computation time bounded by n^{r_2}. We show that for all constants l, r_1, r_2, and t, there exist a constant k and an oracle B such that $AM^B[k, t, t]\nsubseteq AM^B[l, r_1, r_2]$.

The fact that the number of rounds seems to make a difference raises the question whether $AM[2]$ or $AM[Poly]$ is the natural probabilistic version of NP. One vote for $AM[2]$ is the recent result by Nisan and Wigderson [12] that the class of languages which are in NP^B with probability 1 for a random oracle B is equal to $AM[2]$.

1.3. Outline of our proof

Furst, Saxe, Sipser [6] and Sipser [13] were the first to show that oracle separation results involving classes such as the levels of the polynomial time hierarchy could be achieved by proving lower bounds for constant depth circuits. Since then, improved bounds and subsequent separations have been achieved by Yao [14] and Hastad [10]. We use essentially the same paradigm. However, formulating $AM^B[f(n)]$ in a suitable way and deriving lower bounds will require some work. We proceed as follows.

First we describe (in Section 2) a natural Arthur—Merlin game with $f(n)$ interactions in which Arthur makes one query to an oracle B during his polynomial time evaluation of whether to accept or reject the input. We call this the defining game. Let the *value* of the game be the probability that Arthur accepts the input. We define a unary language $L(B)$ based on this game as follows: $1^n\in L(B)$ iff the value of the defining game is greater than 2/3. For some B the value of the defining game will never fall between 1/3 and 2/3. In these cases $L(B)\in AM^B[f(n)]$. For the sake of the outline the goal is to show that of these B's there exists one for which $L(B)\notin\Pi^B_2$ and hence $L(B)\notin AM[2]$. We proceed as follows:

For any oracle Turing machine M^B which has 2 alternations and runs in polynomial time we want to choose B to ensure that $L(B)$ is not accepted by M^B. To do this we choose a sufficiently large n and look at the behavior of M^B on input 1^n. The output of M^B corresponds to the value of a depth 3 circuit of small size as shown in [6]. (The general reduction from machines with $g(n)$ interactions to circuits

of depth $g(n)+1$ will be outlined in Section 3. The proof that $AM[g(n)] \subset \Sigma_{g(n)+1}$ will be given in Section 4.) The inputs to this circuit are Boolean variables y_z^B where $y_z^B=1$ iff $z \in B$. We fix the values of some of the inputs (determining whether some strings are in the oracle set or not) in such a way as to determine the output of the circuit and hence determine the computation of M^B, but so that the value of the defining game played on input 1^n will not be determined. (As in [6], [14] and [10] we do not know how to find such a setting deterministically but rely on probabilistic arguments to show that one exists.) By determining slightly more of the oracle set we can force the value of the game in such a way that if M^B accepted then the value of the game will be less than $1/3$ and if M^B rejected then the value of the game will be greater than $2/3$. In this way we ensure that M^B does not recognize $L(B)$. Finally, to ensure that no machine recognizes $L(B)$ we do a standard diagonalization over all oracle Turing machines.

To keep track of what happens to the value of the game as we fix part of the oracle set we need to introduce some notation. In Section 5 we define a special type of circuit called the $\top$ *circuit.* It has two types of gates: *threshold* gates and *or* gates. It takes 0 and 1 as inputs and outputs 0, 1, and a special symbol †. We introduce a $\top$ circuit that computes the value of the defining game in the following sense. If the $\top$ circuit with $f(n)$ levels evaluates to 1 on a certain setting of input variables then the value of the defining game on 1^n with the oracle corresponding to the setting of the variables is $\geq 2/3$; if the circuit evaluates to 0 then the value of the defining game is $\leq 1/3$; while if the circuit evaluates to † then no useful information about the value of the game is obtained.

In Section 6 we define the new random restrictions we need to construct B. Restrictions assign values to some input variables and hence simplify the functions to which they are applied. In Section 7 we show that with high probability the defining $\top$ circuit hit with a restriction can be written as a $\top$ circuit with a constant fewer levels. In Section 8 we show that with high probability any small constant depth circuit hit with a restriction can be written as a small circuit of depth one less. The technical part of this argument is postponed to Section 10. Since the number of interactions in the defining game grows with n as $f(n)$, the height of the $\top$ circuit grows in the same manner. Hence, for n large enough, a constant number of restrictions completely determines the depth 3 circuit but leaves the $\top$ circuit undetermined. This will give us enough freedom (in Section 8) to set the variables y_z^B by diagonalization so that $L(B)$ is in $AM^B[f(n)]$ but not in Π_2^B. In Section 9 we consider games of fixed size. We conclude in Section 11 with some open problems.

This paper is a complete version of the conference paper [1].

2. *AM* games and the definition of $L(B)$

Let us start by making a definition of *AM*.

An Arthur—Merlin game is played between two Turing-machines A(rthur) and M(erlin). A is probabilistic polynomial time while M has no resource bounds. On input x, A and M interact for a polynomial number of rounds in the following way:

1) A flips a predetermined number of random coins and sends the result to M. Call A's message in the i'th round a_i.
2) M responds with a message b_i based on an arbitrary computation.

After at most a polynomial number of rounds the game terminates and A computes in polynomial time a $\{0, 1\}$-valued function. The function depends on the a_i, the b_i and the input w. If the value of this function is 1 we say that Arthur accepts the input while otherwise he rejects. We will use the phrases "Arthur accepts the input" and "Merlin wins the game" synonymously since (one) can think of the game as Merlin doing his best to make Arthur accept, while Arthur just indifferently flips coins.

Let the value of the game be the probability (over Arthur's coin flips) that Arthur accepts the input when Merlin plays optimally. The value of the game is then a function of the input. Now we define a language L to be in $AM[f(n)]$ if there is a $f(|x|)$ round game such that

1) If $x \in L$ then the value of the game is at least 2/3.
2) If $x \notin L$ then the value of the game is at most 1/3.

The definition of a relativized *AM* game is now straightforward.

Definition. An Arthur—Merlin game with oracle B is an Arthur—Merlin game in which Arthur and Merlin have access to oracle B.

Definition. $L \in AM^B[f(n)]$ if there exists an Arthur—Merlin game with oracle B which on input x makes at most $f(|x|)$ moves such that for all $x \in L$ the value of the game is at least 2/3; and for all $x \notin L$ the value of the game is at most 1/3.

The language $L(B)$ is defined with respect to the following Arthur—Merlin game (which we will often refer to as the defining game). It will be convenient for the game to have an even number of interactions so define $f'(n) = \lfloor f(n)/2 \rfloor$.

On input 1^n, the game has $f'(n)$ rounds starting with Arthur. At the ith round the following happens.

1. Arthur sends an n-bit random string, denoted a_i.
2. Merlin responds with an n-bit string, denoted b_i.

Arthur accepts if the string $a_1 b_1 a_2 b_2 \ldots a_{f'(n)} b_{f'(n)}$ of length $2nf'(n)$ is in oracle B.

Definition. The language $L(B)$ is a unary language such that for all n, $1^n \in L(B)$ iff the value of the above game is $\geq 2/3$.

Note that $1^n \notin L(B)$ only means that the value of the game on 1^n is $< 2/3$. However, for many B's the value of the game is never between 1/3 and 2/3. In these cases $L(B) \in AM^B[f(n)]$. The goal is for a given $g(n)$ which is $o(f(n))$ to find a B for which it is also true that $L(B) \notin \Sigma^B_{g(n)+1}$. Our first step in doing so is to give a circuit formulation of $\Sigma^B_{g(n)+1}$ and of the *AM* protocol recognizing $L(B)$ which we do in Sections 3 and 5 respectively.

3. Relativized complexity and circuits

Let us first state the connection between $\Pi^B_{g(n)}$ and $g(n)$ depth circuits. This was first established in [6] and [13].

Definition. A $\Pi^B_{g(n)}(\Sigma^B_{g(n)})$-*machine* is an alternating Turing machine which runs in polynomial time, has at most $g(n)$ alternations along any computation branch, starts with an $\wedge$ ($\vee$) alternation, and makes polynomial length queries to an oracle for B.

Definition. A language L is said to be in $\Pi^B_{g(n)}(\Sigma^B_{g(n)})$ iff it is accepted by some $\Pi^B_{g(n)}(\Sigma)^B_{g(n)}$ machine.

For the remainder of the paper we will identify an oracle B with the values of the Boolean variables $\{y^B_z | z \in \sum^*\}$ by setting $y^B_z = 1$ iff $z \in B$.

Lemma 3.1. *Let M^B be a $\Pi^B_{g(n)}(\Sigma^B_{g(n)})$ machine which runs in time t on input x for any oracle B. Then there is a depth $g(n)+1$ circuit C of size 2^{2t} which has a subset of the y^B_z as inputs such that for every oracle B, M^B accepts x precisely when C outputs 1 on inputs y^B_z.*

Remark. The structure of the circuit depends on M^B and the input x.

Proofs for the case $g(n) = constant$ can be found in [6] and [11]. The generalization to $g(n)$ unbounded is straightforward.

When we are studying the fine structure of the size of games it is useful to have a refinement of Lemma 3.1. We say that an alternating machine is in *non-alternating mode* if it will make no more alternations during its computation. While it can still make alternations we say it is in *alternating mode.*

Lemma 3.2. *Let M^B be a $\Pi^B_{g(n)}(\Sigma^B_{g(n)})$ machine which runs in time t_1 in alternating mode and time t_2 in non-alternating mode on input x for any oracle B. Then there is a depth $g(n)+1$ circuit C which has at most 2^{2t_1} gates at least two away from the inputs and bottom fanin at most t_1+t_2 which has a subset of the y^B_z as inputs such that for every oracle B, M^B accepts x precisely when C outputs 1 on inputs y^B_z.*

This can again be seen by almost the same argument.

4. *AM* and alternating machines

In this section we will establish the inclusion of $AM[f(n)]$ in $\sum_{f(n)+1}$. This inclusion will be needed in the proof of our main theorem. In Section 10 we will need rather detailed information about the size of games so we introduce some size parameters.

Definition. Let $AM[f(n), r_1, r_2]$ denote the set of languages which are recognizable by an Arthur—Merlin game with the following properties: there are at most $f(n)$ interactions; each interaction has length at most n^{r_1}; and Arthur's decision procedure takes time at most n^{r_2}.

For notational simplicity we will make the assumption that $r_2 > r_1$. This is natural since it is always satisfied if we assume that Arthur looks at the entire conversation.

Using this notation we can state the lemma we will prove in this section.

Lemma 4.1. *If* $L \in AM[f(n), r_1, r_2]$ *then* L *can be recognized by a* $\Sigma^B_{f(n)+1}$ *machine which runs in time* $O(f^2(n)n^{2r_1}\log(f(n)n^{r_1}))$ *in alternating mode and at most time* $O(f(n)n^{r_1+r_2}\log(f(n)n^{r_1}))$ *in non-alternating mode.*

Remark. Lemma 4.1 appears in a slightly different form in [7]. For completeness we give the proof.

Proof. We will convert the game between Arthur and Merlin to a game between two all-powerful players $\forall$ and $\exists$ with one extra move. In a certain sense to be made precise below, $\exists$ will play the role of Merlin and $\forall$ the role of Arthur. By the usual correspondence between such games between optimal players and alternating Turing machines the lemma will follow.

First we will decrease the uncertainty in the Arthur—Merlin game. Play the old game $24\lceil\log(72f(n)n^{r_1})\rceil$ times in parallel and let Arthur accept if old Arthur accepts in a majority of the games. It is easy to see that if $x \in L$ then the probability that Merlin can win the game is at least $1-2^{-\lceil\log(72f(n)n^{r_1})\rceil}$ and if $x \notin L$ then Merlin can win with probability at most $2^{-\lceil\log(72f(n)n^{r_1})\rceil}$. The size of each interaction in this game is bounded by $24n^{r_1}\lceil\log(72f(n)n^{r_1})\rceil$.

Let a_i be Arthur's message in the ith round of this game and let b_i be Merlin's message in the ith round. Let $P(a, b)$ denote the predicate such that $P(a, b)=1$ iff Arthur accepts after the conversation $a_1 b_1 \ldots q_{\lfloor f(n)/2\rfloor}$, where q is b if $f(n)$ is even and a if $f(n)$ is odd. Consider the following game between two all powerful players $\exists$ and $\forall$.

Let $k=36f(n)n^{r_1}$.

Round 0. $\exists$ sends $c^{(1)}, c^{(2)}, \ldots, c^{(k)}$ to $\forall$, where $c^{(j)}=(c_1^{(j)}, c_2^{(j)}, \ldots, c^{(j)}_{\lfloor f(n)/2\rfloor})$ and

$$|c_i^{(j)}| = 24\lceil\log(72f(n)n^{r_1})\rceil n^{r_1}.$$

For $i=1, 2, \ldots, \lfloor f(n)/2\rfloor$.

Round i. $\forall$ sends a_i, $|a_i|=24\lceil\log(72f(n)n^{r_1})\rceil n^{r_1}$.

$\exists$ sends $b_i^{(1)}, b_i^{(2)}, \ldots, b_i^{(k)}$ where $|b_i^{(j)}| = 24\lceil\log(72f(n)n^{r_1})\rceil n^{r_1}$.

If $f(n)$ is odd $\forall$ completes the game by sending $a_{\lceil f(n)/2\rceil}$.

$\exists$ wins the game iff for some j, $P(c^{(j)} \oplus a, b^{(j)})=1$.

Lemma 4.2. $\exists$ *has a winning strategy iff* $x \in L$.

Observe that Lemma 4.1 follows from Lemma 4.2 by just calculating the total size of the messages. Thus we need only establish Lemma 4.2.

Assume first that $x \in L$. Then we know that there is a strategy S for Merlin such that he wins with probability at least $1-2^{-\lceil\log(72f(n)n^{r_1})\rceil}$. Let $\mathcal{A}$ be the set

of a such that Merlin wins if he follows S and Arthur has coins a. Let $d^{(1)}d^{(2)}, \ldots, d^{(k)}$ be a random choice for $\exists$'s first message and let $a_1, a_2, \ldots, a_{\lceil f(n)/2 \rceil}$ be $\forall$'s moves. If $\exists$ follows strategy S on $d^{(j)} \oplus a$ to produce the moves $b^{(j)}$ then $\exists$ wins iff for some j, $d^{(j)} \oplus a \in \mathscr{A}$. For any fixed a the probability that $d^{(j)} \oplus a \notin \mathscr{A}$ for all j is at most $(2^{-\lfloor \log (72f(n)n^{r_1}) \rfloor})^k = 2^{-36f(n)n^{r_1}\lfloor \log (72f(n)n^{r_1}) \rfloor} \leq 2^{-|a|-1}$, since

$$|a| = 24n^{r_1}\lceil \log (72f(n)n^{r_1}) \rceil \lceil f(n)/2 \rceil.$$

Thus with probability at least $1/2$ all a's lead to acceptance. In particular, there is a choice of the $d^{(j)}$ such that $\exists$ wins if he follows strategy S.

Now assume that $x \notin L$. Fix any initial message $c^{(1)}, c^{(2)}, \ldots, c^{(k)}$. If $\forall$ sends random messages a_i, then by the property of the AM game no matter what strategy $\exists$ uses, the probability that $P(a \oplus c^{(j)}, b^{(j)}) = 1$ is bounded by $2^{-\lfloor \log (72f(n)n^{r_1}) \rfloor}$. Thus the probability that $\exists$ wins the game is bounded by $k/2^{\lfloor \log (72f(n)n^{r_1}) \rfloor} \leq 1/2$. In particular, there are choices of a for which $\forall$ wins. This completes the proof of Lemma 4.2. ∎

5. Oracle games and AM circuits

To give the circuit formulation of the defining AM game for $L(B)$ let us first define something slightly more general.

Definition. A weak AM game with oracle B is an AM game with oracle B in which Arthur makes only one oracle query in his polynomial time evaluation of the game. Without loss of generality we can assume that this query is made at the end of the protocol.

Observe that the game defining $L(B)$ is a weak AM game. In all that is to follow we will assume without loss of generality that all interactions in an Arthur—Merlin game are of the same length and also that the number of interactions only depends on the length of the input.

Let us now define a new type of circuit. The circuit will have two types of gates: A gates and M gates. Inputs and outputs of these gates will be rational numbers. A gates take the value which is the average of the values of their inputs and M gates take the value which is the maximum of the values of their inputs.

Definition. An A_d^l circuit is a 2^l-ary tree of height d where the root and nodes at every other level are A gates and the remaining nodes are M gates.

Lemma 5.1. *Let G^B be a weak AM game with oracle B which has d interactions of length l. For every x there is an assignment to the inputs of A_d^l from $\{0,1\} \cup \{(y_z^B, \bar{y}_z^B)\}$, $z \in \{0,1\}^*$, such that for every B the value of G^B on x is equal to the output of A_d^l.*

Proof. There is an obvious mapping between conversations of the AM game (i.e., strings of length dl) and leaves of the A_d^l tree. At each leaf four cases may occur:

1. Arthur accepts without asking an oracle query, in which case we mark the leaf 1.

2. Arthur rejects without asking an oracle query, in which case we mark the leaf 0.
3. Arthur accepts iff oracle query $z \in B$ is true, in which case we mark the leaf by variable y_z^B.
4. Arthur accepts iff oracle query $z \in B$ is false, in which case we mark the leaf by variable $\bar{y}_z^B$.

The lemma follows by an easy induction on d, the number of interactions. ∎

In particular, the value of the defining game on 1^n is equal to the output of $A^n_{2f'(n)}$ on the y_z^B's with $|z|=2f'(n)n$.

We will not work with AM circuits. Instead we will use $\top$ *circuits*. Apart from being of interest on their own, these $\top$ circuits will be easier to work with. There will be two types of gates: or gates, $\vee$, and threshold gates, $\top_\tau$. Both take inputs from $\{0, 1, \dagger\}$ and output $\{0, 1, \dagger\}$ as follows:

$$\vee = \begin{cases} 1, & \text{if there exists a 1 in the input;} \\ \dagger, & \text{if there exists a } \dagger \text{ but no 1;} \\ 0, & \text{if all inputs are 0.} \end{cases}$$

$$\top_\tau = \begin{cases} 1, & \text{if there are a } \tau \text{ fraction} \\ & \text{of 1's in the input;} \\ 0, & \text{if there are } \tau \text{ fraction} \\ & \text{of 0's in the input;} \\ \dagger, & \text{otherwise.} \end{cases}$$

where τ is a threshold parameter which is always greater than $1/2$.

Definition. A $\top^l_{\tau,d}$ circuit is a 2^l-ary tree of height d where the root is a $\top_\tau$ gate and every other level consists of $\top_\tau$ gates and every other of $\vee$ gates.

We will show that there is a useful sense in which the two new types of circuits just defined can simulate each other.

Lemma 5.2. *For* $\frac{1}{2} < \tau \leqq 1-(1-p)^{1/\lceil d/2 \rceil}$

$$A^l_d(x) \geqq p \Rightarrow \top^l_{\tau,d}(x) = 1$$

$$A^l_d(x) \leqq 1-p \Rightarrow \top^l_{\tau,d}(x) = 0.$$

Proof. We prove the first implication, the second being similar. Observe that if $1-(1-p)^{1/\lceil d/2 \rceil} > 1/2$ then $p > 1-2^{-\lceil d/2 \rceil}$ and thus the lemma is only meaningful for p very close to 1.

Say the inputs to an average gate A of height i are $q_i^1 \ldots q_i^{2^l}$. These are the outputs of the corresponding maximum gates of height $i-1$. Suppose the output of the average gate is at least p_i, then at least a fraction τ of the q_i's are at least $(p_i-\tau)/(1-\tau)$. Suppose we know by induction that $A^l_{i-2}(x) \geqq (p_i-\tau)/(1-\tau)$ implies $\top^l_{\tau,i-2}(x)=1$, then $A^l_i(x) \geqq p_i$ implies that at least τ $\vee$ gates at level $i-1$ in $\top^l_{\tau,i}(x)$ are true. Hence, $\top^l_{\tau,i}(x)$ evaluates to one. This gives us the recurrence $p_i \geqq \geqq p_{i-2}(1-\tau)+\tau$ where $p_1=p_2=\tau$ and $p_d=p$. If $\tau \leqq 1-(1-p)^{1/\lceil d/2 \rceil}$, this can be solved with $p_i \leqq p$. ∎

A partial converse is given below.

Lemma 5.3.

$$\top^l_{\tau,d}(x) = 1 \Rightarrow A^l_d(x) \geq \tau^{\lceil d/2 \rceil}$$

$$\top^l_{\tau,d}(x) = 0 \Rightarrow A^l_d(x) \leq 1-\tau^{\lceil d/2 \rceil}.$$

Proof. Again we prove only the first implication. Use induction on d. The base cases, $d=1$ and $d=2$, are straightforward. There are two different cases, d odd and d even. When d is odd the lemma follows immediately from the $d-1$ case by looking at the input to the $\vee$ gate which takes the value 1.

Hence, assume that d is even and that the lemma is true for $d-2$. $\top^l_{\tau,d}(x)=1$ implies that a fraction τ of the $\vee$ gates at height $d-1$ evaluate to 1. Using the induction hypothesis, this implies that a fraction τ of the M gates at height $d-1$ in the A^l_d tree evaluate to a least $\tau^{\lceil (d-2)/2 \rceil}$. Hence, $A^l_d(x)$ is at least $\tau^{\lceil d/2 \rceil}$. ∎

Lemma 5.4. *For τ of the form $1-o(1/f(n))$, if*

$$\forall n, \quad \top^n_{\tau,2f'(n)}(y^B_z) = 0 \quad \text{or} \quad 1$$

where $|z|=2nf'(n)$ then

$$L(B) \in AM^B[f(n)]$$

and the $\top^n_{\tau,2f'(n)}$ circuit evaluates $L(B)$ correctly for sufficiently large n.

Proof. The proof follows from the definition of $L(B)$ and Lemmas 5.1 and 5.3. ∎

Remark. By the usual correspondence between game trees and quantifiers, Lemmas 5.3 and 5.4 are sufficient to show that the set of languages given by a family of formulas which have alternating threshold quantifiers and existential quantifiers followed by a polynomial time predicate is exactly $AM[Poly]$ (provided the threshold is sufficiently large). However if the threshold is smaller, the threshold circuits seem to be more powerful. For example using threshold 2/3 it is possible to recognize any language in PSPACE in polynomial depth.

This type of formula with a constant number of alternations has been studied by several people. See [15] for an overview.

For the remainder of the paper we will assume that $l=n$ and $\tau=1-2^{-n^{1/4}}$ unless otherwise stated and we will write $\top_{2f'(n)}$ as shorthand for $\top^l_{\tau,2f'(n)}$. Later in the proof when we are setting the variables y^B_z (i.e., determining our oracle set B) we will be careful to make $\top_{2f'(n)}=0$ or 1 for all n so that we can claim $L(B) \in AM^B[f(n)]$ using Lemma 5.4.

For most of the remainder of the paper (Sections 6—9) we work to show that $\top_{2f'(n)}$ cannot be computed by small $(o(f(n)))$ depth circuits. More specifically we will show that for all circuit families of depth $g(n)=o(f(n))$ and size 2^{2n^r} there exists n large enough such that there is some input x for which $C_{g(n)}(x) \neq \neq \top_{2f'(n)}(x) \neq \dagger$. Using this fact and Lemma 3.1 we will construct (in Section 8) a setting of the variables $y^B_1, y^B_2, \ldots$ such that no $\sum^B_{g(n)+1}$ machine accepts $L(B)$. At the same time, however, the setting will satisfy the hypothesis of Lemma 5.4. This will give us $L(B)$ in $AM^B[f(n)]$ but not in $\sum^B_{g(n)+1}$ and we will achieve the claimed separation.

6. New random restrictions

Let us start by recalling a definition from [6].

Definition. A restriction ϱ is a function of the variables x_i to the set $\{0, 1, *\}$. $\varrho(x_i)=0$ (1) means we assign the value 0 (1) to x_i while $\varrho(x_i)=*$ means we keep x_i as a variable.

Given a function F we will denote by $F\restriction_\varrho$ the function we obtain by applying ϱ to the variables of F. $F\restriction_\varrho$ will be a function of the variables which were given the value $*$ by ϱ. As in [6], [14] and [10] we will use random restrictions. We define however a new family of random restrictions, $R_{k,n}$.

Definition. For every integer k the random restriction $\varrho \in R_{k,n}$ is defined as follows. Partition the variables into disjoint groups of size 2^{2kn}. Call each group a *k-block*. Let the ith k-block be the set $\{x_{(i-1)2^{2kn}+1}, \ldots, x_{i2^{2kn}}\}$. Now associate the variables in each k-block with the leaves of a $\top_{2k}$ circuit in the natural way. Label the nodes in each $\top_{2k}$ circuit independently in the following way.

Mark the top node (which is a $\top$ node) with a $*$ and mark the children recursively as follows:

1. For a $\top$ node marked 1, mark all the children 1.
2. For a $\top$ node marked $0(*)$, mark each child 1 with probability $2^{-n^{1/3}}$ or $0(*)$ with probability $1-2^{-n^{1/3}}$.
3. For an $\vee$ node marked $1(*)$, mark each child $1(*)$ with probability $2^{-n/2}$ or 0 with probability $1-2^{-n/2}$.
4. For an $\vee$ node marked 0, mark all the children 0.

Finally let $\varrho(x_i)$ be the label assigned to x_i.

Additionally, we need the new concept of an identification. We will not make a general definition but only what we need in the present case.

Definition. For each $\varrho \in R_{k,n}$ let the identification ι work as follows. For all variables given the value $*$ in a k-block E by $\varrho \in R_{k,n}$, ι forces all these variables to be equal. Let y_E be the single variable associated with E.

Thus, given any function F in the original variables, $F\restriction_{\iota\varrho}$ will be a function in the variables $\{y_E\}$.

The idea behind these new identifications is the following. We want the value of the $\top_{2k}\restriction_{\iota\varrho}$ circuit corresponding to a k-block E to be equal to the new variable y_E with high probability. Thus applying a restriction to a game with $2f'(n)$ interactions will result in a game with $2f'(n)-2k$ interactions. We will make this precise in the next section.

7. The effect of $R_{k,n}$ on the $\top$ circuit

Our goal in this section is to show that the function computed by our $\top$ circuit after a restriction from $R_{k,n}$ has been applied is the same with high probability as the function computed by the circuit with $2k$ fewer interactions. In the remainder of the paper we will often write $R_{k,n}$ as R_k leaving the n implicit.

Lemma 7.1. *For any polynomial d and integer n such that $d(n)>2k$, $\top_{d-2k}=\top_d\restriction_{\iota\varrho}$ with probability (taken over $\varrho \in R_k$) at least $1-2^{-2^{n/4}}$ for $n \geqq c \log d$ where c is some absolute constant.*

Proof. Take a k-block in $\top_d$. Recall that $\varrho \in R_k$ first labels the threshold gates at height $2k$ with a $*$ and recursively labels the children by $*$, 1, or 0. After all the variables have been labeled all the starred variables are forced to be equal to a new variable y_E.

Define a *good* gate to be a gate that is either labeled 0(1) and takes the value 0(1) or is labeled $*$ and takes the value y_E. To prove Lemma 7.1 we only need to prove that the top node of every block is a good node. We prove slightly more, namely that with high probability every node is a good node. A node which is not good will be called *bad*. We say that an *error* occurs at a node if the node is bad but all its descendants are good. The key to proving Lemma 7.1 is given below.

Lemma 7.2. *The probability that an error occurs at an individual node is* $\leqq 2^{-2^{n/4}}$ *for* $n > n_0$ *for some absolute constant* n_0.

Proof. First observe that there cannot be an error at nodes at which rules 1 or 4 of $R_{n,k}$ were applied. Thus we only have to investigate rules 2 and 3. Let us start with the simpler rule 3 which deals with $\vee$ gates. Call the fixed node we are interested in l.

If l was marked 1 the only way there can be an error at this node is if no child was marked 1. The probability of this is

$$(1-2^{-n/2})^{2^n} \leqq e^{-2^{n/2}}.$$

The same analysis applies to the case where l is marked $*$.

Next let us investigate rule 2. Assume for definiteness that l is marked 0. Then the probability that there will be an error at l is bounded by the probability that at least a fraction $1-\tau = 2^{-n^{1/4}}$ of the children are marked 1. The probability of this is bounded by

$$\binom{2^n}{2^{n-n^{1/4}}} (2^{-n^{1/3}})^{2^n - n^{1/4}} \leqq \left(\frac{2^n e 2^{-n^{1/3}}}{2^{n-n^{1/4}}}\right)^{2^{n-n^{1/4}}} \leqq 2^{-2^{n/2}}$$

for $n > n_0$ for some constant n_0. This concludes the proof of Lemma 7.2. ∎

Now to prove Lemma 7.1 we just observe that since there are at most 2^{nd+1} nodes in a $\top_d$ circuit, the probability that an error occurs at some gate is at most $2^{nd+1-2^{n/2}}$. So the probability that all $\top_k$ circuits evaluate to y_E is at least $1-2^{-2^{n/4}}$ for $n > c \log d$ for some constant c. ∎

8. The construction of B

Before we can construct our oracle we will first need to show that small depth circuits cannot compute larger depth threshold circuits.

Theorem 8.1. *For all constants* r, *any polynomially bounded functions* $f(n)$ *and* $g(n) = o(f(n))$ *and any circuit* C_n *of depth* $g(n)$ *and size* 2^{2n^r} *where* $n > n_0(r, f, g)$ *there is a setting of the input variables for which* $C_n(x) \neq \top_{2f'(n)} \neq \dagger$.

Proof. To prove the theorem we will first show that if we hit an AND of OR's with small fanin with ι_Q then with high probability we can write the resulting function as an OR of ANDs with small fanin.

To state our main lemma, let $AND(H) \geqq s$ denote the event that the function H cannot be written as an OR of AND's of fanin $< s$.

Main Lemma. *Let* $G = \bigwedge_{i=1}^{w} G_i$, *where* G_i *are OR's of fanin* $\leqq n^r$ *where* $r \leqq \frac{k-2}{3}$. *Let* F *be an arbitrary function and* ϱ *a random restriction in* R_k. *Then for* $s \geqq 1$

$$Pr[AND(G\restriction_{\iota\varrho}) \geqq s \mid F\restriction_{\varrho} \equiv 1] \leqq 2^{-sn/5}$$

for $n > n_0(r)$.

Remark 1. By looking at $\neg G$ one can see that it is possible to convert an OR of ANDs to an AND of ORs with the same probability.

Remark 2. If there is no restriction ϱ satisfying the condition $F\restriction_{\varrho} \equiv 1$ we use the convention that the conditional probability in question is 0.

We will postpone the proof of the main lemma to the last section and for now use it to prove the following lemma.

Lemma 8.2. *Let* g *be any function bounded by a polynomial and let* H *be computed by a circuit of depth* $g(n)$, *bottom fanin at most* n^r, $r \geqq 1$, *and at most* 2^{2n^r} *gates at least distance 2 from the input. For* $k \geqq 3r+2$, $n > n_0(r)$, *and* $\varrho \in R_{k,n}$, *with probability at least* $1-2^{-n}$, $H\restriction_{\iota\varrho}$ *can be computed by a circuit of depth* $g(n)-1$, *bottom fanin* n^r *and at most* 2^{2n^r} *gates at least distance 2 from the input.*

Proof. Consider the given circuit C_n of depth $g(n)$ which computes H. Without loss of generality assume that the gates closest to the inputs are OR gates. By the main lemma with $F \equiv 1$, for $n > n_0(r)$ after applying $\iota\varrho$, each height 2 subcircuit can be rewritten as an OR of ANDs of fanin at most n^r with probability at least $1-2^{-n^{r+1/5}}$. The probability that some depth 2 subcircuit cannot be written as an OR of ANDs of fanin at most n^r is at most $2^{2n^r} 2^{-n^{r+1/5}} < 2^{-n}$ for $n \geqq 15$, $r \geqq 1$. Hence with high probability we can collapse the two consecutive levels of OR gates and write the resulting circuit as a depth $g(n)-1$ circuit of bottom fanin at most n^r with at most 2^{2n^r} gates of distance at least 2 from the inputs. The last fact follows since each gate a distance at least 2 from the inputs in the new circuit corresponds to a gate a distance at least 3 from the inputs in the old circuit. ∎

Now we can prove Theorem 8.1 as follows. Let $k = 2+3r$. Choose $n \geqq 15$ large enough so that the following are all true: Lemma 7.1 holds; the main lemma holds; $g(n)(2^{-n}+2^{-2^{n/4}}) \leqq 1/4$; and $f'(n) > kg(n)$. We will apply a series of $g(n)$ restrictions from R_k to both C_n and to $\top_{2f'(n)}$. C_n will be completely determined but the threshold circuit will be undetermined. Let us make this precise.

First, consider C_n as a depth $g(n)+1$ circuit with bottom fanin 1. After one application of a restriction from R_k with probability at least $1-2^{-n}$ we will be able to write the resulting circuit as a depth $g(n)$ circuit with bottom fanin at most n^r. Note that there will be at most 2^{2n^r} gates at least distance 2 from the input.

Now apply $g(n)-2$ restrictions from R_k in succession. By repeated application of Lemma 8.2 the resulting circuit will be of depth 2 and bottom fanin at most n^r with high probability. Finally, hit this resulting circuit with one more restriction from R_k. By the main lemma with $s=1$ and $F \equiv 1$ this will be a constant function with probability at least $1-2^{-n/5}$.

Now apply $g(n)$ restrictions from R_k to $\top_{2f'(n)}$. By Lemma 7.1 the resulting function will be $\top_t$ where $t=2f'(n)-2kg(n)$ with very high probability.

The probability that a sequence of $g(n)$ restrictions simultaneously determines C_n and reduces the threshold circuit to one with $2kg(n)$ fewer levels is at least $1-g(n)(2^{-n/5}+2^{-2^{n/4}})\geq 3/4$. Hence, such a sequence of restrictions exists. Clearly there exist many settings of the remaining variables such that $\top_t$ is different from the constant evaluated by C_n and is not $\dagger$. In particular set all of the remaining variables to the opposite of the value of C_n. ∎

Now we can prove our main theorem.

Theorem 8.3. *For any polynomially bounded function $f(n)$ computable in polynomial time and any $g(n)=o(f(n))$ there is an oracle B such that $AM^B[f(n)] \not\subset \Sigma^B_{g(n)}$.*

Proof. Assume that f is unbounded, since otherwise there is no integer valued g satisfying the condition of the theorem. We will set y^B_z in rounds. Let $M^B_1, M^B_2, \ldots$ be an enumeration of all alternating Turing machines which have at most $g(n)$ alternations and which run in polynomial time. Assume without loss of generality that M_i runs in time n^i. Let $\mathscr{C}_1, \mathscr{C}_2, \ldots$ be the corresponding families of circuits of depth $g(n)+1$ and size 2^{2n^i} as given by Lemma 3.1. Let n_0 be large enough such that $n_0 \geq 15$ and $(g(n_0)+1)(2^{-n_0/5}+2^{-2^{n_0/4}}) \leq 1/4$. Set y^B_z arbitrarily to 0 for $|z|=1$. Now repeat the following for all i.

Round i. Given M_i the corresponding circuit family $\mathscr{C}_i$ has size bounded by 2^{2n^i} and depth $g(n)+1$. Let n_i be the smallest integer such that $n_i > n_{i-1}$, $2f'(n_i)n_i > \max(n_{i-1}^i, 2f'(n_{i-1})n_{i-1})$, and $n_i > n_0(i, f, g)$ where $n_0(i, f, g)$ is the constant from Theorem 8.1. The second condition ensures that oracle queries of length $2f'(n_i)n_i$ have not been determined in previous rounds. Arbitrarily set to 0 all y^B_z with $n_{i-1}^i < |z| < 2n_i f'(n_i)$ and $2n_i f'(n_i) < |z| \leq \max(n_i^i, 2f'(n_i)n_i)$. Call the resulting circuit $C'_n(y^B_z)$. Now use Theorem 8.1 to set y^B_z with $|z|=2n_i f'(n_i)$ such that $C'_n(y^B_z) \neq \top_{f(n)}(y^B_z) \neq \dagger$.

Fact 1. *B is well defined.*

This follows from the fact that B is uniquely determined by the setting of the variables y^B_z and each of these variables is assigned a value precisely once in the construction.

Fact 2. *M_i^B does not decide $L(B)$ correctly on 1^{n_i}.*

This is by the construction and the correspondence between oracle machines and circuits. To conclude the proof of Theorem 8.3 we need only observe that by Lemma 5.4, $L(B) \in AM[f(n)]$. ∎

Finally using Lemma 4.1 we get

Corollary 8.4. *For any polynomially bounded function $f(n)$ which is computable in polynomial time and any $g(n)=o(f(n))$ there is an oracle B such that $AM^B[f(n)] \not\subset AM^B[g(n)]$.*

9. The hierarchy for restricted size

The proof that $AM[f(n)]=AM[cf(n)]$ for any constant c [3] utilizes in a critical way that the size of each message is only bounded by an arbitrary polynomial. In particular, when decreasing the number of interactions by a factor of two, the size of each message is roughly squared. The obvious question is whether this is necessary.

The answer appears to be yes since if the new size did not depend on c, c could be a function of n. But this contradicts our main theorem. In this section we will make the connection between the size of the game and the number of interactions more explicit.

Theorem 9.1. *Let $f(n)\geqq n^l$ for sufficiently large n and let $f(n)$ be computable in polynomial time. Let $c\geqq 28$ be an even integer such that $f(n)\geqq 2c$ for sufficiently large n and let $t\geqq\max\left(\frac{36l+36r_1}{c}, \frac{18r_1+18r_2+18l+18}{f(n)}\right)$. Then there is an oracle A such that $AM^A[f(n), t, t+l]\not\subset AM^A\left[\frac{1}{c}f(n), r_1, r_2\right]$.*

Remark. Observe that the second term in the bound for t matters only when f is bounded.

Proof. The proof is very close to the proof of the main theorem and hence let us only describe the key points. We will assume that $r_2\geqq r_1+l$ which is the case if Arthur looks at the entire conversation.

Of course, the language which will achieve the separation is identical to $L(B)$, except that in the definition each message is of length n^t. Thus $L(B)$ corresponds to a $\top^{n^t}_{2f'(n)}$ tree.

To any protocol in $AM^A\left[\frac{1}{c}f(n), r_1, r_2\right]$ corresponds by Lemma 4.1 a $\Sigma^A_{(1/c)f(n)+1}$ machine which runs in alternating time $\leqq\frac{1}{10}n^{2l+2r_1+1}$ and non-alternating time $\leqq n^{r_1+r_2+l+1}$ for sufficiently large n. Such a machine corresponds by Lemma 3.2 to a family of circuits C_n of depth $\frac{1}{c}f(n)+2$ with at most $2^{(1/5)n^{2l+2r_1+1}}$ gates at least 2 away from the inputs and of bottom fanin at most $n^{r_1+r_2+l+1}$.

As in the proof of the main theorem we only have to establish that C_n cannot compute the same function as $\top^{n^t}_{2f'(n)}$.

Let R_{k,n^t} be the space of random restrictions similar to $R_{k,n}$ but where n is replaced by n^t in the definition. Of course, the main lemma then is true for R_{k,n^t} with n replaced by n^t. Thus following the proof of the main theorem we can use a R_{k_1} restriction where $k_1=\left\lceil\frac{3(r_1+r_2+l+1)}{t}\right\rceil+2$ to eliminate the bottom level of C_n and obtain a circuit of depth one less with bottom fanin at most n^{2r_1+2l}. After this first round we can use a R_{k_2} with $k_2=\left\lceil\frac{6r_1+6l}{t}\right\rceil+2$ to eliminate a level of C_n and to maintain the fanin. Doing this $\frac{1}{c}f(n)$ rounds we have reduced C_n to a constant

while we have removed $2k_1+\frac{2}{c}f(n)k_2$ levels of the $\top_{2f'(n)}$ circuit. Thus to complete the proof we just have to check that

$$2k_1+\frac{2}{c}f(n)k_2 \leqq 2f'(n).$$

But this follows from

$$2k_1+\frac{2}{c}f(n)k_2 = 2\left\lceil\frac{3(r_1+r_2+l+1)}{t}\right\rceil+4+2\left(\left\lceil\frac{6r_1+6l}{t}\right\rceil+2\right)\frac{1}{c}f(n) <$$

$$< \frac{6(r_1+r_2+l+1)}{t}+6+\left(\frac{12r_1+12l}{t}+6\right)\frac{1}{c}f(n) \leqq 1$$

where the last inequality follows by the conditions on t and c.

Theorem 9.1 gives the following interesting corollary concerning languages that can be recognized in fixed size and constant number of interactions.

Corollary 9.2. *For any constant t there is an oracle A such that the hierarchy $(AM^A[k, t, t])_{k=1}^{\infty}$ contains an infinite number of levels.*

Please observe that Theorem 9.1 is not too far from optimal (except for the values of the constants, which can be improved) since Babai and Moran [3] prove that $AM[f(n), r_1, r_2]\subset AM\left[\frac{1}{c}f(n), kcr_1, r_2+kcr_1\right]$ for some constant k.

10. Main Lemma

In this section we prove our main lemma which is a version of the main lemma in [10], the difference being that we are presently working with the space R_k of random restrictions. Recall that $AND(G\lceil_{\varrho})\geqq s$ denotes the event that the function $G\lceil_{\varrho}$ cannot be written as an OR of ANDs of size $<s$.

Main Lemma 10.1. *Let $G=\bigwedge_{i=1}^{w} G_i$, where G_i are OR's of fanin $\leqq n^{(k-2)/3}$. Let F be an arbitrary function and ϱ a random restriction in R_k. Then for $s\geqq 1$*

$$Pr[AND(G\lceil_{\varrho})\geqq s|F\lceil_{\varrho}\equiv 1] \leqq 2^{-sn/5}$$

for $n>n_0(k)$.

Proof. We prove the lemma by induction on w the number of ORs in G. If $w=0$ the lemma is obvious ($G\equiv 1$). We first study what happens to G_1, the first OR in the circuit. Note that

$$Pr[AND(G\lceil_{\varrho})\geqq s|F\lceil_{\varrho}\equiv 1]$$

is less than the maximum of

$$Pr[AND(G\lceil_{\varrho})\geqq s|F\lceil_{\varrho}\equiv 1\wedge G_1\lceil_{\varrho}\equiv 1]$$

and

$$Pr[AND(G\lceil_{\varrho})\geqq s|F\lceil_{\varrho}\equiv 1\wedge G_1\lceil_{\varrho}\not\equiv 1]$$

The first term is

$$Pr[AND(G\lceil_{\varrho}) \geqq s | (F \wedge G_1)\lceil_{\varrho} \equiv 1].$$

However in this case

$$G\lceil_{\varrho} = \bigwedge_{i=1}^{w} G_i\lceil_{\varrho} = \bigwedge_{i=2}^{w} G_i\lceil_{\varrho}$$

since we are only concerned about ϱ's which forces G_1 to be 1. Thus $AND(G\lceil_{\varrho}) \geqq s$ is equivalent to saying that $\bigwedge_{i=2}^{w} G_i\lceil_{\varrho}$ cannot be written as an OR of ANDs of fanin at most s. But this probability is $\leqq 2^{-sn/5}$ by the inductive hypothesis since we are talking about a product of size $w-1$.

Now consider the second term

$$Pr[AND(G\lceil_{\varrho}) \geqq s | F\lceil_{\varrho} \equiv 1 \wedge G_1\lceil_{\varrho} \not\equiv 1].$$

Since we will be conditioning on these two events often we will denote them by $1_F \wedge \overline{1_{G_1}}$. Let T denote the set of variables occurring in G_1. Since we are looking at the case when G_1 is not made true by ϱ we have two possibilities. Either $G_1\lceil_{\varrho} \equiv 0$ or $G_1\lceil_{\varrho}$ is undetermined. The first case adds nothing to the above probability since in this case $G\lceil_{\varrho} \equiv 0$. Thus we only have to consider the second case. In this case there must be at least one variable $x_i \in T$ which is given the value $*$ by ϱ. We will say that a k-block E is *exposed* if there is a variable x_i such that $x_i \in E$, $x_i \in T$, and $\varrho(x_i) = *$. Let Z denote the set of blocks which have some variable in common with G_1 and let Y denote the set of exposed blocks. We let $exp(Y)$ denote the event that precisely the blocks of Y are exposed. For shorthand we will use $exp(E)$ rather than $exp(\{E\})$ when we are talking about a single block. By the above discussion we have

$$Pr[AND(G\lceil_{\varrho}) \geqq s \mid 1_F \wedge \overline{1_{G_1}}] \leqq \sum_{Y \subset Z, Y \neq \emptyset} Pr[AND(G\lceil_{\varrho}) \geqq s \wedge exp(Y) \mid 1_F \wedge \overline{1_{G_1}}]$$

$$= \sum_{Y \subset Z, Y \neq \emptyset} Pr[exp(Y) \mid 1_F \wedge \overline{1_{G_1}} \times Pr[AND(G\lceil_{\varrho}) \geqq s \mid 1_F \wedge \overline{1_{G_1}} \wedge exp(Y)].$$

The last equality follows by the definition of conditional probability. We derive a bound for the first factor in Lemmas 10.2 and 10.3 and we use induction for the second factor. Assume first for simplicity that only one block is exposed.

Lemma 10.2. *Let E be a k-block. Then*

$$Pr[exp(E) \mid 1_F \wedge \overline{1_{G_1}}] \leqq \frac{1}{4n^k} 2^{-n/5}$$

for sufficiently large n.

Proof. By the definition of conditional probability we want to prove

$$\frac{\sum'_{exp(E)} Pr(\varrho)}{\sum' Pr(\varrho)} \leqq \frac{1}{4n^k} 2^{-n/5}.$$

Here the $'$ indicates that we are only summing over ϱ's satisfying the condition $F\lceil_{\varrho} \equiv 1 \wedge G_1\lceil_{\varrho} \not\equiv 1$. If this quotient is $\frac{0}{0}$ we use the convention that it takes on the value 0.

Let ϱ be any restriction which satisfies $F\lceil_\varrho \equiv 1 \wedge G_1\lceil_\varrho \not\equiv 1 \wedge exp(E)$. To estimate the above quotient we will find a restriction $\tilde{\varrho}$ which only satisfies the first two conditions and gives a large contribution to the denominator. Let N denote the set of variables in $T \cap E$ which appear as $\bar{x}_i$ in G_1 and let P denote the ones that appear without negation.

Let f be the map from ϱ to $\tilde{\varrho}$ defined by the following rules.

1. $\tilde{\varrho}(x_k)=\varrho(x_k)$ for $x_k \notin N \cup P$.
2. $\tilde{\varrho}(x_k)=0$ for x_k in P.
3. $\tilde{\varrho}(x_k)=1$ for x_k in N.

In this way we maintain the condition $G_1\lceil_\varrho \not\equiv 1$. Unfortunately as defined so far $\tilde{\varrho}$ may not be a possible restriction for R_k. Look at the $\top_{2k}$ circuit which was used to label the variables in E. After 3 we may have $\vee$ gates at the bottom level labeled $*$ with children labeled 1. We are forced to relabel the $\vee$ to 1 and remark all remaining starred variables (those not in T) to 1. This in turn will force us to make further global changes to $\top_{2k}$. Starting at the $\vee$ gates of height 1 repeat steps 4 and 5 one level in the $\top_{2k}$ tree at a time until they cannot be applied further. Then percolate changes down all affected subtrees using rules 6—9.

4. If a child of an $\vee$ gate was changed from $*$ to 1 remark the $\vee$ gate and all remaining starred children from $*$ to 1.
5. If fewer than $n^{1/3}$ children (but at least one child) of an $\top$ gate were changed from $*$ to 1 remark the $\top$ gate and all remaining starred children from $*$ to 0.
   If at least $n^{1/3}$ children were changed from $*$ to 1 remark the $\top$ node and all remaining starred children from $*$ to 1.
6. If an $\vee$ gate was changed from $*$ to 1 remark all $*$ children to 1.
7. If an $\vee$ gate was changed from $*$ to 0 remark all $*$ children to 0.
8. If a $\top$ gate was changed from $*$ to 0 remark all $*$ children to 0.
9. If a $\top$ gate was changed from $*$ to 1 remark all $*$ children to 1.

We will establish several properties of $\tilde{\varrho}$.

Fact 1. No gate above level $2(k-1)$ in $\top_{2k}$ is changed.

This follows from the condition in rule 5. The number of $\top$ gates changed to 1 on level $2i$ is at most $n^{r-(i/3)}$ due to the restriction on the fanin of G_1. However, r is bounded by $(k-2)/3$ so that at most one $\top$ gate at level $2(k-2)$ changes from $*$ to 1 and hence at most one $\top$ gate at level $2(k-1)$ changes from $*$ to 0. Note that in particular the top node of $\top_{2k}$ is still marked $*$, and only variables in E are changed.

Fact 2. $\tilde{\varrho}$ satisfies $F\lceil_\varrho \equiv 1$ and $G_1\lceil_\varrho \not\equiv 1$.

Since we only change $*$'s to non $*$'s we cannot violate the first condition. Since we never change the value of variables in T after applying rules 2 and 3 $\tilde{\varrho}$ also satisfies $G_1\lceil_\varrho \not\equiv 1$.

By these two facts it follows that $\tilde{\varrho}$ gives a contribution to the denominator. Note that f is not $1-1$. Let $\bar{\varrho}$ be any restriction in the preimage of $\tilde{\varrho}$. To bound the quotient of the lemma we will bound

$$\frac{\sum_{\bar{\varrho} \in f^{-1}(\tilde{\varrho})} Pr[\bar{\varrho}]}{Pr[\tilde{\varrho}]}.$$

To do this it will be convenient to use the following concept of an atom.

Definition. An *atom* is a mapping from the each node of a T_{2k} tree to subsets of its 2^n children.

An atom a determines a restriction $\varrho \in R_k$ in the following manner: At each node v where a probabilistic choice has to be made, the children corresponding to the set $a(v)$ will be given the marking according to the alternative with the smaller probability. We will write this transformation from atom to restriction as $\varrho = h(a)$. There are two things to observe.

Observation 1. There are several atoms corresponding to the same original restriction. The reason for this is that at the nodes where there is no choice made (i.e., T gates labeled 1 and V gates labeled 0) it does not matter what value that atom takes.

Observation 2. There is a natural way to define probability on these atoms with respect to a given k-block. Let $\mathscr{T}$ and $\mathscr{V}$ be the set of T nodes and V nodes respectively. Then

$$Pr(a) = \prod_{v \in \mathscr{V}} (2^{-(n/2)})^{|a(v)|} (1-2^{-(n/2)})^{2^n - |a(v)|} \prod_{v \in \mathscr{T}} (2^{-n^{1/3}})^{|a(v)|} (1-2^{-n^{1/3}})^{2^n - |a(v)|}.$$

Using this definition, the probability of a labeling of a given k-block is

$$Pr(\varrho) = \sum_{a \in h^{-1}(\varrho)} Pr(a).$$

This can be seen as follows. Let $\mathscr{T}_1$ and $\mathscr{V}_0$ be the set of all T gates labelled 1 and V gates labelled 0 by ϱ respectively. All $a \in h^{-1}(\varrho)$ will agree on $\mathscr{T} - \mathscr{T}_1$ and $\mathscr{V} - \mathscr{V}_0$. Let $t_1, \ldots, t_q$ be an enumeration of $\mathscr{T}_1$ nodes and $v_1, \ldots, v_p$ be an enumeration of $\mathscr{V}_0$ nodes. Let $C(w)$ denote the children of a node w. Using the above definitions we get

$$\sum_{a \in h^{-1}(\varrho)} Pr(a) =$$

$$= \prod_{v \in \mathscr{V} - \mathscr{V}_0} (2^{-(n/2)})^{|a(v)|} (1-2^{-(n/2)})^{2^n - |a(v)|} \prod_{t \in \mathscr{T} - \mathscr{T}_1} (2^{-n^{1/3}})^{|a(v)|} (1-2^{-n^{1/3}})^{2^n - |a(v)|} \times$$

$$\times \Big(\sum_{a(v_1) \subseteq C(v_1)} (2^{-(n/2)})^{|a(v_1)|} (1-2^{-(n/2)})^{2^n - |a(v_1)|} \Big) \ldots$$

$$\ldots \Big(\sum_{a(t_q) \subseteq C(t_q)} (2^{-n^{1/3}})^{|a(t_q)|} (1-2^{-n^{1/3}})^{2^n - |a(t_q)|} \Big).$$

Note that all the sums are equal to 1. Now it is clear that the remaining product is precisely the probability of ϱ.

Obtaining $\tilde{\varrho}$ from $\bar{\varrho}$ can be done by changing the atoms corresponding to $\bar{\varrho}$ into atoms corresponding to $\tilde{\varrho}$ and then obtaining $\tilde{\varrho}$ from these atoms. Translating the mapping f of $\bar{\varrho}$ to $\tilde{\varrho}$ into the mapping g on atoms results in the following rules. We make fewer changes but some nodes become "active" without changing.

1. No operation.
2. Let v be a leaf corresponding to a variable in P. If $v \in a(parent(v))$ (i.e., if v was labelled $*$ by $\bar{\varrho}$) then remove v from $a(parent(v))$.
3. A variable $x_i \in N$ given the value $*$ becomes active.
4. If v corresponds to an V gate and one of its children is active, it becomes active.

5. If v corresponds to a $\top$ gate and has at least one but fewer than $n^{1/3}$ active children then remove v from $a(parent(v))$ and add the active children to $a(v)$. If v has more than $n^{1/3}$ active children it becomes active.

6—9. No operation.

To check that g on atoms corresponds to f on restrictions is a simple but tedious verification which we leave to the reader.

The mapping g is still not $1-1$. However, it is simple enough that we will be able to establish the following claim.

Claim. For all $\tilde{\varrho}$ and all $a \in h^{-1}(\tilde{\varrho})$,

$$\sum_{b \in g^{-1}(a)} Pr(b) \leqq (2^{-n/5}/4n^k) Pr[a].$$

Observe that the claim implies Lemma 10.2:

$$\sum_{\bar{\varrho} \in f^{-1}(\tilde{\varrho})} Pr[\bar{\varrho}] = \sum_{\bar{\varrho} \in f^{-1}(\tilde{\varrho})} \sum_{b \in h^{-1}(\bar{\varrho})} Pr[b] = \sum_{a \in h^{-1}(\tilde{\varrho})} \sum_{b \in g^{-1}(a)} Pr[b] \leqq \frac{2^{-n/5}}{4n^k} \sum_{a \in h^{-1}(\tilde{\varrho})} Pr[a]$$

and the last sum is just $Pr[\tilde{\varrho}]$.

Thus we only have to establish the claim. Consider an atom a such that $h(a)=\tilde{\varrho}$. Say $b \in g^{-1}(a)$. The only nodes at which they may differ are the nodes at which rules 2 and 5 apply. Let M_b be the set of nodes at which rule 5 applies in the map from b to a; let $m_b=|M_b|$. Let L_b, l_b, be defined similarly for rule 2. Also, let $c_{i,b}$ be the number of children changed when rule 5 was applied for the ith time during the map from b to a. Note that $1 \leqq c_{i,b} \leqq n^{1/3}$ and that $l_b+m_b \geqq 1$. Using the above definitions we have

$$Pr(b) = \left(\frac{2^{-n/2}}{1-2^{-n/2}}\right)^{l_b+m_b} \prod_{i=1}^{m_b} \left(\frac{1-2^{-n^{1/3}}}{2^{-n^{1/3}}}\right)^{c_{i,b}} Pr(a).$$

To bound the sum of the probabilities over all $b \in g^{-1}(a)$ let M be the union of all M_b's where $b \in g^{-1}(a)$; let $m=|M|$. Define L and l similarly and let $Ch(i, c_{i,b})$ be the number of ways of choosing the $c_{i,b}$ children changed at the i'th node where rule 5 is applied. We get

$$\sum_{b \in g^{-1}(a)} Pr(b) \leqq \sum \binom{l}{l_b}\binom{m}{m_b} \times$$

$$\times \left(\frac{2^{-n/2}}{1-2^{-n/2}}\right)^{l_b+m_b} \sum_{1 \leqq c_{i,b} \leqq n^{1/3}} \prod_{i=1}^{m_b} Ch(i, c_{i,b}) \left(\frac{1-2^{-n^{1/3}}}{2^{-n^{1/3}}}\right)^{c_{i,b}} Pr(a)$$

where the first sum is taken over $1 \leqq l_b+m_b \leqq l+m$.

To complete the calculation we will need a bound on $l+m$ and $Ch(i, c_{i,b})$. We claim that $l+m \leqq n^r = n^{(k-2)/3}$. To see this, first note that rule 2 might be applied only at variables of P labelled $*$ by $\bar{\varrho}$. Hence, $l \leqq |P|$.

Now let us investigate where rule 5 might apply. Rule 5 of g corresponds to changing the label of a $\top$ gate from $*$ to 0 during the map f. Say we are given $\bar{\varrho} \in f^{-1}(\tilde{\varrho})$ and $v \in N$ labelled $*$ by $\bar{\varrho}$. Observe that for a leaf to be labelled $*$ by $\bar{\varrho}$, $\bar{\varrho}$ must have labelled all its ancestors on the path from the root to the leaf with a $*$. Once v is changed from $*$ to 1, then by application of rules 4 and 5 of f, nodes up the path will change from $*$ to 1 until eventually a $\top$ node, t, changes from $*$ to 0 (note that this is guaranteed to happen eventually). Once t changes from $*$ to 0, none of the changes which occurred in the subtree rooted at t contribute to

any changes higher up in the tree. Hence, for this $\bar{\varrho}$ v contributes to Rule 5 of g being applied only at t. Moreover, for all $\tilde{\varrho}\in f^{-1}(\bar{\varrho})$ which label v with $*$, v changing from $*$ to 1 contributes to exactly t changing from $*$ to 0. This is true since we now know that $\tilde{\varrho}$ must label the path from v to t with all ones then a zero. Hence, there can be no application of rule 5 below t and, again, any application of rule 5 above t cannot be due to changes in the subtree rooted at t. So, for every variable in N which is ever labelled $*$ by some $\bar{\varrho}\in f^{-1}(\bar{\varrho})$ we can associate exactly one node in M. Hence, $m\leqq|N|$ and $l+m\leqq|N|+|P|$ which is bounded by $n^{(k-2)/3}$ by the restriction on the fanin of G_1.

By the same argument, at any node of M there are at most $n^{(k-2)/3}$ children which could change when rule 5 is applied. Thus

$$Ch(i, c_{i,b}) \leqq \binom{n^{(k-2)/3}}{c_{i,b}} \leqq \binom{n^{(k-2)/3}}{n^{1/3}}.$$

Using this and continuing with the calculation we have

$$\sum_{b\in g^{-1}(a)} Pr(b) \leqq$$

$$\leqq \sum_{1\leqq l_b+m_b\leqq l+m} \binom{l}{l_b}\binom{m}{m_b}\left(\frac{2^{-n/2}}{1-2^{-n/2}}\right)^{l_b+m_b}\left(n^{1/3}\binom{n^{(k-2)/3}}{n^{1/3}}\right)^{m_b} 2^{2n^{1/3}m_b} Pr(a) \leqq$$

$$\leqq \sum_{1\leqq l_b+m_b\leqq l+m} \binom{l}{l_b}\binom{m}{m_b}\left(\frac{2^{-n/2}}{1-2^{-n/2}}\right)^{l_b+m_b} 2^{m_b n^{1/3}((k-2)/3\log n+3)} Pr(a) \leqq$$

$$\leqq \sum_{1\leqq l_b+m_b\leqq l+m} \binom{l}{l_b}\binom{m}{m_b}\left(\frac{1}{8n^{3k}}2^{-n/5}\right)^{l_b+m_b} Pr(a) =$$

$$= \sum_{i=1}^{l+m}\binom{l+m}{i}\left(\frac{1}{8n^{3k}}2^{-n/5}\right)^{i} Pr(a) = \left(\left(1+\frac{1}{8n^{3k}}2^{-n/5}\right)^{l+m}-1\right) Pr(a) \leqq$$

$$\leqq \left(\left(1+\frac{1}{8n^{3k}}2^{-n/5}\right)^{n^{(k-2)/3}}-1\right) Pr(a) \leqq \frac{1}{4n^k}2^{-n/5} Pr(a).$$

In the calculation we assumed that n was sufficiently large.

Thus we get the desired estimate for the quotient and we have proved Lemma 10.2. ∎

Next we have

Lemma 10.3. *For sufficiently large n*

$$Pr[exp(Y) \mid 1_F \wedge \overline{1_{G_1}}] \leqq \left(\frac{1}{4n^k}\right)^{|Y|} 2^{-|Y|n/5}.$$

Proof. The proof is almost identical to the proof of Lemma 10.2. Instead of changing ϱ on only one block we do the same changes independently on all blocks in Y. Thus we gain a factor $\frac{1}{4n^k}2^{-n/5}$ for each block and the result follows. ∎

Now we estimate the other factor needed for the main lemma 10.1. Namely,

$$Pr[AND(G\restriction_{t\varrho}) \geqq s \mid 1_F \wedge \overline{1_{G_1}} \wedge exp(Y)].$$

We want to use induction. To do this we have to get rid of the two last conditions. The blocks in Y correspond to $|Y|$ remaining variables after ι. We will try all possibilities of these variables and eliminate these blocks from the future probabilities.

By the condition $exp(Y)$ the variables in G_1 not in the blocks contained in Y were all given non $*$ values. Since these variables do not make G_1 true and do not take the value $*$, they take a fixed value. This conditioning can easily be incorporated in $F\lceil_\varrho \equiv 1$ by changing F.

We have

$$Pr[AND(G\lceil_{\iota\varrho}) \geqq s \mid 1_F \wedge \overline{1_{G_1}} \wedge exp(Y)] \leqq$$

$$\leqq \max_{\varrho_0} Pr[AND(G\lceil_{\iota\varrho}) \geqq s \mid 1_F \wedge \overline{1_{G_1}} \wedge exp(Y) \wedge \varrho_Y = \varrho_0]$$

where we maximize over all behaviors, ϱ_Y, of ϱ on the blocks Y. This is in its turn bounded by:

$$\sum_{\sigma \in \{0,1\}^{|Y|}} Pr[AND(G\lceil_{\iota\varrho\sigma}) \geqq s - |Y| \mid F'\lceil_\varrho \equiv 1]$$

where σ is a part of a possible minterm and F' is a modification of F. The stars of ϱ in the blocks Y are substituted in F' by taking AND of the two formulas resulting from F by substituting 0 and 1. This can be done since making a function 1 even when some variable is undetermined is the same as making the function 1 in the two cases where 0 and 1 are substituted for the variable. Non-$*$ values are just included as usual equalities.

Each of these probabilities can now be estimated by $2^{-(s-|Y|)n/5}$ by the induction hypothesis (with a restriction with fewer blocks). The size of the ANDs we are looking for has decreased by $|Y|$ since we have included the variables corresponding to Y.

Finally, since there are $2^{|Y|}$ possible σ we need to evaluate the sum.

$$\sum_{Y \subset Z, Y \neq \emptyset} 2^{-|Y|n/5} \left(\frac{1}{4n^k}\right)^{|Y|} 2^{|Y|} 2^{-(s-|Y|)n/5} =$$

$$= 2^{-sn/5} \sum_{i=1}^{|Z|} \binom{|Z|}{i} \left(\frac{1}{2n^k}\right)^i = 2^{-sn/5} \left(\left(1+\frac{1}{2n^k}\right)^{|Z|} - 1\right) \leqq 2^{-sn/5}$$

since $|Z| \leqq |T| \leqq n^{(k-2)/3}$.

This finishes the induction step and the proof of the Main Lemma. ∎

11. Discussion

The relation between *co-NP* and *AM*[*Poly*] remains an interesting open problem at this point. Evidence that *co-NP* is not contained in *AM*[2] has been given by Boppana, Hastad and Zachos [4] who showed that if *co-NP*$\subset$*AM*[2] then the polynomial time hierarchy collapses to *AM*[2]. If in fact *co-NP*$\not\subset$*AM*[2] it would be interesting to resolve whether *co-NP*$\subset$*AM*[*Poly*]. As an indication that here the answer might also be no, Fortnow and Sipser [5] proved that there is an oracle A such that *co-NP*$^A \not\subset AM^A$[*Poly*].

One disturbing detail when proving the size hierarchy result when f is con-

stant is that we need to bound the running time of Arthur. This should not really be necessary since intuitively there should be no way to compensate for lost communication by doing more polynomial time computation. However we did not see how to prove this.

Acknowledgments. We would like to thank Oded Goldreich for many valuable comments.

References

[1] W. AIELLO, S. GOLDWASSER and J. HASTAD, On the Power of Interaction, *Proc. of the 27th IEEE Symposium on Foundations of Computer Science*, 368—379, Toronto, **1986**.
[2] L. BABAI, Trading Group Theory for Randomness, *Proc. of the 17th ACM Symposium on Theory of Computing*, 421—429, Providence, **1985**.
[3] L. BABAI and S. MORAN, Arthur—Merlin Games: a Randomized Proof System, and a Hierarchy of Complexity Classes, *JCSS*, **36** (1988), No. 2, 254—276.
[4] R. BOPPANA, J. HASTAD and S. ZACHOS, Does *co-NP* have Short Interactive Proofs?, *Information Processing Letters*, **25** (1987), No. 2, 127—132.
[5] L. FORTNOW and M. SIPSER, Are There Interactive Protocols for *co-NP*?, *Information Processing Letters*, **28** (1988), 249—251.
[6] M. FURST, J. SAXE and M. SIPSER, Parity, Circuits, and the Polynomial Time Hierarchy, *Math. System Theory*, **17** (1984), 13—27.
[7] O. GOLDREICH, Y. MANSOUR and M. SIPSER, Interactive Proof Systems: Provers that Never Fail and Random Selection, *Proc. of the 28th IEEE Symposium on Foundations of Computer Science*, 449—461, Los Angeles, **1987**.
[8] S. GOLDWASSER, S. MICALI and C. RACKOFF, The Knowledge Complexity of Interactive Proofs, *Proc. of the 17th ACM Symposium on Theory of Computing*, 291—305, Providence, **1985**, also in *SIAM J. on Computing*, **18** (1989), No. 1, 186—208.
[9] S. GOLDWASSER and M. SIPSER, Private Coins vs. Public Coins in Interactive Proof Systems, *Proc. of the 18th ACM Symposium on Theory of Computing*, 59—68, Berkeley, **1986**.
[10] J. HASTAD, Almost Optimal Lower Bounds for Small Depth Circuits, *Proc. of the 18th ACM Symposium on Theory of Computing*, 6—20, Berkeley, **1986**.
[11] J. HASTAD, *Computational Limitations of Small Depth Circuits*, Ph. D. thesis, MIT, **1986**.
[12] N. NISAN and A. WIGDERSON, Hardness vs. Randomness, *Proc. of the 29th IEEE Symposium on Foundations of Computer Science*, 2—11, White Plains, **1988**.
[13] M. SIPSER, Borel Sets and Circuit Complexity, *Proc. of the 15th ACM Symposium on Theory of Computing*, 61—69, Boston, **1983**.
[14] A. YAO, Separating the Polynomial-Time Hierarchy by Oracles, *Proc. of the 26th IEEE Symposium on Foundations of Computer Science*, 1—10, Portland, **1985**.
[15] S. ZACHOS, Probabilistic Quantifiers, Adversaries and Complexity Classes; An Overview, *Structures in Complexity Theory*, Lecture Notes in Computer Science, **233** (1986), 383—398.

William Aiello

Bell Communications Research
Morristown, New Jersey, USA

Shafi Goldwasser

Laboratory of Computer Science and
Dept. of Electrical Engineering and Computer Science
Massachussets Institute of Technology
Cambridge, Massachusetts, USA

Johan Hastad

Department of Numerical Analysis and
Computer Science
Royal Institute of Technology, Stockholm, Sweden