Trading Group Theory for Randomness

László Babai

Dept. Algebra
Eötvös University
Budapest
Hungary H-1088

Dept. Computer Science
University of Chicago
1100 E 58th St.
Chicago, IL 60637

Abstract.

In a previous paper [BS] we proved, using the elements of the theory of *nilpotent groups,* that some of the *fundamental computational problems in matrix groups* belong to NP. These problems were also shown to belong to $coNP$, assuming an *unproven hypothesis* concerning *finite simple groups*.

The aim of this paper is to replace most of the (proven and unproven) group theory of [BS] by elementary combinatorial arguments. The result we prove is that relative to a *random oracle B,* the mentioned matrix group problems belong to $(NP\cap coNP)^B$.

The problems we consider are *membership* in and *order* of a matrix group given by a list of generators. These problems can be viewed as multidimensional versions of a close relative of the discrete logarithm problem. Hence $NP\cap coNP$ might be the lowest natural complexity class they may fit in.

We remark that the results remain valid for *black box groups* where group operations are performed by an oracle.

The tools we introduce seem interesting in their own right. We define a new hierarchy of complexity classes $AM(k)$ "just above NP", introducing *Arthur vs. Merlin games,* the bounded-away version of Papadimitriou's *Games against Nature*. We prove that in spite of their analogy with the polynomial time hierarchy, the finite levels of this hierarchy collapse to $AM=AM(2)$. Using a combinatorial lemma on finite groups [BE], we construct a game by which the nondeterministic player (Merlin) is able to convince the random player (Arthur) about the relation $|G|=N$ provided Arthur trusts conclusions based on statistical evidence (such as a Solovay-Strassen type "proof" of primality).

One can prove that AM consists precisely of those languages which belong to NP^B for almost every oracle B.

Our hierarchy has an interesting, still unclarified relation to another hierarchy, obtained by removing the central ingredient from the *User vs. Expert games* of Goldwasser, Micali and Rackoff.

1. Introduction

1.1. Randomness vs. mathematical intractability: a tradeoff

Paul Erdös has taught us that randomness can do miracles as long as we don't insist on explicit constructions. If we do, quite often much heavier mathematics has to be invoked - if there is any help at all. The few cases where randomness has successfully been eliminated, like for expanding graphs, point to the difficulty (cf. [Pin], [Pip] vs. [Mar], [GG]).

A random string can sometimes replace the most formidable mathematical hypothesis. The Solovay-Strassen Monte Carlo primality test [SS] vs. Gary Miller's deterministic primality test, based on the Extended Riemann Hypothesis [Mil], is one of the famous examples.

The objective of this paper is to introduce some new random tools to replace an unproven group theoretic hypothesis.

1.2. Matrix groups

By far the most common way to represent groups is by matrices. This is almost the only way groups are being thought of in science. The term "Representation Theory" refers to matrix representations, a central tool in the theory of finite groups, harmonic analysis, quantum mechanics and other fields.

It appears that the main reason why computational group theory has so far mainly concentrated on permutation groups is that while many of the basic problems in permutation groups are solvable in polynomial time (cf.[Sim], [FHL], [BKL]), even the simplest questions on matrix groups seem computationally infeasible.

The *membership* problem (does a given matrix belong to a group given by a list of generators?) is *undecidable* for 4 by 4 integral matrices [Mih].

It seems therefore wise to restrict our attention to matrix groups over *finite fields*. Here the basic problems (membership, order) are at least finite and in fact easily seen to belong to $PSPACE$. On the other hand, finding a polynomial time algorithm seems hopeless even in the one-dimensional (number theoretic) case. Concerning the place of these problems in the polynomial time hierarchy, the best we may hope for is putting them in $NP \cap coNP$.

We don't quite manage to achieve this goal but we get about as close to it as a Monte Carlo primality test to proving primality.

We shall introduce the complexity class AM which is the randomized version of NP in the same way as Gill's BPP is of P [Gi]. Our main result is this.

Theorem 1.1. *Membership* in, and *order* of matrix groups over finite fields belong to $AM \cap coAM$.

We shall outline the proof of this result in Section 5. For the details see [Ba]. The necessary complexity theoretic machinery will be treated in detail in Sections 2-4.

For comparison as well as for later use, let us quote the two main results of [BS]. They assert that the membership problem belongs to NP and so does the problem of deciding whether an integer divides the order of a group. Moreover, it has been proved in [BS] that both problems belong to $coNP$ as well if we are willing to accept a reasonably plausible but probably very difficult new mathematical hypothesis on finite simple groups (the Short Presentation Conjecture [BS, p.238]).

1.3. The ingredients

We shall define a hierarchy of complexity classes denoted A, M, MA, AM, MAM, AMA, etc.: the *Arthur-Merlin hierarchy*. Trivial inclusions will correspond to substrings, e.g. $M \subseteq MA \subseteq MAM$. Moreover, $A = BPP$ and $M = NP$. It will be straightforward to show that $AM \subseteq NP^B$ for almost every oracle B.

The main component of the proof of Theorem 1.1 is an *approximate upper bound* algorithm of class MAM. This algorithm almost certainly accepts the pair (G,N) if $|G| \leq N$ and almost certainly rejects it if $|G| \geq 2N$.

The algorithm is based on a combinatorial lemma on finite groups [BE].

Another ingredient is the *verification of the divisors* of the order of a group. This is in class NP by [BS, Theorems 9.1 & 10.1]. One can, however, replace the group theoretic methods of [BS] by an elementary and much more general technique, due to Sipser [Sip] and based on the Carter-Wegman universal hash functions [CW], to obtain a slightly weaker, AM class divisibility verification.

A combination of divisibility and approximate upper bound verifications puts verification of the *exact order* of a given group in MAM.

The last crucial ingredient is that $MAM = AM$ or more generally that the hierarchy above AM collapses to AM. This result will be treated in detail in this paper.

2. A hierarchy of complexity classes

2.1. Convincing a distrustful party

King Arthur recognizes the supernatural intellectual abilities of Merlin but doesn't trust him. How should Merlin convince the intelligent but impatient King that a string x belongs to a given language L?

If $L \in NP$, Merlin will be able to present a *witness* which Arthur can check in polynomial time.

We shall define a hierarchy of complexity classes "just above NP" which still allow Merlin to convince Arthur about membership provided Arthur accepts statistical evidence.

We define these relatively low complexity classes in terms of a combinatorial game played by Merlin and Arthur.

2.2. Combinatorial games

The general definition of combinatorial games will be given in 3.1.

Here we consider games whose rules depend in a polynomially computable way on an input string x. The precise definition is this.

In our games, two players alternate moves.

At the beginning of the game, on input x a deterministic polynomial time bounded Turing machine produces a nonnegative integer $t = t(|x|)$ and a sequence of positive integers $n_1, \ldots, n_t$ such that $\sum_{i=1}^{t} n_i < |x|^{const}$.

Each player, when it is his/her turn, outputs a 0-1 string. The player at turn i outputs a string of length n_i.

The history of the game (the sequence of previous moves) is always known to each player.

After $t(|x|)$ moves the game terminates and a deterministic polynomial time bounded Turing machine, known to both players, evaluates x and the sequence of moves and declares the winner.

The *length* of the game is t,the total number of moves. The *size* of the game is $|x| + \sum_{i=1}^{t} n_i$.

2.3. Arthur vs. Merlin games

In a *Game against Nature* [Pa] we require that player A ("Nature") be indifferent:

(i) the moves of A are random (A just rolls the dice and does not care whether he/she wins or loses).

Given such a game, let $W(x)$ denote the probability that a player M, capable of optimizing his/her winning chances at each move, will be able to beat the indifferent player A.

We shall call such a game an *Arthur vs. Merlin game* if, in addition to (i), the following holds:

(ii) for any input string x, one of the following holds:

(a) $W(x) > 2/3$,or
(b) $W(x) < 1/3$.

The language *accepted* by this game consists of those strings x for which alternative (a) holds.

Let $AM(t(n))$ denote the class of languages accepted by Arthur-Merlin games of length $t(|x|)$ with Arthur moving first. Analogously, $MA(t(n))$ corresponds to games where Merlin moves first. Let further

$$AM(P)=MA(P)=\cup\{AM(n^k):k>0\}$$

(games of polynomial length). For $t(n)=c$ (constant) let us use strings of length c to indicate the sequence of players. For example

$$AM(3)=AMA,\ MA(4)=MAMA,\ MA(1)=M.$$

It is the condition that winning probabilities must be *bounded away* from 1/2 that makes such a game a "practical" way for Merlin to convince Arthur that x belongs to L. Clearly, the bounded-awayness condition (ii) makes the classes $AM(t(n))$ much smaller than the corresponding classes defined by Papadimitriou [Pa]. In fact, it seems very unlikely that $coNP$ could be part of $AM(P)$.

2.4. The hierarchy collapses

We shall mainly be concerned with the finite levels of this hierarchy. Clearly, $M=NP$ (Merlin has the power of nondeterminism) and $A=BPP$. Moreover, obviously,

$$AM(k)\cup MA(k)\subseteq AM(k+1)\cap MA(k+1).$$

What may be slightly surprising is that this hierarchy collapses.

Theorem 2.1. For any constant $k\geq 2$,

$$AM = AM(k) = MA(k+1).$$

Sections 3 and 4 are devoted to the proof of this result.

Theorem 2.1 says that the advantage Merlin gains if we force Arthur to reveal all his moves in advance is not too great. Note, however, that the cost of this reduction is a substantial increase of the size of the game. We have to essentially square the game size for each alternation saved. Thus, the following question remains open.

Problem 2.2. Is $AM = AM(P)$?

A short hierarchy still survives:

$$NP\cup BPP\subseteq MA\subseteq AM\subseteq AM(P)\subseteq PSPACE.$$

These inclusions seem more likely to be proper.

2.5. 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 $\Sigma_2\cap\Pi_2$ (P. Gács, see [Sip]). Perhaps the most elegant proof of this fact was given by Clemens Lautemann [Lau]. His proof directly generalizes to AM and MA and gives the following result.

Proposition 2.3. (a) $AM\subseteq\Pi_2$.
(b) $MA\subseteq\Sigma_2\cap\Pi_2$.

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

It remains an open problem whether the unbounded levels of the Arthur-Merlin hierarchy are contained in a finite level of the polynomial time hierarchy. We believe the answer is yes.

Conjecture 2.4. $AM(P)\subseteq\Sigma_k$ for some (finite) k.

Another relation we believe is true is that AM (and even $AM(P)$) does not contain $coNP$. This, of course would imply $NP\neq coNP$. Nevertheless some supporting evidence might be found. For instance one might hope to be able to prove such a separation result relative to a (random) oracle.

2.6. Random oracles

It is straightforward to prove that $AM\subseteq NP^B$ for almost every oracle B.

Using methods standard in recursion theory (cf. [Ku],[Sac, Ch. 10]), one can actually prove the following

Proposition 2.5.
(i) BPP consists of precisely those languages which belong to P^B for almost every oracle B.
(ii) AM consists of precisely those languages which belong to NP^B for almost every oracle B.

This observation shows that AM is a fairly natural complexity class.

2.7. Another hierarchy

Our aim was to put *matrix group order* in as low a complexity class as possible. This is how the class AM arose.

Going in the other direction, one might wonder what is the largest complexity class still giving Merlin (or rather: the Expert) a chance to convince Arthur (or rather: the User) that $x\in L$.

"Interactive proofs with minimum information", a notion recently introduced by Goldwasser, Micali and Rackoff [GMR], motivate the following definition. User and Expert are playing a cardgame (as opposed to the chessgame of Arthur and Merlin). User draws all the cards at random at her first move and hides them from Expert. When it is her turn, User feeds the history of the game (including the input string x) into a polynomial time bounded Turing machine (known to Expert) and reveals the output of the computation to Expert. Expert has unlimited computational power. She prints a string of polynomial length. When the game terminates, a polynomial time bounded Turing machine, known to both players, evaluates the history of the game and declares the winner.

We suggest to call such a game a *User vs. Expert game* if Expert's chances of winning are bounded away from 1/2. Thus we can define the complexity classes $EU(t(n))$ and $UE(t(n))$ in analogy with the corresponding Arthur-Merlin classes.

It is clear that $AM(t(n)) \subseteq UE(t(n))$, $MA(t(n)) \subseteq EU(t(n))$. Further, $E=M=NP$, $U=A=BPP$, $EU=MA$. The first open question is whether the inclusion $AM \subseteq UE$ is proper. Perhaps more intriguing is the question whether there is a collapse in the User-Expert hierarchy. And the final question: is $UE(P)$ (polynomial length User-Expert games) the ultimate random version of NP ? Observe that even $UE(P)$ is unlikely to contain $coNP$.

Finally, can one prove that at least for constant k the class $UE(k)$ is on a finite level of the polynomial time hierarchy? (Recall that $AM(k)=AM \subseteq \Pi_2$.)

3. Arthur-Merlin games

In this section we build the machinery for the proof of Theorem 2.1. The proof will be completed in Section 4.

3.1. Randomized combinatorial games

Let $D_1, \ldots, D_t$ be nonempty finite sets and f a function defined over the Cartesian product

$$dom(f)=D_1 \times \cdots \times D_t.$$

If the range of f is $\{0,1\}$ then f defines a *combinatorial game*. In this game, two players, henceforth denoted M and A, alternate moves. (We identify ourselves with M ; A is the *adversary*.) The i^{th} move of the game consists of picking an element $x_i \in D_i$. The game terminates after the t^{th} move. Player M wins if $f(x_1, \ldots, x_t)=1$. The sequence $(x_1, \ldots, x_t)$ is the *history* of the game and $dom(f)$, the set of all possible histories is the *game space*. The *size* of the game is $\log|dom(f)|$ (base 2 logarithm).

M may or may not be the the first player. In order to properly specify the game we have to tell who moves first. The game is specified by the pair (f,Q) where Q is the first player. ($Q=A$ or M.)

If the range of f is the interval $[0,1]$ then (f,Q) defines a *randomized* game played as follows. The two players make a total of t moves as before and then the referee flips a biased coin and with probability $p=f(x_1, \ldots, x_t)$ declares M to be the winner.

We call f the *payoff function* of this game.

In such games, the strategy of M should aim at maximizing the probability of winning.

3.2. Games against an indifferent adversary

We shall assume that the *adversary* (player A) is *indifferent*, and selects a uniformly distributed random element of D_i for the i^{th} move. On the other hand, M's moves will be assumed to be *optimal* (M has immense computational power). In order to express the winning chance of M in this game, the following formalism will be helpful.

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

$$Axf(x)=\sum\nolimits_{x\in D}\frac{f(x)}{|D|}, \quad Mxf(x)=\max\{f(x)|x\in D\}.$$

Functions $f(x_1,\ldots,x_t)$ defined over the Cartesian product $D_1 \times \cdots \times D_t$ of the respective domains of the variables permit prefixes of the form $Q_1x_1 \cdots Q_tx_t$ where $Q_i=M$ or A.

In a game played optimally against an indifferent adversary, the probability that M wins is clearly

$$Mx_1Ax_2Mx_3 \cdots Qx_tf(x_1,\ldots,x_t)$$

if M moves first (Q is A or M depending on the parity of t), and

$$Ax_1Mx_2Ax_3 \cdots Qx_tf(x_1,\ldots,x_t)$$

if A has the first move.

3.3. Simulation of biased games

We shall be interested in games where one of the players has a significant advantage. These are the games to be played between Arthur and Merlin; Arthur's moves are random. Note that in our model, we know that the game is biased but not in *whose favor*.

Such games will be used to recognize, by a *reliable statistical test*, which of the players has the advantage. We define the *uncertainty* of the game to be

$$unc(f,Q)=\min\{p,1-p\},$$

where p is the probability that M wins. (Q has the first move, f is the payoff function.) The greater the bias, the smaller is the uncertainty (of the outcome of the game).

Our aim is to *simulate* the given game (f,Q) by another biased game (f^*,Q^*) with *fewer moves*, such that the following requirements are met:

(i) The bias of the new game goes in favor of the same player as in the original game.
(ii) The *game space* (the domain of the payoff function) does not increase significantly.
(iii) The uncertainty decreases (or at least does not increase substantially).
(iv) It should be easy to simulate f^* from f. What this means is that given a history u for f^* one can easily compute a small family of histories $v_1,\ldots,v_s$ for f such that the referee's decision α in f^* is an easily computed function of the decisions β_i corresponding to v_i and possibly of additional independent coin tosses. (The β_i and α are random 0-1 variables with expected values $f(v_i)$ and $f^*(u)$, resp.)

Condition (i) expresses that the new game *simulates* the original one. Conditions (ii) - (iv) guarantee that the *complexity* of the game does not increase significantly.

3.4. Increasing the bias

The first step is to turn a *modest* bias 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 domain of f is

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

The game has the same number of moves (t) as (f,Q). Let us denote the i^{th} move by $(x_{i1},\ldots,x_{ik})$ where $x_{ij} \in D_i$. Let $p_j = f(x_{1j},\ldots,x_{tj})$ and let $f^k(x_{11},\ldots,x_{tk})$ be the probability that out of k independent random 0-1 variables ς_j where $E(\varsigma_j)=p_j$, more than half come out to be 1.

Proposition 3.1. Suppose $unc(f,Q) < 1/3$. Then $unc(f^k,Q) < c^k$ where $c=2\sqrt{2}/3 = 0.9428\ldots < 1$.

Proof. The number of boards where the favored player loses is the number of successes in a sequence of k Bernoulli trials each with probability of success less than 1/3. 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 1/3 by any constant less than 1/2.

3.5. Switching moves

In this section we show, how a two-move biased MA game (Merlin first, Arthur second) can be simulated by a two-move AM game in the sense that conditions (i) - (iv) will hold. We describe the simulation after this handy preliminary lemma.

Switching Lemma 3.2. Let X and Y be two nonempty finite sets and let $H(x,y)$ be a non-negative function defined over $X \times Y$. Then

$$AyMxH(x,y) \leq |X| MxAyH(x,y).$$

Proof. For $y \in Y$, let $x(y)$ be Merlin's optimal reply in the AM-game: $MxH(x,y)=H(x(y),y)$. Let $Y(x)=\{y \in Y \mid x(y)=x\}$. Clearly, the $Y(x)$ partition Y.

Let $\psi = MxAyH(x,y)$. It follows that for every $x \in X$,

$$\textstyle\sum_{y \in Y(x)} H(x(y),y) = \sum_{y \in Y(x)} H(x,y) \leq$$

$$\textstyle\sum_{y \in Y} H(x,y) = |Y| AyH(x,y) \leq |Y|\psi.$$

Using this inequality we infer

$$AyMxH(x,y) = AyH(x(y),y) = \frac{\sum_{y \in Y} H(x(y),y)}{|Y|} =$$

$$\sum_{x \in X} \frac{\sum_{y \in Y(x)} H(x(y),y)}{|Y|} \leq \sum_{x \in X} \psi = |X|\psi. \text{ ▮}$$

Of course simply switching the order of moves will give Merlin an unfair advantage, capable of reversing the odds. In order to balance this advantage, we shall ask Arthur to start with independent random first moves on a large number of boards. Merlin will have to give the same response on all boards and still win the majority. We are going to formalize this idea.

Let X and Y again be two nonempty finite sets and $f(x,y)$ a payoff function on $X \times Y$, i.e. $0 \leq f(x,y) \leq 1$. We shall simulate the MA-game (f,M) by the AM-game (F,A) defined as follows.

The game (F,A). We select a positive integer m. The game space will be $dom(F) = Y^m \times X$. A game history is described by a sequence $(\overline{y},x)$ where $x \in X$ and $\overline{y}=(y_1, \cdots, y_m)$, $y_i \in Y$. Merlin is the winner if he wins *more than half* of the (f,M)-games (x,y_i) $(i=1, \cdots, n)$. (These m events are independent; the i^{th} one has probability $f(x,y_i)$.)

Upper Bound Lemma 3.3. Let $\psi = MxAyf(x,y)$ be the probability that Merlin wins in the MA-game (f,M) (the game to be simulated). Then his chance of winning in the simulating AM game (F,A) defined above is

$$A\overline{y}MxF(\overline{y},x) \leq 2^m |X| \psi^{m/2}. \tag{3.1}$$

Proof. Let I denote a subset of size $\lceil m/2 \rceil$ of $\{1,\ldots,m\}$. Clearly, the probability $F(\overline{y},x)$ that Merlin wins after game history $(\overline{y},x)$ is at most the sum over I of the probabilities that he wins all games $\{(x,y_i) : i \in I\}$ in the (f,M)-game. The latter probability is $\prod_{i \in I} f(x,y_i)$. We conclude that for every $x \in X$,

$$\textstyle A\overline{y}F(\overline{y},x) \leq \left(\sum_{\overline{y} \in Y^m} \sum_I \prod_{i \in I} f(x,y_i)\right) / |Y|^m =$$

$$\sum_I \frac{\sum_{\overline{y} \in Y^m} \prod_{i \in I} f(x,y_i)}{|Y|^m} = \sum_I (Ayf(x,y))^{|I|} < 2^m \psi^{m/2}.$$

Since this inequality holds for every x, we obtain that

$$MxA\overline{y}F(\overline{y},x) \leq 2^m \psi^{m/2}.$$

Now an application of the Switching Lemma to F and Y^m in the roles of H and Y, resp., completes the proof. ▮

The proof of the lower bound is more straightforward. We consider the same game (F,A) as above. $1-\psi = 1-MxAyf(x,y)$ is the chance that Merlin loses in the (f,M)-game. We want to show that if $1-\psi$ is small then so is $1-A\overline{y}MxF(\overline{y},x)$.

Lower Bound Lemma 3.4. The probability that Merlin loses in (F,A) is

$$1 - A\overline{y}MxF(\overline{y},x) \leq 2^m (1-\psi)^{m/2}. \tag{3.2}$$

Proof. Let $\Psi = A\overline{y}MxF(\overline{y},x)$. Let x_0 be Merlin's optimal opening move in the (f,M)-game. Clearly, $\psi = Ayf(x_0,y)$ and $\Psi \geq A\overline{y}F(\overline{y},x_0)$. The right hand side in the last inequality is Merlin's chance of winning in (F,A) using the strategy that no matter what Arthur's opening move he selects x_0. The probability that Merlin loses at least half of m independent games under this strategy is less than the sum over all subsets I of $\{1,\ldots,m\}$ of cardinality $\lceil m/2 \rceil$ of the probability that Merlin loses each of the games indexed by $i \in I$, i.e. $1-\Psi \leq \sum_I (1-\psi)^{|I|} \leq 2^m (1-\psi)^{m/2}$. ▮

4. The collapse

In this section we complete the proof of Theorem 2.1. We have to show how to simulate any combinatorial game by an *AM*-game.

Let (g,A) be a game of length $t \geq 3$ starting with Arthur's move. Let $dom(g)=D_1 \times \cdots \times D_t$. We shall switch the second and third moves in this game in the way (F,A) was constructed from (f,M) in Section 3.5. Thus, we select a positive integer m and construct the new game space

$$dom(G)=(D_1 \times D_3^m) \times D_2 \times \prod_{i=4}^{t} D_i^m .$$

The parentheses indicate that the game (G,A) will have one less move than (g,A).

For

$$\overline{w}=((x_1,\overline{x}_3),x_2,\overline{x}_4,\overline{x}_5,\ldots,\overline{x}_t) \in dom(G)$$

we define $G(\overline{w})$ to be the probability that Merlin wins more than half of the m (f,A)-games

$$\overline{w}_i=(x_1,x_2,x_{3i},x_{4i},\ldots,x_{ti}) \quad (i=1,2,\ldots,m).$$

We shall prove that if $unc(g,A)$ is not too large and m is chosen appropriately then G simulates g in the sense of Section 3.3.

Let $n=\max\{6,\lceil \log|D_2| \rceil\}$ (base 2 logarithm).

Recall that the size of a game is the base 2 logarithm of the order of the game space.

Theorem 4.1. Let $m=3n$ and assume $unc(g,A)<1/18$ Then

(a) (G,A) simulates (g,A) (the same player has the advantage in both).

(b) $unc(G,A) \leq 9unc(g,A)$.

(c) $size(G,A) \leq (size(g,A))^2$, provided $size(g,A)>20$.

Proof. Checking (c) is simple arithmetic. To prove (a) and (b), let Q be the favored player in (g,A) and let $\epsilon=unc(g,A)$. Then ϵ is the probability that Q loses. Notice that winning probabilities don't change if we truncate the game at any given level, assigning Merlin's winning probabilities at the leaves of the truncated game tree as payoff function values to the corresponding (short) histories. Moreover, truncating after the third move is interchangeable with the operation of constructing G from g. Therefore we may henceforth assume $t=3$. (Observe that the notion of *randomized* games enables this simplification.)

For greater clarity, we shall write X for D_1, Y for D_2 and Z for D_3.

Let x denote Arthur's opening move in the (g,A)-game. This can also be interpreted as part of the opening move in the (G,A)-game. Once x is fixed, we are left with two two-move residual games: the *MA*-game (g_x,M) and the *AM*-game (G_x,A) defined on $Y \times Z$ and $Z^m \times Y$, resp. They are related to each other precisely in the way (F,A) was constructed from (f,M) in Section 3.5.

Let $u(x)$ and $U(x)$ denote the probabilities that Q loses in (g_x,M) and (G_x,A), resp. Then a combination of the Upper and Lower Bound Lemmas (3.3, 3.4) implies that for every $x \in X$,

$$U(x) \leq 2^m |Y| u(x)^{m/2} \leq 2^{4n} u(x)^{3n/2}. \qquad (4.1)$$

It is easy to see that for $u(x) \leq 1/8$, the right hand side of (4.1) is not greater than $u(x)$. On the other hand, for a random $x \in X$,

$$Prob(u(x)>1/8)<8E(u(x))=8\epsilon \qquad (4.2)$$

and therefore the probability that Q loses in the (G,A)-game is

$$E(U(x)) < E(u(x)) + 8\epsilon = 9\epsilon$$

By our assumption $\epsilon<1/18$ this means that Q is still the favored player, proving (a), and that $unc(G,A)$ is the probability that Q loses, thus less than 9ϵ, proving (b). ▮

Corollary 4.2. Let (f,Q) be a biased t-move combinatorial game. Then (f,Q) can be simulated by a combinatorial *AM*-game (F,A) such that

$$size(F) \leq c_t(size(f))^{2^{t/2}},$$

where c_t is a constant, depending on t, the bias of (f,Q) and the desired bias of (F,Q).

Note that here we insist that the games be combinatorial, i.e. the values of the of the payoff functions are 0,1. This is necessary for the proof of Theorem 2.1.

Proof. First, at a price of an $O(t)$ increase in size we achieve that (f,Q) has uncertainty 9^{-t} (Proposition 3.1). Then we apply Theorem 4.1 repeatedly at most $t/2$-times. (If $Q=M$, we may introduce a dummy first move for Arthur for convenience.) Finally we use Proposition 3.1 again to achieve the desired bias. Clearly both constructions used preserve the combinatorial nature (0-1 payoff) of the games (although the proof required a detour into randomized games). ▮

This result immediately implies Theorem 2.1. ▮

Problem 4.3. Give some evidence, perhaps in a limited model, that *MA*-games are not strong enough to simulate *AM*-games.

5. Statistical verification of the order of a matrix group

5.1. Statement of the problems

We shall consider matrix groups G over a finite field F. Each group will be given by a *list of generators.* We are mainly concerned with the following two problems.

Membership: the set of pairs (g,G) where $g \in G$.

Exact order: the set of pairs (N,G) where $|G|=N$.

In addition, we need the following classes.

p-groups: $\{(p,G)$: $|G|$ is a power of the prime $p\}$.

Divisor of order: $\{(N,G)$: the integer N divides $|G|\}$.

Lower bound: $\{(N,G)$: $N \leq |G|\}$

Upper bound: $\{(N,G)$: $|G| \leq N\}$.

It is easy to see that if *exact order* belongs to *NP* then it also belongs to *coNP* and so does *membership* as well. If *divisor of order* belongs to *NP* then so does *lower bound.* If both *lower* and *upper bound* belong to *NP* then so does *exact order.*

Furthermore, all these statements relativize to any oracle and in particular remain true if *NP* is replaced by *AM* everywhere.

5.2. Approximate bounds

The problems of verifying approximate upper or lower bounds cannot be stated as language recognition problems. Randomized complexity classes such as those related to the Games against Nature - with a "continuous" spectrum of accepatance probabilities - are particularly suited for formalization of approximate verification problems.

Let $C=AM(t(n))$ or $MA(t(n))$ be one of the complexity classes discussed in Section 2.

We shall say that the *approximate upper bound* problem *belongs to class C* if there exists a corresponding *Game against Nature* of length $t(n)$ (not necessarily satisfying 2.3 (ii)) taking input strings of the form (N,G) such that

(i) if $|G| \geq 2N$ then $W(N,G) < 1/3$;
(ii) if $|G| \leq N$ then $W(N,G) \geq 2/3$.

(Recall that $W(N,G)$ denotes the probability that Merlin is able to win on input (N,G).)

We define the complexity of *approximate lower bounds* analogously.

5.3. Main results

We outline those partial results which will add up to a proof of Theorem 1.1. The central part is the following.

Theorem 5.1. The *approximate upper bound* problem belongs to *MAM*.

On the other hand, a technique of Sipser [Sip] implies quite generally (not only for groups, but for level-sets of any *NP*-set of strings) that

Theorem 5.2. The *approximate lower bound* problem belongs to *AM*.

A simple application of Sylow's Theorem and the Reachability Theorem [BS, p.232] shows that Theorem 5.2 automatically implies its stronger version, namely

Corollary 5.3. *Divisor of order* belongs to *AM*.

Proof. In order to verify that N divides the order of G, Merlin guesses and verifies the prime factorization of N. For each prime power dividing N, he guesses a corresponding Sylow p-subgroup P of order, say, p^e. He verifies that P is a subgroup of G (membership test) and that P is a p-group. (Both properties belong to *NP* [BS].) Then he verifies (via Theorem 5.2) that $|P|>p^e/2$. This implies that p^e divides $|G|$. ∎

Corollary 5.3 clearly implies that *lower bound* belongs to *AM*.

We remark that *divisor of order* actually belongs to *NP* (this is the main result of [BS]; its proof occupies the first ten sections of that paper) but we don't need that fact now.

Finally, Corollary 5.3 and Theorem 5.1 immediately imply that *exact order* belongs to *MAM*. As a matter of fact, if we know that the order of G is divisible by N and is less than $2N$ then by Lagrange's theorem we conclude that $|G|=N$.

Therefore, by Theorem 2.1, *exact order* belongs to *AM*. Thus, in view of the comments in Section 5.1, Theorem 1.1 follows. ∎

5.4. A combinatorial lemma

The *approximate upper bound* algorithm of class *MAM* will be based on the following elementary result.

Lemma 5.4. [BE] Let G be a finite group of order N and let

$t=\lfloor \log N + \log \ln N + 3 \rfloor$

Then there exist elements $x_1,\dots,x_t \in G$ such that every member of G occurs among the 2^t subproducts $x_1^{\epsilon_1} \cdots x_t^{\epsilon_t}$ where $\epsilon_i = 0,1$.

(log and ln stand for base 2 and base e logarithms, resp.)

5.5. The "approximate upper bound" game

We outline the proof of Theorem 5.1.

First we note, that for the parameters of Lemma 5.4, $2^t/N < 8 \ln N$. Hence, "on average", each element of G is represented a small number of times; small meaning at most a constant times the length of the input. (In a uniform encoding, the strings representing the elements of G cannot be shorter than $\log N$. - The average referred to above should be interpreted as *harmonic mean*, i. e. the reciprocal of the arithmetic mean of the reciprocals.)

This observation makes it possible for Merlin, having guessed $x_1,\dots,x_t$, to convince Arthur that there are unlikely to be more than, say, $9N/8$ elements of G represented by the set S of the 2^t subproducts of the x_i. To this end, the players proceed as follows.

Merlin declares the values of N and t and exhibits a sequence $x_1,\dots,x_t$ of elements of G. Arthur picks $m=t^2$ random subproducts s_j of the x_i (choosing any of the 2^t possibilities with equal probability each time). For each j, Merlin exhibits a number $1 \leq r_j \leq t^2$ of representations of s_j as products of different subsets of the x_i. The average T of the numbers $2^t/r_j$ is calculated. If $T<17N/16$ then Merlin is declared the winner and Arthur accepts the inequality $|G|<9N/8$.

Clearly, Merlin's optimal strategy is to exhibit as many representations for each s_j as possible. If s_j is represented in n_j different ways as a subproduct, let $\bar{n}_j = \min\{n_j, t^2\}$. Playing optimally, Merlin demonstrates that there are at least $\bar{n}_j$ representations of s_j. It is easy to see that the numbers $1/n_j$ are unbiased estimators of $|S|2^{-t}$. Their variance is clearly less than their expected value (this is true of any random variable with range between 0 and 1). Therefore with increasing m, a constant relative error becomes exponentially unlikely. The effect of replacing n_j by $\bar{n}_j$ is clearly negligible if t is large. This argument proves that the above strategy guarantees an almost certain win for Merlin if $|S| \leq N$

and no strategy will give him a non-negligible winning chance if $|S| \geq 9N/8$.

The other task of Merlin is to convince Arthur that S contains nearly all members of G, say $|G| < 8|S|/7$.

One can prove that for sufficiently large t the following two claims imply $|G| < 8|S|/7$:

(i) the x_i generate G ; and

(ii) for each x_i, $|S - Sx_i| < |S|/t^2$.

Merlin will have to produce a short proof of (i). This is possible because *membership* belongs to NP [BS].

The verification of (ii) requires another AM class statistical test (which can be performed in parallel with the above described statistical verification of the inequality $|S| < 9N/8$). Arthur, as before, selects a large number of random subproducts s_j of the x_i. Merlin responds by presenting, for each k, $1 \leq k \leq t$ and for each of Arthur's s_j's a representation of $s_j x_k$ as a member of S, i.e. a subproduct of the x_i. Merlin wins if he is able to produce such a representation for each pair (j,k). Clearly, if $S=G$ then Merlin will be able to win (always). If, however, (ii) fails, then he has a negligible chance of winning.

Putting this all together we conclude that if Merlin wins in both games then $|G|$ must be less than $9N/7$ unless Merlin was improbably lucky. On the other hand, if $|G| \leq N$ then Merlin is able to win in both games almost always.This proves Theorem 5.1.∎

6. Black box groups

As in [BS], matrices really have little to do with these results. We only need two properties of the matrix groups over finite fields:

(i) The elements of the group are encoded by strings of uniform length.

(ii) Group operations are performed in polynomial time.

Note that integer matrices violate (i).

A *black box group* is defined by (i) and

(ii') Group operations are performed by an oracle.

We should also assume (in contrast to [BS]) that the codes are unique, i.e. each string corresponds to at most one element of the group. (This is true for matrix groups, but not for their factor groups. Induction arguments involving factor groups motivated the omission of this assumption in [BS].)

All results of this paper remain valid under these virtually minimal assumptions.

References

[Ad] L. Adleman, Two theorems on random polynomial time, in: Proc. 19th IEEE Symp. Found. Comp. Sci., 1978, pp. 75-83

[Ba] L. Babai, On the complexity of matrix group problems II, in preparation

[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 Discr. Math. 12 (1982), pp. 27-30

[BS] L. Babai and E. Szemerédi, On the complexity of matrix group problems I, in: Proc. 25th IEEE Symp. Found. Comp. 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. Comp. 10 (1981), 96-113

[BKL] L. Babai, W. M. Kantor and E. M. Luks, Computational complexity and the classification of finite simple groups, in: Proc. 24th IEEE Symp. Found. Comp. Sci., Tucson AZ 1983, pp. 162-171

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

[FHL] M. L. Furst, J. Hopcroft and E. M. Luks, Polynomial-time algorithms for permutation groups, in: Proc. 21st IEEE Symp. Found. Comp. Sci., Syracuse, N. Y. 1980, pp. 36-41

[GG] O. Gaber and Z. Galil, Explicit construction of linear sized superconcentrators, J. Comp. Syst. Sci. 22 (1981), 407-420

[Gi] J. Gill, Computational complexity of probabilistic Turing machines, SIAM J. Comp. 6 (1977), 675-695

[GMR] S. Goldwasser, S. Micali and C. Rackoff, The knowledge complexity of interactive protocols, in: Proc. 17th ACM Symp. Theory of Comp., Providence, R. I. 1985 (this volume)

[Ku] S. A. Kurtz, Randomness and genericity in the degrees of unsolvability, Ph. D. Thesis, Univ. of Illinois at Urbana, 1981

[Lau] C. Lautemann, *BPP* and the polynomial hierarchy, Info. Proc. Letters 17, no.4 (1983) 215-217

[Mar] G. A. Margulis, Explicit constructions of concentrators, Probl. Pered. Info. 9 (1973), 71-80 (English transl. in Probl. Info. Transm. (1975), 325-332)

[Mih] K. A. Mihailova, The occurrence problem for direct products of groups (Russian), Dokl. Akad. Nauk SSSR 119 (1958), 1103-1105 and Mat. Sb. (N. S.) 70 (112) (1966), 241-251

[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: Proc. 24th IEEE Symp. Found. Comp. Sci., Tucson AZ, 1983, pp. 446-450

[Pin] M. Pinsker, On the complexity of a concentrator, 7th Internat. Teletraffic Conf., Stockholm 1973, 318/1-4

[Pip] N. Pippenger, Superconcentrators, SIAM J. Comp. 6 (1977), 298-304

[Sac] Gerald E. Sacks, Degrees of unsolvability, Annals of Math. Studies 55, Princeton Univ. Press, Princeton N.J. 1966 (2nd ed.)

[Sim] C. C. Sims, Some group theoretic algorithms, in: Lect. Notes in Math., Springer, N. Y. 1978, pp. 108-124

[Sip] M. Sipser, A complexity theoretic approach to randomness, in: Proc. 15th ACM Symp. on Theory of Comp., Boston 1983, 330-335

[SS] R. Solovay and V. Strassen, A fast Monte Carlo test for primality, SIAM J. Comp. 6 (1977) 84-85

[St] L. Stockmeyer, The polynomial time hierarchy, Theor. Comp. Sci. 3, no. 1 (1976), 1-22