IITM-CS6840: Advanced Complexity Theory Feb 07, 2012

Lecture 18 : Proof of Toda's Theorem

Lecturer: Jayalal Sarma M.N. *Scribe: Sunil K S.*

THEME: Between P and PSPACE

LECTURE PLAN:In this lecture we will be concluding the lectures on the theme of contrasting the power of counting to that of alternations. Today we will be proving the interesting result that $\mathsf{PH} \subseteq \mathsf{P}^{\#\mathsf{P}}$. To do this, first using the Valiant-Vazirani lemma we will show that $\mathsf{PH} \subseteq \mathsf{BP}(\oplus\mathsf{P})$.

1 Valiant-Vazirani Lemma

We have already saw Valiant-Vazirani theorem which stated that :

Lemma 1. *Valiant-Vazirani Theorem: There exists a probabilistic polynomial-time algorthm f such that for every n-variable boolean formula φ*

$$\begin{aligned} \varphi \in SAT \Rightarrow Pr[f(\chi) \in USAT] &\geq \frac{1}{8n} \\ \varphi \notin SAT \Rightarrow Pr[f(\chi) \in SAT] &= 0 \end{aligned}$$

But to prove that $\mathsf{PH} \subseteq \mathsf{BP}(\oplus\mathsf{P})$ we will need amplified version of this lemma.

Lemma 2. *Valiant-Vazirani lemma (Amplified version): There exists a randomized algorithm which produce φ from a given boolean formula ϕ such that for a polynomila $q(n)$:*

$$\begin{aligned} x \in L \Rightarrow Pr[\varphi \in \oplus SAT] &\geq \left(1 - \frac{1}{2^{q(n)}}\right) \\ x \notin L \Rightarrow Pr[\varphi \notin \oplus SAT] &= 1 \end{aligned}$$

In the above construction, $\varphi = \phi \wedge h$, where h is a hash function. Here φ is independent on ϕ or its structure. To find h, chosen from a good hash family, we just need to know the domain and range, which is 2^n. Here h depends only on n. Let $\tau(x, y)$ denote the choice of h from the hash family based on the random string y. Now we modify the lemma statements given above as:

$$\begin{aligned} (\exists\phi)\phi \in SAT \Rightarrow Pr_y[\oplus_x \phi \wedge \tau(x, y)] &\geq \left(1 - \frac{1}{2^{q(n)}}\right) \\ (\neg\exists\phi)\phi \notin SAT \Rightarrow Pr_y[\oplus_x \phi \wedge \tau(x, y)] &= 1 \end{aligned}$$

where ϕ_x represents the parity of number of satisfying assignments.

Observations: Existing of a satisfying assignment for ϕ is equivalent to saying $\phi \in SAT$. Here we can talk about any ϕ. That is ϕ can even have $\exists$ or $\forall$ quantifiers. This gives a better handle since construction is completely oblivious of what ϕ is.

Claim: $\mathsf{PH} \subseteq \mathsf{BP}(\oplus \mathsf{P})$

Proof. Proof by induction on the number of alterations, k.
Basis: for $k = 0$, we already have the result $\mathsf{NP} \subseteq \mathsf{BP}(\oplus \mathsf{P})$.
Assume the result for k. ie; Σ_k or $SAT_k \in \mathsf{BP}(\oplus \mathsf{P})$.
We need to prove that $SAT_{k+1} \in \mathsf{BP}(\oplus \mathsf{P})$.

$$\phi \in SAT_k \Rightarrow Pr[\oplus_z \phi \wedge \tau(x, z)] \geq (1 - \frac{1}{2^{q(n)}})$$

$$\phi \notin SAT_k \Rightarrow Pr[\oplus_z \phi \wedge \tau(x, z)] = 0$$

Now, any $\phi' \in SAT_{k+1}$ can be written as $\phi' = \exists \sigma$ where $\sigma \in SAT_k$

$$\phi' \in SAT_{k+1} \Rightarrow \exists \sigma \in SAT_k \text{ with } \sigma \in \Pi_k$$

Let $\sigma' = \exists(\neg\varphi)$ where $\sigma = \neg\varphi$

$$\neg(\forall\varphi) \Rightarrow \exists\sigma \Rightarrow Pr[\sigma' \in \oplus SAT] \geq (1 - \frac{1}{2^{q(n)}})$$

$$\forall\varphi \Rightarrow \neg\exists\sigma \Rightarrow Pr[\sigma' \notin \oplus SAT] \geq (1 - \frac{1}{2^{q(n)}})$$

$$\forall\varphi \Rightarrow Pr[\sigma'' \in \oplus SAT] \geq (1 - \frac{1}{2^{q(n)}})$$

$$\neg\forall\varphi \Rightarrow Pr[\sigma'' \notin \oplus SAT] \geq 1$$

$$\varphi \to \varphi \wedge (h(y) = 0^k) \to \varphi \wedge (h(y) = 0^{k_1} \wedge h(x) = 0^{k_2})$$

$$\phi = \exists\forall\varphi$$

□

2 Toda's Theorem: $\mathsf{PH} \subseteq \mathsf{P}^{\#\mathsf{P}}$

Any problem in PH can be solved by a P machine by making queries to a $\#\mathsf{P}$ machine.

Lemma 3. *If $\mathcal{A} \in \oplus\mathsf{P}$ then $\exists B$ such that for any polynomial q and input x of length n,*

$$x \in \mathcal{A} \Rightarrow (\#(x,y) \in B) \equiv -1 \mod 2^{q(n)}$$

$$x \notin \mathcal{A} \Rightarrow (\#(x,y) \in B) \equiv 0 \mod 2^{q(n)}$$

Now we can state the lemma as, Let $\mathcal{A} \in \oplus\mathsf{P}$. Then for any polynomial q, there exists a polynomial-time NTM M such that for any input x of length n,

$$x \in \mathcal{A} \Rightarrow (\chi_M(x)) \equiv -1 \mod 2^{q(n)}$$

$$x \notin \mathcal{A} \Rightarrow (\chi_M(x)) \equiv 0 \mod 2^{q(n)}$$

Here $\chi_M(x)$ denotes the number of accepting computations of M on x.

Proof. Let M_1 be a polynomial time NTM such that

$$\chi_{M_1} \equiv 1 \mod 2 \text{ if } x \in \mathcal{A}$$

$$\chi_{M_1} \equiv 0 \mod 2 \text{ if } x \notin \mathcal{A}$$

In other words,

$$x \in \mathcal{A} \Rightarrow (\#acc_M(x)) \text{ is odd}$$
$$x \notin \mathcal{A} \Rightarrow (\#acc_M(x)) \text{ is even}$$

Define another polynomial time NTM M_2 that repeats M_1 on x a number of times such that

$$x \in \mathcal{A} \Rightarrow (\#acc_{M_2}(x)) \text{ is odd}$$
$$x \notin \mathcal{A} \Rightarrow (\#acc_{M_2}(x)) \text{ is even}$$

Note: Given two NDTMs M_1 and M_2, we know how to get an M_3 with:

- $\#acc_{M_3}(x) = \#acc_{M_1}(x) + \#acc_{M_2}(x)$: By running M_1 and M_2 in parallel.
- $\#acc_{M_3}(x) = \#acc_{M_1}(x) \times \#acc_{M_2}(x)$: By rnning M_1 and M_2 one after other.

Let $f(x,i) = \chi_{M_2}(< x, i >)$. It if clear that f satisfies the recurrence relation given below:

$$f(x, i+1) = 3f(x,i)^4 + 4f(x,i)^3, i \geq 0 \tag{1}$$

From equation 1

$$f(x,0) \text{ is even } \Rightarrow f(x,i) \equiv 0 \mod 2^{2^i}$$

$$f(x,0) \text{ is odd } \Rightarrow f(x,i) \equiv -1 \mod 2^{2^i}$$

Now from the above analysis,

$$x \in \mathcal{A} \Rightarrow (\chi_M(x) \equiv -1 \mod 2^{2^{\log q(n)}} \equiv -1 \mod 2^{q(n)}$$

$$x \notin \mathcal{A} \Rightarrow (\chi_M(x) \equiv 0 \mod 2^{2^{\log q(n)}} \equiv 0 \mod 2^{q(n)}$$

□

Theorem 4. $\mathsf{BP}(\oplus\mathsf{P}) \subseteq \mathsf{P}^{\#\mathsf{P}}$

Proof. $L \in \mathsf{BP}(\oplus\mathsf{P})$ means there exists a set $\mathcal{A} \in \oplus\mathsf{P}$ and a polynomial p such that for all x,

$$x \in L \Rightarrow Pr_y[(x,y) \in \mathcal{A}] \geq \frac{2}{3}$$

$$x \notin L \Rightarrow Pr_y[(x,y) \in \mathcal{A}] \leq \frac{1}{3}$$

where y ranges over all strings of length $p(|x|)$.

By lemma 3, if $\mathcal{A} \in \oplus\mathsf{P}$, there exists a polynomial-time NTM M such that for all (x,y), with $|x| = n$ and $|y| = p(n)$

$$\chi_M(< x, y >) \equiv -1 \mod 2^{p(n)}, \text{ if } < x, y > \in \mathcal{A}$$

$$\chi_M(< x, y >) \equiv 0 \mod 2^{p(n)}, \text{ if } < x, y > \notin \mathcal{A}$$

Let $g(x)$ and $h(x)$ are two functions defined as,

$$g(x) = |\{y : |y| = p(|x|), (x,y) \in \mathcal{A}\}|$$
$$h(x) = \sum_{|y|=p(|x|)} \chi_M(< x, y >)$$

Then for any x of length n,

$$\begin{aligned}
h(x) &= \sum_{<x,y>\in\mathcal{A}} \chi_M(< x, y >) + \sum_{(<x,y>\notin\mathcal{A})} \chi_M(< x, y >) \\
&= (\sum_{<x,y>\in\mathcal{A}} (-1) + \sum_{(<x,y>\notin\mathcal{A})} 0) \mod 2^{q(n)} \\
&\equiv (g(x).(-1) + (2^{p(n)} - g(x)).0) \mod 2^{p(n)} \\
&\equiv (-g(x)) \mod 2^{p(n)}
\end{aligned}$$

Construct a machine N that has $h(x)$ accepting path (Just guess a y and run N). Now make a query to a #P machine to compute $h(x)$. By having $q(n)$ sufficiently large, ie., $2^{q(n)} > 2p(n)$ we can compute $g(x) = (2^{p(n)} - h(x))$.

We know $x \in L$ if and only if $g(x) > 2^{p(n)-1}$. Since $g(x)$ can be computed from $h(x)$ and $p(n)$ it is clear that we can decide whether $x \in L$ from $h(x)$. The function h is in $\#\mathsf{P}$, we can define an NTM M_1 that on input x first nondeterministically guesses a string y of length $p(|x|)$ and then simulate M on $< x, y >$. Hence, $L \in P^{\#\mathsf{P}}$. $\square$