18.405J/6.841J Advanced Complexity Theory	April 23, 1998
Lecture 19	
Lecturer: Daniel A. Spielman	*Scribe: Raj D. Iyer*

Public Coins = Private Coins

In this lecture we will prove that public coins are equivalent to private coins. The general theorem is

Theorem 1 IP$(k(n)) =$ AM$(k(n)+3)$.

which was proved by Goldwasser and Sipser. We will prove the special case

Theorem 2 IP$(1) =$ MAMAMA,

that is, one round of interaction between a prover P and a verifier V is equivalalent to five turns of an MA protocol. Our proof contains all the essential ideas needed to prove Theorem 1, and the analysis is several pages shorter.
For notational convenience, M = Merlin = Σ, and A = Arthur = BP.

What is the difference between public coins and private coins? Obviously, if we can prove a statement by flipping coins in public, we can prove that same statement if we hide our coin flips. Thus, MAMAMA $\subseteq$ IP(1), leaving us to prove the other containment, IP(1) $\subseteq$ MAMAMA, which is not so obviously true.

The IP(1) Protocol

We first describe the IP(1) protocol which we will eventually simulate by the MAMAMA protocol. A language L is in IP(1) if and only if there exist a randomized polynomial time function V (verifier) and an arbitrary function P (prover), such that for every function P' and string w

$$w \in L \quad \Leftrightarrow \quad \Pr\left[(\mathrm{V},\mathrm{P})(w) = \ accept\right] \geq 1 - 2^{-n} \tag{1}$$

$$w \notin L \quad \Leftrightarrow \quad \Pr\left[(\mathrm{V},\mathrm{P}')(w) = \ accept\right] \leq 2^{-n} \tag{2}$$

in one round of interaction, where the probability is taken over V's coin flips (here we have reduced the error probability via the familiar method of parallel repetition).

What do V and P do in this interaction? In general,

V: selects a string r uniformly at random from a set of strings R. Using w and r, V produces a question q and sends it to P.

P: generates an answer a_q without seeing r and sends a_q to V.

V: $\mathrm{V}(w, r, q, a_q)$ decides whether to accept or reject.

It is important that question q does not reveal r, because otherwise we have public coins.

We will see that whether or not V accepts w is determined completely by which random string r V selects from R. Thus we will rephrase statements (1) and (2) about the problity of acceptance in terms of the *number* of strings r which will cause V to accept w. If $w \in L$, we expect that most random strings will cause acceptance, and if $w \notin L$, we hope that few strings will cause acceptance.

In the first round, V acts as a function which takes a word w and a random string r and produces a question q. In other words, q is determined[1] by w and r. Given q, P wants to convince V to accept, so P wants to return an answer a_q in the hope that the r which caused V to generate q will also cause V to believe a_q (and thus accept). Thus P chooses a_q to maximize the probability over all strings in R that $\mathrm{V}(w, r, q, a_q) = accept$,

Definition 3

$$a_q = \ arg\ max_{a'} \left|\{r \in R \mid \ V(w,r) \to q \ and \ V(w,r,q,a') = \text{accept}\,\}\right|.$$

For a question q, let $S_q \subseteq R$ be the set of strings which cause V to accept, and let S be the union over all q,

Definition 4

$$S_q = \{r \in R \mid \mathrm{V}(w,r,q,a_q) = accept\}$$

and

$$S = \bigcup_q S_q.$$

Thus if $r \in S \subseteq R$ then r will cause V to accept w. Therefore,

$$\mathrm{Pr}_{r\in R}\left[(\mathrm{V},\mathrm{P})(w) = accept\right] = |S| \,/\, |R|\,.$$

By our assumptions (1) and (2) about the error probability of the protocol, we know that

$$w \in L \quad \Leftrightarrow \quad |S| \,/\, |R| \geq 1 - 2^{-n} \tag{3}$$
$$w \notin L \quad \Leftrightarrow \quad |S| \,/\, |R| \leq 2^{-n}. \tag{4}$$

The MAMAMA simulation of the IP protocol

From the previous argument, we see that in order to determine if $(\mathrm{V},\mathrm{P})(w) = accept$, we can compute a lower bound on the size of the set S. Using this idea we can give a MAMAMA protocol which simulates the IP protocol.

In the previous lecture we showed how Merlin can prove a lower bound on the size of a set provided that Arthur can verify that an element is a member of the set in polynomial time[2]. To prove that a given string r is in the set S, M needs to compute q and send A a poly-time

[1] Thus V had better not be injective, because then P will be able to discover r.

[2] In the last lecture, the set was the collection of (vertex-labelled) graphs isomorphic to a graph G, denoted $[G]$. The polynomial-time checkable proof that a graph H is in $[G]$ is the permutation mapping the vertices of H to the vertices of G. We used the set to show that GRAPH-NONISOMORPHISM can be solved using public coins.

checkable proof that the answer a_q indeed maximizes the number of random strings which lead to acceptance (see Definition 3). It is not clear that such a proof exists which can be checked in determinstic poly-time. However, M can furnish A with a *probabalistic* poly-time proof[3] of a string's membership in S. This will be a MAMA protocol.

M will attempt to show that $|S|$ is large using the following lemma:

Lemma 5 If $w \in L$ then there exist constants c and d such that there are at least 2^c questions q such that

$$|S_q| \geq 2^d$$

and

$$2^{c+d} \geq |R| / 4 \log |R| .$$

Proof: OK.

If $w \in L$ then by the first claim of the lemma, $|S| \geq 2^{c+d}$, since the S_q's are disjoint. Thus by the second claim,

$$\frac{|S|}{|R|} \geq \frac{1}{4 \log |R|} \geq 2^{-n}$$

for some n.

Thus we see a potential protocol:

M: produces c and d and sends them to A.

Sub protocol: M convinces A that for a given q, S_q is large.

M: uses the sub protocol to show that there are 2^c such q's.

A: accepts if he is convinced.

If $w \in L$, then A will accept with high probability. If $w \notin L$, A will reject with high probability.

The lower bound protocols we will present follow closely those given in the previous lecture.

Sub Protocol

We will give a MAMA protocol in which M tries to convince A that, for a given r which determines q, the set S_q is very large. More precisely, given r which generates q and any $\epsilon > 0$,

- if $|S_q| \geq 2^d$ then $\Pr[\text{A accepts}] \geq 1 - \epsilon$
- if $|S_q| \leq 2^d/2$ then $\Pr[\text{A accepts}] \leq \epsilon$,

[3] These should not be confused with probabalistically checkable proofs (PCPs) which are only vaguely related to the theorem at hand.

where the probability is taken over A's coin flips.

This protocol is a probabilistic proof of membership in $|S_q|$, i.e. a proof that the answer provided by P really is a_q. This protocol is almost exactly the same as the lower bound protocol used in the last lecture. Given r, M generates q and sends a_q. Given a_q and r, A can decide whether or not $\mathrm{V}(w, r, q, a_q) = \textit{accept}$. Now M must convince A that S_q is large by showing a lower bound on $|S_q|$, i.e. the number of r's which cause V to accept.

The method is the same as before: we will use a poly-time hash function h from a universal family $\mathcal{H}$ of hash functions. M sends to A a hash function $h : R \to T$ where $|T| = 2^d$. By the proof in the last lecture, there exists an h such that if $|S_q| \geq 2^d$ then $|h(S_q)| \geq 2^d/2$. We amplify this by considering a function $h : R^k \to T^k$ so that $|T| = 2^{dk}$ and if $|S_q| \geq 2^d$ then $\left|h\left(S_q^k\right)\right| \geq 2^{dk}/2$. And if $|S_q| \leq 2^d/2$ then for all $h \in \mathcal{H}, \left|h\left(S_q^k\right)\right| \geq 2^{dk}/2^k$.

The MAMA protocol follows that of last lecture:

M: sends a hash function $h : R \to T$ to A.

A: selects $t_1, \ldots, t_j$ uniformly at random from T and sends these to M.

M: provides preimages $r_1, \ldots, r_j$.

A: checks that more than $1/4$ of the t_i's, $h(r_i) = t_i$ and also that each $r_i \in S_q$. If both of these test pass, then A accepts.

Note that A can compute $h(r_i)$ in poly-time by our assumption about h. A can check that $r \in S_q$ by seeing if $\mathrm{V}(w, r, q, a_q) = \textit{accept}$.

In conclusion, for all *epsilon* > 0, there exists a j such that

- if $|S_q| \geq 2^d$ then $\exists M$ s.t. $\Pr[\text{A accepts}] \geq 1 - \epsilon$
- if $|S_q| \leq 2^d/2$ then $\forall M, \Pr[\text{A accepts}] \leq \epsilon$.

Main Protocol

Now we come to the main protocol. Remember that the whole point of this is to show that a word w is in a language L. In the beginning, there's just the word[4]. The main protocol is

M: sends c and d and claims that for more than $2^c q$'s, $|S_q| \geq 2^d$.

M: sends $h' : U^k \to T'$ where U is the space of questions q and $|T'| = 2^{ck}$.

A: chooses $t'_1, \ldots, t_{j'}$ uniformly at random from T'.

M: sends preimages $q_1, \ldots, q_{j'}$.

MAMA: begins sub-protocol to prove (in parallel) that each S_{q_i} is large.

A: accepts if more than $1/4$ of the q_i's are preimages of t'_i's and S_{q_i} is large.

[4] "In the beginning was the Word, and the Word was with God, and the Word was God."–*John* 1:1.

Thus we have an MAMAMA protocol.

We choose j and j' large enough so that for all $\alpha > 0$,

- if $|S|/|R| \geq 1 - 2^{-n}$ then $\exists M$ s.t. $\Pr[\text{A accepts}] \geq 1 - \alpha$.
- if $|S|/|R| \leq 1/4 \log |R|$ then $\forall M, \Pr[\text{A accepts}] \leq \alpha$.

e.g. we choose j and j' to be polynomial in n.

1 References

The general proof appears in

- Shafi Goldwasser and Michael Sipser. "Private coins versus public coins in interactive proof systems." *Randomness and Computation.* Advances in Computing Research, vol. 5, JAI Press, 1989, p. 73–90.

To review the definition of the class IP and some properties of it (like the scribe did), see

- Michael Sipser, *Introduction to the Theory of Computation*, PWS Publishing Company, Boston, 1997, pp. 354-366.

In preparing these notes, the scribe consulted and incorporated parts of the notes from lectures 17 and 18 of 1997, scribed by Eric Lehman and John Dunagan, respectively.