6.841 Advanced Complexity Theory	March 14, 2005

Lecture 12

Lecturer: Madhu Sudan — *Scribe: Swastik Kopparty*

1 Today

- Finish $BPP \subset PH$
- Randomized Reductions, Valiant-Vazirani

2 BPP $\subset$ PH

Recall the outline of the proof from last time. For language $L \in BPP$, we are given machine $M(\cdot, \cdot)$ running in polynomial time that recognizes L with completeness $1 - \frac{1}{m^2}$ and soundness $\frac{1}{m^2}$, where m is the length of the random string (which is anyway polynomial in n).

The first prover (trying to show that $x \in L$) first selects $r_1, \ldots, r_l \in \{0,1\}^m$ and gives them to the second prover (who is trying to prove that $x \notin L$). The second prover then selects a $y \in \{0,1\}^m$. The first prover wins if $\exists i \in [l], M(x, r_i \oplus y)$ accepts. The second prover wins if $\forall i, M(x, r_i \oplus y)$ rejects. We want to show that the first prover winning is equivalent to $x \in L$. That is, we want to show that $x \in L$ is equivalent to $\exists r_1, \ldots, r_l \forall y, (M(x, r_1 \oplus y)$ accepts$) \wedge (M(x, r_2 \oplus y)$ accepts$) \wedge \ldots (M(x, r_l \oplus y)$ accepts$)$.

Then $L \in \Sigma_2$.

2.1 Soundness: $x \notin L$

$$\forall r_i, Pr_y[M(x, r_i \oplus y) accepts] \leq \frac{1}{m^2}$$

So, by union bound,

$$\forall r_1, \ldots, r_l, Pr_y\left[\exists i \in [l] : M(x, r_i \oplus y) accepts\right] \leq \frac{l}{m^2}$$

which, if $l < m^2/2$, is less than $\frac{1}{2}$.

2.2 Completeness: $x \in L$

Fix a y.

$$Pr_{r_i}[M(x, r_i \oplus y) rejects] \leq \frac{1}{m^2} < \frac{1}{2}$$

So,

$$Pr_{r_1, \ldots, r_l}\left[\forall i \in [l], M(x, r_i \oplus y) rejects\right] \leq \left(\frac{1}{2}\right)^l$$

Therefore, by union bound,

$$Pr_{r_1, \ldots r_l}\left[\exists y \forall i \in [l], M(x, r_i \oplus y) rejects\right] \leq \frac{2^m}{2^l}$$

which, if $l \geq m + 1$, is $\leq \frac{1}{2}$.

Therefore $\exists r_1, \ldots, r_l \forall y \exists i \in [l], M(x, r_i \oplus y)$ accepts.

This result is due to Lautemann and Sipser.

Note that if $x \in L$, a random choice of r_i will suffice for the proof. If $x \notin L$, a random choice of y will suffice. Thus the job of the correct prover is easy.

Using this idea, Fortnow proved that promise-$RP \subset P \Rightarrow$ promise-$BPP \subset P$.

3 Unique SAT

When public key cryptography was first introduced, 3 problems were given as examples of functions hard to invert.

$USAT = \{\phi : \phi \text{ is uniquely satisfiable }\}$

The problem is, given a USAT formula, to find the satisfying assignment.

$USAT$ is a promise problem with yes instances L_{yes} and no instances L_{no}

$L_{yes} = \{\phi : \phi \text{ has 1 satisfying assignment }\}$

$L_{no} = \{\phi : \phi \text{ has no satisfying assignment }\}$

We will show Valiant and Vazirani's result that says that SAT reduces to $USAT$ under "randomized reductions". Thus, unless $RP = NP$, $USAT$ cannot be solved in randomized polynomial time.

3.1 Randomized reductions

Given 2 promise problems π_1, π_2, we say that π_1 reduces to π_2 under randomized reductions "$\pi_1 \leq_R \pi_2$" if $\exists$ probabilistic polynomial time algorithm A s.t.

- If $x \in \pi_{1,no}$, then $A(x) \in \pi_{2,no}$
- If $x \in \pi_{1,yes}$, then $Pr[A(x) \in \pi_{2,yes}] \geq \frac{1}{poly(n)}$

3.2 The reduction

We will now witness "The Amazing Power of Pairwise Independence" (see the article by Prof. Wigderson).

Suppose we have a SAT formula ϕ with EXACTLY M satisfying assignments. We will produce a ϕ', such that if $M > 0$, then with inverse polynomial probability p has exactly 1 satisfying assignment, and if $M = 0$ then with probability 1, ϕ' has 0 satisfying assignments.

Suppose we knew M. Then construct ϕ' as follows: Pick a random function $h : \{0,1\}^n \rightarrow \{0, \ldots, M\}$. Then, set $\phi'(x) = \phi(x) \wedge [h(x) = 0]$.

If $M = 0$, then ϕ' is not satisfiable. If M was indeed > 0, then we can conclude that $\exists c_1, c_2$ such that:

$$Pr[\exists x : \phi'(x) = 1] \geq c_1$$

and

$$Pr[\exists x_1, x_2 : x_1 \neq x_2 : \phi'(x_1) = \phi'(x_2) = 1] \leq c_2$$

with $c_1 > c_2$. Thus, with constant probability, $\phi'(x)$ will have exactly 1 satisfying assignment.

This is the core of the approach. We now need to remove the dependence on knowing M, as well as the choice of the hash function (right now we are choosing from a doubly exponential space of functions, requiring exponentially many random bits).

3.2.1 Removing the dependence on knowing M

This is done quite easily. Suppose the actual number of satisfying assignments is $\in [2^j, 2^{j+1}]$. Then, by using $M = 2^{j+1}$, and using the construction of ϕ' as before, $\exists c_{1,j}, c_{2,j}$ such that

$$Pr[\exists x : \phi'(x) = 1] \geq c_{1,j}$$

and

$$Pr[\exists x_1, x_2 : x_1 \neq x_2 : \phi'(x_1) = \phi'(x_2) = 1] \leq c_{2,j}$$

with $c_{1,j} > c_{2,j}$.

Thus the following will work: guess j at random from $\{0, \ldots, n\}$, and then do as above. With probability $\frac{1}{n+1}$ we will hit upon an approximately right value of M, and can use the constant probability guarantees of the above sentence to argue correctness of the reduction.

3.2.2 Polynomial time computable h

We have $Pr[\exists x : \phi'(x) = 1] \geq c_1$ and $Pr[\exists x_1 \neq x_2 : \phi'(x_1) = \phi'(x_2) = 1] \leq c_2$. Clearly, the second equation suggests that we want some sort of pairwise independence.

Definition 1 *A family $H = \{h : \{0,1\}^n \to \{0,1\}^m\}$ is pairwise independent if*

$$\forall x_1 \neq x_2 \in \{0,1\}^n, \forall y_1, y_2 \in \{0,1\}^m, Pr_{h \in H}[h(x_1) = y_1 \wedge h(x_2) = y_2] = \frac{1}{2^m} \cdot \frac{1}{2^m}$$

For example, consider $H = \{h_{A,b} : A \in \{0,1\}^{m \times n}, b \in \{0,1\}^m\}$, with $h_{A,b}(x) = Ax + b$. This is a pairwise independent family (exercise). This uses only $mn + m$ random bits to pick a random element of the family$H$.

We claim that using h from H suffices for the reduction. For x_1, x_2 satisfying assignments for ϕ, $x_1 \neq x_2$, define the bad event $B_{x_1,x_2} =$

- 1 if $h(x_1) = h(x_2) = 0$
- 0 otherwise

By pairwise independence of H, $Pr[B_{x_1,x_2} = 1] = \frac{1}{4^m}$. Also, $Pr[h(x_1) = 0] = \frac{1}{2^m}$.

Define $G_{x_1} = 1$ if $h(x_1) = 0$ but $h(x_2) \neq 0 \forall x_2 \in \phi^{-1}(1)$. This is the good event, and we want it to occur for some x_i (this implies a unique satisfying assignment x_i for ϕ').

Then

$$Pr[G_{x_1} = 1] \geq Pr[h(x_1) = 0] - Pr[\bigcup_{x_1,x_2} “B_{x_1,x_2} = 1''] $$

$$\geq Pr[h(x_1) = 0] - \sum_{x_2} Pr[B_{x_1,x_2} = 1] \geq \frac{1}{2^m} - \frac{M-1}{4^m}$$

Observe that G_{x_1} and G_{x_2} are mutually exclusive. Thus the union bound is now the union equality:

$$Pr[\exists x_1 : G_{x_1} = 1] = \sum_{x_1} Pr[G_{x_1} = 1] \geq M(\frac{1}{2^m} - \frac{M-1}{4^m}) \geq \frac{1}{2}$$

where the last inequality holds if $2^{m-3} < M < 2^{m-2}$. Thus, with constant probability (given that m was guessed correctly), we get a USAT formula of the required type.

To recap, the reduction is the following:

- Pick $j \in_R \{0, \ldots, n\}$
- Set $m = j + 3$
- Pick $A \in \{0,1\}^{m \times n}, b \in \{0,1\}^m$
- Output $\phi'(x) = \phi(x) \wedge [h_{A,b}(x) = 0]$

4 Next time

Complexity of Counting Problems