18.405J/6.841J Advanced Complexity Theory April 8, 2000

Lecture ??

Lecturer: Dan Spielman *Scribe: Andrej Bogdanov*

1 PSPACE $\subseteq$ IP

The proof of **PSPACE** $\subseteq$ **IP** makes use of efficient interactive protocols to check polynomial identities. In general, a polynomial $f(x_1, \ldots, x_n)$ of degree $p(n) = O(n^k)$ has roughly $p(n)^n$ coefficients and cannot be handled by an interactive protocol. However, the particular structure of **PSPACE** allows for a sequence of ϵ-reductions which transform a polynomial of degree $p(n)$ which represents the computation of a **PSPACE** machine to a suitably small polynomial.

Two types of reductions are used for the proof. The first one reduces assertions of the form

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

to

$$f(b_1, \ldots, b_n) = \beta.$$

This reduction was analyzed in the last lecture. The second type of reduction replaces assertions about the value of a polynomial at two points with a value of a polynomial at a single point and is analyzed below.

In what follows, $\mathbf{F}$ denotes a finite field. We are interested in the case when $|\mathbf{F}|$ is large (say $|\mathbf{F}| = 2^n$). Arithmetic over a field of this size can be performed in polynomial time.

Lemma 1 (two-for-one) *Let $f(x_1, \ldots, x_n)$ be a polynomial of degree $p(n) = O(n^k)$ in each of its variables. There exists an $\frac{np(n)}{|\mathbf{F}|}$-reduction from the assertion*

$$f(a_1, \ldots, a_n) = \alpha \text{ and } f(b_1, \ldots, b_n) = \beta$$

to an assertion of the form

$$f(c_1, \ldots, c_n) = \gamma.$$

Proof Let $l(t) = (1-t)(a_1, \ldots, a_n) + t(b_1, \ldots, b_n)$, $g(t) = f \circ l(t)$. Consider the following protocol between prover and verifier:

P: Send a polynomial $q(t)$.

V: Reject if $\deg q(t) > p(n)$ or $q(0) \neq \alpha$ or $q(1) \neq \beta$. Otherwise, pick a random $t_0 \in \mathbf{F}$ and ask for proof that $f(l(t_0)) = q(t_0)$.

A good prover sends $q(t) \equiv g(t)$. In this case, $q(t_0) = g(t_0)$ and the verifier accepts.

On the other hand, if $g(0) \neq \alpha$ or $g(1) \neq \beta$ but the prover doesn't reject, it follows that $q(t) \not\equiv g(t)$. By Schwarz' Lemma,

$$\Pr_{t_0}[q(t_0) = g(t_0)] \leq \frac{np(n)}{|\mathbf{F}|}.$$

Therefore, if the prover is crooked, the probability of error is bounded by $\frac{np(n)}{|\mathbf{F}|}$. ■

We are now ready to prove that **PSPACE** $\subseteq$ **IP**. For this purpose, it is instructive to recall the proof of **PSPACE** $\subseteq$ **AP**. In this proof, we look at the computation graph of a **PSPACE** Turing Machine M on input w of size n (assuming, for convenience, that the **accept** state yields itself). We then set up the following predicate:

$$FROMTO(\mathbf{x}, \mathbf{y}, l) = \begin{cases} 1, \text{ if } \mathbf{x} \longrightarrow \mathbf{y} \text{ in } 2^l \text{ steps,} \\ 0, \text{ otherwise.} \end{cases}$$

The assertion $FROMTO(\mathbf{start}, \mathbf{accept}, n^k)$ is checked using the recursive formula

$$FROMTO(\mathbf{x}, \mathbf{y}, l) = \bigvee_{\mathbf{z}} FROMTO(\mathbf{x}, \mathbf{z}, l-1) \wedge FROMTO(\mathbf{z}, \mathbf{y}, l-1).$$

Proof of **PSPACE** $\subseteq$ **IP**: We construct a family of polynomials $P_k(\mathbf{x}, \mathbf{y})$ with the property that $P_k(\mathbf{x}, \mathbf{y}) \equiv FROMTO(\mathbf{x}, \mathbf{y}, k)$ on binary inputs. For $k = 0$, $P_0(\mathbf{x}, \mathbf{y})$ needs to check the validity of a single step transition from state $\mathbf{x}$ to state $\mathbf{y}$. Let ϕ be a $3SAT$ instance representing this transition and obtain P_0 from ϕ using the canonical substitutions $\neg x \rightarrow 1 - x$, $x \wedge y \rightarrow xy$, $x \vee y \rightarrow 1 - (1 - x)(1 - y)$.

For $k > 0$, define P_k inductively by

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

The family P_k has the desired property because for any pair of binary inputs $(\mathbf{x}, \mathbf{y})$, there exists at most one $\mathbf{z} \in \{0,1\}^n$ for which $P_{k-1}(\mathbf{x}, \mathbf{z}) \cdot P_{k-1}(\mathbf{z}, \mathbf{y})$ is nonzero.

We now show that the degree of P_k in each of its variables is polynomial in n, the number of variables. For $k = 0$, this is obvious from the construction. For $k > 0$, we note that each of the variables appears in one of the product terms but not in both. Therefore the degree of variable v_i in P_k is never greater than the degree of v_i in P_{k-1}. We denote this degree by $p(n)$.

Let $2^{T(n)}$ be the running time of M on input w. Then M accepts w if and only if

$$P_{T(n)}(\mathbf{start}, \mathbf{accept}) = 1.$$

By definition, this is equivalent to

$$\sum_{\mathbf{z} \in \{0,1\}^n} P_{T(n)-1}(\mathbf{start}, \mathbf{z}) \cdot P_{T(n)-1}(\mathbf{z}, \mathbf{accept}) = 1.$$

By the summation lemma from last lecture, there exists a $\frac{np(n)}{|\mathbf{F}|}$-reduction of this assertion to an assertion of the form

$$P_{T(n)-1}(\mathbf{a}) \cdot P_{T(n)-1}(\mathbf{b}) = \alpha$$

By the two-for-one lemma, there exists another $\frac{np(n)}{|\mathbf{F}|}$-reduction of the last assertion to

$$P_{T(n)-1}(\mathbf{c}) = \beta.$$

Continuing in this fashion for $T(n)$ steps yields the assertion

$$P_0(\mathbf{x}) = \gamma.$$

This assertion can be checked by a verifier in a polynomial time because by construction, P_0 has a polynomial size formula.

The total probability of error resulting from all the reductions does not exceed $\frac{np(n)T(n)}{|\mathbf{F}|}$. For large fields $\mathbf{F}$, this value is within the error bound for an interactive proof system. ■

2 Multiple prover interactive proof systems

Given the result **IP** = **PSPACE**, it is natural to ask whether we can obtain more computational power by tweaking the structure of interactive proof systems. In particular, we are interested in what happens when we allow multiple provers which operate independently but can be asked only a restricted number of questions.

Definition 2 **MIP**$(p(n), q(n))$ *is the class of languages decidable by an interactive proof system with* $p(n)$ *independent provers and* $q(n)$ *interactions between verifier and provers.*

By *independent provers* we mean that any interaction between the verifier and a prover is hidden to all the other provers. (A good analogy is to think of provers as criminals, the verifier as the police and interactions between the two as interrogations. The police examines the criminals independently and "accepts" the criminals' statements only if they match.)

Multiple prover interactive proofs are much more powerful than **IP**. It has been shown that **MIP**$(2, 1)$ = **NEXPTIME**. We derive a weaker form of this result in the following two lectures.

Some caution must be exercised when working with multiple provers because **MIP** lacks some of the properties which are often used when one analyzes **IP**. One of the useful properties of **IP** is that error probabilities in **IP** can be exponentially amplified by using more random bits. It is surprising that this property does not extend to **IP**, as the following example shows.

Consider the following multiple prover interactive proof system with a verifier V and two provers P_1, P_2. The protocol $V \longleftrightarrow P_1, P_2$ is:

V: Choose two bits a, b independently at random. Send a to P_1 and b to P_2.

P_1: Send bit c to V.

P_2: Send bit d to V.

V: Accept if $a \vee c \neq b \vee d$; reject otherwise.

We show that the probability of accepting cannot exceed $\frac{1}{2}$ independent of the provers' strategy. Prover P_1 has four possible strategies. They are summarized in the following table:

a	c			
0	0	0	1	1
1	0	1	0	1
$a \vee c$	a	a	1	1

It follows that regardless of P_1's strategy, $a \vee c$ evaluates to either a or 1. Similarly, regardless of P_2's strategy, $b \vee d$ evaluates to either b or 1. In any of the four cases, $\Pr[a \vee c \neq b \vee d] \leq \frac{1}{2}$.

Now consider what happens when the protocol is conducted twice (in parallel) – once on bits a_1, b_1 and once on bits a_2, b_2, all chosen independently at random. By analogy with **IP**, we would expect the probability of acceptance to drop to $(\frac{1}{2})^2 = \frac{1}{4}$ regardless of the provers' strategy. However, the provers

P_1: $c_1 = c_2 = a_1 \vee a_2$,

P_2: $d_1 = d_2 = b_1 \vee b_2$

convince V to accept if and only if $a_1 = a_2 = 0$ or $b_1 = b_2 = 0$, but not both. In this case, $Pr[V \text{accepts}] = \frac{3}{8}$, which is greater than $\frac{1}{4}$.