Arthur-Merlin Games: A Randomized Proof System, and a Hierarcy of Complexity Classes

Stefano Passador

University of Udine

passador.stefano@spes.uniud.it

August 10, 2018

Overview

1. Introduction
2. Preliminaries
3. Switching Moves: Proof of the Speedup Theorem
4. Arthur-Merlin Protocols for Approximate Lower Bound Verification
5. An Explicit AM Protocol for Graph Nonisomorphism and Coset Intersection in Permutation Groups.

Arthur-Merlin system is a randomized interactive proof system where 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.
Arthur-Merlin proof system is based on public coint tosses.

Introduction

What is a Proof?

An important party was certainly missing to make the proof process complete: **the verifier**. Intuitively a proof is not a sequence of symbols, but a process of **convincing a distrustful party**.
We shall introduce a notion of **randomized (interactive) proof**, combining the nondeterministic nature of a classical proof system (the prover just prints the proof, the verifier doesn't ask how he got it) with the randomization performed by the verifier. Although this notation itself has nothing to do with how efficient the proof is, the motivation behind it is complexity theoretic and therefore we combine the two aspects (randomization and efficiency) in the definition.

Efficient Provability

Arthur recognises 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?

If $L \in NP$, 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*.

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.

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

- $W(x) > 2/3$ (x is accepted), or
- $W(x) < 1/3$ (x is rejected).

The Arthur-Merlin Hierarchy

In order to take efficiency into account, we require that the total number of moves and the total length of the strings printed of the game as well as the running time of the referee be bounded by a polynomial of length of x. The language defined by such protocol is the set of strings x for which Merlin's winning chances are greater than 1/2 (and thus greater than 2/3). Let $t(n)$ be a polynomially bounded function of $n = |x|$. Languages accepted by Arthur-Merlin games with moves $\leq t(n)$, form the classes $AM(t(n))$ (Arthur moves first) and $MA(t(n))$ (Merlin moves first). Let further:

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

The Arthur-Merlin Hierarchy

For $t(n) = c$ (c constant) we use strings of length c to indicate the sequence of players. (ex. AM(3) = AMA, MA(4) = MAMA. . .).
It shoud be clear that $MA(1) = NP$ and $AM(1) = BPP$. All languages in AM belong to NP^c, for almost every oracle C.

Inheriting Forefathers' Proof

An advantage of AM over higher levels of the hierarchy is that only a limited interaction is required for a proof of membership in languages $L \in AM$.
With Arthur-Merlin, if Merlin has to respond to Arthur's random string they cannot leave the proof to future generation 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. If humanity can agree on a certified random string, such string can replace Arthur's opening move and Merlin's reply might be worth recording for posterity. But this solution is no longer available if Arthur has to move after a move of Merlin and then Merlin has to reply again.

Inheriting Forefathers' Proof

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 can 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.

Games against Nature

“Games agains nature” is used to describe complexity classes arising from polynomially bounded games against an indifferent, randomising adversary. Arthur-Merlin games are particular “games against nature”, the crucial restriction being the condition that the winning chances are always *bounded away* from $1/2$. 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.

Relation to the Polynomial Time Hierarchy

BPP is contained within the polynomial time hierarchy, in fact it is contained in $\Sigma_2^p \cap \Pi_2^p$. The proof directly generalizes to AM and MA and gives the following result:

Proposition 1:

1. $AM \subseteq \Pi_2^p +$
2. $MA \subseteq \Sigma_2^p \cap \Pi_2^p$

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

Relation to the Polynomial Time Hierarchy

The unbounded levels of the Arthur-Merlin hierarchy are not believed to be contained in a finite level of the polynomial time hierarchy. For any unbounded $t(n)$, there exists an oracle C such that $AM(t(n))^c$ is not contained in $PH^c = U_{k>0} \sum_k^{p,c}$.
Another plausible relation si that AM (and even $AM(poly)$) does not contain coNP. This would imply $NP \neq coNP$. Nevertheless, supporting evidence can be found.
Proposition 2: If $coNP \subseteq AM$ then the polynomial time hierarchy collapses to $\Sigma_2^p = \Pi_2^p = AM$.
Another piece of evidence is relativized separation: $AM^c \not\supseteq coNP^c$.
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. One hopes that separation results such as $coNP \not\subseteq AM$ are provable in that model.

Collapse and Speedup

$$AM(t(n)) \cup MA(t(n)) \subseteq AM(t(n)+1) \cap MA(t(n)+1)$$

It is slightly surprising that the finite levels of the Arthur-Merlin hierarchy collapse:

Theorem 3 For any $t(n) \geq 2$ (where $t(n)$ is polynomially bounded, $AM(t(n)) = AM(t(n)+1) = MA(t(n)+1)$.

In particular, for constant $k \geq 2, AM = AM(k) = MA(k+1)$.

Proof of proposition 2 Assume $coNP \subseteq AM$. We claim $\Sigma_2^p = \Pi_2^p = AM$. It suffices to prove $\Sigma_2^p \subseteq AM$, since $AM \subseteq \Pi_2^p$. Let $L \in \Sigma_2^p$. By definition this means that for some $L_2 \in coNP$,

$x \in L$ iff $\exists^p y : (x, y) \in L_1$ (where $\exists^p$ refers to polynomially bounded quantifier).

Now, by assumption, $L_1 \in AM$. Therefore, $L \in MAM$. But $MAM = AM$ by the Collapse Theorem.

Collapse and Speedup

The Collapse Theorem leaves us with the following short hierarchy:

$$NP \cup BPP \subseteq MA \subseteq AM \subseteq AM(poly) \subseteq PSPACE$$

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 $PSPACE$. Oracles separating MA and AM, AM from AM(poly), AM(poly) from PSPACe have been found. And each of these results operates on the relation between members of the Arthur-Merlin hierarchy and the polynomkial time hierarchy.

Collapse and Speedup

Theorem 4: Speedup Theorem For $t(n) \geq 2$,

$$AM(2t(n)) = AM(t(n))$$

. The proof remains valid under any oracle. There exists an oracle C relative to which no unbounded speedup is possible: if $t(n)/s(n)$ is unbounded then $AM(t(n))^c \neq AM(s(n))^c$.

Approximate Lower Bounds

Let the NP-language L consists 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 $|L(x)| \geq f$. 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 $|L(x)| \geq (1 + \epsilon)f$; and gives him almost no chance if $|L(x)| < f$, for any given constant ϵ. (In fact ϵ can even depend on n al long as it decreases at most at a polynomial rate).
The protocol employs universal hashing.

The Class almost-NP and Yet another Hierarchy

Let *almost-NP* be the class of those languages which belong to NP^B (with B any oracle).
It's clear that AM $\subseteq$ almost-NP. By the analogy with BPP (which is equal to almost-P), one would expect that AM $=$ almost-NP. So:
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: $AM^B = NP^B$ for almost every oracle B.
The idea is that first we improve the error probability on input x from $1/3$ 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; these can be repaired by extending the finite control.

The Class almost-NP and Yet another Hierarchy

Nevertheless, it is natural to ask:

Problem 2: Does here exist an oracle separating AM from almost-NP?

Theorem 6: For almost every pair of oracles B, C,

1. $BPP = P^B \cap P^C$ and
2. $almostNP = NP^B \cap NP^C$

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's 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 Class almost-NP and Yet another Hierarchy

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, amost-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 computation paths of 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, $ANP =^{def} ANP(2) = almostNP$.

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 interaction is minimal. If we seek further related proof systems, we believe randomization will be guiding line and interaction a possible 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.

Preliminaries

Formalism

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.

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

Functions $f(x_1, x_2, \dots, x_t)$ defined over the Cartesian product $D_1 \times \dots \times D_t$ of the respective domains of the variables permit prefixes of the form $Q_1 x_1 \dots Q_t x_t$ 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 $f(x)$.

Formalism

We define the random variables $\exists x \phi(x)$ and $Rx\phi(x)$ 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

$$Rx\phi(x) = \phi(\xi)$$

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

$$E(\exists x \phi(x)) = Mxf(x) \textit{ and } E(Rx\phi(x)) = Axf(x).$$

Now, inductively, for $D = D_1 \times \ldots \times D_t$ we can define the random variable (over the sample space $D_2 \times D_3 \times \ldots \times \Omega$:

$$\exists x_1 Rx_2 \ldots Q_t x_t \ \phi(x_1, \ldots, x_t)$$

Its expected value will be:

$$Mx_1 Ax_2 \ldots S_t x_t f(x_1, \ldots, x_t)$$

Formalism

Now, inductively, for $D = D_1 \times \ldots \times D_t$ we can define the random variable (over the sample space $D_2 \times D_3 \times \ldots \times \Omega$:

$$\exists x_1 R x_2 \ldots Q_t x_t \ \phi(x_1, \ldots, x_t)$$

Its expected value will be:

$$M x_1 A x_2 \ldots S_t x_t f(x_1, \ldots, x_t)$$

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

Randomized Combinatorial Games

If $D = D_1 \times \ldots \times D_t$ 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 $\phi(x_1, \ldots, x_t)$ 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.

Randomized Combinatorial Game

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 $D^i = D_1 \times \ldots \times D_i$. The histories form a rooted tree in a natural way, $(x_1, \ldots x_{i-1})$ being the parent of $(x_1, \ldots, x_i)$. This is the *game tree*.
The size of the game is $log|D|$.

Games against an Indifferent Adversary. Truncated Games

Hencefoth we assume that Arthur's moves are random. An AM(t) game has t moves with Arthur moving first; in a MA(t)-game, Merlin moves first. Given a partial game history $(x_1, \ldots, x_i) \in D^i$ and assuming Merlin's moves to be optimal after the ith move (ties are resolved according to a predetermined ordering of D_j, the outcome of the game can be described by the partial payoff variable (again a (0, 1) random variable)

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

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

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

where $S_j = M$ if $Q_j = \exists$ and $S_j = A$ if $Q_j = R$.

Games against an Indifferent Adversary. Truncated Games

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 we mean the assignment of the values $f^i(x)$ to the nodes $x = (x_1, \dots, x_i) \in D^i$ 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: $f^i(x) = Ax_{i+1} f^{i+1}(xx_{i+1})$

For instance, in an AMAM-game defined by the random variable:

$$\phi^0 = Rx_1 \exists x_2 Rx_3 \exists x_4 \phi(x_1, x_2, x_3, x_4)$$

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.

Games against an Indifferent Adversary. Truncated Games

The reason for introducing randomized game is that this family is closed under truncation. The ith truncation (ϕ^i, Q) of a game (ϕ, Q) with game space $D = D_1 \times \ldots \times D_t$ 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 $\phi^i(x_1, \ldots, x_i)$ 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* $(\phi_x, Q_i + 1)$ with game space $D_{i+1} \times \ldots \times D_t$ where Q_{i+1} is the initial of the player making move $i+1$; and

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

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

Biased Games, Arthur-Merlin Protocols

We call a game against an indifferent adversary *biased* if the game value is either $> 1/3$ or $< 1/3$. The game favors Merlin in the former case and Arthur in the latter. The *uncertaintiy* of the game (ϕ, Q) is $unc(\phi, Q) = minf^0, 1 - f^0$, where f^0 is the game value, the probability that Merlin wins. So, a biased game is one with uncertainty $< 1/3$.
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 $t = t(n), n_1, \ldots, n_t$ and thus generates the game space $D_1 \times D_2 \times \ldots \times D_t$ 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.

Simulation of Arthur-Merlin protocols

Let $P1$ and $P2$ be two Arthur-Merlin protocols. 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. 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 will be the (cost of simulation) plus (the number of queries) times (old complexity).

Decreasing the Uncertainty

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 ol dgame. Let $\phi_i(u_i)$ be independent realization of the random variables $\phi(u_i)$ (replacing the sample space Ω by Ω^k).

Decreasing the Uncertainty

We set $\phi^k(u) = maj(\phi_1(u_1), \dots, \phi_k(u_k))$ where "maj" is the strict majority function.

Proposition 7: Suppose $unc(f, Q) < 1/3$. Then $unc(f^k, Q) < c^k$, where $c = 2\sqrt{2/3} < 1$.

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 $1/3$. 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 $1/2$. The cost of this simulation is $k \times (1 + log|D|)$.

Switching Moves: Proof of the Speedup Theorem

Theorem 8. For polynomially bounded $t(n) \geq 2$,
$AM(2t(n)) = AM(t(n) + 1)$.

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 protocol P2 allowing only $t/2 + 1$ moves. Set $\epsilon = (24t)^{-t}$. 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 $< \epsilon$ or $> 1 - \epsilon$ for Merlin. Let s be the (increased) game size and $m = 4st$. We may assume $s > n$.

Simulation

The simulation goeas 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 AMAMA . . . to AAMA AAMA AAMA . . . .

Suppose an MAM segment of P1 consists of Merling selecting a string $x \in X$, then Arthur selecting $y \in Y$ and Merling selecting $z \in Z$. The corresponding part of P2 will consist of Arthur selecting $(y_1, \dots, y_m) \in Y^m$, then Merling selecting $x \in X$ and $(z_1, \dots, z_m) \in Z^m$, finally Arthur selecting $i \in \{1 \dots m\}$. 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 onlly "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 revel 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 counterthis 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).

Motivation

Continuing in this fashion and eventually declaring the winner of the majority to be the winner would suffice for a proof of the Collpase 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 $(y_1, \ldots, y_m)$, 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 where 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.

Analysis

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 (Ry) is exponentially unlikely to give Merlin $> 1/(2t)$ chance that this expected payoff, after Arthur's second move (Ri), will exceed $12t\delta$

Lemma 9. For $y \in Y^m$, let $C(y)$ denote the event

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

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 $x \in X, Ayf^2(x,y) \leq \delta$. Therefore, for every x and random y we have

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

Analysis

The probability of the bad event $B(x, y)$ that $f^2(x, y_i) > 12t\delta$ happens for more than $m/(2t)$ out of the m values $1 \leq i \leq m$ (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 $C(y) = (\exists X)B(x, y)$. We thus have $Prob(C(y)) \leq \sum_{x \in X} ProbB(x, y)$ completing the proof of the lemma.

Analysis

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

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

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

Proposition 10. If the value of an Arhtur node in a game tree is δ then the value of at most $1/(2t)$ fraction of its children exceeds $2t\delta$.

Analysis

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. Let us call the successive nodes in each AAMMA segment u, y, x, z, i-nodes, respectively. A u for instance is an A-node at the beginning of the AAMMA segment. If we label a u-node by its history h, then Arthur's next move, say u takes us to the child node labeled by the history hu.
After every i-move, the history $h = (\ldots, u, y, x, z, i)$ of the P2-game defines a unique history $\pi(h) = (\ldots, u, x, y_i, z_i)$, 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. If h is a projectable node of the P2-tree, let $e(h) = f(\pi(h))$ denote the value of the node $\phi(h)$ in the P1-tree.

Analysis

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 hy of a 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 y-node are lucky.

Finally, we call a child hi of a i-node h lucky, if $h = h'yxz$ for some 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.

Analysis

Leet now $h = (u_1, y_1, \ldots, z_{t/4}, i_{t/4})$ be a P2-game history. If none of the initial segments of the game represent a lucky node, then $f(h) \leq (2t)^{t/4}(12t)^{t/4}\epsilon < 24^{-t/2} < 1$, 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 $Prob(L_i|M_i)$, where L_i denotes the event that the ith move hit a lucky node, while M_i standa 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 $\leq 1/4 + t2^{-n-2} < 1/3$.

Analysis

It is clear that the Speedup Theorem and therefore the Collapse Theorem follow from Theorem 9, 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.

Analysis

Lemma. 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 $Rx\exists yRzf(x,y,z)$. We may assume that the uncertainty of the game is $< 1/6$. Let us define the simulating protocol P2 as $Rx\exists ymaj_z f(x,y,z)$. Here $maj_z g(z)$ takes the value 1 if $\sum_{z \in Z} g(z) \geq |Z|/2$; 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.

Analysis

We have to show that the simulation is correct. **Claim.** For any $(0,1)$-valued function $f(x,y,z)$ over the finite domain $X \times Y \times Z$,

$$AxMymaj_z f(x,y,z) \leq 2AxMyAzf(x,y,z)$$

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 $g(z)$, $maj_z g(z) \leq 2Azg(z)$. Moreover, the average, maximum and minimum operators are semihomogeneous.

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 $1/6$. This completes the proof of the Speedup and Collapse Theorems.

Arthur-Merlin Protocols for Approximate Lower Bound Verification

Approximate lower bound verification

For a language L consisting of paris (x, y) of strings such that $|x| = |y|$, let $L(x) = \{y|(x, y) \in L\}$.
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 $C = AM(t(n))$ or $MA(t(n))$.
Fix some $\epsilon > 0$. 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:

- if $|L(x)| \geq (1 + \epsilon)N$ then $W(N, x) > 2/3$
- if $|L(x) < N$ then $W(N, x) < 1/3$

Approximate lower bound verification

(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 based on universal hashing, one can show:
Theorem 12. For any $L \in NP$ and $\epsilon > 0$, an ϵ-approximate lower bound protocol of class AM exists. **Proof.** First of all we remark, that the Collapse and Speedup Theorems apply to approximate lower bound protocols as well. Therefore it suffies 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 $\{0,1\}^n$. In this case Arthur selects m random stringsy$_i$ 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 $m|L(x)|2^{-n}$ cases, we declare him the winner; otherwise Arthur wins.

Approximate lower bound verification

Obviously, choosing any $m > c/\epsilon$ guarantees the validity of the two points for some absolute constant c.

Now we turn to the general case. We identify $\{0,1\}^n$ with the n-dimensional space other 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 $2?k \geq |S|/\alpha$. Then there exists a $k \times n$ $\{0,1\}$-matrix C s.t. $|C(S)| \geq (1-\alpha)|S|$, where $C(S) \subseteq \{0,1\}^k$ is the image of S under the linear map C.

Approximate lower bound verification

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 $Prob(Cz = 0) = 2^{-k}$. Therefore, for any two distinct $u, v, \in \{0,1\}^n$, $Prob(Cu = Cv) = Prob(C(u - v) = 0) = 2^{-k}$. 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 $\leq |S|2^{-k}$. Therefore the expected number of mateless members of $|S|$ is $\geq (1 - \alpha)|S|$, and this, clearly, is a lower bound on the expected size of $C(S)$. Consequently, for some C we have $|C(S)| \geq (1 - \alpha)|S|$.

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

Approximate lower bound verification

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

If $|S| \geq (1+\epsilon)N$ then Merlin can assure (by the lemma) that $|C(S)| \geq (1+\delta)N$. 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 choses, $|C(S)| \leq |S| < N$, so Merlin is likely to lose.

Approximate lower bound verification

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.

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

Graph nonisomorphism

Theorem. 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:

1. if X and Y are isomorphic, then $c = 2ab$
2. 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 (2), it suffices, however, to verify an approximate lower bopund for a and b and an approximate upper bound for c, each with factor of $2^{1/3}$.

Graph nonisomorphism

The approximate lower bound follow immediately from the result stated in the previous section.

We remark that actually, this AM-class approximate lower bound estimate can be replaced by a stronger scheme, namely the inequality $|AutX| \geq a$ (or even the relation that a devides$|AutX|$) 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 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

$$n!/|AutZ|$$

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.

Coset intersection

Assume that we are given two permutation groups G, H acting on the same set S and two permutation g, h of S. The coset intersection problem asks wheter or not the cosets Gg and Hh intersect (permutation groups are given by a list of generators).
Graph isomorphism can be reduced to this problem. 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 $s \in S$.

Coset intersection

Theorem 15. The coset intersection problem is in $NP \cap coAM$. **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 $|\hat{\mathrm{G}}| = 2|G|^2$. 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 $\hat{\mathrm{G}}$. We again have alternatives (1) and (2), 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 $|\hat{\mathrm{G}}|$.

The End