Professor Luca Trevisan — February 21, 2001
Scribe: Lawrence Ip — Last revised February 24, 2001

Lecture 9

1 BPP $\subseteq \Sigma_2$

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

Lemma 1
*If L is in **BPP** then there is an algorithm A such that for every x,*

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

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, $\mathbf{Pr}_r(\hat{A}(x,r) = \text{wrong answer}) \leq \frac{1}{3}$. and $\hat{A}$ uses $\hat{m}(n)$ 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 $\mathbf{Pr}_r(A(x,r) = \text{wrong answer}) \leq 2^{-ck(n)}$. We can then find $k(n)$ with $k(n) = \Theta(\log \hat{m}(n))$ such that $\frac{1}{2^{ck(n)}} \leq \frac{1}{3k(n)\hat{m(n)}}$. $\Box$

Theorem 2
BPP $\subseteq \Sigma_2$.

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

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

where m is the number of random bits used by A on input x.
Suppose $x \in L$. Then

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

So

$$\mathbf{Pr}_{y_1,\dots,y_m}(\forall z \bigvee_i A(x, y_i \oplus z)) = 1 - \mathbf{Pr}_{y_1,\dots,y_m}(\exists z A(x, y_1 \oplus z) = \cdots = A(x, y_m \oplus z) = 0)$$
$$> 0.$$

So $(y_1, \dots, y_m)$ exists.
Conversely suppose $x \notin L$. Then

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

So

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

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

2 Counting Classes

Definition 1 *R is an* $\mathbf{NP}$-relation, *if there is a polynomial time algorithm A such that $(x, y) \in R \Leftrightarrow A(x, y) = 1$ and there is a polynomial p such that $(x, y) \in R \Rightarrow |y| \leq p(|x|)$.*

$\#R$ is the problem that, given x, asks how many y satisfy $(x, y) \in R$.

Definition 2 *$\#\mathbf{P}$ is the class of all problems of the form $\#R$, where R is an $\mathbf{NP}$-relation.*

Unlike for decision problems there is no canonical way to define reductions for counting classes. There are two common definitions.

Definition 3 *We say there is a* parsimonious reduction *from $\#A$ to $\#B$ (written $\#A \leq_{par} \#B$) if there is a polynomial time transformation f such that for all x, $|\{y, (x, y) \in A\}| = |\{z : (f(x), z) \in B\}|$.*

Often this definition is a little too restrictive and we use the following definition instead.

Definition 4 *$\#A \leq \#B$ if there is a polynomial time algorithm for $\#A$ given an oracle that solves $\#B$.*

#CIRCUITSAT is the problem where given a circuit, we want to count the number of inputs that make the circuit output 1.

THEOREM 3
*#CIRCUITSAT is #**P**-complete under parsimonious reductions.*

PROOF: Let $\#R$ be in #**P** and A and p be as in the definition. Given x we want to construct a circuit C such that $|\{z : C(z)\}| = |\{y : |y| \leq p(|x|), A(x, y) = 1\}|$. We then construct $\hat{C}_n$ that on input x, y simulates $A(x, y)$. From earlier arguments we know that this can be done with a circuit with size about the square of the running time of A. Thus $\hat{C}_n$ will have size polynomial in the running time of A and so polynomial in x. Then let $C(y) = \hat{C}(x, y)$. $\Box$

THEOREM 4
*#3SAT is #**P**-complete.*

PROOF: We show that there is a parsimonious reduction from #CIRCUITSAT to #3-SAT. That is, given a circuit C we construct a Boolean formula ϕ such that the number of satisfying assignments for ϕ is equal to the number of inputs for which C outputs 1. Suppose C has inputs $x_1, \ldots, x_n$ and gates $1, \ldots, m$ and ϕ has inputs $x_1, \ldots, x_n, g_1, \ldots, g_m$, where the g_i represent the output of gate i. Now each gate has two input variables and one output variable. Thus a gate can be complete described by mimicking the output for each of the 4 possible inputs. Thus each gate can be simulated using at most 4 clauses. In this way we have reduced C to a formula ϕ with $n + m$ variables and $4m$ clauses. So there is a parsimonious reduction from #CIRCUITSAT to #3SAT. $\Box$