CSC 2421: Approximation Algorithms Winter 2003

Lecture 13 : April 8th

Lecturer: Charles Rackoff *Scribe: S. Alam and Y. Ahmed*

1 Today's Topics

- Brief review of the proof that $P^{\#P} \subseteq IP$.
- Application: Interactive proofs of Tautologies.
- Proof outline of $NEXP \subseteq PCP(n^{O(1)}, n^{O(1)})$.
- Proof outline of NP $\subseteq PCP(O(logn), (logn)^{O(1)})$.

2 Brief review of the proof that $P^{\#P} \subseteq IP$

In the previous lecture, we saw that given an arithmetic expression $E(x_1, ..., x_n)$ of (total) degree d, over field F, and given a field element α, then there is a probabilistic polynomial time verifier to interact with a prover that is supposedly proving that

$$\sum_{x_1,....,x_n \in \{0,1\}} E(x_1, ..., x_n) = \alpha \in F.$$

- If it is true, then there exists a prover such that the verifier accepts with probability 1.
- If it is false, then for all provers, the verifier accepts with probability $\leq \frac{n \cdot d}{|F|}$.

3 Application: Interactive proofs of Tautologies

Say we have a polynomial time Turing machine $M(\vec{x}, \vec{y})$, where $|\vec{x}| = |\vec{y}|$. Given $\vec{x} = \vec{a}$, we want to interact with a prover to prove that $M(\vec{a}, \vec{y})$ accepts for all $\vec{y}$ of the right length. Now first transforming the problem to a formula and then to an arithmetic integer expression[1] $E(\vec{y})$, for all $\vec{y} \in \{0, 1\}^m$, we have

$$\begin{aligned} E(\vec{y}) = &\ 1 \quad \text{if the formula is satisfied by } \vec{y} \\ = &\ 0 \quad \text{if the formula is falsified by } \vec{y} \end{aligned}$$

[1] Arithmetic integer expressions consist of integer constants, and operators $+$, $-$ and $\times$.

Let us exchange 0 and 1 for convenience. Then we get

$$\begin{aligned} E(\vec{y}) = &\ 0 \quad \text{if the formula is satisfied by } \vec{y} \\ = &\ 1 \quad \text{if the formula is falsified by } \vec{y} \end{aligned}$$

We want to be convinced that

$$E(\vec{y}) = 0 \quad \text{for all } \vec{y} \in \{0,1\}^m.$$

Say, $y = y_1, y_2, ..., y_n$. We are interested in proofs that $E(\vec{y})$ is 0 for all of the 2^n values of $\vec{y}$.

3.1 Method 1

If $E(\vec{y})$ is not 0, then it is 1. So we can use a field of size $> 2^n$ and prove that

$$\sum_{\vec{y} \in \{0,1\}^n} E(\vec{y}) = 0 \in F.$$

(To prove this, just set $\alpha = 0$ in proof of $P^{\#} \subseteq IP$, given in the previous lecture notes.) Here the field F is large. For reasons to be seen later, we wish to use a much smaller field. There are different ways of doing this. Now we will talk about a standard way.

3.2 Method 2

We want a small F such that $|F| \in (n \cdot d)^{O(1)}$, where d is the degree of E.

Let us consider the polynomial in variables $u_1, u_2, ..., u_n$ over F:

$$q(u_1, u_2, ..., u_n) = \sum_{y_1, .., y_n \in \{0,1\}} E(y_1, .., y_n) u_1^{y_1} u_2^{y_2} ... u_n^{y_n}.$$

The total degree of q is $\leq n$, since the degree in each variable is at most 1. We want a proof that q is identically 0.

Now we need the following famous lemma [3].

Lemma 1 (Schwartz's Lemma) *If $f(x_1, x_2, ..., x_n)$ is a nonzero polynomial of total degree $\leq d$, and the values x_i $(i = 1, 2, ..n)$ are chosen in the interval $[0, N-1]$ independently of each other according to the uniform distribution then*

$$Pr[f(x_1, x_2, ..., x_n) = 0] \leq \frac{d}{N}.$$

By Schwartz's lemma, if q is not identically 0, then it is equal to 0 for at most a fraction $\frac{n}{|F|}$ of input values for $(u_1, ..., u_n)$.

We now choose random values $r_1, r_2, ..., r_n \in F$ for $u_1, u_2, ..., u_n$ and our new goal is to prove that

$$\sum_{y_1,..,y_n \in \{0,1\}} E(y_1, .., y_n) r_1^{y_1} r_2^{y_2} ... r_n^{y_n} = 0 \in F.$$

This is equivalent to proving[2]

$$\sum_{y_1,..,y_n \in \{0,1\}} E(y_1, .., y_n)(y_1 r_1 + 1 - y_1)(y_2 r_2 + 1 - y_2)..(y_n r_n + 1 - y_n) = 0 \in F.$$

Now our changed goal is to prove

$$\sum_{y_1,..,y_n \in \{0,1\}} E'(y_1, .., y_n) = 0 \in F,$$

where E' has degree $\leq n + d$. Now we can prove this as before. If d is polynomial in n, we can use a field of size polynomial in n.

The probability of being cheated is $\leq \frac{n \cdot (n+d)}{|F|} + \frac{n}{|F|}$. The last term comes from the error introduced by the choice of $r_1, \ldots, r_n$.

3.3 A different view

Let us consider the situation where we have a polynomial time Turing machine $M(\vec{x}, \vec{y})$ as before, but M has access to an oracle $B : \{0,1\}^n \to \{0,1\}, n = |\vec{x}| = |\vec{y}|$. The prover and the verifier both have access to B. Now the question is

"Can we have an interactive proof system which, for $\vec{x} = \vec{a}$, is able to prove that $M(\vec{a}, \vec{y})$ accepts for all $\vec{y}$, with respect to oracle B?"

Intuitively the answer is No. For example, $M(\vec{a}, \vec{y})$ might accept if and only if $B(\vec{y}) = 0$. So now we are asking the prover to prove that B maps everything to 0. B is of size 2^n. How can the prover prove that B maps everything to 0? It seems pretty obvious that the prover can't do it.

But the prover can convince us in a different way. Say that it gives us another oracle A which is an extension of this oracle B, $A : F^n \to F$, such that A is of degree $\leq n$. Such an A will exist; in fact, such an A exists that is of degree $\leq$1 in each variable.

Now, we can do the above interactive proof problem. We will do as before, but now our expression E will involve atomic expressions of the form $A(y_{i_1}, y_{i_2}, ...)$ in addition to what we had before.

[2]If $y \in \{0,1\}$, then $r^y = y \cdot r + (1 - y)$.

If E before had degree d, then it will now have degree $\leq deg(A) \cdot d \leq n \cdot d$. Nothing has been changed except that now at the end of the interactive proof, the verifier will have to make an oracle query to A. So all the stuff we had before now works, for $|F| \in n^{O(1)}$.

Now we are in a good shape to prove our main theorem.

4 Proof of $NEXP \subseteq PCP(n^{O(1)}, n^{O(1)})$

Theorem 2 *$NEXP \subseteq PCP_{1,s}(n^{O(1)}, n^{O(1)})$, for some $s < 1$.*

Proof Let $M(\vec{x})$ be a 2^n time non-deterministic Turing machine. Given $\vec{a}$ of length n, we want an oracle to convince us that $M(\vec{a})$ accepts.

First try: We will view the proof as an oracle $B : \{0,1\}^{cn} \rightarrow \{0,1\}$, where B is supposedly a representation of an accepting computation of M on $\vec{a}$. An accepting computation is the concatenation of successive ID's (instantaneous descriptions), for some accepting running of M. Now the verifier has to check that it is indeed an accepting computation. Formally, the verifier wants to check that $M'(\vec{a}, \vec{y})$ accepts all $\vec{y}$, using B as an oracle, for some M' related to M. (The idea is that M' will interpret y as the index of one of the $2^{O(n)}$ checks that has to be performed on B.)

In fact, the verifier actually can't do this, as explained in Section 3.3.

Second try: The oracle will contain, supposedly, a function $A : F^{cn} \rightarrow F$ that is of degree $\leq d = c \cdot n$ and that extends a $B : \{0,1\}^{cn} \rightarrow \{0,1\}$.

If we trust A to have the above mentioned properties, then the verifier can interact as before[3], using another part of the proof to answer all the "interactive questions". Note that $|F| = n^{O(1)}$, and the verifier only has to query A in a constant number of places (since M' need only query A in a constant number of places).

Now the question arises,

How do we check that A is a degree $\leq d$ polynomial extending a $B : \{0,1\}^{cn} \rightarrow \{0,1\}$?

The answer is that it is not possible to check its total correctness. However, we may proceed in a different way.

Let us give a definition first.

Definition 3 *Given two functions $A, A' : F^m \rightarrow F$, we say A and A' are "δ-close" if they differ on a fraction $\leq \delta$ of the inputs, i.e., $Pr_{\vec{x} \in F^m}[A(\vec{x}) = A'(\vec{x})] \geq 1 - \delta$.*

Then we would like to check the following statement

For some small constant $\delta > 0$, A is δ-close to some $A' : F^{cn} \rightarrow F$ that is of degree $\leq d$ and that extends some $B : \{0,1\}^{cn} \rightarrow \{0,1\}$.

[3]See Subsection 3.3 and then Section 3

Then we will (somehow) query A' instead of A when we do the interactive part.

Remark A can't be δ-close to two different such A', say A'' and A''', for the following reason. Assume otherwise. Then $A'' - A'''$ is a degree $\leq cn$, non-zero polynomial that is 2δ-close to 0, and therefore is equal to 0 on a fraction of inputs $\geq 1 - 2\delta$. By Schwartz's lemma we would have $1 - 2\delta \leq \frac{cn}{|F|}$, which is impossible if δ is a small enough constant that $2\delta < 1 - \frac{cn}{|F|}$.

Low degree test Let us choose random $\vec{x}, \vec{y} \in F^{cn}$, and define the polynomial in one variable $p(t) = A(\vec{x} + t \cdot \vec{y})$ where p should be of degree $\leq d = cn$ in t. Now evaluate $A(\vec{x} + t \cdot \vec{y})$ for $d+1$ fixed values of t and use interpolation to compute a degree $\leq d$ polynomial p' that is supposedly equal to p, and then randomly choose $t_0 \in F$ and check that $p'(t_0) = A(\vec{x} + t_0 \cdot \vec{y})$.

Now we will state a theorem (without giving a proof) about the reliability of this test.

Theorem 4 *If A is not δ close to some A' of degree d, then the above test rejects with probability $> \delta/2$.*

If, for a constant k, we repeat this test $> 2k/\delta$ times, it is easy to see that if A is *not* δ close to a degree $\leq d$ polynomial, then we will discover this fact with probability $> (1 - \frac{1}{2^k})$. So from now on, assume that A is δ close to a (unique) degree $\leq d$ polynomial A'.

To check that A' extends a $B : \{0,1\}^{cn} \to \{0,1\}$, we just check that $A'(\vec{y}) \cdot (1 - A'(\vec{y}))$ is 0 for every $\vec{y} \in \{0,1\}^{cn}$ (assuming we had access to A'). We do this the same way we check that $M'(\vec{a}, \vec{y})$ accepts for all $\vec{y}$, by using part of the oracle as an interactive prover.

In order to perform the checks in the above paragraph, we have to be able to evaluate A' a constant number of times, even though we only have direct access to A. So how do we compute the value of A' on a particular input, say $\vec{b}$? We use ideas similar to those in the low degree test.

How to evaluate $A'(\vec{b})$
Recall that we have access to an oracle $A : F^{cn} \to F$ that (we assume) is δ-close (where δ is a small constant) to the degree $\leq d = cn$ polynomial A'.

We first choose a random $\vec{b'} \in F^{cn}$.
Define the degree $\leq d$ polynomial p' in one variable $p'(t) = A'(\vec{b} + t \cdot \vec{b'})$.
Evaluate $A(\vec{b} + t \cdot \vec{b'})$ for $d+1$ fixed values of t and use interpolation to compute a degree $\leq d$ polynomial p that is supposedly equal to p'.
Next, choose a random nonzero $t_0 \in F$ and check that $p(t_0) = A(\vec{b} + t_0 \cdot \vec{b'})$;
if the check fails, then REJECT; if the check succeeds, then output $p(0)$.

It is clear that if $A = A'$ (which is supposed to be the case) then this procedure is guaranteed to output $A'(\vec{b})$.
So say that A is not necessarily equal to A'; we want to show that with high (constant)

probability, the procedure outputs either REJECT or $A'(\vec{b})$. This will be the case if $A(\vec{b}+t_0 \cdot \vec{b'}) = A'(\vec{b}+t_0 \cdot \vec{b'})$, and either $p = p'$ or $p(t_0) \neq p'(t_0)$. That is, we want to show it is unlikely that either one of the following two events will occur:

- **E1:** $A(\vec{b} + t_0 \cdot \vec{b'}) \neq A'(\vec{b} + t_0 \cdot \vec{b'})$
- **E2:** $p \neq p'$, but $p(t_0) = p'(t_0)$

Since t_0 is nonzero, and since $\vec{b'}$ is chosen randomly and independently of t_0, $\vec{b} + t_0 \cdot \vec{b'}$ is randomly distributed, and so event E1 occurs with probability $\leq \delta$.

If $p \neq p'$, then $p-p'$ is a nonzero polynomial in t of degree $\leq d$, and so the probability that a random nonzero t_0 will be a root of $p-p'$, that is, that event E2 occurs, is $\leq d/(|F|-1)$.
■

5 A PCP result about NP

We have,

$$NEXP \subseteq PCP_{1,s}(n^{O(1)}, n^{O(1)})$$

Bringing this down to NP in the obvious way, we would have:

$$B : \{0,1\}^{c \log n} \to \{0,1\}$$

$$A : F^{c \log n} \to F$$

where $|F| = (\log n)^{O(1)}$.

At a constant number of points in the above proof checking we would have to choose $c \log n$ values from F; this would use $\theta(\log n \log \log n)$ random bits, which is too many.

So instead of using $B : \{0,1\}^{c \log n} \to \{0,1\}$, we use $B : H^{\frac{c \log n}{\log \log n}} \to \{0,1\}$ where $H = \{0, 1, ..., \log n\}$, and $A : F^{\frac{c \log n}{\log \log n}} \to F$. (Note that B still represents n^c bits.) Now when we choose at random $\frac{c \log n}{\log \log n}$ members of F, we are only using $O(\log n)$ random bits. Note that we cannot hope that A is (close to) a degree $\frac{c \log n}{\log \log n}$ polynomial, but we can hope that A is (close to) a degree $((|H|-1)\frac{c \log n}{\log \log n})$ polynomial. With some care we will wind up with

$$NP \subseteq PCP_{1,s}(O(\log n), (\log n)^{O(1)}).$$

References

[1] László Babai, Lance Fortnow, and Carsten Lund. Non-deterministic exponential time has two-prover interactive protocols. *Computational Complexity*, 1:3-40, 1991. Preliminary version in *Proceedings of FOCS'90*.

[2] Carsten Lund, Lance Fortnow, Howard Karloff, and Noam Nisan. Algebraic methods for interactive proof systems. *Journal of the ACM*, 39(4):859-868, 1992.

[3] Jacob Schwartz. Fast probabilistic algorithms for verification of polynomial identities, *JACM*, 27:701-707, 1980.