CS 880: Advanced Complexity Theory 3/30/06

Lecture 22: Constraint Graph Alphabet Reduction

Instructor: Dieter van Melkebeek Scribe: Bess Berg and Jurgen Van Gael

The proof of the PCP theorem which we have discussed relies on the gap amplification of constraint graphs (CG from now on): we build a CG transformation such that if the original CG is satisfiable then the transformed CG is satisfiable, and if the original CG is unsatisfiable then the number of unsatisfied constraints in the transformed CG goes up by a constant factor. Remember from last time that the alphabet size of the transformed CG is enlarged; today we want to address this issue: we will design another transformation that can reduce the size of the alphabet in a CG and only deteriorating the gap by a constant factor.

1 Proof idea

We will prove the following lemma.

Lemma 1. *There exists an alphabet Σ' and a constant $\delta > 0$ such that in polytime we can transform any constraint graph G over alphabet Σ into a constraint graph G' over the alphabet Σ' such that,*

- $UNSAT(G) = 0 \Rightarrow UNSAT(G') = 0$,
- $UNSAT(G') \geq \delta \cdot UNSAT(G)$,
- $SIZE(G') = O(f(|\Sigma|)) \cdot SIZE(G)$.

The idea of the proof is a concept called *composition*. We will view each of the edge conditions c_e of the original graph G as a boolean function φ_e on a large but constant (depending on $|\Sigma|$) number of variables. Each vertex in G can take on $|\Sigma|$ possible values, and we can encode each value with $\log |\Sigma|$ Boolean variables; then φ_e is just a translation of the original edge constraint in terms of the encoding of the boolean variables. Now the main idea is to transform each boolean formula φ_e into a CG G_{φ_e} over the alphabet Σ' and satisfying the desired gap property, so that our final graph G' is the union of all of these G_{φ_e}.

Several issues come up however:

1. Circularity: If we look at the method for gap amplification outlined above, it seems that our last step relies on the correctness of this method! However, since our φ_e are of constant size, it is sufficient to prove that constant size CG's with the required gap exist.

2. Consistency: If we evaluate an assignment on a particular constraint c_e, it is not sufficient to check whether the assignment satisfies φ_e. We must also check whether all the 'local' assignments are consistent across all constraints. The solution to this problem will be to consider a global assignment on all boolean variables, and verify for each constraint e that the relevant portion of the global assignment satisfies φ_e.

3. For any $e = (u, v)$ and φ_e, the vertices in G_{φ_e} can take on only a fixed number $|\Sigma'|$ of values. $|\Sigma'|$ is independent of $|\Sigma|$, and therefore independent of the number of Boolean variables corresponding to u and v. So each constraint in G_{φ_e} must be insensitive to many of the Boolean variables corresponding to u and v, so that changing the value of one such variable may only affect a constant number of constraints. Then if we wish G_{φ_e} to reject with probability at least δ we must impose the condition that the given assignment is far in Hamming distance from any satisfying assignment. We will be able to achieve something like this by using a good error-correcting code.

2 Assignment Testers

For now we define an assignment tester without explaining how to actually construct such a thing. We will discuss the construction a little later.

Definition 1 (Assignment Tester)**.** *An assignment tester (AT) is a transformation that takes a Boolean formula φ on variables V and outputs a CG G_φ on variables V' with $V \subseteq V'$ such that for any assignment σ to V:*

- $\varphi(\sigma) = 1 \Rightarrow UNSAT(G_{\varphi|V \leftarrow \sigma}) = 0$,
- $UNSAT(G_{\varphi|V \leftarrow \sigma}) \geq \rho \cdot h(\sigma, SAT(\varphi))$,

where $SAT(\varphi)$ is the set of all satisfying assignments to φ and $h(x, y)$ is the relative hamming distance between x and y.

We now wish to make use of such an AT over a fixed alphabet Σ'. First we will encode the values of the original alphabet Σ using a good error-correcting code $C : \Sigma \to \{0, 1\}^m$ with relative distance d (m is just some function of $|\Sigma|$). For any vertex u of the original CG G, let $[u]$ denote the set of m Boolean variables corresponding to the encoded value of u. Now, for each edge $e = (u, v)$ consider a formula φ_e on variable set $[u] \cup [v]$. Recall that c_e is the constraint function for edge e, and define φ_e by

$$\varphi_e \equiv \begin{cases} \sigma'([u]) \text{ is a valid codeword in C, and} \\ \sigma'([v]) \text{ is a valid codeword in C, and} \\ c_e\left(C^{-1}(\sigma'([u])),\ C^{-1}(\sigma'([v]))\right) = 1 \end{cases}$$

We apply the AT to each such φ_e to obtain the CG G_{φ_e} on a variable set containing $[u]$ and $[v]$. Our final G' is the union of these G_{φ_e}:

$$G' = \bigcup_e G_{\varphi_e}.$$

Note that each G_{φ_e} involves common variables. In particular, if $e_1 = (u, v)$ and $e_2 = (u, w)$ then both $G_{\varphi_{e_1}}$ and $G_{\varphi_{e_2}}$ contain the vertices $[u]$.

To prove that G' satisfies the gap property, first consider the case where G is satisfiable, say by some satisfying assignment σ. Then by the definition of the AT, G' will also be satisfiable by some assignment σ' such that $\sigma'([u]) = C(\sigma(u))$. Next, assume G is not satisfiable. We need to show that any assignment σ' to G' violates many constraints. To see this, fix some σ' and construct an assignment σ to the original graph G. We will set $\sigma(u)$ to be an element $s_u \in \Sigma$ such that

$C(s_u)$ is as close as possible to $\sigma'([u])$; that is, given the assignment $\sigma'([u])$, do error-correcting on $\sigma'([u])$ and then decode to obtain $\sigma(u)$. Now, σ cannot satisfy G because we have said that G is unsatisfiable, so let $e = (u, v)$ be some edge in G such that the formula φ_e is not satisfied under σ. Every satisfying assignment to φ_e is a pair of valid codewords from C, say (C_u, C_v). Either the Hamming distance between C_u and $\sigma'([u])$ must be at least $\frac{1}{2}dm$, or the Hamming distance between C_v and $\sigma'([v])$ must be at least $\frac{1}{2}dm$, otherwise by the definition we gave for σ we would have $(\sigma(u), \sigma(v)) = (C_u, C_v)$. Then we have

$$h\left(\sigma'([u] \cup [v]), SAT(\varphi_e)\right) \geq \frac{d}{4}.$$

Then by the properties of the AT we get

$$UNSAT(G') \geq \frac{\rho \cdot d}{4} \cdot UNSAT(G),$$

and setting $\delta = \frac{\rho d}{4}$ we obtain our desired result,

$$UNSAT(G') \geq \delta \cdot UNSAT(G).$$