Lecture 18

Lecturer: Daniel A. Spielman *Scribe: Sergi Elizalde*

1 $\sharp$P $\subseteq$ IP

The main idea we will use to prove that $\sharp\mathbf{P} \subseteq \mathbf{IP}$ will be arithmetization. Using that $\sharp$**3SAT** is complete for $\sharp\mathbf{P}$, the proof consists in reducing $\sharp$**3SAT** to a language that is in **IP** .

Define an **arithmetic formula** as an expression containing variables $(x_1, x_2, \ldots)$, constants, and symbols $*, +, -, (,)$, in the intuitive way. We can convert any Boolean formula into an arithmetic formula using the following rules:

$$\begin{aligned} \phi_1 \wedge \phi_2 &\longrightarrow \phi_1 * \phi_2 \\ \neg\phi_1 &\longrightarrow 1 - \phi_1 \\ \phi_1 \vee \phi_2 &\longrightarrow 1 - (1-\phi_1) * (1-\phi_2) \end{aligned}$$

A Boolean formula and its arithmetization take the same values on binary inputs. The advantage of the arithmetic formula is that it can be evaluated on any input.

Definition 1 *Define the language* **Arithmetic Formula Sum** *to be*

$$AFS = \{(f(x_1, \ldots, x_n), s) : \sum_{x_1 \in \{0,1\}} \sum_{x_2 \in \{0,1\}} \cdots \sum_{x_n \in \{0,1\}} f(x_1, \ldots, x_n) = s\}$$

Note that if $f(x_1, \ldots, x_n)$ has been obtained by arithmetization of a Boolean formula ϕ, then $(f(x_1, \ldots, x_n), s) \in AFS$ iff ϕ has s satisfying assignments. Now, to show that $\sharp\mathbf{P} \subseteq \mathbf{IP}$ it will suffice to prove the following theorem.

Theorem 2 $AFS \in \mathbf{IP}$

The idea of the proof is to apply a sequence of reductions to AFS. However, our notion of reduction has to be generalized first so that it can be adapted to interactive proofs.

Definition 3 *Let A_1,A_2 be two languages, and let $\varepsilon > 0$. An ε-***reduction** *from A_1 to A_2 is an* **IP** *subprotocol in which* **V** *outputs* reject *or w_2 in such a way that*
$w_1 \in A_1 \Longrightarrow \exists$ *a prover* **P** *s.t.* $\Pr[(\mathbf{V} \longleftrightarrow \mathbf{P}) = w_2 \in A_2] = 1$
$w_1 \notin A_1 \Longrightarrow \forall$ *prover* **P** , $\Pr[(\mathbf{V} \longleftrightarrow \mathbf{P}) = w_2 \in A_2] \leq \varepsilon$

Intuitively, this means that, up to a probability ε of error, if we can convince that $w_2 \in A_2$, then we can convince that $w_1 \in A_1$. Note that a 0-reduction is a standard reduction. The following claim will help us understand this definition better.

Claim 4 *If $A_2 \in$* **P** *and there exists a $\frac{1}{3}$-reduction from A_1 to A_2, then $A_1 \in$* **IP**.

Proof of Claim: The **IP** protocol for A_1 is as follows. On input w_1, run the subprotocol corresponding to the reduction. If **V** rejects, reject. Otherwise, assuming that w_2 is the output, accept if $w_2 \in A_2$ and reject otherwise.

Clearly this proves that $A_1 \in \mathbf{IP}$, because if $w_1 \in A_1$ then there exists a prover for which **V** accepts with probability 1, and if $w_1 \notin A_1$ then, for all provers, **V** rejects with probability at least 2/3. ■

Proof Let $L_n \subseteq AFS$ be the subset of those elements in which the formula has n variables. To prove the theorem, we will construct a chain of ε-reductions, for an appropriate $\varepsilon \leq \frac{1}{3m}$, each one from L_n to L_{n-1}, so that all together give a $\frac{1}{3}$-reduction from L_m to L_0. Then the result will be implied from the claim and the fact that $L_0 \in \mathbf{P}$. Now we give the reduction from L_n to L_{n-1}.

The reduction

Assume that the input is $(f(x_1, \ldots, x_n), s)$.
Let d be the length of f. Note that since we gave the definition of arithmetic formulas without using exponentiation, d is an upper bound on the degree of f. Take $\varepsilon = \frac{1}{3m}$. Let

$$f_1(x_1) = \sum_{x_2 \in \{0,1\}} \cdots \sum_{x_n \in \{0,1\}} f(x_1, \ldots, x_n)$$

Consider the following protocol.

P : Send $g_1(x_1)$, a polynomial of degree at most d.
V : Check if $g_1(0) + g_1(1) = s$. If not, *reject.*
Choose a constant $c_1 \in \{1, \ldots, \lceil \frac{d}{\varepsilon} \rceil\}$ at random.
Output "$(f(c_1, x_2, \ldots, x_n), g_1(c_1)) \in L_{n-1}$?"

We have to show that this is indeed an ε-reduction. If $(f(x_1, \ldots, x_n), s) \in L_n$, consider the prover that sends $g_1(x_1) = f_1(x_1)$. Then, **V** will output $(f(c_1, x_2, \ldots, x_n), f_1(c_1))$, which is clearly in L_{n-1} by the definition of f_1.

Suppose now that $(f(x_1, \ldots, x_n), s) \notin L_n$. This is equivalent to $f_1(0) + f_1(1) \neq s$. If the polynomial $g_1(x_1)$ that **P** sends verifies $g_1(0) + g_1(1) \neq s$, then **V** rejects, so the prover can't win this way. If, on the contrary, $g_1(0) + g_1(1) = s$, then $g_1 \not\equiv f_1$, because $f_1(0) + f_1(1) \neq s$. So, since the non-zero polynomial $f_1 - g_1$ has degree at most d, there are at most d choices for c_1 such that $g_1(c_1) = f_1(c_1)$. Hence,

$$\Pr_{c_1}[g_1(c_1) = f_1(c_1)] \leq \frac{d}{\lceil \frac{d}{\varepsilon} \rceil} \leq \varepsilon$$

When $g_1(c_1) \neq f_1(c_1)$, the output is not in L_{n-1}. So,

$$\Pr_{c_1}[(f(c_1, x_2, \ldots, x_n), g_1(c_1)) \in L_{n-1}] \leq \varepsilon$$

This proves that what we have given is an ε-reduction from L_n to L_{n-1}.

The proof of the main result $\sharp\mathbf{P} \subseteq \mathbf{IP}$ follows easily now. Assume that **V** has to decide if a given input instance of the form $(f(x_1, \ldots, x_m), s)$ (for some m) is in AFS or not. To do that, first it runs $\frac{1}{3m}$-reductions from L_n to L_{n-1}, for $n = m, m-1, \ldots, 1$, as the one described above. The output of these reductions is of the form "$w_{n-1} \in L_{n-1}$?". If **V** has not rejected in any of them, then **V** checks if $w_0 \in L_0$, which can be done in polynomial time. If that is the case, **V** accepts, otherwise rejects.

This gives an **IP** protocol for AFS. Indeed, if $w_n \in L_n$, then $\exists\mathbf{P}$ such that $\Pr[(\mathbf{V} \longleftrightarrow \mathbf{P})\ \textit{accept}] = 1$. And if $w_n \notin L_n$, then $\forall\mathbf{P}$ $\Pr[(\mathbf{V} \longleftrightarrow \mathbf{P})\ \textit{accept}] \leq \frac{1}{3}$. Here we use that in the chain of reductions the errors just add up, because $\Pr[w_m \notin L_m \wedge w_0 \in L_0] = \sum_n \Pr[w_n \notin L_n \wedge w_{n-1} \in L_{n-1}] \leq m\ \frac{1}{3m} = \frac{1}{3}$. ■

Observe that the chain of reductions is in fact an $\frac{1}{3}$-reduction from L_n to L_0.

2 PSPACE $\subseteq$ IP

The proof of this result will use again the same ideas, namely arithmetization and ε-reductions. When trying to find an interactive protocol for **PSPACE** , we will have to deal with polynomials in implicit form, like, as we had before,

$$f_1(x_1) = \sum_{x_2 \in \{0,1\}} \cdots \sum_{x_n \in \{0,1\}} f(x_1, \ldots, x_n)$$

This is certainly a description of a polynomial, but it cannot be evaluated efficiently. This problem will be overcome using arithmetization and reductions as the previous ones.

Another difficulty that we will encounter is that, in some stage of the protocol, the prover will try to convince the verifier about the value that a polynomial takes in two certain points. The following lemma will let us reduce this case to that of one single point, which we are more familiar with.

Lemma 5 (two-for-one) *Let $f(y_1, \ldots, y_k)$ be an implicitly represented polynomial. There exists an ε-reduction that reduces the verification of $f(a_1, \ldots, a_k) = \alpha$ and $f(b_1, \ldots, b_k) = \beta$ to a single verification of the form $f(c_1, \ldots, c_k) = \gamma$.*

Proof of Lemma Let d be an upper bound on $deg(f)$. We will evaluate the polynomial in a line. Let $l(t) = (1-t)(a_1, \ldots, a_k) + t(b_1, \ldots, b_k)$, and let $f_1(t) = f(l(t))$. Note that $deg(f_1) \leq d$. Consider the following protocol.

P : Send $g_1(t)$, a polynomial of degree at most d.
V : Check if $g_1(0) = \alpha$ and $g_1(1) = \beta$. If not, *reject.*
Choose a constant $c \in \{1, \ldots, \lceil \frac{d}{\varepsilon} \rceil\}$ at random.
Ask for a proof that $f(l(c)) = g_1(c)$

This is indeed an ε-reduction, by the same reasoning used above. ■

Now we can proof that **PSPACE** $\subseteq$ **IP**.

Proof Let M be any given **PSPACE** Turing Machine. On input w, consider the graph of configurations of M on w. Recall that when we proved that **PSPACE** = **APTIME** the second day of class, we defined the function

$$FromTo(\mathbf{x}, \mathbf{y}, k) = \begin{cases} 1 & \text{if } \exists \text{ a path from } \mathbf{x} \text{ to } \mathbf{y} \text{ of length } k \\ 0 & \text{otherwise} \end{cases}$$

for any two configurations $\mathbf{x}$ and $\mathbf{y}$. Then, for a big enough $K \sim 2^{poly(|w|)}$, $FromTo(start, accept, K)$ tells us whether M accepts w or not. This function has the property that

$$FromTo(\mathbf{x}, \mathbf{y}, k) = \exists \mathbf{z} : FromTo(\mathbf{x}, \mathbf{z}, \lceil k/2 \rceil) \wedge FromTo(\mathbf{z}, \mathbf{y}, \lfloor k/2 \rfloor)$$

Now we will define several polynomials that will be the arithmetization of these functions. For the first one, $FT_1(\mathbf{x}, \mathbf{y})$, we can make a small arithmetic formula satisfying

$$FT_1(\mathbf{x}, \mathbf{y}) = \begin{cases} 1 & \text{if } M \text{ can go from } \mathbf{x} \text{ to } \mathbf{y} \text{ in one step} \\ 0 & \text{otherwise} \end{cases}$$

The other polynomials are defined recursively, and are given implicitly.

$$FT_k(\mathbf{x}, \mathbf{y}) = \sum_{z_1 \in \{0,1\}} \cdots \sum_{z_k \in \{0,1\}} FT_{k/2}(\mathbf{x}, \mathbf{z}) \ FT_{k/2}(\mathbf{z}, \mathbf{y}) \tag{1}$$

Note that the degree of FT_k in each variable is at most the degree in the same variable of $FT_{k/2}$. For binary $\mathbf{x}$ and $\mathbf{y}$, $FT_k(\mathbf{x}, \mathbf{y})$ evaluates as $FromTo(\mathbf{x}, \mathbf{y}, k)$, with the advantage that it can be evaluated in general points. In principle, the coefficients of the polynomials can grow as k increases, but this can be solved working over a finite field.

Next we will describe an ε-reduction from a verification of the value of FT_k in a certain point to a verification of the value of $FT_{k/2}$ in some other point. Suppose that **P** wants to convince **V** of the fact that $FT_k(\mathbf{x}, \mathbf{y}) = s$ for some fixed $\mathbf{x}$ and $\mathbf{y}$. Using the implicit expression (1) for FT_k, we can apply a chain of ε'-reductions as we did in the proof of $AFS \in \mathbf{IP}$ to reduce it, up to a small enough error, to an assertion of the form $FT_{k/2}(\mathbf{x}, \mathbf{c})\ FT_{k/2}(\mathbf{c}, \mathbf{y}) = s'$.

Now, **P** sends α and β such that $\alpha\beta = s'$ (otherwise **V** would reject immediately), trying to convince **V** that $FT_{k/2}(\mathbf{x}, \mathbf{c}) = \alpha$ and $FT_{k/2}(\mathbf{c}, \mathbf{y}) = \beta$. By the two-for-one lemma, this can be reduced to an assertion of the form $FT_{k/2}(l(t)) = \gamma$, which only depends on one evaluation.

Applying these reduction repeatedly, we eventually get down to some verification of the form $FT_1(\mathbf{a}, \mathbf{b}) = r$. So, after a polynomial number of repetitions, checking that $FT_K(\mathit{start}, \mathit{accept}) = 1$ has been reduced to an evaluation of FT_1 in some particular point, which can be done in polynomial time. This proves that we can decide whether M accepts w or not using an interactive protocol.

Although we did not specify how to choose the ε and ε', we can make them exponentially small if we work in a big enough field, so that the total error of the general reduction is small. ■