6.841 Advanced Complexity Theory April 15, 2009

Lecture 19

Lecturer: Madhu Sudan *Scribe: Alex Arkhipov*

1. Review of Last Class

Last class we gave a formulation of Probabilistically Checkable Proofs as a coloring of a graph that satisfies certain constraints.

Definition 1.1. The *graph k-coloring problem* is as follows:

> Given a graph $G = (E, V)$, does there exist a coloring $\chi : V \to \{1, 2, \dots, k\}$ such that for each $(u, v) \in E$, $\chi(u) \neq \chi(v)$?

In generalized graph coloring, each edge restricts the coloring of its endpoints by an arbitary relation, described by an admissibility function Π.

Definition 1.2. The generalized *k-coloring problem* is as follows:

> Given a graph $G = (E, V, \Pi)$, which includes map $\Pi : E \times \{1, 2, \dots, k\} \times \{1, 2, \dots, k\} \to \{0, 1\}$, does there exist a coloring $\chi : V \to \{1, 2, \dots, k\}$ such that for each $e = (u, v) \in E$, $\Pi(e, \chi(u), \chi(v)) = 1$?

In a coloring χ, we say an edge $e = (u, v)$ is *invalid* if it does not satisfy the constraint $\Pi(e, \chi(u), \chi(v)) = 1$. The unsatisfiability $\mathrm{UNSAT}(G, \chi)$ is fraction of invalid edges in G, and the unsatisfiability $\mathrm{UNSAT}(G)$ of a graph is the minimum $\mathrm{UNSAT}(G, \chi)$ over all colorings χ.

Recall the Lemma that we wished to prove that would allow us to reduce the number of colors:

Lemma 1.3. *There exists a k and $\delta > 0$, so that for any K, there is a reduction function f from K-coloring instances to k-coloring instances so that for any G and $\tilde{G} = f(G)$,*

- *If $\mathrm{UNSAT}(G) = 0$, then $\mathrm{UNSAT}(\tilde{G}) = 0$.*
- *$\mathrm{UNSAT}(\tilde{G}) \geq \delta\, \mathrm{UNSAT}(G)$*

We'll first see look at a naive attempt to perform this reduction, and see how it can lead to unsatisfiability falling by more than a constant factor δ.

1.1. Attempt at reduction from K-coloring to 3-coloring.

To illustrate the obstacle to showing Lemma 1.3, we'll sketch a linear time reduction for standard K-coloring to standard k-coloring, with $k = 3$. We'll convert K-coloring instance G to a 3-coloring instance $\tilde{G}$ by replacing each edge of G with a gadget of $\tilde{G}$ that encodes the same restriction. However, we'll find that if $\mathrm{UNSAT}(G) = \varepsilon$, then $\mathrm{UNSAT}\left(\tilde{G}\right) \leq \frac{\varepsilon}{K}$, and thus cannot satisfy $\mathrm{UNSAT}(\tilde{G}) \geq \delta \mathrm{UNSAT}(G)$ for any constant δ.

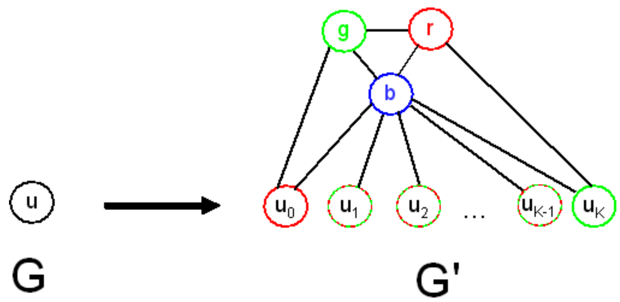

Figure 1.1. The conversion of a vertex of G to one of $\tilde{G}$. The vertices are marked with their possible colors.

1.1.1. *Construction of* $\tilde{G}$. Make three special nodes $\{r, g, b\}$ and connect them with edges, so that they must be different colors which we'll label red, green, and blue, which we will also call the three possible colors of the nodes. We may restrict the possible colors of a node in $\tilde{G}$ by connecting it to each of $\{r, g, b\}$ we want to exclude.

For each node $u \in G$, make $K+1$ nodes $u_0, \ldots u_K$ in $\tilde{G}$. Restrict them to be each red or green (by connecting each by an edge to the blue node), and furthermore restrict u_0 to be red and u_K to be green. Then, in any coloring of $\tilde{G}$, there is some first node u_j that marks a switch from red to green; i.e. the first $i \in \{1, \ldots, K\}$ so that u_{j-1} is red and u_j is green. (We may assume that u_i is red for $i < j$ and green for $i \geq j$, as we'll see that allowing additional "switches" won't give any advantage in coloring the graph). To such a coloring of the u_i nodes in G, we associate the node $u \in G$ being colored with color i.

We need to enforce the restriction that for each edge $(u, v) \in G$, u and v have different colors. Correspondingly in $\tilde{G}$, we put in gadgets to ensure that u and v don't both switch colors at the same value, that there for no $i \in \{1, \ldots, K\}$ so that u_{i-1} is red and u_i green, and also v_{i-1} red and v_i is green. To do so, for each i, we put in a gadget with two additional nodes x_{uvi} and y_{uvi}, which are edge-connected to each other and to u_{i-1} and u_i, and to v_{i-1} and v_i, respectively.

If both u and v have color j, then both the added node x_{uv} and y_{uv} can't be red or green, and are forced to be blue, which is disallowed. In any other case, valid colorings exist for the two added nodes. So, we have encoded the restriction for the edge $(u, v) \in G$.

1.1.2. *Analysis of Reduction.* From the construction of $\tilde{G}$, it's easy to see that $\tilde{G}$ is 3-colorable if and only G is K-colorable. How does the unsatisfiability of G compare to that of $\tilde{G}$?

Suppose the best K-coloring of $\tilde{G}$ fails on d edges, so that $\mathrm{UNSAT}(G) = d/|G|$ (where the size of a graph is its number of edges). Then, for each invalid edge $(u, v) \in G$, say with $\chi(u) = \chi(v) = i$, we may color the (u, v) edge gadget in $\tilde{G}$ so that is valid for all but one edge by allowing both x_{uvi} and y_{uvi} to be blue. So, $\tilde{G}$ fails on only d edges and $\mathrm{UNSAT}\left(\tilde{G}\right) \leq d/\left|\tilde{G}\right|$. (It is possible that $\mathrm{UNSAT}(\tilde{G})$ is smaller, if multiple invalid edges in G share a vertex u, illegally coloring $u_0, \ldots, u_K \in \tilde{G}$ gives only one invalid edge in $\tilde{G}$.) Since $\left|\tilde{G}\right| / |G| = \Theta(K)$,

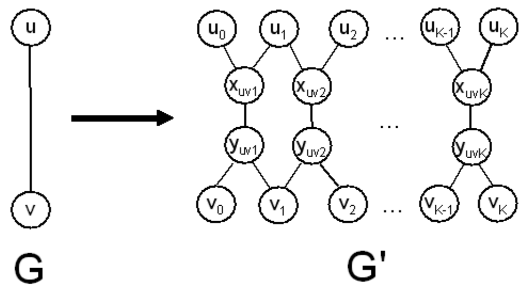

Figure 1.2. Each edge of G is repaced by a gadget in $\tilde{G}$ that enforces the constraint the u and v are different colors by enforcing the fact that the sequences of colors u_i and v_i cannot switch at the same point.

UNSAT($\tilde{G}$) $= O\left(\frac{1}{K}\right)$ UNSAT(G). So, we fail to produce an at most constant reduction of unsatisfiability.

In the hopes of finding a better reduction, we look at a way to construct exponentially long PCP's that we will use to create a better K-coloring to k-coloring reduction.

2. Quadratic Equation Solvability

We give a scheme for giving a Probabilistically Checkable Proof of an NP-complete problem that is exponentially-sized and requires a constant number of queries. This scheme is due to Arora, Lund, Motwani, Sudan, and Szegedy.

2.1. **Problem Definition.** The Quadratic Equation Solvability Problem is like SAT, in that it asks whether a given formula, consisting of the AND of m clauses, is satisfiable. except each of these clauses may be an arbitrary second-degree polynomial in the variables $x_1, \ldots, x_n$.

Definition 2.1. The *Quadratic Equation Solvability Problem* takes as input a formula ϕ of n Boolean variables $x_1, \ldots, x_n$ that is the AND of m clauses that are degree-2 polynomials $p_1, \ldots, p_m$ of the variables $x_1, \ldots, x_n$ over $\mathbb{Z}_2$, and asks whether ϕ has a satisfying assignment $\bar{a}$ (so that $p_i(\bar{a}) = 0$ for each i).

Since in a circuit, we may express the relations given by a AND, OR, and NOT gates by a quadratic expressions, this problem is at least as hard a circuit-SAT, and thus NP-hard. It is NP hard, since it's easy to verify a satisfying assignment.

2.2. **PCP Scheme.** Let Z_1 be sets of all homogenous linear polynomials in n variables $x_1 \ldots, x_n$, and let Z_2 be all homogenous quadratic polynomials, respectively. The PCP will contain information about a satisfying assignment of the QES formula in the form of two tables: T_1 and T_2, of size 2^n and 2^{n^2}, that are claimed to be list the values of Z_1 and Z_2, respectively, on some single satisfying assignment $\bar{a}$. However, since the prover may cheat and put any values in the tables, to check that the proof is valid, it is up to the verifier to see that there exists an $\bar{a}$ so that the following three conditions hold:

(1) For each linear polynomial L, $T_1(L) = L(\bar{a})$.
(2) For each quadratic polynomial Q, $Q_1(L) = Q(\bar{a})$

(3) For each of the polynomial clauses P_i, $P_i(\bar{a}) = 0$.

The first two check the validity of the tables, and then the third trusts the values in the tables to be valid (i.e. to contain the value of the state polynomial as $\bar{a}$) in order to check that the assignment claimed is satisfying.

Since we, as the verifier, may only make a constant number of queries, we cannot confirm these conditions with certainty. So, we need to make a scheme so that if the QES formula cannot be satisfied, an adversary making the tables is forced to commit to a large number of discrepancies that we can catch.

We will show that the following verification algorithm works:

Algorithm 2.2. *We verification the PCP in three verification steps, corresponding to the three properties given above:*

(1) *Pick $L_1, L_2 \in Z_1$ at random, and check that*

$$T_1(L_1) + T_1(L_2) = T_1(L_1 + L_2)$$

(all computations are modulo 2).

(2)

(a) *Pick $Q_1, Q_2 \in Z_2$ at random, and check that*

$$T_2(Q_1) + T_2(Q_2) = T_2(Q_1 + Q_2)$$

.

(b) *Pick $L_1, L_2 \in Z_1$ and $S \in Z_2$ at random, and check that*

$$T_2(S + L_1 L_2) = T_2(S) + T_1(L_1)\, T_1(L_2)$$

.

(3) *Pick $r \in \{0,1\}^m$ at random and let $A_r = \sum r_j P_j$. Write*

$$A_r = Q_r + L_r + C_r$$

with $Q_r \in Z_2$, $L_r \in Z_1$, and C constant. Pick $S \in Z_2$ at random and check that

$$T_2(Q_r + S) - T_2(S) + T_1(L_r) + C_r = 0$$

.

If any of these checks rejects, then reject. Otherwise, accept.

This scheme uses a constant number of queries (specifically ten).

We would like to show the following.

Theorem 2.3. *The given verification scheme V has the following properties:*

- *If ϕ is satisfiable, then for tables T_1 and T_2 as described, then V returns YES.*
- *If ϕ is not satisfiable, then for any tables T_1 and T_2, then V returns YES with probability $F \leq \frac{8}{9}$.*

It should be easy to see that the first part is true, since giving tables $T_1(L) = L(\bar{a})$, $Q_1(L) = Q(\bar{a})$ for a valid satisfying assignment $\bar{a}$ will pass the three checks regardless of the random choices made.

For the remainder of the section, will put a bound on F, the probability that we are falsely led to accept when ϕ is not satisfiable.

Define δ_1 and δ_2 be the fraction of entries in T_1 for which $T_1(L) \neq L(\bar{a})$ (we'll call these invalid entries), minimized over all satisfying assignments $\bar{a}$, and let δ_2 be fraction of invalid entries of T_2.

$$\delta_1 = \min_{\bar{a}} \left[\Pr_L \left[T_1(L) \neq L(\bar{a}) \right] \right]$$

$$\delta_2 = \min_{\bar{a}} \left[\Pr_Q \left[T_2(Q) \neq Q(\bar{a}) \right] \right]$$

We will also use another discrepancy measure that says how nonlinear T_2 is

$$\delta_3 = \min_{(c_{ij})} \left[\Pr_{L_1,L_2} \left[T_2(c_{11},\ldots,c_{nn}) \neq \sum_{i,j} c_{ij} b_{ij} \right] \right]$$

Define F_1, F_{2a}, F_{2b}, and F_3 to be maximum probability that verifications steps (1), (2a), (2b), and (3), respectively, accept, given that ϕ has no satisfying assignment. Then, $F \leq F_1 F_2 F_3$. We'll show that $F \leq 8/9$ by bounding F_1, F_2, and F_3 in terms of δ_1, δ_2, and δ_3.

Lemma 2.4. *The probability F_1 that verification step (1) accepts has $F_1 \leq 1 - \frac{2}{9}\delta_1$.*

Proof. For any two polynomials L_1 and L_2, it must be true that $(L_1 + L_2)(\bar{a}) = L_1(\bar{a}) + L_2(\bar{a})$ (taken modulo 2). In the first verification step, we check that T_1 respects this property on a single instance of two linear functions. By a theorem that we won't prove, for any table T_1 and any $\bar{a}$

$$\Pr_{L_1,L_2} \left[T_1(L_1) + T_1(L_2) \neq T_1(L_1 + L_2) \right] \geq \frac{2}{9}\delta_1$$

So, if the table has at least fraction δ_1 invalid entries, this probability that we will reject is at least $\frac{2}{9}\delta_1$. So, the chance of being fooled $F_1 \leq 1 - \frac{2}{9}\delta_1$. □

Lemma 2.5. *The probability F_{2a} that verification step (2a) accepts has $F_{2a} \leq 1 - \frac{2}{9}\delta_3$.*

Proof. As in the verification of (1), check the linearity condition $T_2(Q_1) + T_2(Q_2) = T_2(Q_1 + Q_2)$. If we each $Q \in Z_2$ as a matrix of coefficients (c_{ij}), and think of T_2 as a function of these coefficients, the linearity property is satisfied for all Q_1 and Q_2 if and only if the function $T_2(c_{11},\ldots,c_{nn})$ is a linear function of the coefficients (c_{ij}), i.e. there exists coefficients b_{ij} so that

$$Q(\bar{a}) = \sum_{i,j} c_{ij} b_{ij}$$

By the theorem we used without proof in the verification for (1), if T_2 does not satisfy this linearity property for some fraction of entries

$$\delta_3 = \min_{(b_{ij})} \left[\Pr_{(c_{ij}) \in \{0,1\}^{n \times n}} \left[T_2(c_{11},\ldots,c_{nn}) \neq \sum_{i,j} c_{ij} b_{ij} \right] \right]$$

, then the probability of us catching the error by the linearity check

$$\Pr_{Q_1,Q_2} \left[T_2(Q_1) + T_2(Q_2) \neq T_2(Q_1 + Q_2) \right] \geq \frac{2}{9}\delta_3$$

So, the probability this step accepts is

$$F_{2a} \leq 1 - \frac{2}{9}\delta_3$$

$\square$

Lemma 2.6. *The probability F_{2b} that verification step (2b) accepts has $F_{2b} \leq 1 - \delta_2\left(2\delta_1 + \delta_3 + \frac{3}{4}\right)$.*

Proof. We write out a polynomial $Q \in Z_2$ in terms of its coefficents (c_{ij}) as

$$Q(\bar{a}) = \sum_{i,j} c_{ij} a_i a_j$$

Using the coefficient representation from the verification for (1b), we associate Q with its matrix of n^2 coefficients $C = (c_{ij})$ so that

$$Q(\bar{a}) = \sum_{i,j} c_{ij} b_{ij}$$

So, to check that Q is a quadratic polynomial, we need to check that the row vector $\bar{a} = (a_1, a_2, \ldots, a_n)$ has $b_{ij} = a_i a_j$ for each i, j, or equivalently that the matrix $B = \bar{a}^T \bar{a}$.

We do this using a well-known probabilistic test of matrix equality by picking two random (row) vectors $x, y \in \{0,1\}^n$ and checking that $xBy^T = x\bar{a}^T \bar{a} y^T$. If matrices B and $\bar{a}^T\bar{a}$ differ in some entry i, j, then with probability at least $\frac{1}{4}$, $xBy^T \neq x\bar{a}^T\bar{a}y^T$, since the equality depends on whether $x_i y_j = 1$ or 0, which happen with probabilities $1/4$ and $3/4$. The condition $xBy^T = \left(x\bar{a}^T\right)\left(\bar{a}y^T\right)$ is equivalent to

$$Q_{xy}(\bar{a}) = L_x(\bar{a})\, L_y(\bar{a})$$

where L_x and L_y are linear functions with coefficients given by x and y, and Q_{xy} is the quadratic polynomial with $c_{ij} = x_i y_j$.

We check this by querying the tables, using the indirect query $T_2(L_x L_y + S) - T_2(S)$ with random S for $T_2(L_x L_y)$. The check

$$T_2(S + L_1 L_2) = T_2(S) + T_1(L_1)\, T_1(L_2)$$

will be false if T_2 reports an incorrect value for $S + L_1 L_2$ or S, which happens with probability at least δ_2, and none of the following problems occur:

- $T_1(L_1) \neq L_1(\bar{a})$ (probability δ_1)
- $T_1(L_2) \neq L_2(\bar{a})$ (probability δ_1)
- $\sum_{i,j} c_{ij} b_{ij} = T_2(c_{11}, \ldots, c_{nn})$ (probability δ_3)
- $xBy^T = x\bar{a}^T\bar{a}y^T$ (probability $3/4$ if $B \neq \bar{a}^T\bar{a}$)

So, the probability F_{2b} of rejecting at this step satisfies

$$F_{2b} \leq 1 - \delta_2\left(2\delta_1 + \delta_3 + \frac{3}{4}\right)$$

$\square$

Lemma 2.7. *If ϕ has no satisfying assignment, the probability F_3 that verification step (3) accepts has $F_3 \leq \frac{1}{2} + \delta_1 + 2\delta_2$.*

Proof. If ϕ has no satisfying assignment, then for any $\bar{a}$, $P_i(\bar{a}) = 1$ for some i. With a constant number of queries, we can't even check whether the tables claim that $P_i = 0$ for each of the m polynomials P_i, or even a substantial fraction. However, we can use a trick from the Razborov-Smolensky proof of the circuit lower bound of evaluating a random linear combination $A_r = \sum r_j p_j$, which has $A_r(\bar{a}) = \sum r_j p_j(\bar{a})$, for random $r \in \{0,1\}^m$. Since the coefficient r_i of a failing polynomial P_i with $P_i(\bar{a}) = 1$ is equally likely to be 0 or 1, so

$$\Pr_r [A_r(\bar{a}) = 0] = \frac{1}{2}$$

The polynomial A_r may not be homogenous. We can uniquely decompose it as $A_r = Q_r + L_r + C_r$, with $Q_r \in Z_2$, $L_r \in Z_1$, and C_r constant. Then, checking if $A_r(a) = 0$ equates to checking that $Q_r + L_r + C_r = 0$.

We do this by checking whether $T_2(Q_r + S) - T_2(S) + T_1(L_r) + C_r = 0$. Since $A_r(\bar{a}) = Q_r(\bar{a}) + L_r(\bar{a}) + C_r(\bar{a})$ and $(Q_r + S)(\bar{a}) - S(\bar{a}) = Q_r(\bar{a})$, if ϕ is not satisfiable, by union bound

$$\begin{aligned}
F_1 &= \Pr_r [T_2(Q_r + S) - T_2(S) + T_1(L_r) + C_r \neq 0] \\
&\leq \Pr_r [A_r(\bar{a}) = 0] + \Pr_r [T_1(L_r) \neq L_r(\bar{a})] + \Pr_{r,S} [T_2(Q_r + S) \neq (Q_r + S)(\bar{a})] + \Pr_S [T_2(S) \neq S(\bar{a})] \\
&= \frac{1}{2} + \delta_1 + 2\delta_2.
\end{aligned}$$

□

Proof. Note that $F \leq F_1 F_{2a} F_{2b} F_3$ (this is an inequality, since in F we have the restriction of using the same T_1 and T_2 for all the verification steps). So, from the Lemmas,

$$F \leq \left(\left[1 - \frac{2}{9}\delta_1\right]\right)\left(\left[1 - \frac{2}{9}\delta_3\right]\right)\left(\left[1 - \delta_2\left(2\delta_1 + \delta_3 + \frac{3}{4}\right)\right]\right)\left(\left[\frac{1}{2} + \delta_1 + 2\delta_2\right]\right)$$

where, $[x]$ denotes $\min(x, 1)$, since each of the probabilities F_1, F_{2a}, F_{2b}, F_3 are at most 1. □

A computer-aided calculation of the maximum value F over $\delta_1, \delta_2, \delta_3 \in [0,1]$ gives a maximum value of $\frac{8}{9}$. This proves the theorem.

3. Next class

Next class, we'll use a PCP scheme like the one shown to create a reduction from K-coloring to k-coloring which has $\mathrm{UNSAT}(\tilde{G}) \geq \delta \mathrm{UNSAT}(G)$ for constant δ, thus satisfying Lemma 1.3. Proving this Lemma will complete the proof for the existence of a polynomial-size PCP.