(67902) Topics in Theory and Complexity	January 3rd, 2007

Lecture 10a

Lecturer: Irit Dinur — *Scribe: Yonatan Amit, Ezra Hoch*

1 Overview

Last lesson we discussed constraint graphs. Our aim was to prove the following theorem. Previously, we proved that the PCP theorem easily follows. That is, by proving the following theorem we conclude that the PCP theorem is proved.

Theorem 1 *There exists Σ_0, s.t. for all Σ there is some $c > 0$ and $\alpha > 0$ s.t. given a constraint graph $G = \langle (V, E), \Sigma, \mathcal{C} \rangle$ we can generate $G' = \langle (V', E'), \Sigma_0, \mathcal{C}' \rangle$ that maintains*

- $size(G') \leq c \cdot size(G)$ *where size is defined as* $(|V| + |E|)$.
- $\textsc{Unsat}(G) = 0 \Rightarrow \textsc{Unsat}(G') = 0$
- $\textsc{Unsat}(G) > 0 \Rightarrow \textsc{Unsat}(G') = \min \{\alpha, 2 \cdot \textsc{Unsat}(G)\}$

We now advance to proving this theorem (which will take a long time). Proof sketch:

$$G \to \textsc{Prep}(G) = H_1 \to H_1^t = H_2 \to H_2 \circ \Phi$$

1. (Lemma 2 next) $G \to \textsc{Prep}(G) = H_1$. This stage is a beautification of the graph: The $\textsc{Unsat}$ level decreases by a constant factor , Σ stays constant, but the combinatorial properties are better: H_1 will be a d-regular expander with constant d.
2. $H_1 \to (H_1)^t = H_2$ The $\textsc{Unsat}$ level will increase by a constant (as large as we wish). Σ will increase (regrettably).
3. $H_2 \to H_2 \circ \Phi = H_3 = G'$. We correct the alphabet $\Sigma \to \Sigma_0$, and $\textsc{Unsat}$ changes only by a constant factor.

We emphasize that each phase is linear in the size of the input. In stage 2 we can increase the $\textsc{Unsat}$ as we wish, thus negating the decrease from stages 1 and 3.

To sum up, we take a general constraint graph G, we make it a d-regular expander, so it is easy to work with. This operation reduces $\textsc{Unsat}$. Then we enlarge the $\textsc{Unsat}$ on the expense of the size of Σ. Finally, in the last stage, we reduce the size of Σ, changing $\textsc{Unsat}$ by only a constant factor. And we have a graph G' that is linear in the size of G but has twice as much $\textsc{Unsat}$ (if we didn't reach the maximal value α).

2 First Operation – Preprocessing

In this lesson we will discuss mainly the first operation, marked by $\textsc{Prep}(G)$. Following is a lemma that claims operation $\textsc{Prep}(G)$ does what it needs.

Lemma 2 *There exists constants $\beta_I > 0$ and $0 < \lambda < d$ and c_I s.t. given a constraint graph $G = \langle (V, E), \Sigma, \mathcal{C} \rangle$ we can compute a constraint graph $G' = \textsc{Prep}(G)$ in polynomial time such that*

- G' *is d-regular. (Constant for us)* $\lambda(G') \leq \lambda$
- G' *has the same alphabet as* G, *and* $size(G') \leq c_I \cdot size(G)$

- $\beta_I \cdot \textsc{Unsat}(G) \leq \textsc{Unsat}(G') \leq \textsc{Unsat}(G)$

We now move to the proof of this lemma. We will actually do so in two phases

$$G \to \textsc{Prep}_1(G) \to \textsc{Prep}_2(\textsc{Prep}_1(G)).$$

$\textsc{Prep}_1$ will turn G into a d-regular, and $\textsc{Prep}_2$ will make this graph an expander.

3 First Preprocessing Step - "Regularization"

We start be defining the first preprocessing step. The idea is to "blow up" each vertex of G into a cloud of vertices whose size is the degree of that vertex. So high degree vertices get large clouds and low degree vertices get small clouds. Each edge now connects two vertices from two different clouds, such that the degree of each vertex is now 1. Furthermore, we interconnect vertices inside a cloud by expander edges.

Definition 3 *Let $G = \langle (V, E), \Sigma, \mathcal{C} \rangle$. We define $H = \textsc{Prep}_1(G)$ as the following constraint graph:*

First, we replace each vertex v in G with a cloud representing the neighbors of v. Namely, for each $v \in V$ let $[v] = \{(v, e) \,|\, e$ is an edge of $v\}$. The vertex set V' of H is defined as $V' = \bigcup_v [v]$.

For each $v \in V$ we denote its degree in G by d_v and let X_v be a (d_v, d, λ)-expander [1]*. Each such expander is used to connect the edges inside the cloud $[v]$.*

We define

$$E_1 = \bigcup_v E(X_v) \quad \text{and} \quad E_2 = \{\{(v, e), (u, e)\} \,|\, e = (u, v) \in E\}.$$

Thus we may define $E' = E_1 \cup E_2$. Note that E_1 are all the edges of the different X_v's expanders, and E_2 are the edges induced by the original graph.

We now define the constraints on H. There are two types of edges in H, those originating from E_1 and those from E_2. The constraint on each edge is defined according to the type of this edge.

- *$e \in E_1$: These constraints try to enforce an identical assignment across each cloud of H. Formally*

$$\forall a, b: \ \mathcal{C}'(e)(a, b) = \begin{cases} 1 & a = b \\ 0 & \text{otherwise} \end{cases}$$

- *$((v, e), (u, e)) = e' \in E_2$: These constraints preserve the original constraints on G. Formally*

$$\mathcal{C}'(e') = \mathcal{C}(e)$$

We note that $|V'| = 2|E|$. Also clearly the graph (V', E') is a regular graph of degree $d + 1$, thus $\text{size}(H) \leq \text{constant} \cdot \text{size}(G)$.

Claim 4 $\textsc{Unsat}(H) \leq \textsc{Unsat}(G)$

Proof Let $a : V \to \Sigma$ denote the best assignment to G. We define $b(v, e) = a(v)$, that is, each cloud gets a single color. In H there are two types of edges. Since the constraints of edges from E_1 require that every node cloud has the same color, we have that $|E_1|$ of the constraints hold. On the other hand, all constraints of edges from E_2 hold (or don't hold) according to the assignment a as applied to G (since E_2 are edges induced by G on H). Hence, the number of constraints that are violated in H is $|E| \cdot \textsc{Unsat}_a(G)$. Thus,

$$\textsc{Unsat}_b(H) = \frac{\textsc{Unsat}_a(G) \cdot |E|}{|E'|} \underbrace{\leq}_{|E| \leq |E'|} \textsc{Unsat}_a(G) = \textsc{Unsat}(G).$$

[1] We are relying on the fact that there exists $0 < \lambda < d$ s.t. for all n there exists an (n, d, λ) expander.

Considering the best assignment b' for H gives us

$$\textsc{Unsat}\left(H\right) = \textsc{Unsat}_{b'}\left(H\right) \leq \textsc{Unsat}_{b}\left(H\right) \leq \textsc{Unsat}\left(G\right)$$

■

Claim 5 *There exists* $0 < \beta_{I_1} < 1$ *such that for every assignment* $b : V' \to \Sigma$*, it holds that* $\textsc{Unsat}_b\left(H\right) \geq \beta_{I_1} \cdot \textsc{Unsat}_a\left(G\right)$*, for an assignment* $a : V \to \Sigma$ *defined by*

$$a(v) = argmax_{\sigma\in\Sigma}\left\{Prob_{(v,e)\in[v]}\left(b(v,e) = \sigma\right)\right\}.$$

(a is defined such that $a(v)$ *will be the most popular value of* b *on nodes in the cloud of* $[v]$*).*

By taking b to be the assignment that gives $\textsc{Unsat}\left(H\right) = \textsc{Unsat}_b\left(H\right)$, this claim implies that

$$\textsc{Unsat}\left(H\right) = \textsc{Unsat}_b\left(H\right) \geq \beta_{I_1} \cdot \textsc{Unsat}_a\left(G\right) \geq \beta_{I_1} \cdot \textsc{Unsat}\left(G\right).$$

Proof Let $F \subseteq E$ be the set of all edges that are not satisfied by a, and let $S \subset V'$ be the set of all nodes that "do not agree" with the majority choice of assignment. That is, $S = \{(v,e)|b(v,e) \neq a(v)\}$. Also, let $F' \subset E'$ be the set of edges that are not satisfied by b. We note that $\textsc{Unsat}_a\left(G\right) = \frac{|F|}{|E|}$, and $\textsc{Unsat}_b\left(H\right) = \frac{|F'|}{|E'|}$.

We now show that $|F| \leq |F'| + |S|$. This holds because each edge e that is violated (not satisfied) by a in G, means one of two things: either e is violated by b in H, or at least one of the nodes of e do not agree on the choice of the majority. Hence, for every $(v,u) \in F$, either $(v,u) \in F'$ or ($v \in S$ or $u \in S$). And we have shown that $|F| \leq |F'| + |S|$.

We will also rely on the fact that $|E'| = (d+1)\,|E|$. Consider the case where $|F'| \geq \frac{|F|}{2}$. In this case we have

$$\textsc{Unsat}_b\left(H\right) = \frac{|F'|}{|E'|} \underbrace{\geq}_{|F'|\geq\frac{|F|}{2}} \frac{|F|}{2\cdot|E'|} \underbrace{\geq}_{|E'|\leq(d+1)\cdot|E|} \frac{|F|}{2\cdot(d+1)\cdot|E|} = \frac{1}{2\cdot(d+1)} \cdot \textsc{Unsat}_a\left(G\right)$$

Now, consider the case where $|F'| < \frac{|F|}{2}$ this implies that $|S| \geq \frac{|F|}{2}$. For each $v \in V$ we define $S^v = [v] \cap S$, that is, S^v contains all nodes in $[v]$ which disagree on the majority assignment. We also define $S^{v,\sigma} = \{(v,e) \in S^v | b(v,e) = \sigma\}$, that is $S^{v,\sigma}$ contains all nodes in S^v with assignment of σ. Since $\forall\sigma \neq a(v)$ it holds that $S^{v,\sigma} \leq \frac{|[v]|}{2}$ (otherwise, it would be the majority assignment) there are at least $h \cdot |S^{v,\sigma}|$ edges coming out of $S^{v,\sigma}$ into other nodes in $[v]$ (where h is the expansion of X_v). For each such edge, its constraint is violated by b. Hence, the number of violated constraints is at least

$$\sum_{v\in V}\sum_{\sigma\in\Sigma\setminus\{a(v)\}} |S^{v,\sigma}| \cdot \frac{h}{2} = \frac{h}{2} \cdot \sum_{v\in V} |S^v| = \frac{h}{2} \cdot |S|$$

Therefore,

$$\textsc{Unsat}_b\left(H\right) \geq \frac{|S|\cdot h/2}{|E'|} \underbrace{\geq}_{|S|\geq\frac{|F|}{2}} \frac{|F|\cdot h}{4\cdot|E'|} \underbrace{\geq}_{|E'|\leq(d+1)\cdot|E|} \frac{h}{4\cdot(d+1)} \cdot \frac{|F|}{|E|} = \frac{h}{4\cdot(d+1)} \cdot \textsc{Unsat}_a\left(G\right)$$

setting $\beta_{I_1} = min\left\{\frac{h}{4\cdot(d+1)}, \frac{1}{2\cdot(d+1)}\right\}$, we have that for every assignment b, it holds that $\textsc{Unsat}_b\left(H\right) \geq \beta_{I_1} \cdot \textsc{Unsat}_a\left(G\right)$. ■

Summing up this $\textsc{Prep}_1$ step, we have taken a general graph G and turned it into a d-regular graph H, keeping the size of H linear in G and not ruining the $\textsc{Unsat}$ by more than β_{I_1}. In the next section we will make H an expander.

4 Second Preprocessing Step – Expanderizing

We would like now to preform $G \to \textsc{Prep}_2(G) = H$. Remember that this operation is intended to add expansion to a d-regular graph, such that it becomes an expander. More formally:

Lemma 6 *There exist $d' > \lambda > 0$, $c_{I_2}, \beta_{I_2} > 0$, such that there is a transformation from d-regular constraint graph $G = \langle (V, E), \Sigma, \mathcal{C} \rangle$ to $H = \textsc{Prep}_2(G)$, such that $H = \langle (V, E'), \Sigma, \mathcal{C}' \rangle$ and:*

- *H is $(d + d')$-regular, and $\lambda(H) < \lambda + d$.*
- *$\beta_{I_2} \cdot \textsc{Unsat}(G) \leq \textsc{Unsat}(H) \leq \textsc{Unsat}(G)$.*
- *$size(H) \leq c_{I_2} \cdot size(G)$.*

Proof We consider a d'-regular expander $X = (V, E_X)$ on $|V|$ nodes, with a given λ. We define (V', E') as follows: $V' = V$, $E' = E \cup E_X$. It remains to define the constraints. For each $e \in E$, $\mathcal{C}'(e) = \mathcal{C}(e)$. For each $e \in E_X$ we let $\mathcal{C}'(e)$ be the trivial constraint that is *always* satisfied.

Note that H is $(d+d')$-regular, since each node has d edges from G and d' edges from X. In addition, $size(H)$ is not to large. More specifically, H has the same amount of vertices as in G, and

$$|E'| = |V'| \cdot (d + d') \underbrace{=}_{V=V'} |V| \cdot (d + d') \underbrace{=}_{\frac{2|E|}{d}=|V|} \frac{2 \cdot |E|}{d} \cdot (d + d')$$

Therefore, there is some $c_{I_2} > 0$ s.t. $size(H) \leq c_{I_2} \cdot size(G)$.

In addition, according to Exercise 2, question 2, we have that $\lambda(H) \leq \lambda(G) + \lambda(X)$. Since $\lambda(X) = \lambda$ and $\lambda(G) \leq d$ we have that $\lambda(H) \leq \lambda + d$.

We are now left with the second requirement. Fix some assignment $a : V \to \Sigma$ for G, it is also an assignment for H. Since the additional edges in H have constraints that are always satisfied, we have that $\textsc{Unsat}_a(H) \leq \textsc{Unsat}_a(G)$. This holds for any assignments a, hence we have that $\textsc{Unsat}(H) \leq \textsc{Unsat}(G)$.

In addition if a violates $\alpha \cdot |E|$ edges in G, then it violates $\alpha \cdot |E|$ edges in H. Since, $|E'| = \frac{|E|}{2 \cdot d} \cdot (d+d')$, we have that

$$\textsc{Unsat}_a(H) = \frac{\alpha\,|E|}{|E'|} = \frac{\alpha \cdot |E|}{\frac{d+d'}{d} \cdot |E|} = \alpha \cdot \frac{d}{d+d'} = \frac{d}{d+d'} \textsc{Unsat}_a(G).$$

Setting $\beta_{I_2} = \frac{d}{d+d'}$, we have that for every assignments a, it holds that $\textsc{Unsat}_a(H) \geq \beta_{I_2} \cdot \textsc{Unsat}_a(G)$. This completes the proof, since it implies that $\textsc{Unsat}(H) \geq \beta_{I_2} \cdot \textsc{Unsat}(G)$ ■

5 Back to the First Operation

Combining the two previous lemmas, we have that $\textsc{Prep}(G) = \textsc{Prep}_2(\textsc{Prep}_1(G))$ does what is expected of the first operation. That is, it keeps the size linear ($c_I = c_{I_1} \cdot c_{I_2}$), and doesn't reduce the $\textsc{Unsat}$ too much ($\beta_I = \beta_{I_1} \cdot \beta_{I_2}$). Lastly, $\textsc{Prep}(G)$ transforms G into a d-regular expander.

Remark We have seen that the *PCP* theorem claims that given a language in $L \in NP$ then there exists a way of writing L's proofs in such a way that they are *PCP* checkable. However, we did not see how one can transform a proof for L into a proof for the *PCP* equivalent of L.

Looking back at theorem 1(in this scribe), if we can show that when constructing G' from G we know how to transform an assignment for G into an assignment for G', then we have actually shown how to transform a proof for L into its *PCP* equivalent. We note that during the above constructions, we know how to transform the coloring (constraint assignment) of the input graph into a coloring of the output graph. Which means we can construct a *PCP* proof from a *NP* proof.

6 The Second Operation - Powering

We now raise a constraint graph to the t-th *power* in order to increase its UNSAT. This operation also increases the alphabet size. We will not show the proof in this lesson, just the construction.

Definition 7 *Let $G = \langle (V, E), \Sigma, \mathcal{C} \rangle$ be a d-regular constraint graph, and let $t \in \mathbb{N}$ be some even number. We define $G^t = \langle (V, E'), \Sigma', \mathcal{C}' \rangle$ to be the following constraint graph:*

In E' there are exactly k parallel edges between u and v if the number of walks between u and v in G is k. In other words, if we denote by A the adjacency matrix of G, then the adjacency matrix of G^t is A^t.

We define $\Sigma' = \Sigma^{d^0+d^1+...+d^{\frac{t}{2}}}$ ($d^0 + d^1 + ... + d^{\frac{t}{2}}$ is the bound on the number of nodes in a ball of radius $\frac{t}{2}$). We denote $B(v, \frac{t}{2}) = \left\{u | dist_G(u, v) \leq \frac{t}{2}\right\}$, and note that $d^0 + d^1 + ... + d^{\frac{t}{2}}$ is an upper bound on the number of nodes in a ball of radius $\frac{t}{2}$ around v. We think of $\sigma \in \Sigma'$ as an assignment $B(v, \frac{t}{2}) \to \Sigma$.

Next, for each $(v_1, v_2) = \bar{e} \in E'$ we define $\mathcal{C}'(\bar{e}) : \Sigma' \times \Sigma' \to \{0, 1\}$ to "check as much as you can". Formally, for any $\sigma_1, \sigma_2 \in \Sigma'$, consider

$$\sigma_1 : \overbrace{B(v_1, \frac{t}{2})}^{B_1} \to \Sigma \qquad \sigma_2 : \overbrace{B(v_2, \frac{t}{2})}^{B_2} \to \Sigma$$

$\mathcal{C}'(\bar{e})(\sigma_1, \sigma_2)$ will work as follows: if there is some $w \in B_1 \cap B_2$ such that $\sigma_1(w) \neq \sigma_2(w)$ then $\mathcal{C}'$ returns 0 (this checks that nodes in the intersection of the balls B_1, B_2 have a well defined assignment). Otherwise, let $\sigma : B_1 \cup B_2 \to \Sigma$ such that $\sigma |_{B_i} = \sigma_i$. The constraint is satisfied if there is no edge in $E \cap (B_1 \cup B_2) \times (B_1 \cup B_2)$ that is violated by σ.

A few remarks about the above definition:

- G^t is a d^t-regular graph.
- If UNSAT $(G) = 0$ then UNSAT $(G^t) = 0$, since if there is an assignment that satisfies G, we can construct an assignment that satisfies G^t.
- $size(G^t) \leq d^{t-1} \cdot Size(G)$. In particular, if t, d are constants, then $Size(G^t) = O(Size(G))$.

7 Future Lessons

In the future, we will show that UNSAT $(G^t) \geq$ constant $\cdot\, t \cdot$ UNSAT (G). Actually, for the definition we gave, the proof would show that UNSAT $(G^t) \geq$ constant $\cdot \sqrt{t} \cdot$ UNSAT (G). However, in next lesson we will construct a graph that chooses the different walks from G in a non-uniform way. This will create a graph G' such that we will achieve t and not $\sqrt{t}$.