JOURNAL OF COMPUTER AND SYSTEM SCIENCES 36, 254–276 (1988)

Arthur–Merlin Games: A Randomized Proof System, and a Hierarchy of Complexity Classes

LÁSZLÓ BABAI

Eötvös University, Budapest, Hungary and
University of Chicago, Chicago Illinois

AND

SHLOMO MORAN

Technion, Haifa, Israel

Received June 24, 1986; revised August 3, 1987

One can view NP as the complexity class that captures the notion of efficient provability by classical (formal) proofs. We consider broader complexity classes (still "just above NP"), in the hope to formalize the notion of efficient provability by *overwhelming statistical evidence.* Such a concept should combine the nondeterministic nature of (classical) proofs and the statistical nature of conclusions via Monte Carlo algorithms such as a Solovay–Strassen style "proof" of primality. To accomplish this goal, two randomized interactive proof systems have recently been offered independently by S. Goldwasser, S. Micali, and C. Rackoff (GMR system) (*in* "Proceedings, 17th ACM Symp. Theory of Comput., Providence, RI, 1985," pp. 291–304) and by L. Babai (Arthur–Merlin system) (*in* "Proceedings, 17th ACM Symp. Theory of Comput., Providence, RI, 1985," pp. 421–429), respectively. The proving power of the two systems has subsequently been shown by S. Goldwasser and M. Sipser (*in* "Proceedings, 18th ACM Symp. Theory of Comput., Berkeley, CA, 1986") to be equivalent. In both systems, a nondeterministic prover (Merlin) tries to convince a randomizing verifier (Arthur) that a certain string x belongs to a language L. The verifier operates under polynomial time constraint. The GMR system uses private coin tosses whereas in the Arthur–Merlin proof system, coin tosses are public. In this paper we give an exposition of the Arthur–Merlin system. We describe the resulting hierarchy of complexity classes $\mathrm{AM}(t(n))$, where $t(n)$ is the number of rounds of interaction on inputs of length n. The "Collapse Theorem" (Babai, *loc. cit.*) states that for $t(n) \geqslant 2$, $\mathrm{AM}(t(n)+1) = \mathrm{AM}(t(n))$. In particular, the finite levels of the hierarchy collapse to $\mathrm{AM} \stackrel{\text{def}}{=} \mathrm{AM}(2)$. We prove the following stronger version ("Speedup Theorem"): for $t(n) \geqslant 2$, $\mathrm{AM}(2t(n)) = \mathrm{AM}(t(n))$. This complements a result of W. Aiello, J. Hastad, and S. Goldwasser (*in* "Proceedings, 27th IEEE Symp. Found. of Comput. Sci., 1986," pp. 368–379), saying that in a relativized world, no unbounded reduction of the number of rounds is possible. R. Boppana, J. Hastad, and S. Zachos, (*Inform. Process. Lett.*, in press) provided a strong piece of evidence to support the view that AM is "not much larger" than NP by showing that if AM contains coNP then the polynomial time hierarchy collapses to $\sum_2^p = \prod_2^p = \mathrm{AM}$. We show that this is an immediate consequence of the Collapse Theorem. A combination of the result of S. Goldwasser and M. Sipser, a striking observation by O. Goldreich, S. Micali, and A. Widgerson, and the Collapse Theorem imply that graph non-isomorphism belongs to AM. We give a direct proof of this fact and generalize it to the coset intersection problem for permutation groups. In view of the Boppana–Hastad–Zachos

0022-0000/88 $3.00

result, one obtains the strongest evidence yet against NP-completeness of graph isomorphism and coset intersection. Using random oracles, we define the class *almost*-NP. This class, another natural randomized extension of NP, has, as an immediate consequence of recent work by Noam Nissan, turned out to be equal to AM. This provides additional evidence of the robustness of the class AM.

1. INTRODUCTION

1.1. *What Is a Proof?*

Maybe Fermat had a proof. But an important party was certainly missing to make the proof process complete: the *verifier*. Each time rumor gets around that a student somewhere proved $P = \text{NP}$, people ask "Has Karp seen the proof?" (they hardly ever ask the student's name). Perhaps the verifier *is* more important than the *prover*. (Should Dick Karp himself come up with such a proof, we would perhaps be inclined to trust the news, not because we believe he could conceivably prove such a result, but rather on the assumption that he must have handed the proof over to Karp, the trusted verifier.)

Since the creation of formal systems, the element of *interaction* in the proof process has been ignored in mathematics. Intuitively, though, everyone appears to be aware that a proof is not a sequence of symbols, but a process of *convincing a distrustful party*. Phrases like "we leave the other 27 cases to the reader" would certainly not be admissible in a formal system, but this does not appear to worry mathematical authors too much. Exceedingly long proofs (like that of the Classification of Finite Simple Groups, admittedly over 5000 pages scattered over a literature of 100 years) and proofs depending on an excessive amount of machine computation (like the Four Color Theorem) pose particular philosophical problems and raise the question that (for different reasons) there might be a positive probability that a proof is in error. It would, however, be very difficult to quantify this probability. Undetected hardware failure is highly unlikely, a software error is quite possible and an oversight in some parts of a long proof is very likey. In the last case, however, the question is rather, what are the chances of a *significant* error, an elusive notion.

All these philosophical difficulties are avoided in another nonclassical method of proving mathematical assertions "beyond reasonable (or even utterly unreasonable) doubt." Imagine that a 300-digit number N is being tested for compositeness via a Solovay–Strassen style Monte Carlo test. Suppose 1000 independent applications of the test fail to come up with a *witness* of compositeness. On the other hand, if N were composite, each test would have more than 50% chance of producing a witness so the probability of such outcome would be less than 2^{-1000}. (This is much less than the probability of multiple hardware failures or of cosmic rays changing critical bits.) The odds against N being composite are so overwhelming that it should be considered quite safe to declare N a prime—certainly much safer than claiming the truth of either of the above mentioned, widely accepted theorems, (Not

to speak of the degree of certainty gained from Gary Miller's deterministic primality test which depends on the extended Riemann hypothesis.)

Problems of philosophical nature will hardly ever be completely eliminated in this area. Here, for instance, the notion of truly independent random choices can be brought into question. This problem will not concern us in the present discussion but we point out that in order for such a randomized proof to be valid, *the coins must be tossed by the verifier.* This fact alone requires a greater degree of interaction than do classical proofs and in particular presents difficulties for third parties to accept the outcome.

We shall introduce a notion of *randomized (interactive) proofs*, combining the nondeterministic nature of a classical proof system (the prover just prints the proof, the verifier does not ask how he got it) with the randomization performed by the verifier as in the Solovay–Strassen test. Although this notion itself has nothing to do with how *efficient* the proof is (i.e., how long it takes to verify it), the motivation behind it is complexity theoretic and therefore we combine the two aspects (randomization and efficiency) in the definition.

1.2. *Efficient Provability*

King Arthur recognizes the supernatural intellectual abilities of Merlin but does not trust him. How should Merlin convince the intelligent but impatient King that a string x belongs to a given language L (presumably a matter of life or death)?

If $L \in \mathrm{NP}$, Merlin will be able to present a *witness* which Arthur can check in polynomial time. This corresponds to the classical notion of proof, combined with a criterion of efficiency.

We shall introduce a proof system which permits Merlin to efficiently prove a conceivably broader class of propositions to Arthur provided Arthur accepts proofs by overwhelming statistical evidence.

As a result, we obtain a hierarchy of complexity classes "just above NP." Membership in languages belonging to these classes have efficient proofs by *Arthur–Merlin protocols.*

1.3. *The Arthur–Merlin Hierarchy*

An *Arthur–Merlin protocol* defines a combinatorial game, to be played by Arthur, whose moves are random, and Merlin, who is capable of making optimal moves.

The input is a string x. Merlin intends to convince Arthur that x belongs to some language L. (Any theorem can be thought of as such a statement.)

In a predetermined number of moves, Arthur and Merlin take turns in printing finite strings on a tape. In the end, a deterministic Turing machine, the *referee*, known in advance to both players, evaluates the input and the moves and declares the winner. Arthur's moves being random, if Merlin plays optimally, his winning chance $W(x)$ depends on x only. It is required that for every x, either

(a) $W(x) > \frac{2}{3}$ (x is accepted), or

(b) $W(x) < \frac{1}{3}$ (x is rejected).

The *length* of the game is the total number of moves; the *size* of the game is the total length of the strings printed.

In order to take efficiency into account, we require that the length and the size of the game as well as the running time of the referee be bounded by a polynomial of the length of x.

The language defined by such a protocol is the set of strings x for which Merlin's winning chances are greater than $\frac{1}{2}$ (and thus greater than $\frac{2}{3}$).

Let $t(n)$ be a polynomially bounded function of n. Throughout this paper, n will denote the length of the input: $n = |x|$. Languages accepted by Arthur–Merlin games of length $\leqslant t(n)$ form the classes $\mathrm{AM}(t(n))$ (Arthur moves first) and $\mathrm{MA}(t(n))$ (Merlin moves first). Let further

$$\mathrm{AM}(\mathrm{poly}) = \mathrm{MA}(\mathrm{poly}) = \bigcup \{\mathrm{AM}(n^k): k > 0\}$$

(games of polynomial length). These complexity classes form the Arthur–Merlin hierarchy.

For $t(n) = c$ (constant) we use strings of length c to indicate the sequence of players. For example,

$$\mathrm{AM}(3) = \mathrm{AMA}, \qquad \mathrm{MA}(4) = \mathrm{MAMA}, \qquad \mathrm{MA}(1) = \mathrm{M}.$$

It should be clear that $\mathrm{MA}(1) = \mathrm{NP}$ and $\mathrm{AM}(1) = \mathrm{BPP}$. The class $\mathrm{AM} = \mathrm{AM}(2)$ will play a particularly important role. We shall prove that the finite levels of the hierarchy collapse to AM. All known natural examples of languages in AM(poly) actually belong to AM. Moreover, all languages in AM belong to NP^C for almost every oracle C. (We use the second half of this sentence to define the class *almost*-NP, see Subsection 1.11).

While the definition of the classes of the Arthur–Merlin hierarchy bears strong resemblance to the polynomial time hierarchy of Meyer and Stockmeyer [MS, St76, CKS, cf. GJ, p. 162], the differences seem more prominent than the analogies. The interplay between the two hierarchies, nevertheless, plays a key role in relativized separation arguments involving the Arthur–Merlin classes (cf. subsections 1.8 and 1.9).

1.4. *Inheriting Forefathers' Proofs*

Another advantage of the complexity class AM over higher levels of the hierarchy is that still, only a limited interaction is required for a proof of membership in languages $L \in \mathrm{AM}$. Classical proofs have the advantage that they can be written up and later generations of eager verifiers can convince themselves of the correctness of old proofs. This is no longer the case if Merlin has to respond to Arthur's random string. Nevertheless, if humanity can agree on a certified random string (such as the output of a Geiger–Müller counter), such a string can replace Arthur's opening move and Merlin's reply might be worth recording for posterity. This solution is,

however, no longer available, if Arthur has to move *after* a move of Merlin and then Merlin has to reply again.

An alternative, for the case when either no such universal random string has been agreed upon, or the protocol involves an MAM move sequence, is that the verifier be a generally trusted coin tosser. Coin tosses could also be generated by a committee which is guaranteed to have at least one trustworthy member. In this case, results of Santha, Vazirani, Vazirani [SV, VV], Chor, and Goldreich [CG] concerning slightly random ("probability bounded") sources become relevant. They prove that BPP can be recognized using slightly random sources; one might expect their result to generalize to AM(poly).

1.5. *The Goldwasser–Micali–Rackoff Hierarchy: Another Randomized Proof System*

Arthur–Merlin protocols were introduced in [Ba1] with the aim of constructing slight randomized extensions of NP in order to accommodate certain languages in as low complexity classes as possible. Such languages include nonmembership in, order, and isomorphism of matrix groups over finite fields, neither of which is known to belong to NP. (Cf. [Ba 2] for details.)

In contrast to this "minimal" approach, Goldwasser, Micali, and Rackof [GMR] independently introduced a randomized interactive proof system, hoping to capture the widest possible class of "efficiently verifiable" languages.

GMR protocols are similar to Arthur–Merlin protocols. The difference is that the GMR verifier employs private coins (not seen by the prover, but, of course, seen by the referee). Instead of revealing the random string, at each move the verifier feeds the history of the game (fully known to him) into a polynomial time bounded Turing machine and passes the output on to the prover. The prover has to make an optimal move, based on this partial information.

The resulting complexity classes $\mathrm{IP}(t(n))$ ($t(n)$ moves, verifier moves first) seem at first wider than the corresponding classes $\mathrm{AM}(t(n))$; the inclusion $\mathrm{AM}(t(n)) \subseteq \mathrm{IP}(t(n))$ is immediate. A surprising recent result of Goldwasser and Sipser [GS] states that $\mathrm{IP}(t(n)) \subseteq \mathrm{AM}(t(n)+2)$, which, combined with the Collapse Theorem yields the pleasing conclusion that $\mathrm{AM}(t(n)) = \mathrm{IP}(t(n))$, proving the full equivalence of the two proof systems.

When studying inclusions involving these complexity classes (as we do in this paper), the simplicity of the Arthur-Merlin system makes it the system of choice [Ba1, AHG, FS, BHZ]. On the other hand, very elegant and simple protocols have been described in the GMR system, with the added feature of providing "zero knowledge proofs," a conceptual leap of great significance, which Arthur–Merlin protocols do not seem to be able to simulate.

1.6. AM *Protocols for Graph Nonisomorphism and Coset Disjointness*

The most convincing example of a randomized interactive proof has been found by Goldreich, Micali and Wigderson [GMW]. They describe a strikingly simple GMR protocol of class IP(2) for the graph nonisomorphism problem. This problem is not known to belong to NP. The result, combined with [GS, Ba1], implies that

there exists an AM = AM(2) protocol to verify graph nonisomorphism. We shall give a direct proof of this fact in the last section and generalize it to the coset disjointness problem for permutation groups.

Other natural problems in AM include nonmembership in and order of matrix groups over finite fields as well as isomorphism of such groups. Some of these results are outlined in [Ba1]; the full proofs will appear elsewhere [Ba2, BLS].

1.7. *Games against Nature*

Man wants to win, nature is indifferent. Papadimitriou [Pa] introduced the term "games against nature" to describe complexity classes arising from polynomially bounded games against an indifferent, randomizing adversary. Arthur–Merlin games are particular "games against nature," the crucial restriction being the condition that the winning chances are always *bounded away* from $\frac{1}{2}$ (conditions (a), (b) in Subsection 1.3). In the absence of such restriction one obtains substantially more powerful complexity classes. The finite levels of Papadimitriou's hierarchy are equivalent to the polynomial time hierarchy; and with a polynomial number of rounds one obtains another description of PSPACE [Pa].

1.8. *Relation to the Polynomial Time Hierarchy*

It is known, that BPP is contained within the polynomial time hierarchy [Sip]; in fact it is contained in $\sum_2^p \cap \prod_2^p$ (P. Gács, see [Sip]). A very elegant proof of this fact was given by Clemens Lautemann [Lau]. The proof directly generalizes to AM and MA and gives the following result.

PROPOSITION 1. (a) $\mathrm{AM} \subseteq \prod_2^p$.

(b) $\mathrm{MA} \subseteq \sum_2^p \cap \prod_2^p$.

The idea of the proof is, that, as in the proof of the result on BPP, the "random" quantifier ($\mathcal{R}$) can be replaced by an existential quantifier and a universal quantifier, in either order. Membership of a string x in a language $L \in \mathrm{AM}$ can be defined by an expression of the form $\mathcal{R}y\exists z\phi(x, y, z)$, hence in this case $\mathcal{R}y$ has to be replaced by $\forall u \exists v$ to yield (a). The proof of (b) goes analogously, using, in addition, our result that $\mathrm{MA} \subseteq \mathrm{AM}$. We omit the details. ∎

The unbounded levels of the Arthur–Merlin hierarchy are not believed to be contained in a finite level of the polynomial time hierarchy. Evidence to this effect has been furnished by Aiello, Hastad, and Goldwasser [AHG]. They prove that for any unbounded $t(n)$, there exists an oracle C such that $\mathrm{AM}(t(n))^C$ is not contained in $PH^C = \bigcup_{k>0} \sum_k^{P,C}$.

Another plausible relation is that AM (and even AM(poly)) does not contain coNP. This, of course, would imply NP ≠ coNP. Nevertheless, supporting evidence can be found. Boppana, Hastad, and Zachos have recently proved the following:

PROPOSITION 2. [BHZ] *If* $\text{coNP} \subseteq \text{AM}$ *then the polynomial time hierarchy collapses to* $\sum_2^P = \prod_2^P = \text{AM}$.

(In the next subsection we derive this result as an immediate consequence of the Collapse Theorem).

Another piece of evidence is relativized separation. Fortnow and Sipser [FS] have constructed an oracle C such that $\text{AM}^C \not\supseteq \text{coNP}^C$.

Another source of analogous situations arises in communication complexity theory. One can define the communication complexity analogs of the Arthur–Merlin hierarchy in a natural way [BFS]. One hopes that separation results such as $\text{coNP} \not\subseteq \text{AM}$ are provable in that model.

1.9. *Collapse and Speedup*

Obviously,

$$\text{AM}(t(n)) \cup \text{MA}(t(n)) \subseteq \text{AM}(t(n)+1) \cap \text{MA}(t(n)+1).$$

What may be slightly surprising is that the finite levels of the Arthur–Merlin hierarchy collapse.

THEOREM 3 (Collapse Theorem [Ba1]). *For any* $t(n) \geqslant 2$ *(where* $t(n)$ *is polynomially bounded),*

$$\text{AM}(t(n)) = \text{AM}(t(n)+1) = \text{MA}(t(n)+1).$$

In particular, for constant $k \geqslant 2$,

$$\text{AM} = \text{AM}(k) = \text{MA}(k+1).$$

After introducing a formalism and preliminary results in Section 2, we complete the proof of this result in Section 3.

Next we show, how the Collapse Theorem implies Proposition 2.

Proof of Proposition 2. Assume $\text{coNP} \subseteq \text{AM}$. We claim $\sum_2^P = \prod_2^P = \text{AM}$. It suffices to prove $\sum_2^P \subseteq \text{AM}$, since $\text{AM} \subseteq \prod_2^P$.

Let $L \in \sum_2^P$. By definition this means that for some $L_1 \in \text{coNP}$,

$$x \in L \quad \text{if and only if} \quad \exists^P y\colon (x, y) \in L_1 \qquad (*)$$

(where $\exists^P$ refers to polynomially bounded quantifier). Now, by assumption, $L_1 \in \text{AM}$. Therefore, by $(*)$, $L \in \text{MAM}$. But $\text{MAM} = \text{AM}$ by the Collapse Theorem. ∎

The Collapse Theorem leaves us with the following short hierarchy:

$$\text{NP} \cup \text{BPP} \subseteq \text{MA} \subseteq \text{AM} \subseteq \text{AM(poly)} \subseteq \text{PSPACE}.$$

These inclusions seem more likely to be proper. Of course, this cannot be proven as long as we do not know how to separate P from PSPACE. As is common in such cases, relativized separation results are sought. Oracles separating MA and AM have been found by Sántha [San], AM from AM(poly) by Aiello, Hastad, and Goldwasser [AHG], AM(poly) from PSPACE by Fortnow and Sipser. Each of these results operates on the relation between members of the Arthur–Merlin hierarchy and the polynomial time hierarchy. Under Sántha's oracle, AM is not contained in $\sum_2^P$ (whereas MA always is). Under the [AHG] oracle, AM(poly) is not in PH (whereas AM always remains inside $\prod_2^P$). Under the oracle of Fortnow and Sipser, AM(poly) $\not\subseteq$ coNP.

The main new result of this paper is the following stronger version of the Collapse Theorem.

THEOREM 4 (Speedup Theorem). *For* $t(n) \geqslant 2$,

$$\mathrm{AM}(2t(n)) = \mathrm{AM}(t(n)).$$

We shall prove this result in Section 3. It will be clear that the proof remains valid under any oracle. Theorem 4 thus complements the result of Aiello, Goldwasser, and Hastad [AGH], that there exists an oracle C relative to which no unbounded speedup is possible: if $t(n)/s(n)$ is unbounded then $\mathrm{AM}(t(n))^C \neq \mathrm{AM}(s(n))^C$.

1.10. *Approximate Lower Bounds*

Let the NP-language L consist of a pairs (x, y) of strings of equal length. Let $L(x) = \{y: (x, y) \in L\}$. Let f be an integer. Suppose Merlin wants to convince Arthur that $|L(x)| \geqslant f$. This problem may be $\#$ P-hard but one can achieve a more modest goal at the level of Arthur–Merlin protocols. There exists such a protocol which allows Merlin to win almost certainly if actually $|L(x)| \geqslant (1+\varepsilon) f$; and gives him almost no chance if $|L(x)| < f$, for any given constant ε. (In fact, ε can even depend on n as long as it decreases at most at a polynomial rate.) The protocol employs universal hashing [CW]. The idea of using universal hashing for approximate lower bound verification is due to Sipser [Sip]; it has been employed in the context of Arthur–Merlin protocols in [Bal, GS]. The protocol will be outlined in Section 4.

1.11. *The Class* almost-NP *and Yet Another Hierarchy*

Let *almost-NP* denote the class of those languages which belong to NP^B for almost every oracle B.

It is clear that AM $\subseteq$ almost-NP. By analogy with BPP which is known to be equal to "almost-P" [BG, Ku], one would expect that AM = almost-NP. This, unfortunately, is not clear at all, bringing some disharmony into the neatly simplifying picture of randomized versions of NP. So a problem of considerable interest remains:

Problem 1. Is AM equal to almost-NP?*

Under a random oracle C, both sides collapse to NP^C. This follows from the next observation.

PROPOSITION 5. $\mathrm{AM}^B = \mathrm{NP}^B$ *for almost every oracle* B.

The proof of this statement is analogous to the proof that $\mathrm{BPP}^B = P^B$ for almost every oracle B [Gi]. The idea is that first we improve the error probability on input x from $\frac{1}{3}$ to $2^{-2|x|}$ and then use (deterministically selected) distant bits from the oracle in place of coin tosses. Since $\sum_x 2^{-2|x|} = 2$ is finite, almost surely only a finite number of input strings x will thus be misjudged (Borel–Cantelli lemma [Ré]); these can be repaired by extending the finite control. ∎

Nevertheless, it is natural to ask:

Problem 2. Does there exist an oracle separating AM from almost-NP?

We mention the following interesting results of Kurtz [Ku] regarding computability relative to two independent random oracles.

THEOREM 6 (S. Kurtz [Ku]). *For almost every pair of oracles* B, C,

(i) $\mathrm{BPP} = P^B \cap P^C$ *and*

(ii) almost-$\mathrm{NP} = \mathrm{NP}^B \cap \mathrm{NP}^C$.

The relation of the unbounded levels of the Arthur–Merlin hierarchy and almost-NP is completely obscure.

Problem 3. Does almost-NP include AM(poly)?

It is easy to see that AM has polynomial size nondeterministic circuits.

Problem 4. Does almost-NP have polynomial size nondeterministic circuits?

Problem 5. Does AM(poly) have polynomial size nondeterministic circuits?

The result of Boppana, Hastad, and Zachos [BHZ, Proposition 2] motivates the following question.

Problem 6. If almost-NP contains coNP, does it follow that the polynomial time hierarchy collapses?

Problem 7. Is almost-NP contained on a finite level of the polynomial time hierarchy?

As one would expect, almost-NP is just the first member of yet another hierarchy of randomized extensions of NP.

Let us imagine a super-Arthur, capable of flipping an exponential number ($\exp(n^c)$) of coins. This is a fair substitute for a random oracle, since all com-

* This problem and several others on this page are no longer open. See the comments added in proof at the end of this paper.

putation paths of the nondeterministic Merlin are polynomially bounded. Games between super-Arthur and Merlin (judged as before by a polynomial time bounded referee) form what we propose to call the *almost*-NP *hierarchy*. (The referee will review those coins of Arthur pointed to by Merlin only.) Possible notation: ANP($t(n)$) and NPA($t(n)$), NP referring to Merlin and A to super-Arthur. Then ANP(1) = BPP, NPA(1) = NP, NPA(2) = MA, and the first interesting class, ANP $\stackrel{\text{def}}{=}$ ANP(2) = almost-NP.

1.12. *A Comment on Terminology*

While *interaction* is a more prominent element of the new proof systems than of old ones, *randomization* is the crucial new ingredient in our view.

In AM (arguably the most important one of the new classes), interaction is minimal. If we seek further related proof systems, we believe randomization will be the guiding line and interaction a possible (not necessarily desirable) side-effect. It might be up to debate whether or not almost-NP represents a randomized proof system; it would be difficult to argue that it is interactive in a more significant way than NP is.

It is for these reasons that we have preferred the term "randomized proof system" to the nice phrase "interactive proof system," coined by Sipser. We note, however, that at present, the two terms cover the same systems.

2. Preliminaries

2.1. *Formalism*

For a function f taking real values over the nonempty finite domain $D = \mathrm{dom}(f)$, we shall use the notation $Axf(x)$ and $Mxf(x)$ for the average and maximum operators

$$Axf(x) = \frac{1}{|D|} \sum_{x \in D} f(x), \qquad Mxf(x) = \max\{ f(x) \mid x \in D \}.$$

Functions $f(x_1, ..., x_t)$ defined over the Cartesian product $D_1 \times \cdots \times D_t$ of the respective domains of the variables permit prefixes of the form $Q_1 x_1 \cdots Q_t x_t$ where $Q_i = M$ or A.

Let D be a nonempty finite set and for every $x \in D$, let $\phi(x)$ be a random (0, 1)-variable over some sample space Ω with expected value $f(x)$. We define the random variables $\exists x \phi(x)$ and $\mathit{Я} x \phi(x)$ over the sample spaces Ω and $D \times \Omega$, respectively, by

$$\exists x \phi(x) = \phi(x_0),$$

where x_0 maximizes the expected value of $\phi(x_0)$; ties are resolved according to a predetermined ordering of D; and

$$\mathit{Я} x \phi(x) = \phi(\xi),$$

where ξ is chosen uniformly from D. Then, by definition,

$$E(\exists x \phi(x)) = Mxf(x)$$

and

$$E(\mathcal{A}x\phi(x)) = Axf(x).$$

Now, inductively, for $D = D_1 \times \cdots \times D_t$ we can define the random variable

$$\exists x_1 \mathcal{A} x_2 \cdots Q_t x_t \phi(x_1, ..., x_t)$$

over the sample space $D_2 \times D_4 \times \cdots \times \Omega$. Its expected value will be

$$Mx_1 Ax_2 \cdots S_t x_t f(x_1, ..., x_t),$$

where $Q_t = \exists$ and $S_t = M$ if t is odd; $Q_t = \mathcal{A}$ and $S_t = A$ if t is even. The variable $\mathcal{A}x_1 \exists x_2 \cdots Q_t x_t \phi(x_1, ..., x_t)$ is analogously defined over $D_1 \times D_3 \times \cdots \times \Omega$.

2.2. *Randomized Combinatorial Games*

If $D = D_1 \times \cdots \times D_t$ is a nonempty, finite set and for each $x \in D$ we are given a random (0, 1)-variable $\phi(x)$ over the same sample space Ω then ϕ defines a "randomized combinatorial game" played as follows.

Two players, henceforth called Merlin and Arthur, alternate moves; the ith move consists of picking an element $x_i \in D_i$. After the tth move the game terminates and a referee flips a (biased) coin representing the variable $\phi(x_1, ..., x_t)$ to decide the winner. Merlin wins if $\phi(x) = 1$, Arthur otherwise. Merlin may or may not be the first player so it takes the pair (ϕ, Q) to properly specify the game, where Q is the initial of the player with the first move.

We call $\phi(x)$ the *payoff variable* corresponding to the sequence $x \in D$ of moves. The expected value $f(x) = E(\phi(x))$ is the *payoff function.* In a *pure combinatorial game*, $f(x)$ takes the values 0 and 1 only (no randomization).

A (partial) *history* of the game after i moves is a member of $D^i = D_1 \times \cdots \times D_i$. The histories form a rooted tree in a natural way, $(x_1, ..., x_{i-1})$ being the parent of $(x_1, ..., x_i)$. This is the *game tree.*

The *size* of the game is $\log|D|$ (the length of the binary string describing a game history).

2.3. *Games against an Indifferent Adversary. Truncated Games*

Henceforth we assume that Arthur's moves are random (uniformly selected from D_i at the ith move). An AM(t)-game has t moves with Arthur moving first; in an MA(t)-game, Merlin moves first.

Given a partial game history $(x_1, ..., x_i) \in D^i$ and assuming Merlin's moves to be optimal after the ith move (ties are resolved according to a predetermined ordering

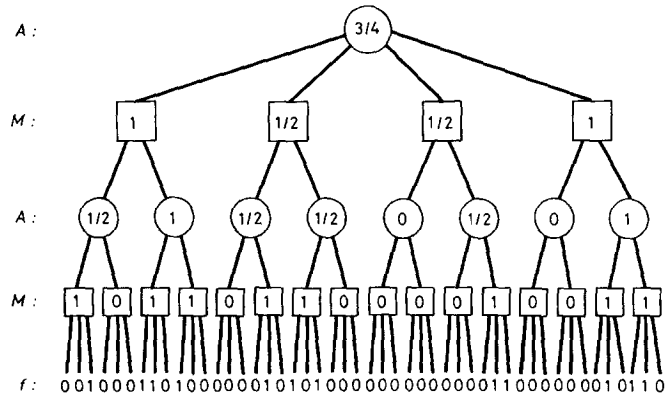

FIG. 1. Evaluation of an AMAM game tree.

of D_j), the outcome of the game can be described by the *partial payoff variable* (again a (0, 1) random variable)

$$\phi^i(x_1, ..., x_i) = Q_{i+1}x_{i+1}Q_{i+2}x_{i+2} \cdots Q_t x_t \phi(x_1, ..., x_t),$$

where Q_j alternates between the existential ($\exists$) and the random ($\mathcal{R}$) quantifiers; Q_{i+1} is $\exists$ if Merlin moves next and $\mathcal{R}$ otherwise. The probability that Merlin wins is

$$f^i(x_1, ..., x_i) = E(\phi^i(x_1, ..., x_i)) = S_{i+1}x_{i+1}S_{i+2}x_{i+2} \cdots S_t x_t f(x_1, ..., x_t),$$

where $S_j = M$ if $Q_j = \exists$ and $S_j = A$ if $Q_j = \mathcal{R}$. This is the *value* of the node $(x_1, ..., x_i)$.

The probability that Merlin wins the game is f^0, the *value of the root* which we also refer to as the *value of the game.*

By the *evaluation of the game tree* (see Fig. 1) we mean the assignment of the values $f^i(x)$ to the nodes $x = (x_1, ..., x_i) \in D^i$ for every i. This is done recursively from bottom up:

$f^t(x) = f(x)$;
on a Merlin level, $f^i(x) = Mx_{i+1}f^{i+1}(xx_{i+1})$;
on an Arthur level, $f^i(x) = Ax_{i+1}f^{i+1}(xx_{i+1})$

(a fortuituous mnemonic coincidence). For instance, in an AMAM-game defined by the random variable

$$\phi^0 = \mathcal{R}x_1 \exists x_2 \mathcal{R}x_3 \exists x_4 \phi(x_1, x_2, x_3, x_4),$$

the value of the game (Merlin's winning chance) is

$$f^0 = E(\phi^0) = Ax_1 Mx_2 Ax_3 Mx_4 f(x_1, x_2, x_3, x_4),$$

where f is the payoff function.

The reason for introducing *randomized* games is that this family is closed under *truncation.* The ith truncation (ϕ^i, Q) of a game (ϕ, Q) with game space $D = D_1 \times \cdots \times D_t$ is defined on the game space D^i; we simply remove the levels $i+1$, $i+2$, etc. from the tree. The new payoff variables will be the partial payoff variables $\phi^i(x_1, ..., x_i)$ at level i; the new payoff function is the value of the nodes on the ith level. The values of the nodes up to level i do not change.

The notion of truncation enables us to reduce the analysis of certain simulations of randomized combinatorial games to trees of small depth.

With each node $x \in D^i$ we associate the *residual game* (ϕ_x, Q_{i+1}) with game space $D_{i+1} \times \cdots \times D_t$, where Q_{i+1} is the initial of the player making move $i+1$; and

$$\phi_x(x_{i+1}, ..., x_t) = \phi(x, x_{i+1}, ..., x_t).$$

The corresponding game is the branch with x as the root.

2.4. *Biased Games, Arthur–Merlin Protocols*

We call a game against an indifferent adversary *biased* if the game value is either $> \frac{2}{3}$ or $< \frac{1}{3}$. The game favors Merlin in the former case and Arthur in the latter. The *uncertainty* of the game (ϕ, Q) is

$$\mathrm{unc}(\phi, Q) = \min\{f^0, 1 - f^0\},$$

where f^0 is the game value, i.e., the probability that Merlin wins. So, a biased game is one with uncertainty $< \frac{1}{3}$.

We shall consider *uniform families* of biased, purely combinatorial Arthur–Merlin games, which we call Arthur–Merlin *protocols.* Such a protocol is determined by a polynomial time Turing machine, the *referee.* On input x the referee computes the polynomially bounded nonnegative integers $t = t(n), n_1, ..., n_t$ and thus generates the game space $D_1 \times D_2 \times \cdots \times D_t$, where $D_i = \{0, 1\}^{n_i}$. Subsequently the referee accepts a game history and declares the winner. For every input, the resulting game must be biased.

These definitions give a precise meaning to the concepts introduced in Subsection 1.3.

2.5. *Simulation of Arthur–Merlin Protocols*

Let P1 and P2 be two Arthur–Merlin protocols, the "old" and the "new" protocol. We say that P2 simulates P1 if

for every input x, the same player is favored in both games. (*)

In the simulations we describe below, the new referee acts in polynomial time, using the old referee as an oracle, with the restriction that

on input x, the new referee queries the old referee only for values on the same input. (**)

Queries are free (i.e., their cost is the computation of the question). By the *cost of the simulation* we mean the time complexity of the computation by the new referee, allowing free queries satisfying (**).

It is then clear that the actual time complexity of the new protocol (when using the old one as a subroutine) will be the cost of simulation plus # queries × old complexity.

2.6. *Decreasing the Uncertainty*

The following simple and useful example illustrates our notions of simulation and complexity.

Often, we have to turn a *modest* advantage into an *overwhelming* one. This is easily accomplished by letting the players play the game in parallel on several "boards" and declaring M the winner if he wins on *more than half* of the boards.

In order to formalize this, let us define the game (f^k, Q) as follows. The new game space is

$$D^k = D_1^k \times \cdots \times D_t^k.$$

Each history $u \in D^k$ can be thought of as the combination of k parallel histories $u_1, \ldots, u_k \in D$ for the old game. Let $\phi_i(u_i)$ be independent realizations of the random variables $\phi(u_i)$ (replacing the sample space Ω by Ω^k). We set $\phi^k(u) = \text{maj}(\phi_1(u_1), \ldots, \phi_k(u_k))$ where "maj" is the (strict) majority function.

PROPOSITION 7. *Suppose* $\text{unc}(f, Q) < \frac{1}{3}$. *Then* $\text{unc}(f^k, Q) < c^k$, *where* $c = 2\sqrt{2}/3 = 0.9428\cdots < 1$.

Proof. The number of boards where the favored player loses is the number of successes in a sequence of k independent Bernoulli trials each with probability of success less than $\frac{1}{3}$ (assuming, as we may, that Merlin's moves are always optimal and uniquely determined). Standard calculation shows that the probability that this number is at least $k/2$ is less than c^k. ∎

Of course, similar result holds if we replace $\frac{1}{3}$ by any constant less than $\frac{1}{2}$. The cost of this simulation is $k \times (1 + \log|D|)$.

3. Switching Moves. Proof of the Speedup Theorem

We begin with a slightly weaker result. Repeated application of this result proves the Speedup and Collapse Theorems with the exception of one case which will be handled at the end of this section.

THEOREM 8. *For polynomially bounded* $t(n) \geq 2$, $\text{AM}(2t(n)) = \text{AM}(t(n) + 1)$.

Proof. Let P1 be the protocol to be simulated. On inputs of length n, P1 allows $t = 2t(n)$ moves in the AM-game. We may assume t is divisible by 4. We simulate

P1 by a protocol P2 allowing only $t/2 + 1$ moves. Set $\varepsilon = (24t)^{-t}$. At the cost of a polynomial increase in the game size (total number of bits communicated), we may assume that P1 yields, on every input of length n, winning probability $< \varepsilon$ or $> 1 - \varepsilon$ for Merlin. Let s be the (increased) game size and $m = 4st$. We may assume $s > n$.

The *simulation* goes as follows. We divide the sequence of moves into $t/4$ AMAM segments. We replace the MAM part of each such segment by an AMA part, thus reducing AMAM AMAM AMAM $\cdots$ to AAMA AAMA AAMA $\cdots$, i.e., to AM AM AM A $\cdots$.

Suppose an MAM segment of P1 consists of Merlin selecting a string $x \in X$, then Arthur selecting $y \in Y$ and Merlin selecting $z \in Z$. The corresponding part of P2 will consist of Arthur selecting $(y_1, ..., y_m) \in Y^m$, then Merlin selecting $x \in X$ and $(z_1, ..., z_m) \in Z^m$, finally Arthur selecting $i \in \{1, ..., m\}$. In evaluating the game, the old deterministic referee is being fed the moves (x, y_i, z_i). In other words, for a little while we pretend the game continues in parallel on m boards, but then Arthur selects one board on which the only "valid" game will continue.

Motivation. In order to reduce the number of turns, we switch some of the moves of Merlin (x) and Arthur (y). We periodically ask Arthur to reveal his move y before it would be his turn. This, of course, could lead to Merlin gaining decisive advantage, sufficient to reverse the odds. To counter this advantage we proceed as follows. We multiply the number of boards and ask the players to play m copies of the game in parallel. Now Arthur makes a separate independent move on each of the m boards (**y**). Next we ask Merlin to make his move x, the same move on each board. We hope that this way he will not be able to do much better than if he had had to move first. Subsequently we let the players continue playing on the m boards in parallel (thus Merlin making his next move **z**). Continuing in this fashion and eventually declaring the winner of the majority to be the winner would suffice for a proof of the Collapse Theorem. For the Speedup Theorem, however, we want to repeat the switching procedure a polynomial number of times. This would blow up the number of boards exponentially. To prevent this, we eliminate all but one of the boards by letting Arthur randomly select one on which to continue. We shall prove that Arthur's random move **y** is exponentially unlikely to result in a position where Merlin has greatly improved chances on a substantial fraction of the boards. This makes it likely that, if Arthur was the favored player, he retains this status after his move i (board selection).

Analysis. Clearly, the size of the new game is still polynomially bounded in n, and so is the running time of the new referee.

It is clear that Merlin's chances can only improve under the new protocol. Indeed, if it is Merlin's turn to select x, he can simply ignore the now available information of what Arthur had selected for $(y_1, ..., y_m)$, and make what would have been his optimal move in the original game. Similarly, for z_i he can select his optimal response to Arthur's y_i. With such choice, Merlin's chances will be precisely what they were in the original game.

So what we have to worry about is whether or not Merlin's chances may improve too much. In proving that they do not we have to be a little more technical. First we concentrate on a single MAM segment, a truncated residual game, which is a game in its own right according to our definitions.

Let $f(x, y, z)$ be the payoff function of an arbitrary MAM-game; let $\delta = f^0$ be the game value. Let m and t be positive integers, $m > 2t$. Simulate this ("old") game by a "new" AMA-game as described above. The next lemma asserts that Arthur's first move in the new game ($\mathcal{A}\mathbf{y}$) is exponentially unlikely to give Merlin $> 1/(2t)$ chance that this expected payoff, after Arthur's second move ($\mathcal{A}i$), will exceed $12t\delta$.

LEMMA 9. *For* $\mathbf{y} \in Y^m$, *let* $C(\mathbf{y})$ *denote the event*

$$(\exists x \in X)(|\{i: Mz_i f(x, y_i, z_i) > 12t\delta\}| > m/(2t)).$$

Then, for random $\mathbf{y} \in Y^m$,

$$\text{Prob}(C(\mathbf{y})) < |X|\, 2^{-m/(2t)}.$$

Proof. Recall that $f^2(x, y)$ denotes Merlin's winning probability after the MA moves x, y in the old game. Now, for every $x \in X$, $Ayf^2(x, y) \leqslant \delta$. Therefore, for every x and random y we have

$$\text{Prob}(f^2(x, y) > 12t\delta) < 1/(12t).$$

The probability of the bad event $B(x, \mathbf{y})$ that $f^2(x, y_i) > 12t\delta$ happens for more than $m/(2t)$ out of the m values $1 \leqslant i \leqslant m$ (for fixed x and randomly chosen y_i) is less than

$$\binom{m}{m/(2t)} (12t)^{-m/(2t)} < (2et/12t)^{m/(2t)} < 2^{-m/(2t)}.$$

Finally, we note that $C(\mathbf{y}) = (\exists x)\, B(x, \mathbf{y})$. We thus have $\text{Prob}(C(\mathbf{y})) \leqslant \sum_{x \in X} \text{Prob}\, B(x, \mathbf{y})$, completing the proof of the lemma. ∎

We remark, that, with the parameters chosen as above, we have $|X| \leqslant 2^s$ and $s > n$; therefore the probability estimate of the lemma implies the upper bound

$$|X|\, 2^{-m/(2t)} \leqslant 2^{s - m/(2t)} = 2^{-s} < 2^{-n}.$$

In order to take the first move of each AMAM segment into consideration, we observe

PROPOSITION 10. *If the value of an Arthur node in a game tree is* δ *than the value of at most* $1/(2t)$ *fraction of its children exceeds* $2t\delta$. ∎

Now we can return to the analysis of the new protocol P2. To each of the $t/4$ AMAM segments of P1, there corresponds an AAMMA segment of P2. We

shall examine the corresponding game-tree of depth t, i.e., without collapsing neighboring levels of the same kind. Let us call the successive nodes in each AAMMA segment u, $\mathbf{y}$, x, $\mathbf{z}$, i-nodes, respectively. A u-node, for instance, is an A-node at the beginning of the AAMMA segment; we call the move Arthur makes here a u-move. If we label a u-node by its history h, then Arthur's next move, say u (a u-move) takes us to the child node labeled by the history hu.

After every i-move (i.e., at the time of arriving at a u-node), the history $h=(\cdots, u, \mathbf{y}, x, \mathbf{z}, i)$ of the P2-game defines a unique history $\pi(h)=(\cdots, u, x, y_i, z_i)$, the *projection* of h. The same holds after every u-move. Let us call the children of the i-nodes and of the u-nodes projectable. If h is a projectable node of the P2-tree, let $e(h)=f(\pi(h))$ denote the value of the node $\pi(h)$ in the P1-tree. (We omit the superscript of f because it is redundant.)

Let us call a child hu of a u-node h lucky (for Merlin), if $e(hu)>2te(h)$. By the proposition, at most a $1/(2t)$ fraction of the children of a u-node can be lucky.

Let us call a child $h\mathbf{y}$ of a $\mathbf{y}$-node h lucky, if

$$(\exists x)(|\{i: Mz_i f(\pi(h), x, y_i, z_i)>12te(h)\}|>m/(2t).$$

By the lemma and the subsequent remark, at most a 2^{-n} fraction of the children of a $\mathbf{y}$-node are lucky.

Finally, we call a child hi of an i-node h lucky, if $h=h'\mathbf{y}x\mathbf{z}$ for some $\mathbf{y}$-node h', and $e(hi)>12te(h')$. If h' is not lucky, then, by definition, at most a $1/(2t)$ fraction of the children of h are lucky.

Let now $h=(u_1, \mathbf{y}_1, \ldots, \mathbf{z}_{t/4}, i_{t/4})$ be a P2-game history. If none of the initial segments of the game represent a lucky node, then $f(h)\leqslant(2t)^{t/4}(12t)^{t/2}\varepsilon<24^{-t/2}<1$, Merlin loses. Hence Merlin's winning probability is not greater than the probability that it hits a lucky node during the game. This probability is bounded above by the sum of conditional probabilities $\text{Prob}(L_i|M_i)$, where L_i denotes the event that the ith move hit a lucky node, while M_i stands for the event that no previous node along the path was lucky.

Each of the three kinds of levels with prospective lucky nodes is encountered $t/4$ times. The corresponding conditional probabilities are bounded by $1/(2t)$, 2^{-n}, and again $1/(2t)$, respectively. The sum is thus $\leqslant\frac{1}{4}+t2^{-n-2}<\frac{1}{3}$. ∎

It is clear that the Speedup Theorem and therefore the Collapse Theorem follow from Theorem 8, with one slight exception. We do not immediately get a reduction from AMA to AM. The simulation described above does, however, yield a simulation of any bounded depth Arthur–Merlin protocol by an AMA protocol such that Arthur's last move is restricted to a *polynomial size domain* (i.e., $O(\log n)$ length).

LEMMA 11. *An* AMA*-protocol with polynomially bounded domain for the last move can be simulated by an* AM*-protocol.*

Proof. Let $X\times Y\times Z$ denote the game space for protocol P1, described as

$\mathrm{Я}x\exists y\mathrm{Я}zf(x, y, z)$. We may assume that the uncertainty of the game is $<\frac{1}{6}$. Let us define the simulating protocol P2 as $\mathrm{Я}x\exists y \operatorname{maj}_z f(x, y, z)$. Here $\operatorname{maj}_z g(z)$ takes the value 1 if $\sum_{z\in Z} g(z) \geqslant |Z|/2$; and 0 otherwise. We thus replace Arthur's last move by a majority vote to be computed by the referee over all possible choices of Arthur. This is feasible in polynomial time because Z is small.

We have to show that the simulation is correct.

CLAIM. *For any* $(0, 1)$*-valued function* $f(x, y, z)$ *over the finite domain* $X \times Y \times Z$,

$$AxMy \operatorname{maj}_z f(x, y, z) \leqslant 2AxMyAzf(x, y, z).$$

The same inequality applies if M is taken to mean the minimum rather than the maximum operator.

Proof. Clearly, for any $(0, 1)$-valued function $g(z)$, $\operatorname{maj}_z g(z) \leqslant 2Azg(z)$. Moreover, the average, maximum, and minimum operators are semihomogeneous; i.e., if Q denotes any of these operators, $\lambda \geqslant 0$, and $h(u)$ is a real-valued function over the finite domain U then $Qu\lambda h(u) = \lambda Quh(u)$. ∎

It follows that the winning chance of neither player will more than double, yielding the correctness of simulation, given that the uncertainty was less than $\frac{1}{6}$. This completes the proof of the Speedup and Collapse Theorems. ∎

4. ARTHUR–MERLIN PROTOCOLS FOR APPROXIMATE LOWER BOUND VERIFICATION

For a language L consisting of pairs (x, y) of strings such that $|x| = |y|$, let $L(x) = \{y \mid (x, y) \in L\}$.

The problems of verifying approximate upper and lower bounds for $|L(x)|$ cannot be stated as language recognition problems. Randomized complexity classes with a "continuous spectrum" of acceptance probabilities are particularly suited for formalization of approximate verification problems.

Let $C = \mathrm{AM}(t(n))$ or $\mathrm{MA}(t(n))$.

Fix some $\varepsilon > 0$. An *ε-approximate lower bound protocol* of class C is an Arthur–Merlin protocol, depending on an input pair (N, x) such that, letting $W(N, x)$ denote Merlin's winning chances,

(i) if $|L(x)| \geqslant (1+\varepsilon)N$ then $W(N, x) > \frac{2}{3}$;

(ii) if $|L(x)| < N$ then $W(N, x) < \frac{1}{3}$.

(Merlin has only small chances if N is not a lower bound and very good chances if N is a generous lower bound. If N is a lower bound but not quite so generous, we do not require any specific behavior of the protocol.)

Using a technique of Sipser [Sip] based on universal hashing [CW], one can show

THEOREM 12. *For any $L \in$ NP and $\varepsilon > 0$, an ε-approximate lower bound protocol of class* AM *exists.*

This fact can be used [Ba2] to replace involved group theoretic arguments or unproven hypotheses, to derive that certain problems for matrix groups over finite fields belong to AM (nonmembership in, order, and isomorphism of such groups). It is also at the heart of the proof of the result of Goldwasser and Sipser [GS]. For completeness, we describe the proof.

Proof. First of all we remark, that the Collapse and Speedup Theorems apply to approximate lower bound protocols as well (with no change in the proof). Therefore it suffices to present an MAM protocol.

Let $n = |x|$; then, by assumption, $n = |y|$ for every $y \in L(x)$.

First we consider the special case when Merlin claims $L(x)$ to be *dense* in $\{0,1\}^n$, i.e., $N > \alpha 2^n$ for some fixed $\alpha > 0$. In this case Arthur selects m random strings y_i of length n; and Merlin supplies a witness whenever one exists that $(x, y_i) \in L$. Merlin's expected number of successes will be $m|L(x)|\,2^{-n}$. If Merlin succeeds in at least $(1+\varepsilon/2)\,mN2^{-n}$ cases, we declare him the winner; otherwise Arthur wins.

Obviously, choosing any $m > c/\varepsilon$ guarantees the validity of (i) and (ii) for some absolute constant c.

Now we turn to the general case. We identify $\{0,1\}^n$ with the n-dimensional space over the field GF(2). Linear maps from $\{0,1\}^n$ to $\{0,1\}^k$ are represented by $k \times n$ (0, 1)-matrices.

LEMMA 13. *Let $S \subseteq \{0,1\}^n$. Let $\alpha > 0$ and $2^k \geqslant |S|/\alpha$. Then there exists a $k \times n$ (0, 1)-matrix C such that $|C(S)| \geqslant (1-\alpha)|S|$, where $C(S) \subseteq \{0,1\}^k$ is the image of S under the linear map C.*

Proof. Let us choose C randomly with uniform distribution over the 2^{kn} matrices. For any $z \in \{0,1\}^n$, if $z \neq 0$ then $\text{Prob}(Cz = 0) = 2^{-k}$. Therefore, for any two distinct $u, v \in \{0,1\}^n$, $\text{Prob}(Cu = Cv) = \text{Prob}(C(u-v) = 0) = 2^{-k}$. Let us call $u, v \in S$ *mates*, if $u \neq v$ and $Cu = Cv$. We conclude that for any $v \in S$, the probability that v has a mate (in S) is $\leqslant |S|\,2^{-k} \leqslant \alpha$. Therefore the expected number of mateless members of $|S|$ is $\geqslant (1-\alpha)|S|$, and this, clearly, is a lower bound on the expected size of $C(S)$. Consequently, for some C we have $|C(S)| \geqslant (1-\alpha)|S|$. ∎

We shall use this lemma with the following additional constraints on the parameters. Given $0 < \varepsilon < \frac{1}{2}$, let $\alpha = \varepsilon/3$, and select k such that $2^{k-1} < (1+\varepsilon)\,N/\alpha \leqslant 2^k$. Let $S = L(x)$.

The protocol runs as follows. Merlin exhibits a $k \times n$ (0, 1)-matrix C (preferably one satisfying $C(S) \geqslant (1-\alpha)|S|$). Let $\delta = (1+\varepsilon)(1-\alpha) - 1$. Next, a δ-approximate lower bound AM protocol, as above, tests the claim that $|C(S)| \geqslant N$. (Merlin verifies $v \in C(S)$ by exhibiting $u \in S$ such that $Cu = v$.)

If $|S| \geqslant (1+\varepsilon)\,N$ then Merlin can assure (by the lemma) that $|C(S)| \geqslant (1+\delta)\,N$. By the choice of k, $C(S)$ is thus dense in $\{0,1\}^k$, and Merlin will win the game with large probability.

On the other hand, if $|S| < N$, then whatever C Merlin chooses, $|C(S)| \leqslant |S| < N$, so Merlin is likely to lose. ∎

We remark that this MAM protocol could easily be transformed into an AM protocol without an appeal to the Collapse Theorem (although it should also be pointed out that the case MAM = AM of the Collapse Theorem can be proved in just a few lines). Indeed, instead of Merlin selecting the matrix C, Arthur can randomly select s matrices, from which Merlin would later pick his favorite. It is clear, that the probability that for all members C of Arthur's collection, $|C(S)| < (1-2\alpha)|S|$, decreases exponentially with increasing s. So, replacing α by 2α, the rest of the proof works.

5. An Explicit AM Protocol for Graph Nonisomorphism and Coset Intersection in Permutation Groups

Nonisomorphism of graphs is not known to belong to NP. In fact, what we know about the length of the shortest proof of nonisomorphism of two graphs is no better than the running time of the best deterministic isomorphism test, i.e., $\exp(C\sqrt{n \log n})$ [Lu2; cf. BL].

Goldreich, Micali, and Wigderson [GMW] found a strikingly simple GMR protocol of class IP(2) for graph nonisomorphism. By the result of Goldwasser and Sipser [GS], it follows that graph nonisomorphism is in AM AM, and then, by the Collapse Theorem, in AM. We give a direct proof of this result.

THEOREM 14. *Graph nonisomorphism belongs to* AM.

Proof. It suffices to solve the problem for connected graphs. If X and Y are connected, let Z be their disjoint union. Assume the number of automorphisms of X, Y, and Z are a, b, and c, respectively. Clearly,

(a) if X and Y are isomorphic, then $c = 2ab$;

(b) if X and Y are not isomorphic, then $c = ab$.

So all we have to do is to decide between the two alternatives $c = 2ab$ and $c = ab$ (not knowing the value of either of these numbers). In order to verify alternative (b), it suffices, however, to verify an *approximate lower bound* for a and b and an *approximate upper bound* for c, each within a factor of $2^{1/3}$.

The approximate lower bounds follow immediately from the result stated in the previous section. (L should consist of pairs (x, y), where x is a string which encodes a graph and y an automorphism of x.)

(We remark that actually, this AM-class approximate lower bound estimate can be replaced by a stronger scheme, namely the inequality $|\mathrm{Aut}\, X| \geqslant a$ (or even the relation that a divides $|\mathrm{Aut}\, X|$) belongs to NP. To verify that X has at least a automorphisms, we just guess generators for the automorphism group and deterministically compute the order of this group (by Sims' algorithm [Sim; cf. FHL]) in polynomial time.

For the approximate upper bound verification we observe that the number of distinct isomorphic copies of Z on its own vertex set of n elements is precisely

$$n!/|\mathrm{Aut}\, Z|.$$

This reduces the upper bound problem to an approximate lower bound verification for the set of isomorphic copies of Z. Thus, again, the hashing technique applies. ∎

An important class of polynomically equivalent problems above the graph isomorphism problem was detected by E. M. Luks in 1980.

Assume that we are given two permutation groups G, H acting on the same set S and two permutations g, h of S. The *coset intersection problem* asks whether or not the cosets Gg and Hh intersect. (Permutation groups are given by a list of generators.)

Luks called the attention to this problem [Lu1] by reducing graph isomorphism to it. Another equivalent problem is the color-isomorphism problem: given a permutation group G acting on S and two colorings α and β of S, does there exist $g \in G$ transforming one coloring into the other:

$$\alpha(s) = \beta(s^g)$$

for every $s \in S$. These and further equivalent problems (including the centralizer of an element in a permutation group) were mentioned by Luks partly in [Lu1], partly in private correspondence.

THEOREM 15. *The coset intersection problem is in* $\mathrm{NP} \cap \mathrm{coAM}$.

Proof. The NP part is clear. For the coAM claim, it suffices to consider the equivalent color isomorphism problem. Let us take two disjoint copies of S and the permutation group $\hat{G}$ generated by G acting on one copy and by the involution switching the two copies. So, $|\hat{G}| = 2|G|^2$. Apply one coloring in each copy. Now, as before, we look at the color-automorphisms of the colorings on each half with respect to G as well as of the coloring of the union with respect to $\hat{G}$. We again have alternatives (a) and (b), thus the verification of approximate bounds on the number of color-automorphism of a colored set suffices. The only difference compared to the previous argument is that the quantity $n!$ will be replaced by $|\hat{G}|$. We leave the details to the reader. ∎

ACKNOWLEDGMENTS

The "simple observation" that AM = almost-NP appears without proof as Proposition 2.5(ii) in [Ba1]. We are grateful to Mike Sipser for detecting that this assertion was unjustified, thus turning it into an intriguing open problem (Problem 1). Other conversations of the first named author with Mike Sipser as well as with Shafi Goldwasser, Silvio Micali, Oded Goldreich, Stuart Kurtz, Gene Luks, and Janos Simon have been inspiring. Our particular debt is due to Johan Hastad whose comment led to a considerable simplification of the proof of Theorem 4.

Note added in proof. A recent fundamental discovery by Noam Nissan reduces the number of independent random bits required for polynomial size, bounded depth randomized Boolean circuits to $(\log n)^C$. This result implies a positive solution to Section 1.11, Problem 1: almost-NP = AM, thus confirming AM as a particularly natural, robust randomized extension of NP. A negative answer to Problem 2 and positive answers to Problems 4, 6, and 7 are immediate. See forthcoming papers by N. Nissan and by N. Nissan and A. Wigderson.

REFERENCES

[AHG] W. AIELLO, S. GOLDWASSER, AND J. HASTAD, On the power of interaction, *in* "Proceedings, 27th IEEE Symp. Found. of Comput. Sci. 1986, pp. 368–379. Full version to appear in *Combinatorica.*

[Ad] L. ADLEMAN, Two theorems on random polynomial time, *in* "Proceedings, 19th IEEE Symp. Found. of Comput. Sci., 1978," pp. 75–83.

[Ba1] L. BABAI, Trading group theory for randomness, *in* "Proceedings, 17th ACM Symp. on Theory of Comput. Providence, RI, (1985)," pp. 421–429.

[Ba2] L. BABAI, Interactive proofs in finite groups, *in* "Randomness and complexity" (S. Micali and F. Preparata, Eds.), to appear.

[BE] L. BABAI AND P. ERDÖS, Representation of group elements as short products, *in* "Theory and Practice of Combinatorics (A. Rosa *et al.*, Eds.)," Annals of Discrete Math. Vol. 12, pp. 27–30, North Holland, Amsterdam, 1982.

[BFS] L. BABAI, P. FRANKL AND J. SIMON, Complexity classes in communication complexity theory, *in* "Proceedings, 27th IEEE Symp. Found. of Comput. Sci. 1986," pp. 337–347.

[BL] L. BABAI AND E. M. LUKS, Canonical labeling of graphs, *in* "Proceedings, 15th ACM Symp. Theory of Comput., 1983," pp. 171–183.

[BLS] L. BABAI, E. M. LUKS, AND E. SZEMERÉDI, On the complexity of matrix group problems, in preparation.

[BS] L. BABAI AND E. SZEMERÉDI, On the complexity of matrix group problems, (preliminary version) *in* "Proceedings, 25th IEEE Symp. Found. of Comput. Sci., Palm Beach, FL, 1984," pp. 229–240.

[BG] C. H. BENNETT AND J. GILL, Relative to a random oracle A, $P^A \neq NP^A \neq coNP^A$ with probability 1, *SIAM J. Comput.* **10** (1981), 96–113.

[BHZ] R. BOPPANA, J. HASTAD, AND S. ZACHOS, Does coNP have short interactive proofs? *Inform. Process. Lett.* in press.

[CW] J. L. CARTER AND M. N. WEGMAN, Universal classes of hash functions, *J. Comput. System Sci.* **18**, No. 2 (1979), 143–154.

[CKS] A. CHANDRA, D. KOZEN, AND L. STOCKMEYER, Alternation, *J. Assoc. Comput. Mach.* **28** (1981), 114–133.

[Co] S. A. COOK, The complexity of theorem proving procedures, *in* "Proceedings, 3rd ACM Symp. Theory of Comput., pp. 151–158.

[FS] L. FORTNOW AND M. SIPSER, private communication.

[FHL] M. L. FURST, J. HOPCROFT, AND E. M. LUKS, Polynomial-time algorithms for permutation groups, *in* "Proceedings, 21st IEEE Symp. Found. of Comput. Sci., Syracuse, NY, 1980," pp. 36–41.

[GG] O. GABER AND Z. GALIL, Explicit construction of linear sized superconcentrators, *J. Comput. System Sci.* **22** (1981), 407–420.

[GJ] M. GAREY AND D. S. JOHNSON, "Computers and Intractability: A Guide to the Theory of NP-Competeness," Freeman, San Francisco, 1979.

[Gi] J. GILL, Computational complexity of probabilistic Turing machines, *SIAM J. Comput.* **6** (1977), 675–695.

[GMW] O. GOLDREICH, S. MICALI, AND A. WIGDERSON, Proofs that yield nothing but their validity

and a methodology of cryptographic protocol design, *in* "27th IEEE Symp. Found. of Comput. Sci., 1986," pp. 174–187.

[GMR] S. GOLDWASSER, S. MICALI, AND C. RACKOFF, The knowledge complexity of interactive proofs, *in* "Proceedings, 17th ACM Symp. Theory of Comput., Providence, RI, 1985," pp. 291–304.

[GS] S. GOLDWASSER AND M. SIPSER, Private coins versus public coins in interactive proof systems, *in* "Proceedings, 18th ACM Symp. Theory of Comput., Berkeley, CA, 1986," pp. 59–68.

[Ku1] S. A. KURTZ, "Randomness and Genericity in the Degrees of Unsolvability," Ph.D. thesis, Univ. of Illinois at Urbana, 1981.

[Ku2] S. A. KURTZ, A note on randomized polynomial time, *SIAM J. Comput.*, in press.

[Lau] C. LAUTEMANN, BPP and the polynomial hierarchy, *Inform. Process. Lett.* **17**, No. 4 (1983), 215–217.

[Lu1] E. M. LUKS, Isomorphism of graphs of bounded valence can be tested in polynomial time, *J. Comput. System Sci.* **25** (1982), 42–65.

[Lu2] E. M. LUKS, The complexity of fixed valence graph isomorphism and the implications for general graph isomorphism, in preparation.

[Mar] G. A. MARGULIS, Explicit constructions of concentrators, *Problemy. Peredachi Informatsii* **9** (1973), 71–80; (English transl. in *Problems Inform. Transmission* (1975), 325–332).

[MS] A. R. MEYER AND L. J. STOCKMEYER, The equivalence problem for regular expressions with squaring requires exponential time, *in* "Proceedings, IEEE Annu. Symp. Switching and Automata Theory, 1972," pp. 125–129.

[Mil] G. L. MILLER, Riemann's hypothesis and tests for primality, *J. Comput. System Sci.* **13** (1976), 300–317.

[Pa] C. H. PAPADIMITRIOU, Games against nature, *in* "Proceedings, 24th IEEE Symp. Found. of Comput. Sci., Tucson, AZ, 1983," pp. 446–450.

[Pin] M. PINSKER, On the complexity of a concentrator, *in* "7th Internat. Teletraffic Conf., Stockholm, 1973," 318/1–4.

[Pip] N. PIPPENGER, Superconcentrators, *SIAM J. Comput.* **6** (1977), 298–304.

[Ré] A. RÉNYI, "Probability Theory," North-Holland, 1976.

[Sac] G. E. SACKS, Degrees of unsolvability, Annals of Math. Studies Vol. 55, 2nd ed., Princeton Univ. Press, Princeton, NJ, 1966.

[San] M. SANTHA, Relativized Arthur–Merlin versus Merlin–Arthur games, *Information and Computation*, to appear.

[SV] M. SANTHA AND U. V. VAZIRANI, Generating quasi-random sequences from semi-random sources, *J. Comput. System Sci.* **33** (1986), 75–87.

[Sim] C. C. SIMS, Some group theoretic algorithms, *in* Lecture Notes in Math. Vol. 697, pp. 108–124, Springer–Verlag, New York, 1978.

[Sip] M. SIPSER, A complexity theoretic approach to randomness, *in* "Proceedings, 15th ACM Symp. Theory of Comput. Boston, 1983," pp. 330–335.

[SS] R. SOLOVAY AND V. STRASSEN, A fast Monte Carlo test for primality, *SIAM J. Comput.* **6** (1977), 84–85.

[St1] L. STOCKMEYER, The polynomial time hierarchy, *Theoret. Comput. Sci.* **3**, No. 1 (1976), 1–22.

[St2] L. STOCKMEYER, The complexity of approximate counting, *in* "Proceedings, 15th ACM Symp. Theory of Comput., 1983," pp. 118–126.

[VV] U. V. VAZIRANI AND V. V. VAZIRANI, Random polynomial time is equal to semi-random polynomial time, *in* "Proceedings, 26th IEEE Symp. Found. of Comput. Sci., 1985," pp. 417–428.

[WR] A. N. WHITEHEAD AND B. RUSSELL, "Principia Mathematica," Cambridge Univ. Press, London, 1910, 1927.

[ZF] S. ZACHOS AND M. FURER, Probabilistic quantifiers vs distrustful adversaries, to appear.

Printed in Belgium