

Computational Complexity Theory – Course Notes

CCT Team

December 11, 2006

1 Lecture 6/12-2006

1.1 PCPs and gap creating reductions

Theorem 1. $\text{EXP} \subseteq \text{P/poly} \Rightarrow \text{EXP} = \text{MA}$

Proof. (sketch) Merlin sends Arthur PCP of the existence of accepting transcript of EXP-machine on input of interest. \square

It seems that Arthur may need exponential time. But this can be repaired by redoing the proof of the PCP-theorem.

Definition 2. *Non-adaptive PCP: The verifier must pose all questions to the oracle before getting an answer.* $\text{PCP}_{\text{n.a.}}(r(n), q(n))$ is the corresponding set of languages.

Proposition 3. $\text{PCP}(r(n), q(n)) \leq \text{PCP}_{\text{n.a.}}(r(n), 2^{q(n)})$

Proof. Some bit has to be read first. Depending on this bit another bit is read. We can construct a tree expressing this. This tree will have height $q(n)$ and a total of $2^{q(n)} - 1$ nodes. \square

Corollary 4. $\text{NP} = \text{PCP}_{\text{n.a.}}(O(\log(n)), O(1))$

Definition 5. *Gap creating reduction for SAT:*

- $f \in \text{SAT} \Rightarrow r(f)$ is true.
- $f \notin \text{SAT} \Rightarrow$ any assignment must falsify at least ϵ fraction of clauses.

Proposition 6. *A gap creating reduction from 3SAT to MAXkSAT is the same as a $\text{PCP}_{\text{n.a.}}(O(\log(n)), k)$ for 3SAT.*

Proof. First we show that given a $\text{PCP}_{\text{n.a.}}(O(\log(n)), k)$ and a 3SAT-instance f we can construct a MAXkSAT instance. The instance will be constructed as follows:

- The variables will be locations in the proof string, (y_1, y_2, \dots, y_l) .
- For the clauses we for each setting of the random bits look at the bits queried. We define $g : \{0, 1\}^k \rightarrow \{0, 1\}$ so that $g(y_{i_1}, y_{i_2}, \dots, y_{i_k}) = 1$ if and only if the verifier accepts when these are the bits read. Take the CNF for g and add corresponding clauses.
 - If f is satisfiable, the MAX k SAT instance is satisfiable.
 - If f is not satisfiable, for any assignment to y and at least half the settings of the random bits, one clause will be false; i.e., at least $\frac{1}{2} \cdot \frac{1}{2^k}$ fraction of clauses. So we have a gap as intended.

If we on the other hand are given a gap creating reduction and a 3SAT-instance f , we do the following: The verifier (V) and the proof preparer (P) computes $f' = r(f)$ on input. P presents satisfying assignment to f' . To verify V picks random clause and checks that this clause has been satisfied. To get error-probability $\frac{1}{2}$ just repeat. \square

Corollary 7. *For some constant k there is no polynomial time approximation scheme for MAX k SAT unless $\mathbf{P}=\mathbf{NP}$.*

Corollary 8. *Same for MAX3SAT and MAX2SAT.*

Proof. The standard reductions from SAT to MAX3SAT and to MAX2SAT are gap preserving. \square

Definition 9 (Gap preserving reduction). *To each maximization problem associate a promise problem:*

- $L =$ the instances where the optimal solutions has value $\geq \alpha$.
- $U-L =$ the instances where the optimal solutions has value $\leq (1 - \epsilon)\alpha$.

A gap preserving reduction is a reduction from one such problem to another.

- $L \rightarrow L'$
- $U-L \rightarrow U'-L'$

Theorem 10 (Håstad). *If $\mathbf{P} \neq \mathbf{NP}$, MAX3SAT has no $\frac{8}{7} - \epsilon$ approximation algorithm. Also holds for MAXE3SAT.*

1.2 Derandomization

In computation where more than one part is involved, Shamir's theorem, the PCP theorem, and cryptography has shown that randomization is very powerful.

1.2.1 Peter uses slides