Arthur-Merlin protocols

January 21 (and 28), 2004

1 Introduction

In the previous lecture we considered games of perfect information between two players P_1 and P_2, monitored by a verifier V. We observed that the optimal strategy for the players in such games is deterministic, and asked what is the complexity of finding the optimal strategy as a function of the complexity of V. That is, which languages L have a game in which if $x \in L$, P_1 has a winning strategy in the game indexed by x, and if $x \notin L$, P_2 has a winning strategy. We showed that for polynomial time V we get PSPACE, and for polynomial space V we get EXPTIME. If V is polynomial time the number of moves in the game is bounded by k, we get Σ_k, the kth level of the polynomial hierarchy (and Π_k for the complement of these languages). It is not known whether the finite levels of the polynomial hierarchy are distinct, or whether the polynomial hierarchy collapses.

We shall consider games in which P_2 is indifferent to the result of the game, and just plays randomly. In this case, P_2 can be simulated by V (if we augment V with randomness), and we get the class known as Arthur-Merlin protocols. Arthur-Merlin protocols that take a constant number of rounds give the complexity class known as AM.

2 Definitions

An *Arthur Merlin* protocol (introduced by Babai) is a game played between a computationally unbounded prover Merlin (M), and a random polynomial time verifier Arthur (A). On common input x, the parties exchange messages, where Arthur chooses his messages at random (just sends sequences of random bits). After some prespecified number of rounds the protocol ends, and Arthur evaluates a publicly known predicate on x and the sequence of messages, and announced *accept* or *reject*. A round in the protocol is a message of Arthur followed by a message of Merlin.

The class AM_k contains all languages L that have a k-round Arthur-Merlin protocol with the following properties:

- If $x \in L$, there is a strategy for Merlin s.t. Arthur accepts with probability at least 2/3.
- If $x \notin L$, for every strategy for Merlin, Arthur accepts with probability at most 1/3.

The classes MA_k are defined similarly, with the difference that a round is a message of Merlin followed by a message of Arthur. The complexity classes AM_1 and MA_1 are simply denoted as AM and MA. They may be viewed as a combination of NP and BPP. There are some interesting languages in AM that are not known to be neither in NP nor in BPP, but we postpone the discussion of this to a later section.

Let us note that the constants 2/3 and 1/3 are somewhat arbitrary. Similar to the case of BPP, the error can be decreased exponentially by repeating the protocol l times with independent coin flips, and taking a majority vote over the outcomes. Some extra care needs to be taken here, because we wish to repeat the protocol in parallel so as to preserve the number of rounds. The problem is that the executions of the parallel copies are not independent, because Merlin can base his messages in one copy on the messages he sees from Arthur in other copies. Nevertheless, the proof goes through (e.g., by induction on the number of rounds). We remark here that in other contexts (perhaps the most famous one is that of two prover one round protocols) parallel repetition does not work as expected.

3 A hierarchy that collapses

For the polynomial hierarchy, it is an open question whether one of Σ_2 and Π_2 contains the other (in which case they are in fact equal, and moreover, the PH then collapses).

For Arthur-Merlin protocols, the situation is easier. We have:

Theorem 1 *$MA \subset AM$.*

Proof: Let $L \in MA$. Then after error reduction we have an MA protocol with verifier V and message lengths $2 < m < n^{O(1)}$ such that:

- If $x \in L$, there is some w such that for at least a $1-2^{-n}$ fraction of strings r, $V(x, w, r)$ accepts.
- If $x \notin L$, for every w, for at most a 2^{-n} fraction of strings r, $V(x, w, r)$ accepts.

We now construct an AM protocol for L.

1. Arthur picks at random strings $r_1, \dots r_m$ and sends them to Merlin.
2. Merlin returns a pair w, r'.
3. Arthur accepts if all of $V(x, w, r' \oplus r_i)$ accept, where $r' \oplus r_i$ denotes the bitwise *exclusive or* of the strings r' and r_i.

If $x \in L$, a simple counting argument shows that if Merlin sends the same w that he would have sent in the original MA protocol, he can always find a string r' that would cause Arthur to accept (because cycling over all r', $r' \oplus r_i$ cycles over all r, and there was only an exponentially small fraction of r (and hence r') that would not cause acceptance).

If $x \notin L$, then for every w, r', the probability that r_i leads to accept is at most 2^{-n}. Hence the probability of acceptance is 2^{-mn}. As there are only 2^{2m} choices for w, r', the union bound shows that for almost all choices of $r_1, \dots, r_m$ Arthur rejects. $\square$

We remark that the protocol that we designed in the proof has *perfect completeness*. That is, when $x \in L$, Arthur always accepts, rather than accept with probability at least

2/3. Similar principles can be used to show that in general, we can assume that AM and Arthur-Merlin protocols have perfect completeness. The issue of perfect soundness is different. If we want Arthur to always reject when $x \notin L$, then every language that has an Arthur-Merlin protocol would be in NP (because showing an accepting transcript would be a witness for $x \in L$).

We have shown that $MA \subset AM$. With slightly more work, it can be shown that

$$AM_2 = AMAM = A(MA)M \subset A(AM)M = AAMM = AM$$

More generally, it can be shown that for every fixed k, $AM_k = AM$.
Homework. Prove that for every fixed k, $AM_k = AM$.

It is not known whether the result extends to k that may depend on n (e.g., to $k = n$).

Observe that essentially the same argument that shows that AM protocols can be assumed to have perfect completeness shows that $AM \subset \Pi_2$, and hence more generally, every finite level of the Arthur-Merlin hierarchy (which collapses) is in the second level of the polynomial hierarchy. Note that if $P = NP$, then also $PH = P$, and this implies that also $AM = P$ and in particular $BPP = P$. (The fact that BPP is in the polynomial hierarchy was first proved by Sipser) This can be contrasted with the fact that it is currently not known whether $BPP \subset NP$.

4 A private coin protocol for GNI

Interactive proofs (introduced by Goldwasser, Micali and Rackoff) are similar to Arthur Merlin games, but with one difference. The verifier (formerly, Arthur), need not just send sequences of random bits. Instead, he may toss coins in private (without the prover, formerly Merlin) seeing the outcome, and then send a message that depends on the outcome. Hence the prover has only partial information about the current state of the verifier. IP_k is the class of languages that have k-round interactive proofs. IP denotes the class of languages that have interactive proofs with a polynomial number of rounds (unlike the convention for AM).

For some languages, it is easier to design interactive proofs than Arthur-Merlin protocols. The most well known example is the problem of Graph Nonisormophism (GNI). The input is a pair of graphs that we denote by G_1 and G_2. The graphs are isomorphic if there is a bijection from the vertex set of G_1 to that of G_2 such that edges are mapped to edges and nonedges are mapped to nonedges. All graphs are partitioned into equivalence classes under the isomorphism relation. $(G_1, G_2) \in GNI$ iff G_1 is not isomorphic to G_2.

Clearly, graph isomorphism is in NP. This is not known to be the case regarding GNI. Goldreich, Micali and Wigderson show that GNI has an interactive proof.

1. The verifier chooses $i \in \{1, 2\}$ at random and a permutation π over the vertex set. Both choices are made in private.

2. The verifier sends the permutaed graph $G = \pi(G_i)$ to the prover.

3. The prover returns $j \in \{1, 2\}$.

4. The verifier accepts if $i = j$.

If the input graphs are not isomorphic, then G is isomorphic to exactly one of them, and the prover in his unbounded computing powers can tell which one, and need never fail. Hence the protocol has perfect completeness.

If the input graphs are isomorphic, then there is exactly the same number of permutations π that would create G from G_1 and G from G_2 (this is true for every three graphs that lie in the same isomorphism class, and the number is determined by the automorphism group). Hence the choices of $i = 1$ and $i = 2$ are indistinguishable to the prover, and he has probability exactly $1/2$ of succeeding (not more and not less, regardless of his strategy).

The use of private coins for the above protocol is essential. But is there a different protocol for GNI that uses only public coins. Perhaps surprisingly, the answer is yes. Goldwasser and Sipser show that $IP_k \subset AM_{k+2}$. As we can collapse levels of AM_k, it follows that for every finite k, $IP_k \subset AM$. We shall sketch the proof that $IP_1 = AM$, which suffices to show that $GNI \in AM$. We first describe a "lower bound protocol" that will be used in the proof that $IP_1 \subset AM_3$.

5 The AM lower bound protocol

Let U be the set of all vectors in $\{0,1\}^m$ except for the 0^m vector. (Removing the vector 0^m is a technicality that simplifies the presentation without hurting the generality of the result.) Let $S \subset U$ be a set that is given implicitly, via a decision procedure for Arthur that for every string s decides whether $s \in S$. For simplicity, one may think of this procedure as a polynomial time algorithm (polynomial in m), but more generally, it may be an AM protocol (a fact that we will use later). Let $k < m$, and let $c > 1$ be some constant. We wish to have an AM protocol with the following properties:

1. If $|S| \geq c2^k$, Arthur accepts with high probability.
2. If $|S| \leq 2^k/c$, Arthur rejects with high probability.

This will be called a *lower bound protocol.* We shall assume for simplicity that c is large, e.g., $c = 10$. However, we note that there are ways to handle every $c > 1$, e.g., by an amplification procedure based on taking products of sets.

The lower bound protocol works by using what is known as a *universal family of hash functions*, a notion with many appliactions that was first introduced by Carter and Wegman.

1. Arthur chooses at random a k by n Boolean matrix B, and a vector $z \in \{0,1\}^k$, and sends them to Merlin.
2. Merlin sends a nonzero vector $y \in \{0,1\}^m$.
3. Arthur accepts if $By = z$ and $y \in S$. Here products are taken over GF_2 (namely, the product of two vectors is the sum modulo 2 of their bitwise *and*). The test $y \in S$ may either be performed by Arthur alone, or possibly by an auxiliary AM protocol.

If $|S| \leq 2^k/c$, then regardless of the choice of B, at most a $1/c$ fraction of the $z \in \{0,1\}^k$ have a preimage in S. Hence arthur accepts with probability at most $1/c$ over the choice of z. (This assumes that Arthur can check reliably whether $y \in S$. If an auxiliary AM protocol is needed for this, the error of the auxiliary protocol should also be added to the total error of the protocol.)

If $|S| = c2^k$ (or larger), f ix an arbitrary z, and let us compute the probability that Arthur accepts over the choice of B. This is exactly the probability that z has a preimage in S. (Here, in principle, checking that $y \in S$ can be assumed to be errorless even if done by an AM protocol, because AM can be assumed to have perfect completeness.)

Let Y_i be an indicator 0/1 random variable (over the choice of random B) that for $y_i \in S$ specifies whether $By_i = z$. Let $Y = \sum_{y_i \in S} Y_i$ be a random variable counting the number of preimages of z.

Proposition 2 *The expectation of Y is $E[Y] = c$.*

Proof: First observe that $Pr[Y_i = 1] = 2^{-k}$. This follows from the fact that $y_i \in \{0,1\}^m$ is a nonzero vector. Hence in some coordinate, say l, it has value 1. Fix all columns of B expect for the lth column. Then every choice of the 2^k choices for column l of B leads to a different outcome for By_i. As this column is chosen uniformly at random, we indeed have that $Pr[Y_i = 1] = 2^{-k}$. As Y_i is a 0/1 indicator random variable, $E[Y_i] = 2^{-k}$. Using the linearity of the expectation we have that

$$E[Y] = E[\sum_{y_i \in S} Y_i] = \sum_{y_i \in S} E[Y_i] = 2^{-k}|S| = c$$

proving the proposition. $\square$

The following is the central property for universal families of hash functions (the family of matrices B is such a family).

Proposition 3 *The indicator random variables are pairwise independent.*

Proof: Consider arbitrarily $y_i, y_j \in U$, with $y_i \neq y_j$, and $z_i, z_k \in \{0,1\}^k$. We show that the probability (over the choice of B) that $By_i = z_i$ and $By_j = z_j$ is exactly 2^{-2k}, which proves pairwise independence. There is a coordinate l where y_i and y_j differ. Assume that y_i is 0 on coordinate l. Then there is some other coordinate l' on which y_i is 1. The two by two matrix defined by the entries of y_i and y_j in columns l and l' is invertible over GF_2, meaning that multiplying it by different vectors in $\{0,1\}^2$ gives different answers. If follows that regardless of the value of B in all columns other than l and l', cyclying over all 2^{2k} possibilities for the columns l and l' gives all possible 2^{2k} values for the pair of products By_i and By_j. $\square$

Proposition 4 *The variance $E[(Y - E[Y])^2]$ of Y is smaller than its expectation c.*

Proof: For the sum of pairwise independent variables, the variance is the sum of the individual variances. The variance of every indicator random variable Y_i is $E[Y_i](1-E[Y_i]) < E[Y_i]$. $\square$

Following the above propositions, the probability that z does not have a preimage in S (namely, $Pr[Y = 0]$) can be upper bounded by Chebyschev's inequality. Or in more detail, when $Y = 0$ then $(Y - E[Y])^2 = c^2$, and because of nonnegativity, this event has probability at most $var[Y]/c^2 < 1/c$, by Markov's inequality.

6 Public coins are as good as private coins

We prove here that $IP_1 \subset AM_3$. The basic idea is as follows. Consider a language $L \in IP_1$, and a protocol (P, V) for it (with m private coin tosses, messages of length m, and with error probability $1/2^{13}m$, for simplicity). We construct an AM protocol (A, M) for it. On input x, let S be the set of messages that V could have sent. S is partitioned into sets S_i, for $0 \le i \le m-1$. Message s is in set S_i if for the best reply m_s of P there are more than 2^i sequences of private coin tosses r of V that would have led to s being send and $V(x, r, s, m_s)$ accepting. Placing every message s only at the highest S_i in which it may belong, and rounding down the number of accepting sequences of coin tosses to the nearest power of 2, we still have that one of the S_i must contribute at least a $1/2m$ fraction of the private random strings on which V accepts. Clearly, no S_i can contribute more than the total number of private random strings on which V accepts.

1. Merlin sends i, and a number S' (that is claimed to be $|S_i|$).
2. Arthur checks that $2^i S' \ge 2^m/2m$ and rejects otherwise.
3. Arthur and Merlin perform an AM lower bound protocol to certify that $|S_i| \ge S'/8$. The outcome of the protocol is a string y for which Merlin claims that $y \in S_i$, and a corresponding (optimal) reply m_y for the original interactive proof (P, V).
4. Arthur and Merlin perform an AM lower bound protocol to certify that for y, m_y there are at least $2^i/8$ private random strings that would lead V to send y and accept m_y.

If $x \in L$ then V accepts with probability close to 1 in the IP_1 protocol. Hence there is a set S_i for which $|S_i|2^i \ge (1 - 1/2^{13}m)2^m/2m$ and Merlin can send i and a value $S' = |S_i|/(1 - 1/2^{13}m)$ as his first message. Thereafter, the two lower bound protocols allow sufficient slackness in the sizes of the underlying sets of which they are applied so that Merlin fails in each one of them with probability at most $1/8$. Hence Arthur accepts with probability at least $3/4$.

If $x \notin L$, then V accepts with probability at most $1/2^{13}m$ in the IP_1 protocol. Hence there are at most $S'/2^6$ strings s in S_{i-6}. In the first lower bound protocol, Merlin has probability at most $1/8$ of producing a string $y \in S_{i-6}$. If this fails, then in the second lower bound protocol Merlin has probability at most $1/8$ of convincing Arthur that $y \in S_i$. If neither one of these two possibilities happen, Arthur rejects. Hence Arthur accepts with probability at most $3/4$.

7 Is graph nonisomorphism NP-complete?

From the above it follows that $GNI \in AM$. Can graph isomorphism be NP-complete? If yes, then GNI is co-NP complete, and co-NP $\subset$ AM. This in turn would lead to the collapse of the polynomial hierarchy, as was shown by Boppana, hastad and Zachos.

Theorem 5 *If $\Pi_1 \subset AM$ then the polynomial hierarchy collapses to its second level.*

We only sketch the main steps of the proof, without proving them. $\Pi_1 \subset AM$ implies co-AM $\subset$ AM, which then implies two things:

- If $\Sigma_k \subset AM$ then also $\Pi_k \subset AM$.
- If $\Pi_k \subset AM$ then also $\Sigma_{k+1} \subset AM$.

Now the whole polynomial hierarchy can be placed in AM by induction, and the proof of the theorem follows from the fact that $AM \in \Pi_2$.

8 Unbounded number of rounds

As we have seen, if co-NP is contained in AM (or AM_k with constant k) then the polynomial hierarchy collapses. A surprising result (at the time) of Lund, Fortnow, Karloff and Nissan (LFKN) showed that co-NP has Arthur-Merlin protocols with an unbounded number of rounds. Let us briefly sketch some of the developments related to this result.

The *permanent* of a square matrix is defined similar to its determinant, but without the alternating sign rule. Unlike the determinant, which is polynomial time computable using Gaussian elimination, there is no known polynomial time algorithm that computes the permanent. Valiant showed that computing the permanent is $\#P$-hard, that is, as hard as counting the number of satisfying assignments of a 3CNF formula (and hence also NP-hard and coNP-hard). Toda showed that any problem in the polynomial hierarchy reduces in polynomial time to a problem in $\#P$. LFKN gave an Arthur-Merlin protocol for computing the permanent, and hence indirectly, for the whole polynomial hierarchy, including coNP. Shortly afterwards, a version of their protocol was shown that directly counts the number of satisfying assignments of 3CNF formulas. Finally, Shamir showed that in fact IP=PSPACE. (The direction $IP \subset PSPACE$ is relatively easy and was known.) In fact, his protocol for PSPACE was an Arthur-Merlin protocol with perfect completeness.

A sequence of works (most notably, by Babai, Fortnow, Lund; Feige, Goldwasser, Lovasz, Safra, Szegedy; Arora, Safra) then led to the PCP theorem (Arora, Lund, Motwani, Sudan, Szegedy), which was the subject of a course in the last semester (with lecture notes on my homepage).