18.405J/6.841J Advanced Complexity Theory	March 14, 2000
Lecture 12	
Lecturer: Dan Spielman	*Scribe: Michael Rosenblum*

Toda's Theorem Continued

We will use ideas from last lecture's proof that AC_0 circuits cannot compute parity to prove Toda's theorem, that $PH \subseteq P^{\#P}$.

Recall that we showed $\#P = \textit{Gap-P}$ last lecture. In order to prove $PH \subseteq P^{\textit{Gap-P}}$ we will first need to show that the function class *Gap-P* is closed under certain operations. For example, last time we showed:

for $f, g \in \textit{Gap-P}$, $h \in FP$

- $f + g \in \textit{Gap-P}$
- $f \cdot g \in \textit{Gap-P}$
- $-f \in \textit{Gap-P}$
- $h \in \textit{Gap-P}$

Today we extend these results, showing that certain polynomials of functions in *Gap-P* are in *Gap-P*.

Lemma 1 *For $f(x, y) \in \textit{Gap-P}$, $q(n)$ a polynomial, we have,*

a. $g(x) \stackrel{\text{def}}{=} \sum_{y:|y|=q(|x|)} f(x, y) \in \textit{Gap-P}$

b. $g'(x) \stackrel{\text{def}}{=} \prod_{j=1}^{q(|x|)} f(x, j) \in \textit{Gap-P}$

c. *For $n = |x|$, with $c_i, \vec{d_i}$ computable in polynomial time in n, i (written in binary), and the family of polynomials $P_n(a_1, \ldots, a_{2^n}) \stackrel{\text{def}}{=} \sum_{i=1}^{2^{n^{O(1)}}} c_i \cdot \vec{a}^{\vec{d_i}}$ with total degree $n^{O(1)}$, we have $P_n(f(x, 1), \ldots, f(x, 2^n)) \in \textit{Gap-P}$.*

d. *For R_n, Q_n families of polynomials as in the previous lemma, we have that their composition is in Gap-P.*

Proof of Lemma 1

a. By hypothesis, there exists a non-deterministic polynomial time machine M s.t. $f(x, y) = acc_M(x, y) - rej_M(x, y)$, where $acc_M(x, y)$ is the number of accepting paths in M's computation on input (x, y), and $rej_M(x, y)$ counts the number of M's rejecting paths on input (x, y). Create a new non-deterministic polynomial time machine M' that on input x, non-deterministically guesses y of length $q(|x|)$, then runs M on (x, y). Then $acc_{M'}(x) - rej_{M'}(x) = \sum_{y:|y|=q(|x|)} [acc_M(x, y) - rej_M(x, y)] = \sum_{y:|y|=q(|x|)} f(x, y)$

b. Using M as in (a.), create a new machine that, for $j = 1$ to $q(|x|)$ runs $M(x, j)$ and rejects iff $M(x, j)$ rejected for an odd number of j.

c. Given M s.t. $f(x,y) = acc_M(x,y) - rej_M(x,y)$, and M_1 computing $(c_i, \vec{d_i})$ on input (i,n), create a new machine M' as follows: choose i non-deterministically between 1 and $2^{n^{O(1)}}$; compute $(c_i, \vec{d_i})$ using M_1; branch c_i times and apply the algorithm from part (b.) to the product $\vec{a}^{\vec{d_i}}$.

d. We need to show for $f(x,i,j) \in \textit{Gap-P}$, and R_n, Q_n as in (c.) that the following polynomial is in *Gap-P*:

$$R_n(Q_n(f(x,1,1),\ldots,f(x,1,2^n)), Q_n(f(x,2,1),\ldots,f(x,2,2^n)),\ldots,Q_n(f(x,2^n,1),\ldots,f(x,2^n,2^n)))$$

First, from (c.) we get a machine $M' \in \textit{Gap-P}$ that on input (x,i) computes $Q_n(f(x,i,1),\ldots,f(x,i,2^n))$, where $n = |x|$. We now apply (c.) again to the *Gap-P* function $g(x,i) \stackrel{\text{def}}{=} acc_{M'}(x,i) - rej_{M'}(x,i)$ using the parameters $(c_i, \vec{d_i})$ from R_n, to get that the above function is in *Gap-P*.

■

Now I'll show how to apply the above lemma to prove Toda's theorem.

Theorem 2 $PH \subseteq P^{\textit{Gap-P}}$

Proof Given $L \in \sum_k^P$ decided by machine M, and an input x with $n = |x|$. M can be standardized so that on every computation path it first chooses its $\forall$ and $\exists$ bits, then does a polynomial time computation, and finally accepts or rejects. The computation tree of M on input x (starting in $\exists$-states, then branching to $\forall$-states, ..., alternating a total of $k-1$ times, then doing a polynomial time computation) can be viewed as a very large circuit. Its inputs are queries of the form $(x, y_1, \ldots, y_k) \in A$? for A a fixed polynomially balanced relation, where y_i are vectors of variables chosen by the machine as it passed through its $\exists$ and $\forall$-states. These inputs feed into AND gates corresponding to $\forall$-states and OR gates corresponding to $\exists$-states. Note that the function
$f(x, y_1, \ldots, y_k) \stackrel{\text{def}}{=} 1$ if $(x, y_1, \ldots, y_n) \in A$ and 0 otherwise
is in *Gap-P*. We will next show how to convert this circuit into a (random) polynomial that can be evaluated using an oracle query to *Gap-P*.

We know from last lecture's proof that we can approximate AND and OR gates with polynomials of low degree. Replace each cluster of $\exists$ ($\forall$)-states (represented in the circuit by an AND (OR) gate) with a randomized polynomial Q_n of degree $poly(|x|)$ and using a polynomial number of random bits. Composing polynomials (at most $k-1$ times) according to the structure of the computation tree gives a polynomial P_n which approximates the entire computation tree of M on input x. P_n has degree $poly(|x|)$ with inputs of the form $f(x, y_1, ..., y_k)$ ranging over all values of $(y_1, ..., y_k)$. As shown in last lecture, the resulting polynomial P_n agrees with the computation it is approximating with probability $> 3/4$. Thus, letting r represent the random bits used in the polynomial,

- $x \in L \Longrightarrow Pr_r[P_n(x,r) = 1] > 3/4$
- $x \notin L \Longrightarrow Pr_r[P_n(x,r) = 0] > 3/4$

Applying Lemma 1.d. ($k-1$ times) gives that the function $P_n \in \textit{Gap-P}$, so it is immediate that $PH \subseteq BPP^{\textit{Gap-P}}$.

To derandomize the above algorithm and get $PH \subseteq P^{\textit{Gap-P}}$, we'll use a trick of Toda's. We first define,

Definition 3 $h(x) = 3x^2 - 2x^3$

The following can be easily proved from the definitions of modular arithmetic:

Lemma 4 *For $c \geq 1$,*

$$x \equiv 1 \ mod \ 2^c \Longrightarrow h(x) \equiv 1 \ mod \ 2^{2c}$$
$$x \equiv 0 \ mod \ 2^c \Longrightarrow h(x) \equiv 0 \ mod \ 2^{2c}$$

Now recursively define, for $c \geq 1$,

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

Upon applying Lemma 4 to $h^{(c)}(x)$, we immediately get the following lemma:

Lemma 5

$$x \equiv 1 \ mod \ 2 \Longrightarrow h^{(c)}(x) \equiv 1 \ mod \ 2^{2^c}$$
$$x \equiv 0 \ mod \ 2 \Longrightarrow h^{(c)}(x) \equiv 0 \ mod \ 2^{2^c}$$

Now consider $g(x) \stackrel{\text{def}}{=} \sum_r h^{(2\log|r|)}(P_n(x,r))$. Then $g(x) \in Gap\text{-}P$ by Lemma 1.d and 1.a. Also, taking $c = 2\log|r|$ in Lemma 5, $Pr_r[P_n(x,r) = 1] > 3/4$ implies $g(x)$ mod $2^{2^{2\log|r|}} > \frac{3}{4}2^{|r|}$, and similarly, $Pr_r[P_n(x,r) = 0] > 3/4$ implies $g(x)$ mod $2^{2^{2\log|r|}} < \frac{1}{4}2^{|r|}$. Thus we can determine whether M accepts on x by making an oracle call asking for $g(x)$ and then taking the answer modulo $2^{2^{2\log|r|}} = 2^{\log|r|^2}$. If the result is $< \frac{1}{2}2^{|r|}$ we reject, otherwise we accept. We have shown $PH \subseteq P^{Gap\text{-}P}$, completing the proof of Toda's theorem. ■