Lecture 11

IP, PH, and PSPACE

May 4, 2004
Lecturer: Paul Beame
Notes: Daniel Lowd

11.1 IP and PH

Theorem 11.1 (Lund-Fortnow-Karloff-Nisan). *There is a polynomial length interactive proof for the predicate PERM$(A) = k$.*

Before we prove this theorem we note a number of corollaries.

Corollary 11.2. *There exist polynomial length interactive proofs for all of $\#\mathsf{P}$.*

Corollary 11.3. $\mathsf{PH} \subseteq \mathsf{IP}$

This is surprising, because a constant number of alternations is equivalent to two alternations, but an unbounded number yields PH. Later, we will prove that there exist interactive proofs for everything in PSPACE.

Proof of Corollaries. These follow from the easy observation that IP is closed under polynomial-time (Turing) reduction and by Toda's theorem. □

Remark. In the journal version of their paper, Lund, Fortnow, Karloff, and Nisan gave an alternative direct protocol proving Corollary 11.2. This proof is given in full in Sipser's text. We present the protocol for the permanent directly both because it is interesting in its own right and because the permanent problem motivates the whole approach for these proofs.

Proof of Theorem 11.1. We perform the computation of PERM over some finite field $\mathbb{F}$, where $|\mathbb{F}| > 10n^3$. We also assume that $\{1, \ldots, n\} \subset \mathbb{F}$, although we can remove this restriction by simply choosing n distinct field elements of $\mathbb{F}$ that can substitute for $\{1, \ldots, n\}$.

Recall the definition of PERM:

$$\text{PERM}(A) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} a_{i,\sigma(i)}$$

where $a_{i,j}$ is the element in the ith row and jth column of matrix A and S_n is the set of all permutations from $\{1, \ldots, n\}$ to itself.

Note that PERM is a multivariate polynomial in the inputs, with total degree n and degree at most 1 in each variable $a_{i,j}$. Furthermore, we can define PERM recursively via the following self-reduction:

Definition 11.1. Let $A(i|j)$ be the $(n-1) \times (n-1)$ matrix equal to A with its ith row and jth column removed.

Claim 2. $\text{PERM}(A) = \sum_{\ell=1}^{n} a_{1,\ell}\text{PERM}(A(1|\ell))$.

The proof for this reduction is by direct application of the definition:

$$\begin{aligned}\sum_{\ell=1}^{n} \sum_{\substack{\sigma \in S_n, \\ \sigma(1)=\ell}} \prod_{i=1}^{n} a_{i,\sigma(i)} &= \sum_{\ell=1}^{n} a_{1,\ell} \sum_{\sigma \in S_n} \prod_{i=2}^{n} a_{i,\sigma(i)} \\ &= \sum_{\ell=1}^{n} a_{1,\ell} \sum_{\sigma \in S_{n-1}} \prod_{i=1}^{n-1} a_{i',\sigma'(i')}\end{aligned}$$

where $i' = i+1$ and $\sigma'(i') = \begin{cases} \sigma(i') & \text{for } \sigma(i') < \ell \\ \sigma(i')+1 & \text{for } \sigma(i') \geq \ell \end{cases}$

$$= \sum_{\ell=1}^{n} a_{1,\ell} \cdot \text{PERM}(A(1|\ell)).$$

To prove that $\text{PERM}(A) = k$, it suffices to prove the values of $\text{PERM}(A(1|\ell))$ for $\ell = 1, \ldots n$. Of course, a fully recursive algorithm would yield $n!$ subproblems, saving us nothing over direct computation. Instead, the prover will give *one* proof that gives allows us to recursively prove values for all $\text{PERM}(A(1|\ell))$ via a single proof rather than n separate proofs.

To do this we will use a representation of these values of the permanent as polynomials. The following are the two basic properties of polynomials that we will use.

Proposition 11.4. *If $p \neq 0$ is a degree d univariate polynomial over $\mathbb{F}$ then p has at most d roots.*

Corollary 11.5. *For any degree d univariate polynomial $f \neq 0$ over $\mathbb{F}$, $Pr_{r \in_R \mathbb{F}}[f(r) = 0] \leq d/|\mathbb{F}|$.*

Proposition 11.6. *(Interpolation Lemma) Given any distinct set of points $\{b_1, \ldots b_n\} \subset \mathbb{F}$ and any (not necessarily distinct) $\{c_1, \ldots, c_n\} \subset \mathbb{F}$ there is a degree n univariate polynomial $p \in \mathbb{F}[x]$ such that $p(b_i) = c_i$ for $i = 1, \ldots, n$.*

Basic Idea Write $B(\ell) = A(1|\ell)$. Then

$$B(\ell) = \begin{bmatrix} b_{1,1}(\ell) & \cdots & b_{1,n-1}(\ell) \\ \vdots & \ddots & \vdots \\ b_{n-1,1}(\ell) & \cdots & b_{n-1,n-1}(\ell) \end{bmatrix}$$

Each $b_{i,j}(\ell)$ is a function: $\{1, \ldots, n\} \to \mathbb{F}$. By the interpolation lemma, there are degree n polynomials $p_{i,j}(z)$ such that $p_{i,j}(\ell) = b_{i,j}(\ell)$ for $\ell = 1, \ldots n$. Write $B(z)$ for this matrix of polynomials. Then $B(\ell) = A(1|\ell)$ for $\ell = 1, \ldots, n$ but is also defined for other values of z.

Given this matrix of polynomials $B(z)$,

$$\text{PERM}(A) = \sum_{\ell=1}^{n} a_{1,\ell}\text{PERM}(B(\ell))$$

Observe that:

1. $\text{PERM}(B(z))$ is a degree $(n-1)n$ univariate polynomial over $\mathbb{F}$.
2. Given A, both players can compute what $B(z)$ is.

Interactive protocol to prove that $\text{PERM}(A) = k$

- Prover computes $f(z) = \text{PERM}(B(z))$ and sends the $n(n-1)+1$ coefficients of f to the verifier.
- Verifier checks that $\sum_{\ell=1}^{n} a_{1,\ell} f(\ell) = k$. If good, chooses $r_1 \in_R \mathbb{F}$ and sends it to the prover.
- Prover continues with a proof that $\text{PERM}(B(r_1)) = f(r_1)$.

The proof continues until matrix size is one. In that case the Verifier computes the permanent directly by checking that the single entry of the matrix is equal to the claimed value for the permanent.

Note. Note that is very important in this protocol r_i could take on a value larger than n. In other words, $B(r_i)$ might not be a submatrix of A at all.

Clearly, if $\text{PERM}(A) = k$ then the prover can always convince the verifier.

Suppose that $\text{PERM}(A) \neq k$. At each round there is an $i \times i$ matrix A_i and an associated claimed value k_i for the permanent of A_i where $A_n = A$ and $k_n = k$. The Prover can cause the Verifier to accept only if $\text{PERM}(A_1) = k_1$. Therefore in this case there is some round i such that $\text{PERM}(A_i) \neq k_i$ but $\text{PERM}(A_{i-1}) = k_{i-1}$. Let $B(z)$ be the $(i-1) \times (i-1)$ polynomial matrix associated with A_i and $f(z)$ be the degree $i(i-1)$ polynomial sent by the Prover in this round. Either $f(z) = \text{PERM}(B(z))$ as polynomials or not.
If $f(z) = \text{PERM}(B(z))$, then $\sum_{\ell=1}^{n} a_{1,\ell} f(\ell) = \sum_{\ell=1}^{n} a_{1,\ell} \text{PERM}(B(\ell)) \neq k$, and the verifier will reject the proof immediately.
If $f(z) \neq \text{PERM}(B(z))$, then $f(z) - \text{PERM}B(z)) \neq 0$ and therefore $Pr_{r \in_R \mathbb{F}}[f(r) = \text{PERM}(B(r))] \leq i(i-1)/|\mathbb{F}|$. In other words, the probability that the prover can "fool" the verifier in this round is at most $i(i-1)/|\mathbb{F}|$. Therefore, the total probability that the Prover succeeds in convincing the Verifier of an incorrect value is at most $\sum_{i=2}^{n} i(i-1)/\mathbb{F} < n^3/|\mathbb{F}| \leq 1/10$ for $|\mathbb{F}| \geq 10n^3$. (In fact, the sum is at most $(n^3 - n)/(3||)$ so $|\mathbb{F}| \geq n^3$ suffices.) □

The above proof shows that there are interactive proofs for coNP. A constant number of rounds is unlikely unless the polynomial-time hierarchy collapses. However, in the above protocol the prover requires the ability to solve a #P-hard problem.

Open Problem 11.1. What prover power is required to prove coNP $\subseteq$ IP?

11.1.1 Low Degree Polynomial Extensions

A key idea of the above argument was to use *low degree polynomial extensions*. This involves taking a function $f : I \to \mathbb{F}$, in this case $I = \{1, \ldots, n\}$, extending it to a polynomial $P_f : \mathbb{F} \to \mathbb{F}$, and checking P_f on random points of $\mathbb{F}$.

To apply this we used the fact that the function in question we wished to compute could be expressed as a multivariate polynomial of low total degree.

11.2 IP equals PSPACE

In this section we prove the following characterization theorem for IP.

Theorem 11.7 (Shamir, Shen). IP = PSPACE

Proof. (Following Shen.) We will prove the hard direction, namely that PSPACE $\subseteq$ IP; the other direction is left as an exercise.

The key idea of this proof will also involve low degree polynomial extensions. In order to use this we need the following facts about finite fields.

1. For any integer n, there exists a prime p such that $n \leq p \leq 2n$.
2. For any prime p and integer $k \geq 0$, there exists a finite field $\mathbb{F}_{p^k}$ with p^k elements.

We construct an IP protocol for TQBF using low-degree polynomial extensions over a small finite field $\mathbb{F}$. Specifically, we can choose a small field $\mathbb{F}$ with $n^3 m \leq |\mathbb{F}| \leq 2n^3 m$, where m is the number of 3-CNF clauses and n is the number of variables in the TQBF formula $\Psi = \exists x_1 \forall x_2 \cdots Q_n x_n \psi(x_1, \ldots, x_n)$ where ψ is a 3-CNF formula.

11.2.1 Arithmetization of Boolean formulas

Create multivariate polynomial extensionds for Boolean formulas as follows:

$$\begin{aligned} f \wedge g &\mapsto P_f \cdot P_g \\ x_i &\mapsto x_i \\ \neg f &\mapsto (1 - P_f) \\ (f \vee g) = \neg(\neg f \wedge \neg g) &\mapsto 1 - (1 - P_f)(1 - P_g). \end{aligned}$$

We use the notation $P_f \circledast P_g$ as a shorthand for $1 - (1 - p_f)(1 - p_g)$. Applying these operations $P_\psi(x_1, \ldots, x_n)$ is of degree $\leq m$ in each variable, with a total degree of $\leq 3m$.

Continuing this in the obvious way we obtain that

$$P_{\forall x_n f}(x_1, \ldots, x_{n-1}) = P_f(x_1, \ldots, x_{n-1}, 0) \cdot P_f(x_1, \ldots, x_{n-1}, 1)$$

and

$$P_{\exists x_n f}(x_1, \ldots, x_{n-1}) = P_f(x_1, \ldots, x_{n-1}, 0) \circledast P_f(x_1, \ldots, x_{n-1}, 1).$$

We want to know if $P_\Psi() = 1$. The obvious analog to our proof for PERM has a problem: the degree of the polynomial doubles at each quantification step and thus the univariate polynomials we will create will have exponential degree. The solution rests on the fact that our polynomial need only be correct on inputs over $\{0, 1\}$, which yields only two points per variable. Thus a polynomial of linear degree in each variable will suffice. For this purpose we introduce a new degree reduction operation Rx_i.

Definition 11.2. $P_{Rx_i f}(x_1, \ldots, x_n) = x_i \cdot P_f(x_1, \ldots, x_{i-1}, 1, x_{i+1}, \ldots, x_n) + (1 - x_i) \cdot P_f(x_1, \ldots, x_{i-1}, 1, x_{i+1}, \ldots, x_n)$

We now replace Ψ by the formal sequence

$$\Psi_0 = \exists x_1 R x_1 \forall x_2 R x_1 R x_2 \exists x_3 R x_1 R x_2 R x_3 \cdots Q_n x_n R x_1 R x_2 \cdots R x_n \psi(x_1, \ldots, x_n).$$

While the $\exists$ and $\forall$ operators increase the polynomial degree, the R operators bring it back down to at most one in each variable.

11.2.2 An Interactive Protcol for PSPACE

Using the arithmetization discussed earlier, we now show that the prover can convince the verifier in polynomial time.

First, the Prover claims to the Verifier that $P_{\Psi_0}() = 1$. At stage j of the interactive proof there will be some fixed values $r_1, \ldots, r_k \in \mathbb{F}$ chosen by the Verifier so far and a value $a_j \in \mathbb{F}$ for which the Prover will be trying to convince the Verifier that $P_{\Psi_j}(r_1, \ldots, r_k) = a_j$.

There are several different cases, depending on the form of Ψ_j.

$\Psi_j = \forall x_{k+1} \Psi_{j+1}$: In this case, the Prover computes $f_{j+1}(z) = P_{\Psi_{j+1}}(r_1, \ldots, r_k, z)$ and transmits the coefficients of f_{j+1}. The Verifier checks that $f_{j+1}(0) \cdot f_{j+1}(1) = a_j$ (which should be $P_{\Psi_j}(r_1, \ldots, r_k)$). If not, the Verifier rejects; otherwise, the Verifier chooses $r_{k+1} \in_R \mathbb{F}$ and sends r_{k+1} to the Prover. The new value $a_{j+1} = f_{j+1}(r_{k+1})$ and the protocol continues as the Prover tries to convince the Verifier that $P_{\Psi_{j+1}}(r_1, \ldots, r_{k+1}) = a_{j+1}$.

$\Psi_j = \forall x_{k+1} \Psi_{j+1}$: In this case, the Prover computes $f_{j+1}(z) = P_{\Psi_{j+1}}(r_1, \ldots, r_k, z)$ and transmits the coefficients of f_{j+1}. The Verifier checks that $f_{j+1}(0) \circledast f_{j+1}(1) = a_j$ (which should be $P_{\Psi_j}(r_1, \ldots, r_k)$). If not, the Verifier rejects; otherwise, the Verifier chooses $r_{k+1} \in_R \mathbb{F}$ and sends r_{k+1} to the Prover. The new value $a_{j+1} = f_{j+1}(r_{k+1})$ and the protocol continues as the Prover tries to convince the Verifier that $P_{\Psi_{j+1}}(r_1, \ldots, r_{k+1}) = a_{j+1}$.

$\Psi_j = R x_i \Psi_{j+1}$: In this case, the Prover computes $f_{j+1}(z) = P_{\Psi_{j+1}}(r_1, \ldots, r_{i-1}, z, r_{i+1}, \ldots, r_{k-1})$ and transmits the coefficients of f_{j+1}. (Unlike the other two cases there may be many coefficients and not just two coefficients.) The verifier checks that $(1 - r_i) f_{j+1}(0) + r_i f_{j+1}(1) = a_j$ (which should be $P_{\Psi_j}(r_1, \ldots, r_k)$). If not, the Verifier rejects; otherwise, the Verifier chooses $r'_i \in_R \mathbb{F}$ and sends r'_i to the Prover. The new value $a_{j+1} = f_{j+1}(r'_i)$ and the protocol continues as the Prover tries to convince the Verifier that $P_{\Psi_{j+1}}(r_1, \ldots, r_{i-1}, r'_i, r_{i+1}, \ldots, r_k) = a_{j+1}$.

In the base case when there are no quantifiers, the Verifier simply evaluates $P_\psi(r_1, \ldots, r_n)$ and accepts if and only if the result is correct.

The total number of stages is $n + n(n-1)/2$: one stage for each existential or universal quantifier, plus $\sum_{i=1}^{n} i$ stages for the R quantifiers. The maximum degree in any stage is no more than the greater of 2 and m, the number of clauses in ψ.

Clearly, if the values are correct, then the Prover can convince the Verifier at each stage by sending the correct polynomial for f_{j+1}.

In case $P_{\Psi_0}() \neq 1$, if the Verifier accepts then there is some stage at which $P_{\Psi_j}(\vec{r}) \neq a_j$ but $P_{\Psi_{j+1}}(\vec{r}') = a_{j+1}$. If the Prover sends the correct coefficients of f_{j+1} then the Verifier will immediately reject because the Verifier directly checks that the Prover's answers reflect the recursive definition of P_{Ψ_j}. If the Prover

sends the incorrect coefficients for f_{j+1} then the chance that the Verifier chooses a random value r on which $a_{j+1} = f_{j+1}(r) = P_{Psi_{j+1}}(\ldots, r, \ldots)$ is at most the degree of f_{j+1} divided by $|\mathbb{F}|$ which is at most $m/|\mathbb{F}|$.

By the union bound, the total failure probability is therefore less than:

$$\frac{(\frac{n(n-1)}{2} + n)m}{|\mathbb{F}|}$$

which for $|\mathbb{F}| \geq mn^3$ yields failure probability less than $1/n$. □