18.405J/6.841J Advanced Complexity Theory	April 22, 1999
Lecture 19	
Lecturer: Daniel A. Spielman	*Scribe: Chris Peikert*

Loose Ends

First, a quick proof that $\mathrm{MA}(k) \subseteq \mathrm{AM}$: in the homework, we proved that $\mathrm{MA} \subseteq \mathrm{AM}$, except it was in the form of $\Sigma \cdot \mathrm{BP} \cdot \mathrm{P} \subseteq \mathrm{BP} \cdot \Sigma \cdot \mathrm{P}$. We can use essentially the same proof to show that $\mathrm{MAM} \subseteq \mathrm{AM}$. Now consider a language like MAMAM. Since $\mathrm{MAM} \subseteq \mathrm{AM}$, $\mathrm{MAMAM} \subseteq \mathrm{MAAM} = \mathrm{MAM} \subseteq \mathrm{AM}$. This process can be done for arbitrarily long alternating strings, which proves the claim.

See last year's notes for a proof that $\mathrm{MA}(k(n)) \subseteq \mathrm{MA}(k(n)/2 + 1)$.

New Topic: PSPACE $\subseteq$ IP(poly)

As a warmup, we prove that $\#\mathrm{P} \subseteq \mathrm{IP}$, and in particular, $\#\mathrm{SAT} \in \mathrm{IP}$, where we remain formal by treating IP as a function class (instead of a language).

Arithmetization

If we have a boolean formula $\phi(x_1, \ldots, x_n)$, then normally each x_i is either 0 or 1. We want to extend the meaning of the formula to allow x_i to take on arbitrary values from some field $\mathbf{F}$. In the original formula, we have three operations: $\wedge, \vee, \neg$. We make the following replacements in the formula:

$$\begin{aligned} a \wedge b &\rightarrow a \cdot b \\ a \vee b &\rightarrow 1 - (1-a)(1-b) \\ \neg a &\rightarrow 1 - a \end{aligned}$$

Note that these replacements preserve De Morgan's laws. In addition, if $a, b \in \{0,1\}$, then arithmetization does not change the value of the function. Finally, notice that if we arithmetize a formula $\phi(x_1, \ldots, x_n)$ to a polynomial $p(x_1, \ldots, x_n)$, then the degree of p is at most the number of leaves in ϕ, when interpreted as a tree.

Given a 3CNF ϕ, define $\#\phi$ to be the number of satisfying assignments of ϕ, which equals

$$\sum_{x_1 \in \{0,1\}} \cdots \sum_{x_n \in \{0,1\}} p(x_1, \ldots, x_n).$$

Now given $a_1, \ldots, a_n \in \mathbf{F}$, a verifier can compute $p(a_1, \ldots, a_n)$. In our interactive proof, the prover will assert what $\#\phi$ is, and the verifier will end up checking a single value $p(a_1, \ldots, a_n)$, using intermediate subassertions.

To make this notion of "intermediate subassertions" formal, we define an ϵ-step to be a sub-protocol/proof such that, beginning with an assertion A,

- either V rejects, or
- we end with an assertion B having the following properties:

$$\begin{aligned} A \text{ is true} &\Rightarrow \exists \text{ a prover s.t. } B \text{ is true} \\ A \text{ is false} &\Rightarrow \Pr B \text{ is false, given V didn't reject} > 1 - \epsilon. \end{aligned}$$

In other words, if V "believes" B with probability $\alpha \approx 1$, then V believes A with probability $\alpha - \epsilon$.

Now assume that we begin with the following assertion:

$$\sum_{x_j \in \{0,1\}} \cdots \sum_{x_n \in \{0,1\}} p(a_1, \ldots, a_{j-1}, x_j, \ldots, x_n) = \alpha \tag{1}$$

The prover sends a polynomial $q(x_j)$ in x_j only, with $\deg(q) \leq \deg(p)$, which is supposedly

$$\sum_{x_{j+1} \in \{0,1\}} \cdots \sum_{x_n \in \{0,1\}} p(a_1, \ldots, a_{j-1}, x_j, x_{j+1}, \ldots, x_n). \tag{2}$$

Then V rejects automatically if $q(0) + q(1) \neq \alpha$, otherwise V chooses $a_j \in \mathbf{F}$ at random, lets $\beta = q(a_j)$, and sends a_j to the prover. The new assertion is then

$$\sum_{x_{j+1} \in \{0,1\}} \cdots \sum_{x_n \in \{0,1\}} p(a_1, \ldots, a_j, x_{j+1}, \ldots, x_n) = \beta. \tag{3}$$

We claim that this is a $\frac{\deg(p)}{|\mathbf{F}|}$-step. To prove this, note that three assertions are involved (they are marked above). If (1) is true, then there exists some $q(x_j)$ with proper degree and some prover that sends it, so (2) is true. Of course, if (1) and (2) are true, then so is (3).

Alternatively, if (1) is false and V does not reject, then (2) must be false. In this case, because q and p have degree at most $\deg(p)$, there are at most $\deg(p)$ choices for a_j such that (3) is true (by Schwartz's lemma), so the probability of choosing one of these values is at most $\frac{\deg(p)}{|\mathbf{F}|}$.

Now we present the full interactive proof for #3SAT: the prover begins with the assertion

$$\sum_{x_1 \in \{0,1\}} \cdots \sum_{x_n \in \{0,1\}} p(x_1, \ldots, x_n) = \beta_0$$

which, via a $\frac{\deg(p)}{|\mathbf{F}|}$-step, is reduced to the assertion

$$\sum_{x_2 \in \{0,1\}} \cdots \sum_{x_n \in \{0,1\}} p(a_1, x_2, \ldots, x_n) = \beta_1$$

which is further reduced, over n similar steps, to the assertion

$$p(a_1, \ldots, a_n) = \beta_n.$$

The verifier takes polynomial time to check this assertion, and accepts if it is true. The entire protocol is an $\frac{n \cdot \deg(p)}{|\mathbf{F}|}$-step, by a probability bound of going from a false assertion to a true one.

For the entire protocol, we can use $|\mathbf{F}| \approx 2^n$, and $\deg(p)$ polynomial in n.

For the proof that PSPACE $\subseteq$ IP(poly), note that V need not truly evaluate p; it just needs to be convinced of the value. Recall the outline of the proof that PSPACE $\subseteq$ AP: let M run in polynomial space, and see that the graph of configurations of M on w has at most 2^{n^k} nodes, for some k, and every node has outdegree 1 since M is deterministic. Then M accepts $w \iff \text{FromTo}(\text{start}, \text{accept}, 2^{n^k}) = 1$, where the definition of FromTo is

$$\text{FromTo}(x, y, 2^j) = \bigvee_z (\text{FromTo}(x, z, 2^{j-1}) \wedge \text{FromTo}(z, y, 2^{j-1})) \tag{4}$$

and where $\text{FromTo}(x, y, 1)$ has a polynomial-sized formula.

Now we arithmetize: let $P_1(x_1, \ldots, x_{n^k}, y_1, \ldots, y_{n^k})$ be an arithmetization of $\text{FromTo}(x, y, 1)$. Furthermore, recursively define

$$P_k(\overline{x}, \overline{y}) = \sum_{z_1 \in \{0,1\}} \cdots \sum_{z_{n^k} \in \{0,1\}} P_{k-1}(\overline{x}, \overline{z}) \cdot P_{k-1}(\overline{z}, \overline{y}).$$

We claim that for $x, y \in \{0,1\}^{n^k}$, $P_k(\overline{x}, \overline{y}) = \text{FromTo}(\overline{x}, \overline{y}, 2^k)$. We've arithmetized the $\wedge$ of FromTos in (4) to multiplication, but we've only replaced the $\vee_z$ by summation. However, this is OK because of determinism - in the $\vee_z$, there is a *unique* node z for which both FromTos are true, if we define FromTo to require *exact* lengths of paths, and make a loop from the accept node to itself. Therefore the sum is either zero or one, depending on whether there is a path of proper length to the accept node.

Now we outline the interactive proof of any PSPACE problem. The first assertion is $P_{n^k}(\text{start}, \text{accept}) = 1$. The general assertion is of the form $P_j(x_1, \ldots, x_n, y_1, \ldots, y_n) = c$ with $x_i, y_i \in \mathbf{F}$.

First, $P_{n^k}(\text{start}, \text{accept}) = 1$ is written out a sum over z_i's. We interactively apply a $\frac{n \cdot \deg(p)}{|\mathbf{F}|}$-step (as above) to get an assertion like

$$P_{j-1}(x_1, \ldots, x_{n^k}, a_1, \ldots, a_{n^k}) \cdot P_{j-1}(a_1, \ldots, a_{n^k}, y_1, \ldots, y_{n^k}) = c'$$

which can be converted into another assertion $P_{j-1}(\overline{u}, \overline{v}) = c''$ by another ϵ-step (as we will show). We then repeat the process until we get an assertion about P_1, which the verifier can then check directly.

To complete the proof, we need only show that an assertion about a product $P_k \cdot P_k$ can be converted to an assertion about a single value of P_k. Define a function that takes an element from the field $\mathbf{F}$ and returns a vector,

$$T(\alpha) = \alpha(x_1, \ldots, x_{n^k}, a_1, \ldots, a_{n^k}) + (1 - \alpha)(a_1, \ldots, a_{n^k}, y_1, \ldots, y_{n^k}).$$

This can be viewed as "linear interpolation" (but in the field $\mathbf{F}$) between the two vectors in the original product of P_k's. Then the prover sends a function $q(\alpha)$, which is supposedly the function $P_k(T(\alpha))$. The verifier immediately rejects if $q(0) \cdot q(1) \neq c'$, otherwise it chooses $b \in \mathbf{F}$ at random, and computes $c'' = q(b)$. The new assertion is now $P_k(T(b)) = c''$. This is an ϵ-step, and the analysis is similar to what appears above. This completes the proof.