6.841 Advanced Complexity Theory Mar 14, 2007

Lecture 11

Lecturer: Madhu Sudan *Scribe: Igor Ginzburg, Adam Lerer*

1 Administrative

- PS2 is graded and available from Swastik.
- PS3 is out and due Friday 4/6.

2 Overview

In this lecture we prove the two containments of Toda's Theorem:

1. $\mathrm{PH} \subseteq \mathrm{BP} \cdot \oplus \cdot \mathrm{P}$
2. $\mathrm{BP} \cdot \oplus \cdot \mathrm{P} \subseteq \mathrm{P}^{\#\mathrm{P}}$

3 Operator Definitions

The alternation operators acting on a language L or complexity class C are defined as follows:

$$\begin{array}{ll} \exists \cdot L = \{x | \exists y \text{ s.t. } (x,y) \in L\} & \exists \cdot C = \{\exists \cdot L | L \in C\} \\ \forall \cdot L = \{x | \forall y \text{ s.t. } (x,y) \in L\} & \forall \cdot C = \{\forall \cdot L | L \in C\} \\ \oplus \cdot L = \{x | \#y \text{ s.t. } (x,y) \in L \text{ is even}\} & \oplus \cdot C = \{\oplus \cdot L | L \in C\} \\ \mathrm{BP} \cdot L = (\Pi_{YES}, \Pi_{NO}) & \mathrm{BP} \cdot C = \{\mathrm{BP} \cdot L | L \in C\} \end{array}$$

$$\Pi_{YES} = \{x | Pr_y[(x,y) \in L] \geq 1 - \frac{1}{2^{q(n)}}\}$$

$$\Pi_{NO} = \{x | Pr_y[(x,y) \in L] \leq \frac{1}{2^{q(n)}}\}$$

The classes in PH can be defined using these operators: $\Sigma_k^{\mathrm{P}} \exists \cdot \forall \cdot \exists \cdot \ldots \cdot \mathrm{P}$ and $\Pi_k^{\mathrm{P}} = \forall \cdot \exists \cdot \forall \cdot \ldots \cdot \mathrm{P}$.

4 Toda's Theorem 1: $\mathrm{PH} \subseteq \mathrm{BP} \cdot \oplus \cdot \mathrm{P}$

Lemma 1 $\mathrm{BP} \cdot \oplus \mathrm{P} = \mathrm{BP} \cdot \oplus \cdot \mathrm{BP} \cdot \oplus \cdot \mathrm{P}$

First we note that for a complexity class C closed under complement, $\mathrm{BP} \cdot C$ is closed under complement since the error bounds for BP are symmetric. $\oplus \cdot C$ is also closed under complement since by a method from Lecture 12 we can add exactly one satisfying assignment to a formula, flipping the parity. Therefore all the classes we're dealing with are closed under complement. The method from Lecture 12 also allows us to exchange 'odd' for 'even' in the definition of $\oplus$.

We see that $\mathrm{BP} \cdot \oplus \mathrm{P} \subseteq \mathrm{BP} \cdot \oplus \cdot \mathrm{BP} \cdot \oplus \cdot \mathrm{P}$ trivially. The opposite containment can be shown based on some general properties:

Property 1.1 $\oplus \cdot \oplus \cdot C = \oplus \cdot C$.

Proof For $L \in C$, let $L' = \oplus \cdot L$ and $L'' = \oplus \cdot \oplus \cdot L$. Now,

$$\begin{aligned} x \in L'' &\iff \#y_1 \text{ s.t. } (x, y_1) \in L' \text{ is odd} \\ &\iff \sum_{y_1} L'(x, y_1) = 1 (\text{mod} 2) \\ &\iff \sum_{y_1} \sum_{y_2} L(x, y_1, y_2) = 1 (\text{mod} 2) \\ &\iff \sum_{(y_1, y_2)} L(x, y_1, y_2) = 1 (\text{mod} 2) \\ &\iff \#(y_1, y_2) \text{ s.t. } (x, y_1, y_2) \in L \text{ is odd} \\ &\iff x \in L' \end{aligned}$$

So, $L'' = L'$. ■

Property 1.2 $\text{BP} \cdot \text{BP} \cdot C \subseteq \text{BP} \cdot C$

Proof Again, for $L \in C$ let $L'' = \text{BP} \cdot \text{BP} \cdot L$ and $L' = \text{BP} \cdot L$. Now,

$$\begin{aligned} x \in L'' &\Rightarrow Pr_{y_1}[(x, y_1) \in L'] \geq 1 - 2^{-q_1(n)} \\ &\Rightarrow Pr_{y_1}[Pr_{y_2}[(x, y_1, y_2) \in L] \geq 1 - 2^{-q_2(n)}] \geq 1 - 2^{-q_1(n)} \\ &\Rightarrow Pr_{y_1, y_2}[(x, y_1, y_2) \in L] \geq 1 - 2^{-q_1(n)} - 2^{-q_2(n)} \\ &\Rightarrow x \in L' \end{aligned}$$

Since we can similarly show that $x \notin L'' \Rightarrow x \notin L'$, $L'' = L'$. ■

Property 1.3 $\oplus \cdot \text{BP} \cdot C \subseteq \text{BP} \oplus \cdot C$

Proof
Fix $L \in C$. We will show that $\oplus \cdot \text{BP} \cdot Ł \subseteq \text{BP} \cdot \oplus \cdot C$.
Now, $\oplus \cdot \text{BP} \cdot L = \{x | \#y \text{ s.t. } \{Pr_z[(x, y, z) \in L] \geq 1 - 2^{-q_1(n)}\} \text{ is even}\}$.
We define $\widetilde{L} = \{x | Pr_z[\#y \text{ s.t. } (x, y, z) \in L \text{ is even}] \geq 1 - 2^{-q_2(n)}\}$. $\widetilde{L}$ is clearly $\in \text{BP} \cdot \oplus \cdot C$. We will show that for proper $q_1(n)$ and $q_2(n)$, $\oplus \cdot \text{BP} \cdot L = \widetilde{L}$:

Define $L' = \text{BP} \cdot L = \{(x, y) | Pr_z[(x, y, z) \in L] \geq 1 - 2^{-q(n)}\}$.
Fix x. We say that z is bad for y if $L(x, y, z) \neq L'(x, y)$.
When we fix y, we see that:

$$Pr_z[z \text{ is bad for } y] \leq 2^{-q(n)}$$

$$Pr_z[\exists y \text{ s.t. } z \text{ is bad for } y] \leq 2^l 2^{-q(n)} (\text{where } l = |y|)$$

$$Pr_z[z \text{ is good for all } y] \geq 1 - 2^l 2^{-q(n)}$$

$$Pr_z[\forall y L(x, y, z) = L'(x, y)] \geq 1 - 2^l 2^{-q(n)}$$

By choosing $q(n)$ sufficiently large, we can get that for most z's, $\forall y L(x, y, z) = L'(x, y)$. So, for most z's, $\oplus \cdot L(x, z) = \oplus \cdot Ł'(x)$. Now, this is equivalent to saying that $\widetilde{L} = \oplus \cdot Ł'(x) = \oplus \cdot \text{BP} \cdot Ł(x)$.
■

Lemma 2 $\exists \cdot C \subseteq BP \cdot \oplus \cdot C$

Proof Let's look at some language $L \in C$. By the same argument given by Valiant-Varizani (an RP reduction), there is a language $L' \in C$ such that

$$\exists y : (x, y) \in L \Leftrightarrow Pr_z[\#y : (x, y, z) \in L' \; is \; even] \geq 1/p(n) \tag{1}$$

and

$$\forall y : (x, y) \notin L \Leftrightarrow Pr_z[\#y : (x, y, z) \in L' \; is \; even] = 0 \tag{2}$$

Now, the only problem is that RP only needs a probability of $1/p(n)$ when $x \in L$, but strong-BP needs a probability $1 - frac12^{-q(n)}$. But it is easy to get the stronger probability.

$$L'^k = \{(x, y | \forall i : (x, y_i, z_i) \in L'\} \tag{3}$$

Clearly, L'^k will still be in C. So

$$\exists y : (x, y) \in L \Leftrightarrow Pr_{z_1..z_k}[\#y_1..y_k : (\overline{x}, \overline{y}, z) \in L' \; is \; even] \geq 1 - (1 - 1/p(n))^k. \tag{4}$$

Therefore,

$$\exists y : (x, y) \in L \Leftrightarrow Pr_z[\#y : (x, y, z) \in L' \; is \; even] \geq 1/p(n) \tag{5}$$

So $L \in BP \cdot \oplus \cdot C$.
C is closed under complement, so we can easily make the same argument for $\forall \cdot C$. ■

Theorem 3 (Toda's Theorem 1) $\forall k : \Sigma_k^{\mathrm{P}}, \; \Pi_k^{\mathrm{P}} \subseteq \mathrm{BP} \cdot \oplus \cdot P$

Proof We will prove by induction on k, where the base case of $k = 0$ is trivial since $\Sigma_0^{\mathrm{P}} = \Pi_0^{\mathrm{P}} = \mathrm{P}$. We assume the induction hypothesis that $\Sigma_{k-1}^{\mathrm{P}}, \; \Pi_{k-1}^{\mathrm{P}} \subseteq \mathrm{BP} \cdot \oplus \cdot P$. We will now show the inductive case using a series of containments:

$$\Sigma_k^{\mathrm{P}} = \exists \cdot \Pi_{k-1}^{\mathrm{P}} \subseteq \exists \cdot \mathrm{BP} \cdot \oplus \cdot \mathrm{P} \subseteq \mathrm{BP} \cdot \oplus \cdot \mathrm{BP} \cdot \oplus \cdot \mathrm{P} \subseteq \mathrm{BP} \cdot \oplus \cdot \mathrm{P}$$

The first containment, $\Sigma_k^{\mathrm{P}} = \exists \cdot \Pi_{k-1}^{\mathrm{P}}$ is true by definition. The second containment is proven in Lemma 2. The last containment is true by the following:

$$\begin{aligned} \mathrm{BP} \cdot \oplus \cdot \mathrm{BP} \cdot \oplus \cdot \mathrm{P} &\subseteq \oplus \cdot \oplus \cdot \mathrm{BP} \cdot \mathrm{BP} \cdot \mathrm{P} \quad \text{(by Property 1.3)} \\ &\subseteq \oplus \cdot \mathrm{BP} \cdot \mathrm{BP} \cdot \mathrm{P} \quad \text{(by Property 1.1)} \\ &\subseteq \oplus \cdot \mathrm{BP} \cdot \mathrm{P} \quad \text{(by Property 1.2)} \end{aligned}$$

■

5 Theorem 2: $BP \cdot \oplus \cdot P \in P^{\#P}$

Toda's theorem states that we can reduce any language in $BP \cdot \oplus \cdot P$ to a language of the form

$$L_{\#} = \{x \mid 0 \leq \#y \; s.t. \; (x, y) \in L \leq a (mod \; b)\} \tag{6}$$

for some language L and integers a and b.

$L_{\#} \in P^{\#P}$, because it's just counting the satisfying y values modulo some constant. In fact, if we prove this reduction, we will show that any language in $BP \cdot \oplus \cdot P$ can be decided with just one query to a $\#P$ oracle.

Let L be some language in P and let $L' \in BP \cdot \oplus \cdot P$ be defined as:

$$x \in L' \Rightarrow Pr_y[\#z\{(x, y, z) \in L\} = 1 (mod \; 2)] \geq \frac{2}{3} \tag{7}$$

$$x \notin L' \Rightarrow Pr_y[\#z\{(x, y, z) \in L\} = 0 (mod \; 2)] \leq \frac{1}{3} \tag{8}$$

We are using a weak version of BP because it is all we'll need for this proof.

Let's suppose that the number of y values that we're enumerating over is $\leq 2^k$.

Now, what if we were somehow able to "pump up" the modulus to be 2^k? In other words, what if we could convert L' into a language defined as:

$$x \in L' \Rightarrow Pr_y[\#z\{(x, y, z) \in L\} = 1 (mod \; 2^k)] \geq \frac{2}{3} \tag{9}$$

$$x \notin L' \Rightarrow Pr_y[\#z\{(x, y, z) \in L\} = 0 (mod \; 2^k)] \leq \frac{1}{3} \tag{10}$$

Then, we could remove the probability notation and write the definition as

$$x \in L' \Rightarrow \leq \#(y, z)\{(x, y, z) \in L\} \geq \frac{2}{3} \cdot 2^k (mod \; 2^k) \tag{11}$$

$$x \notin L' \Rightarrow \#(y, z)\{(x, y, z) \in L\} \leq \frac{1}{3} \cdot 2^k (mod \; 2^k) \tag{12}$$

Now *this* can be calculated by a language of the form described in (6)!

So how are we going to boost this modulus up to 2^k?

Given that $\#y : \{(x, y) \in L_1\} = N_1$ and $\#y : \{(x, y) \in L_2\} = N_2$ for some x, we can construct languages L_+ and $L_\times$:

$$L_+ = \{(x, by) | (b = 0 \; \& \; (x, y) \in L_1) \; or \; (b = 1 \; \& \; (x, y) \in L_2)\} \tag{13}$$

$$\#y : \{(x, y) \in L_+\} = N_1 + N_2 \tag{14}$$

$$L_\times = \{(x,(y_1,y_2))|((x,y_1) \in L_1) \; and \; ((x,y_2) \in L_2)\} \tag{15}$$

$$\#y : \{(x,y) \in L_\times\} = N_1 \times N_2 \tag{16}$$

Combining these constructions, it is clear that if $N(x) = \#y : \{(x,y) \in L\}$, then for any f, we can construct a language L_f such that $\#y : \{(x,y) \in L_f\} = f(N(x))$. So, we just need a function f such that:

$$x = 0 \; (mod \; 2^z) \Rightarrow f(x) = 0 (mod \; 2^{2z}) \tag{17}$$

$$x = 1 \; (mod \; 2^z) \Rightarrow f(x) = 1 (mod \; 2^{2z}) \tag{18}$$

With such an f, we can iteratively apply f to x k times, which will pump $(mod 2)$ up to $(mod 2^k)$. Well, it turns out there's no function with these properties. However, if we replace 1 with -1, then there is such a function, namely $f(x) = 4x^3 + 3x^4$.

$$x = 0 \; (mod \; 2^z) \Rightarrow 4x^3 + 3x^4 = 0 \; (mod \; 2^{2z}) \tag{19}$$

$$x = -1 \; (mod \; 2^z) \Rightarrow 4x^3 + 3x^4 = -1 \; (mod \; 2^{2z}) \tag{20}$$

So now we can construct a language from L' that we can use in (6) with $a = 2^{k-1}$ and $b = 2^k$, thus solving L' in $P^{\#P}$.