Last revised 4/29/2010

In this lecture, we first continue to talk about polynomial hierarchy. Then we prove the Gács-Sipser-Lautemann theorem that BPP is contained in the second level of the hierarchy.

1 The hierarchy

Definition 1 (Polynomial hierarchy) *$L \in \Sigma_k$ iff there are polynomials $p_1, \ldots, p_k$ and a polynomial time computable function F such that*

$$x \in L \Leftrightarrow \exists y_1.\forall y_2. \ \ldots Q_k y_k.F(x, y_1, \ldots, y_k) = 1 \qquad \text{where } Q_k = \begin{cases} \forall \text{ if } k \text{ is even} \\ \exists \text{ if } k \text{ is odd} \end{cases}$$

$L \in \Pi_k$ iff there are polynomials $p_1, \ldots, p_k$ and a polynomial time computable function F such that

$$x \in L \Leftrightarrow \forall y_1.\exists y_2. \ \ldots Q'_k y_k.F(x, y_1, \ldots, y_k) = 1 \qquad \text{where } Q'_k = \begin{cases} \exists \text{ if } k \text{ is even} \\ \forall \text{ if } k \text{ is odd} \end{cases}$$

For clarity, we omitted the conditions that each string y_i must be of polynomial length ($y_i \in \{0,1\}^{p_i(|x|)}$).

One thing that is easy to see is that $\Pi_k = \text{co}\Sigma_k$. Also, note that, for all $i \leq k-1$, $\Pi_i \subseteq \Sigma_k$, $\Sigma_i \subseteq \Sigma_k$, $\Pi_i \subseteq \Pi_k$, $\Sigma_i \subseteq \Pi_k$. This can be seen by noticing that the predicates F do not need to "pay attention to" all of their arguments, and so a statement involving k quantifiers can "simulate" a statement using less than k quantifiers.

Theorem 2 *Suppose $\Pi_k = \Sigma_k$. Then $\Pi_{k+1} = \Sigma_{k+1} = \Sigma_k$.*

Proof: For any language $L \in \Sigma_{k+1}$, we have that there exist polynomials $p_1, \ldots, p_{k+1}$ and a polynomial time computable function F such that

$$x \in L \Leftrightarrow \exists y_1.\forall y_2. \ \ldots Q_{k+1} y_{k+1}.F(x, y_1, \ldots, y_{k+1}) = 1$$

where we did not explicitly stated the conditions $y_i \in \{0,1\}^{p_i(|x|)}$. Let us look at the right hand side of the equation. What is following $\exists y_1$ is a Π_k statement. Thus, there is a $L' \in \Pi_k$ such that

$$x \in L \Leftrightarrow \exists y_1 \in \{0,1\}^{p_1(|x|)}.(x, y_1) \in L'$$

Under the assumption that $\Pi_k = \Sigma_k$, we have $L' \in \Sigma_k$, which means that there are polynomials $p'_1, \ldots, p'_k$ and a polynomial time computable F' such that

$$(x, y_1) \in L' \Leftrightarrow \exists z_1.\forall z_2.\ \ldots Q_k z_k.F'((x, y_1), z_1, \ldots, z_k) = 1$$

where we omitted the conditions $z_i \in \{0,1\}^{p'_i(|x|)}$. So now we can show that

$$\begin{aligned} x \in L &\Leftrightarrow \exists y_1.(x, y_1) \in L' \\ &\Leftrightarrow \exists y_1.(\exists z_1.\forall z_2.\ \ldots Q_k z_k.F'((x, y_1), z_1, \ldots, z_k) = 1) \\ &\Leftrightarrow \exists (y_1, z_1).\forall z_2.\ \ldots .Q_k z_k.F''(x, (y_1, z_1), z_2, \ldots, z_k) = 1) \end{aligned}$$

And so $L \in \Sigma_k$.

Now notice that if $\mathcal{C}_1$ and $\mathcal{C}_2$ are two complexity classes, then $\mathcal{C}_1 = \mathcal{C}_2$ implies co$\mathcal{C}_1$ = co$\mathcal{C}_2$. Thus, we have $\Pi_{k+1} = \text{co}\Sigma_{k+1} = \text{co}\Sigma_k = \Pi_k = \Sigma_k$. So we have $\Pi_{k+1} = \Sigma_{k+1} = \Sigma_k$. $\square$

2 BPP $\subseteq \Sigma_2$

This result was first shown by Sipser and Gács. Lautemann gave a much simpler proof which we give below.

Lemma 3 *If L is in* **BPP** *then there is an algorithm A such that for every x,*

$$\mathop{\mathbb{P}}_r(A(x, r) = \textit{right answer}) \geq 1 - \tfrac{1}{3m},$$

where the number of random bits $|r| = m = |x|^{O(1)}$ and A runs in time $|x|^{O(1)}$.

Proof: Let $\hat{A}$ be a **BPP** algorithm for L. Then for every x,

$$\mathop{\mathbb{P}}_r(\hat{A}(x, r) = \text{wrong answer}) \leq \tfrac{1}{3},$$

and $\hat{A}$ uses $\hat{m}(n) = n^{o(1)}$ random bits where $n = |x|$.

Do $k(n)$ repetitions of $\hat{A}$ and accept if and only if at least $\dfrac{k(n)}{2}$ executions of $\hat{A}$ accept. Call the new algorithm A. Then A uses $k(n)\hat{m}(n)$ random bits and

$$\mathop{\mathbb{P}}_r(A(x, r) = \text{wrong answer}) \leq 2^{-ck(n)}.$$

We can then find $k(n)$ with $k(n) = \Theta(\log \hat{m}(n))$ such that $\frac{1}{2^{ck(n)}} \leq \frac{1}{3k(n)\hat{m(n)}}$. $\square$

Theorem 4 **BPP** $\subseteq \Sigma_2$.

PROOF: Let L be in **BPP** and A as in the claim. Then we want to show that

$$x \in L \iff \exists y_1, \ldots, y_m \in \{0,1\}^m \forall z \in \{0,1\}^m \bigvee_{i=1}^{m} A(x, y_i \oplus z) = 1$$

where m is the number of random bits used by A on input x.

Suppose $x \in L$. Then

$$\begin{aligned} &\mathbb{P}_{y_1,\ldots,y_m}(\exists z A(x, y_1 \oplus z) = \cdots = A(x, y_m \oplus z) = 0) \\ &\leq \sum_{z \in \{0,1\}^m} \mathbb{P}_{y_1,\ldots,y_m}(A(x, y_1 \oplus z) = \cdots = A(x, y_m \oplus z) = 0) \\ &\leq 2^m \frac{1}{(3m)^m} \\ &< 1. \end{aligned}$$

So

$$\begin{aligned} \mathbb{P}_{y_1,\ldots,y_m}\left(\forall z \bigvee_i A(x, y_i \oplus z)\right) &= 1 - \mathbb{P}_{y_1,\ldots,y_m}(\exists z A(x, y_1 \oplus z) = \cdots = A(x, y_m \oplus z) = 0) \\ &> 0. \end{aligned}$$

So a sequence $(y_1, \ldots, y_m)$ exists, such that $\forall z. \bigvee_i A(x, y_i \oplus z) = 1$.

Conversely suppose $x \notin L$. Then fix a sequence $(y_1, \ldots, y_m)$. We have

$$\begin{aligned} \mathbb{P}_z\left(\bigvee_i A(x, y_i \oplus z)\right) &\leq \sum_i \mathbb{P}_z(A(x, y_i \oplus z) = 1) \\ &\leq m \cdot \frac{1}{3m} \\ &= \frac{1}{3}. \end{aligned}$$

So

$$\begin{aligned} \mathbb{P}_z(A(x, y_1 \oplus z) = \cdots = A(x, y_m \oplus z) = 0) &= \mathbb{P}_z\left(\bigvee_i A(x, y_i \oplus z) = 0\right) \\ &\geq \frac{2}{3} \\ &> 0. \end{aligned}$$

So for all $y_1, \ldots, y_m \in \{0,1\}^m$ there is a z such that $\bigvee_i A(x, y_i \oplus z) = 0$. $\square$