| **18.405J/6.841J Advanced Complexity Theory** | March 18, 1999 |
|---|---|
| Lecture 12 | |
| *Lecturer: Dan Spielman* | *Scribe: Jamie Rackmill* |

# Toda's Theorem Continued

We continue in our proof of Toda's theorem, that $PH \subseteq P^{\#P}$. In the last lecture we proved that $PH \subseteq \mathrm{BP} \cdot \bigoplus \cdot \mathrm{P}$. Here we complete the proof by showing that $\mathrm{BP} \cdot \bigoplus \cdot \mathrm{P} \subseteq \mathrm{P}^{\#\mathrm{P}}$ using different techniques from last time.

We first recall the meaning of $L \in \mathrm{BP} \cdot \bigoplus \cdot \mathrm{P}$. This expression means that there exists a language $A \in P$, functions $a(n)$, $b(n) \subseteq n^{\varnothing(n)}$ such that

- $w \in L \Longrightarrow Pr[|\{y \text{ such that} |y| = b(|w|), (w,x,y) \in A\}| \text{ is odd}] \geq \frac{2}{3}$
- $w \notin L \Longrightarrow Pr[|\{y \text{ such that} |y| = b(|w|), (w,x,y) \in A\}| \text{ is odd}] \leq \frac{1}{3}$ ,

where the probabilities are over $x$ chosen uniformly from $\{0,1\}^{a(n)}$.

We will determine by an oracle call which of the two cases we are in. We let $M$ be a nondeterministic Turing Machine such that on input $(w, x)$, it nondeterministically chooses $y$ and accepts if $(w, x, y) \in A$. We then let $f = \#M$, the number of accepting computations of $M$ on its input. We see that

- $w \in L \Longrightarrow Pr_{|x|=a(|w|)}[f(w,x) \text{is odd}] \geq \frac{2}{3}$
- $w \notin L \Longrightarrow Pr_{|x|=a(|w|)}[f(w,x) \text{is odd}] \leq \frac{1}{3}$

We are interested only in whether $f(w, x)$ is even or odd. This corresponds to the last bit in the binary representation of $f(w, x)$. We want to create a function $f'$ that puts zeros between the least significant digit and all of the other digits, such that we have enough zeros when we add up $f'(w, x)$, for all values of $x$, the rightmost bits of the sum will be the number of ones in the righthand column. To accomplish this we examine the manipulation of $\#P$ functions.

**Lemma 1** *Let $f$, $g \in \#P$. Then*

**a.** $f + g \in \#P$

**b.** $f \cdot g \in \#P$

**c.** $c(|w|) \cdot f(w) \in \#P$ *where $c(|w|)$ is a positive integer constant that can be computed in time polynomial in the length of $w$*

**d.** $f(w)^{d(|w|)} \in \#P$ *where $d(|w|)$ is a positive integer that can be computed in time polynomial in the length of $w$, and $d(|w|) \subseteq |w|^{O(1)}$*

**Proof of Lemma 1** Let $f = \#M_1$ and $g = \#M_2$

**a.** We create a machine where the first step of the machine is to nondeterministically choose $M_1$ or $M_2$. Then the total number of accepting paths in this new machine is the number of accepting paths in $M_1$ plus the number of accepting paths in $M_2$.

**b.** We create a machine that runs $M_1$. If a computation path accepts, it then runs $M_2$ and accepts only if both machines accept. Then the total number of accepting paths is the product of the number of accepting paths of $M_1$ times the number of accepting paths of $M_2$.

**c.** We create a machine that computes $c(|w|)$ in polynomial time. The machine then nondeterministically chooses $i$, where $1 \leq i \leq c(|w|)$. This takes time $\lceil \log c(|w|) \rceil$ as $c(|w|)$ is at most exponential in the size of $w$. The machine then ignores $i$ and runs M. The number of accepting paths is thus $c(|w|)$ times the number of accepting paths of $M_1$.

**d.** We create a machine that computes $d(|w|)$. Then for $i = 1$ to $d(|w|)$ this machine runs $M_1$ and rejects if $M_1$ rejects at any point, and accepts if along the computation path $M_1$ always accepts. This machine has $f^{d(|w|)}$ accepting computation paths.

■

Given $f \in \#P$ we want to design a function in $\#P$ that is some polynomial in $f$.

**Definition 2** $h(x) = 3x^4 + 4x^3$

**Claim 3**

$$x \equiv 0 \ mod \ 2^c \Longrightarrow h(x) \equiv 0 \ mod \ 2^{2c}$$

$$x \equiv -1 \ mod \ 2^c \Longrightarrow h(x) \equiv -1 \ mod \ 2^{2c}$$

**Proof** 3

$$\begin{aligned} x &= -1 + a2^c \text{ implies} \\ h(x) &= 3 - 12a2^c + 2^{2c}? - 4 + 12a2^c + 2^{2c}? = -1 + 2^{2c}? \end{aligned}$$

where ? represents other factors ■

**Definition 4**

$$h^{(c)}(x) = h^{(c-1)}(h(x))$$

$$h^{(1)}(x) = h(x)$$

**Corollary 5**

$$x \equiv 1 \ mod \ 2 \Longrightarrow h^{(c)}(x) \equiv -1 \ mod \ 2^{2^c}$$

$$x \equiv 0 \ mod \ 2 \Longrightarrow h^{(c)}(x) \equiv 0 \ mod \ 2^{2^c}$$

Recall that we choose $x$, $|x| = a(|w|)$, at random and consider whether $f(w, x)$ is odd. For $w \in L$ the probability that $x$ is odd is $\geq \frac{2}{3}$. For $w \notin L$ the probability that $x$ is odd is $\leq \frac{1}{3}$.

We consider $h^{(\lceil \log a(|w|) \rceil)}(f(w, x))$. If $f(w, x)$ is odd, then this is equivalent to $-1 \ mod \ 2^{2^{\lceil \log a(|w|) \rceil}}$ and if $f(w, x)$ is even, then this is equivalent to $0 \ mod \ 2^{2^{\lceil \log a(|w|) \rceil}}$. We sum this over all choices for $x$. There are $2^{a(|w|)}$ choices for $x$ so there are enough zeroes to count how many ones we have here.

Assume that we have proved that $h^{\lceil \log a(|w|) \rceil}(f(w, x)) = \#M'(w, x)$, where Turing Machine $M'$ computes $f'$. We can then construct $M''$ to choose $x$ nondeterministically and run $M'(w, x)$. Then $\#M'' = \sum_{|x|=a(|w|)} \#M'(w, x) = \sum_{|x|=a(|w|)} h^{(\lceil \log a(|w|) \rceil)}(w, x)$. Therefore, $\#M'' = -$ the number of $x$ for which $w(x)$ is odd $mod \ 2^{2^{\lceil \log a(|w|) \rceil}}$.

The $P^{\#P}$ machine makes one query. It counts the number of accepting paths of $M''(w)$ and takes the answer $mod 2^{2^{\lceil \log a(|w|) \rceil}} = m$. It then treats the answer as a number between $-m$ and 0 and accepts if the answer is less than $-\frac{2}{3}2^{a(|w|)}$.

We still need to prove that $M'$ is a polynomial time nondeterministic Turing Machine. To do this, we need to show that if $f \in \#P$, where $f = \#M$, $M$ a nondeterministic polynomial time Turing Machine,then for any polynomial function $a(|w|)$, $h^{\lceil \log a(|w|) \rceil}(f(w,x)) \in \#P$. The expression $h^{\lceil \log a(|w|) \rceil}$ is a one variable polynomial, and its degree is polynomial in the length of $w$. Therefore, it is computable in polynomial time. $M'$ can nondeterministically chooose a mononomial $c_i X^i$, nondeterministically choose $1 \leq j \leq c_i$ and run $M$ on $(w,x)$ $i$ times, and accept if it always accepts.