

# Computational Complexity Theory - Course Notes

CCT Team

December 5, 2006

## 19 Lecture 19, 1/12-2006

### 19.1 PCP Theorem (continued)

Ingredients in improvement the from  $\mathbf{NP} = \mathbf{PCP}[\text{polylog } n, \text{polylog } n]$  to  $\mathbf{NP} = \mathbf{PCP}[O(\log n), O(1)]$ :

- More compact proofs: multilinear extension  $\rightarrow$  low-degree extension: For  $a : \{0, 1\}^m \rightarrow \{0, 1\}$  we group the input by words of length  $k$  so  $a : (\{0, 1\}^k)^{m/k} \rightarrow \{0, 1\}$ . We can think of  $\{0, 1\}^k \simeq \{0, 1, \dots, 2^k - 1\}$ , but actually it is  $\text{GF}(2^k)$ .  
Claim:  $\exists$  polynomial  $\tilde{a}$  of total degree  $2^k \frac{m}{k}$  so that  $\tilde{a}$  agrees with  $a$  on  $\text{domain}(a)$ .  $\tilde{a}$  is tabulated on a domain not much larger than  $\{0, 1, \dots, 2^k - 1\}^m$ , so the blow up is polynomial.  $\tilde{a}$  is a Reed-Müller Code, which is a locally checkable code, so we replace the multilinearity test with a low-degree test. This brings us to  $\mathbf{PCP}[O(\log n), O(\log n)]$ .
- Assume  $\mathbf{NP} \subseteq \mathbf{PCP}[O(\log n), \text{polylog } n]$ . We will use the composition: Convince  $V$  that he will be convinced if he checks the proof. This gives the sequence

$$\begin{aligned}\mathbf{NP} &\subseteq \mathbf{PCP}[O(\log n), \text{polylog } n] \\ \mathbf{NP} &\subseteq \mathbf{PCP}[O(\log n), \text{polylog } \log n] \\ \mathbf{NP} &\subseteq \mathbf{PCP}[O(\log n), \text{polylog } \log \log n] \\ &\vdots\end{aligned}$$

At the  $\mathbf{NP} \subseteq \mathbf{PCP}[O(\log n), \text{polylog } \log \log n]$  level we replace the Reed-Müller Code with a Hadamard Code which takes  $n \rightarrow 2^n$  and is checkable using 2 bit queries. We then get the wanted  $\mathbf{NP} \subseteq \mathbf{PCP}[O(\log n), O(1)]$ .

### 19.2 Improvement of Meier's theorem at the EXP-level

**Thm. 1** |

$$\mathbf{EXP} \subseteq \mathbf{P}/\text{poly} \Rightarrow \mathbf{EXP} = \mathbf{MA}$$

*Proof.* A PCP of "This tableau is an accepting tableau on input  $x$ " can be constructed in exponential time in  $|x|$ . By assumption there is a small circuit  $C$  so that  $C(i) = i$ 'th bit in this PCP. Merlin sends Arthur this  $C$ , Arthur can be convinced by "PCPing" using this  $C$ .  $\square$

*Note:* A technical modification may be needed to ensure that Arthur does not use exponential time in  $|\text{tableau}|$  to query  $C$ .

### 19.3 Theorem of non-approximability

Consider an optimization (assume maximization) problem. For each input  $x \in \{0, 1\}^*$  we have a set of feasible solutions  $F_x \subseteq \{0, 1\}^*$ . We want to find  $y \in F_x$  so that

$$\frac{\text{val}(y)}{\max_{y^* \in F_x} \{\text{val}(y^*)\}} \geq \alpha.$$

For non-approximability we (often) use a ‘‘Gap-creating reduction’’: Assume  $f \in 3\text{CNF} \xrightarrow{r}$  instance  $x$  such that

$$\begin{aligned} f \text{ satisfiable} &\Rightarrow \exists y \in F_x : \text{val}(y) \geq \beta \\ f \text{ not satisfiable} &\Rightarrow \forall y \in F_x : \text{val}(y) < \alpha\beta \end{aligned}$$

*Example:* Hamiltonian Cycle  $\rightarrow$  TSP.

The problem with gap-creating reductions is that they are difficult to construct, and it was open for 20 years whether  $\exists r : \text{SAT} \xrightarrow{r} \text{MAXCLIQUE}$ .

Using  $\mathbf{NP} \subseteq \mathbf{PCP}[\text{polylog } n, \text{polylog } n]$  we get a gap creating reduction from 3SAT to MAXCLIQUE running in quasi-polynomial ( $2^{\text{polylog } n}$ ) time:

We make the following reduction from  $f \in 3\text{SAT}$  to  $G = (V, E) \in \text{CLIQUE}$ : A transcript of the PCP-verifier  $M$  on  $f$  consists of

- the  $r$  random bits
- queries to the string
- bit values on the queried positions

We let

$$\begin{aligned} V &= \{\text{transcripts that make } M \text{ accept}\} \\ E &= \{(t_1, t_2) \mid t_1 \text{ and } t_2 \text{ could happen on the same tape}\} \end{aligned}$$

then

$$\begin{aligned} f \text{ satisfiable} &\Rightarrow \exists \text{ clique } C = \{\text{transcripts arising from communicating with} \\ &\quad \text{correct proof of } f(a) = 1 \text{ (for a fixed } a)\}, |C| = 2^r \\ f \text{ not satisfiable} &\Rightarrow \text{no clique of size } > \frac{2^r}{2} : \\ &\quad \text{suppose } \exists C, |C| > \frac{2^r}{2} \Rightarrow \\ &\quad \exists \text{ string } y \text{ making } V \text{ accept with probability } > \frac{1}{2} \end{aligned}$$

**Cor. 2** | Non-approximability of MAXCLIQUE

$$\neg(\mathbf{NP} \subseteq \mathbf{DTIME}(2^{\text{polylog } n})) \Rightarrow \nexists \text{ efficient approximation algorithm for MAXCLIQUE with } \alpha = 2$$

If we use the real PCP theorem, we get  $\mathbf{P} \neq \mathbf{NP}$  instead of  $\neg(\mathbf{NP} \subseteq \mathbf{DTIME}(2^{\text{polylog } n}))$  and by amplification,  $\alpha$  can not be a constant, since  $\alpha$  corresponds to the probability of the PCP-verifier.