**18.405J/6.841J Advanced Complexity Theory** April 12, 2001

# Lecture 17

*Lecturer: Dan Spielman* *Scribe: Ronnie Misra*

## Recap

Last time, we discussed the class of "Arthur/Merln games". Recall that:

**Definition 1** *$L \in AM(k(n))$ if $\exists$ a PTIME verifier $A$ and a polynomial $p(n)$ s.t.*
$x \in L \Longrightarrow \exists$ *a $p(n)$-prover $P$ s.t. $Pr[(A \longleftrightarrow P)$ accepts$] > \frac{2}{3}$*
$x \notin L \Longrightarrow \forall$ *$p(n)$-provers $P$, $Pr[(A \longleftrightarrow P)$ accepts$] < \frac{1}{3}$*

In this lecture, we will show some applications of IP, and discuss the class PCP.

## Graph Isomorphism

Although there are no known PTIME algorithms for graph isomorphism, we can show that ISO is probably not NP-complete:

**Theorem 2** *If ISO is NP-complete, then $\Sigma_3^P = \Pi_3^P$.*

**Proof** In the problem set, we showed that MA $\subseteq$ AM (proved as NP $\cdot$ BP $\cdot$ P $\subseteq$ BP $\cdot$ NP $\cdot$ P). In fact, the AM speedup theorem says that:
AM(k(n)) $\subseteq$ AM($\frac{k(n)}{2} + 1$)
$\Longrightarrow$ AM(k) $\subseteq$ AM $\forall$ constants k
$\Longrightarrow$ MAM $\subseteq$ AM
We also showed in the problem set that AM ( = BP $\cdot$ NP $\cdot$ P ) $\subseteq$ NP/poly.

ISO is NP-complete
$\Longrightarrow$ NISO is coNP-complete
$\Longrightarrow$ coNP $\subseteq$ MAM
$\Longrightarrow$ coNP $\subseteq$ AM
$\Longrightarrow$ coNP $\subseteq$ NP/poly
$\Longrightarrow \Sigma_3^P = \Pi_3^P$.
■

## Probabilistically checkable proofs

We would like to describe a "tough verifier" that accepts a proof that can be "spot checked".

**Definition 3** *A $(R(n), Q(n))$ verifier is a probabilistic, PTIME oracle TM $M^P$ that*

- *gets $R(n)$ random bits*
- *is limited to $Q(n)$ bits from its oracle $P$*

**Definition 4** *$L \in PCP(R(n), Q(n))$ if $\exists$ a $(R(n), Q(n))$ verifier $V$ s.t.*
$x \in L \Longrightarrow \exists\ \Pi$ *s.t. $Pr[V^{\Pi}(x)$ accepts$] = 1$*
$x \notin L \Longrightarrow \forall\ \Pi$, *$Pr[V^{\Pi}(x)$ accepts$] < \frac{1}{2}$*

Here, $\Pi$ is the probabilistically checkable proof (PCP). Note that we don't really think of $\Pi$ as an oracle. Instead, we are using OTM notation just as a way to show that V gets access to random bits of $\Pi$.

**Theorem 5** *NEXP = PCP(Poly(n), Poly(n))*

Even though proofs for languages in NEXP are exponentially long, randomized access allows such proofs to be verified in PTIME.

**Theorem 6** *NP = PCP(O(log n), O(1))*

As a consequence, we can use PCP to show that some approximation problems are NP-hard.

# 3SAT approximation is NP-hard

3SAT $\subseteq$ NP = PCP(O(log n), O(1))

$\Longrightarrow \exists$ a (O(log n), O(1)) verifier for 3SAT.

Assume V is non-adaptive, or that it flips R = O(log n) coins, then queries the proof at Q = O(1) places determined only by these random bits. (If V is adaptive, we can construct a non-adaptive version; this version may use exponentially more queries, but this is still a constant.)

On input $\Phi$, assume V gets random bits r $\in \{0,1\}^R$. Let $q_{r,1} \ldots q_{r,Q}$ denote the indices of the Q bits of $\Pi$ that V queries. Thus, if $\Pi = \Pi_1\Pi_2 \ldots \Pi_N$, V reads $\Pi_{q_{r,1}} \ldots \Pi_{q_{r,Q}}$.

For each r $\in \{0,1\}^R$, construct a small 3SAT instance $\phi_r$ on inputs $\Pi_{q_{r,1}} \ldots \Pi_{q_{r,Q}}$ and some auxilliary bits. Assume $\phi_r$ has at most S clauses. Since Q = O(1), S = O(1). $\phi_r$ should be satisfied when $\Pi_{q_{r,1}} \ldots \Pi_{q_{r,Q}}$ have values that would cause V to accept.

Let $\Phi' = \bigwedge_{r \in \{0,1\}^R} \phi_r$

$\Phi'$ is polynomial in the size of $\Phi$, and has inputs $\Pi_{q_{r,1}} \ldots \Pi_{q_{r,Q}}$ as well as all the auxilliary bits. $\Phi'$ also has some special properties:

- $\Phi \in$ 3SAT $\Longrightarrow \Phi' \in$ 3SAT, since V will accept $\Phi$ for any r $\in \{0,1\}^R$
- $\Phi \notin$ 3SAT $\Longrightarrow \underset{r \in \{0,1\}^R}{Pr}$[V rejects] $> \frac{1}{2}$

  $\Longrightarrow \forall\, \Pi$, at most $\frac{1}{2}$ of the $\phi_r$ clauses are satisfiable (from aux bits)

  $\Longrightarrow \forall\, \Pi$, $\forall\, aux$, there is at least one unsatisfied clause in at least $\frac{1}{2}$ of the $\phi_r$ clauses

  $\Longrightarrow$ at least a $\frac{1}{2S}$ fraction of the clauses of $\Phi'$ are unsatisfiable

  $\Longrightarrow$ the maximum fraction of satisfiable clauses of $\Phi'$ is (1 - $\frac{1}{2S}$) $\Longrightarrow$ approximating 3SAT within a factor of $\frac{1}{4S}$ is NP-hard, where S is some constant dependant on Q