18.405J/6.841J Advanced Complexity Theory	8 March 2001
Lecture 9	
Lecturer: Dan Spielman	*Scribe: David Liben-Nowell*

Today, we will explore the class #P, as part of an investigation of classes that fall between PH and PSPACE. We will also outline the proof of Toda's Theorem, skipping over some of the details, showing that $\mathsf{PH} \subseteq \mathsf{P}^{\#\mathsf{P}}$.

1 Counting Classes

The class #P [Valiant c. 1979] (say "sharp P" or "number P") is a class of functions counting the number of accepting computations in a Turing machine:

$$\#\mathsf{P} \stackrel{\text{def}}{=} \{f : \{0,1\}^* \longrightarrow \mathbb{Z} \mid \exists \text{ a nondeterministic polynomial time Turing machine } M \text{ s.t. } f(x) = \text{the number of accepting paths of } M \text{ on input } x\}.$$

Note there is also a class GapP, in which we count the difference between the number of accepting computations and the number of rejecting computations. Sometimes it's easier to work with GapP than #P. (In #P, e.g., we can only map inputs into the non-negative integers.)

Note that $\mathsf{NP} \subseteq \#\mathsf{P}$ ("is there some accepting path?") and $\mathsf{coNP} \subseteq \#\mathsf{P}$ ("are all paths accepting?"). Today we'll also sketch Toda's theorem:

Theorem 1 (Toda) $\mathsf{PH} \subseteq \mathsf{P}^{\#\mathsf{P}}$.

First, though, some more about #P. A function f is #P-*Complete* if

(i) $f \in \#\mathsf{P}$,

(ii) For all $g \in \#\mathsf{P}$, there exists a polynomial time oracle Turing machine $M^?$ such that $M^F(x)$ outputs $g(x)$.

Interestingly, #P-Complete problems do not necessarily correspond to hard decision problems.

For a circuit C, define CIRCUITSAT(C) to be the number of input setting that satify the circuit:

$$\#\text{CIRCUITSAT}(C) \stackrel{\text{def}}{=} |\{x : C(x) = 1\}|\,.$$

#CIRCUITSAT is #P-Complete.

A *matching* is a graph is a subset of the edges using each node exactly once. (The number of matchings is the permanent of the adjacency matrix, where $\mathsf{perm}(M) = \sum_\pi \prod_{i=1}^n m_{i,\pi(i)}$. Despite its superficial similarity to the determinant $\mathsf{det}(M) = \sum_\pi \prod_{i=1}^n (-1)^{\text{sign}(\pi)} m_{i,\pi(i)}$, it is much harder to compute.) Valiant proved that counting the number of matchings in a graph is #P-Complete. Finding a single matching is easy (unlike #CIRCUITSAT, where finding one is NP-hard).

See Papadimitriou for more details, and Jerrum, Sinclar, and Vigoda [ECCC/STOC'01] for a randomized PTAS for approximately counting the number of matchings.

2 Toda's Theorem

For the rest of the lecture, we will discuss some of the ideas behind Toda's Theorem.

To get an intuition for Toda's theorem, we need to define some new complexity classes. For class $\mathcal{C}$, we will define four class operators. We can use these to get all kinds of new classes.

Definition 2 *For a complexity class $\mathcal{C}$,*

$$\begin{aligned}
\Sigma \cdot \mathcal{C} &\stackrel{\text{def}}{=} \{L : \exists \text{ polynomial } p \ \ \exists A \in \mathcal{C} \\
&\qquad\qquad x \in L \iff \exists y \in \{0,1\}^{p(|x|)} \langle x, y\rangle \in A\} \\
\Pi \cdot \mathcal{C} &\stackrel{\text{def}}{=} \{L : \exists \text{ polynomial } p \ \ \exists A \in \mathcal{C} \\
&\qquad\qquad x \in L \iff \forall y \in \{0,1\}^{p(|x|)} \langle x, y\rangle \in A\} \\
\oplus \cdot \mathcal{C} &\stackrel{\text{def}}{=} \Bigg\{L : \exists \text{ polynomial } p \ \ \exists A \in \mathcal{C} \\
&\qquad\qquad x \in L \iff \left|\left\{y \in \{0,1\}^{p(|x|)} : \langle x, y\rangle \in A\right\}\right| \text{ is odd}\Bigg\} \\
\mathsf{BP} \cdot \mathcal{C} &\stackrel{\text{def}}{=} \Bigg\{L : \exists \text{ polynomial } p \ \ \exists A \in \mathcal{C} \\
&\qquad\qquad x \in L \implies \Pr_{|y|=p(|x|)}[\langle x, y\rangle \in A] > 2/3 \\
&\qquad\qquad x \notin L \implies \Pr_{|y|=p(|x|)}[\langle x, y\rangle \in A] < 1/3\Bigg\}.
\end{aligned}$$

(Read $\oplus \cdot \mathcal{C}$ as "parity $\mathcal{C}$".)

Using these class operators, we can state the key elements of Toda's theorem.

Claim 3 $\mathsf{NP} \subseteq \mathsf{BP} \cdot \oplus \cdot \mathsf{P}$, *and more generally,* $\Sigma_k^P \subseteq (\mathsf{BP} \cdot \oplus)^k \cdot \mathsf{P}$.

Claim 4 *For any "reasonable class" $\mathcal{C}$,*

$$\begin{aligned}
\oplus \cdot \mathsf{BP} \cdot \mathcal{C} &\subseteq \mathsf{BP} \cdot \oplus \cdot \mathcal{C} \\
\oplus \cdot \oplus \cdot \mathcal{C} &= \oplus \cdot \mathcal{C} \\
\mathsf{BP} \cdot \mathsf{BP} \cdot \mathcal{C} &= \mathsf{BP} \cdot \mathcal{C}.
\end{aligned}$$

So $\Sigma_k^{\mathsf{P}} \subseteq \mathsf{BP} \cdot \oplus \cdot \mathsf{P}$.

Claim 5 $\mathsf{BP} \cdot \oplus \cdot \mathsf{P} \subseteq \mathsf{P}^{\#\mathsf{P}}$.

Toda's theorem then follows from these claims. We will give the main ideas of these proofs today, skipping over the hairy details which, we are assured, are hairy.

Lemma 6 $\mathsf{NP} \subseteq \mathsf{BP} \cdot \oplus \cdot \mathsf{P}$.

Proof We show that the NP-Complete problem CIRCUITSAT $\in \mathsf{BP} \cdot \oplus \cdot \mathsf{P}$.

Recall the theorem of Valiant and Vazirani from last lecture: there exists a random polynomial time procedure that takes a circuit C and returns $n+1$ circuits $C_1, ..., C_{n+1}$ such that if C is satistfiable,

$$\texttt{Pr}[\exists i \ C_i \text{ has exactly one satisfying assignment}] \quad > \quad 1/8,$$

and otherwise no C_i has a satisfying assignment. BP allows us to get random bits; when there is only zero or one satisfying assignment, $\oplus$ lets us determine which is the case.

Two things are not so good about this. First, the $1/8$ should be a $2/3$. Second, we have $n+1$ circuits and we want to have 1.

To handle the first problem, we amplify the probability by repetition. Say we output $C_1^i, ..., C_{n+1}^i$ for $i = 1, 2, ..., 20$. Then

$$\texttt{Pr}\left[\exists i, j\ C_j^i \text{ has exactly one satisfying assignment}\right] \quad > \quad 1-(7/8)^{20} \quad \approx \quad 0.930 \quad > \quad 2/3.$$

How can we handle the second problem? We do something like "AND them all togther" — but not quite, because that doesn't work. Write $\#C$ for the number of satisfying assignments for a circuit C. We want to build a circuit C' so that: (1) $\#C'$ is odd if any $\#C_i$ is odd, $\#C'$ is even if all $\#C_i$s are even.

The function $\#C' = 1 + \prod_{i=1}^{m}(\#C_i + 1)$ yields the desired properties. But how do be build the circuit C'? Toda's insight was that we can play arithmetic games with the number of satisfying assignments.

Given two circuits $C_1(x)$ and $C_2(y)$, we can build a circuit C' with $\#C' = \#C_1 + \#C_2$, as follows. Add a new input t, and let the circuit be $C'(x,y) = (C_1(x) \wedge t) \vee (C_2(y) \wedge \neg t)$. Similarly, we can build a circuit C' with $\#C' = \#C_1 \#C_2$: $C'(x,y) = C_1(x) \wedge C_2(y)$. It is easy to build a circuit with any constant number of satisfying assignments (e.g., 1).

Thus we can build C' so that $\#C' = 1 + \prod_{i=1}^{m}(\#C_i + 1)$. So we have a randomized polynomial time procedure which works as follows: (1) apply Valiant/Vazirani and amplify; (2) with probability at least 9/10, we have one C_i with an odd number of satisfying assignments if C is satisfiable, and with probability zero if not; (3) $\#C'$ is odd iff C is satisfiable, with good probability. ∎

Lemma 7 $\mathsf{BP} \cdot \oplus \cdot \mathsf{P} \subseteq \mathsf{P}^{\#\mathsf{P}}$.

Proof Consider a language $L \in \mathsf{BP} \cdot \oplus \cdot \mathsf{P}$. That is, unwinding the definition, there exists a polynomial time Turing machine M so that $M(w)$ outputs a circuit C_w so that

(i) if $w \in L$, then $\texttt{Pr}_x[|\{y : C_w(x,y) = 1\}| \text{ is odd}] > 2/3$.

(ii) if $w \notin L$, then $\texttt{Pr}_x[|\{y : C_w(x,y) = 1\}| \text{ is odd}] < 1/3$.

So deciding whether $w \in L$ is equivalent to differentiating whether $\texttt{Pr}_x[|\{y : C_w(x,y) = 1\}| \text{ is odd}]$ is more than 2/3 or less than 1/3.

We need to show how to do this differentiation in $\mathsf{P}^{\#\mathsf{P}}$. Idea #1: try $\sum_x \sum_y \#C_w(x,y)$. But the sum of two odds is even, so this doesn't help. But: if we're given $y_1, \ldots, y_N \in \{0,1\} \bmod 4N$, then summing them $\bmod 4N$ tells us how many are odd.

$$\begin{aligned} h(a) &\stackrel{\text{def}}{=} 4a^3 + 3a^4 \\ h^{(1)}(a) &\stackrel{\text{def}}{=} h(a) \\ h^{(c)}(a) &\stackrel{\text{def}}{=} h(h^{(c-1)}(a)). \end{aligned}$$

If $a \equiv 0 \bmod 2$ then $h^{(c)}(a) \equiv 0 \bmod 2^{2^c}$. If $a \equiv 1 \bmod 2$ then $h^{(c)}(a) \equiv -1 \bmod 2^{2^c}$. (To prove this, show that if $a \equiv 0 \bmod 2^c$ then $h(a) \equiv 0 \bmod 2^{2c}$ and if $a \equiv 1 \bmod 2^c$ then $h(a) \equiv -1 \bmod 2^{2c}$, and then use induction.)

Instead of $\sum_x \sum_y \#C(x,y)$, we compute $\sum_x h^{(c)}\left(\sum_y \#C(x,y)\right)$. For a fixed x, the value $h^{(c)}\left(\sum_y \#C(x,y)\right)$ is either 0 or $-1 \bmod 2^{2^c}$. We choose $c = \log m$, so 2^{2^c} is large enough that we can count the number of x so for which this value is -1. Now we can check whether $\texttt{Pr}_x\left[|\{y : C_w(x,y)\}| \equiv -1 \bmod 2^{2^c}\right]$ is more than 2/3 or less than 1/3 by nondeterministically choosing x. $h^{(c)}$ now has degree 4^c, which is polynomial in m. ∎

The rest of Toda's theorem isn't so hard — this lecture has sketched almost all of the hard ideas.