Limits of Approximation Algorithms 12 Apr, 2010 (TIFR)

Lec. 10: Query reduction via robust PCP composition

Lecturer: Prahladh Harsha *Scribe: Ramprasad Saptharishi*

10.1 Introduction

We have been gearing towards proving the PCP theorem and last class we came quite close to what we wanted. Using the low-degree test, we constructed a *Zero-On-Subcube* test which was used to give a PCP for 3SAT. The PCP was in fact a *robust* PCP for 3SAT and had the following parameters:

Proof Size	:	$\text{poly}(n)$
Randomness	:	$O(\log n)$
Robust soundness error	:	$1/\text{polylog}(n)$
Query Complexity	:	$\text{polylog}(n)$
Alphabet Size	:	$\text{poly}(n)$

We are doing good in the proof size, randomness used and robust soundness error. In fact, the robust soundness error is even sub-constant! However, the alphabet size is too large and so is the query complexity. How do we reduce these two parameter without affecting the others too much?

Alphabet Reduction: Alphabet reduction can be easily achieved by simple code concatenation: encode each alphabet of the proof using a code over a smaller alphabet but with very good distance, say $1-\rho$. It can be easily shown that the robustness of the new PCP over the smaller alphabet worsens only by an additive factor of $\text{poly}(\rho)$. The details of this reduction is defered to Appendix A[1].

Query reduction: This will be the main focus of this lecture. For this, let us first recall the definition of a robust PCP.

Let Φ be an instance for $\mathsf{L} = \mathsf{3SAT}$ and let $\Pi \in \Sigma^m$ be the (purported) proof provided by the prover to show that $\Phi \in \mathsf{L}$. The general structure for the verifier for L was the following:

- Using Φ and some random bits R, choose a window $I_R \subset [m]$ in the proof and an acceptance predicate φ_R. (we shall drop the subscript R when the context is clear)
- Accept if and only if $\varphi_R(\Pi_{I_R})$ is true.

(Completeness) If $\Phi \in \mathsf{L}$, then there exists a proof Π such that

$$\Pr_R[\varphi(\Pi_I) = 1] = 1$$

[1] Prahladh: These notes are far more detailed than the lecture it corresponds to. Thanks to the scribe Ramprasad for filling in all the missing details in the lecture (especially Appendix A and Appendix B)

(Robust Soundness) If $\Phi \notin \mathsf{L}$, then for every Π,

$$\mathop{\mathbb{E}}_{R}[\mathsf{agr}(\Pi_I, \mathsf{SAT}(\varphi))] \leq \delta$$

For simplicity, we shall assume some regularity conditions on the PCP. We'll assume that all the subsets I_R are of the same size and each $i \in [m]$ is present in the same number of I_R's[2].

In order to reduce the query complexity, we have to devise a way by which the verifier can check if Π_I satisfies φ or not without reading all of Π_I. This is very reminiscent of the philosphy of PCP which is to probe an encoding at very few places to decide if φ is satisfiable or not. This is what we are going to do – run an inner PCP to check if Π_I satisfies φ or not. This idea of using an inner verifier to reduce query complexity is refered to as composition in the PCP literature and is due to Arora and Safra [AS98].

Let us see if we can perform this composition. Note that we cannot perform recursion or composition. The situation here is slightly different; we do not want to know if φ is satisfiable or not but we want to know if φ is satisfiable by this specific Π_I. This is also crucial because, of course, φ is going to be satisfied by some assignment (well if it was always false, why would the verifier be interested in checking it!). We need more from the inner verifier than the outer verifier to perform this consistency test. We will be able to perform this consistency test if the "inner" proof were something more than just a "locally checkable encoding" of a satisfying assignment; we need it to be a "locally decodable encoding".

10.2 Decodable PCPs

The goal is to check if the window Π_I satisfies the predicate φ or not. As before, we would be encoding this Π_I using a proof π and run the "inner PCP" on it. And we want to augment the inner PCP with some more structure so that we can check if the encoded satisfying assignment is indeed Π_I. One possible approach is to expect the inner verifier to decode locations of the satisfying assignment whenever to accepts its proof. This is implemented as follows: the inner PCP verifier is given an additional input i, which is an index into the satsifying assignment and it is expected to return the i-th index of the satisfying assignment it encodes or reject.

10.2.1 Attempted definition

Let π be a (purported) encoding of a satisfying assignment to φ. The verifier on randomness r and input index i picks a random window $J = J(i, r)$ and computes a function $f = f(i, r)$ on it. We'll say that this protocol is a dPCP if the following conditions hold:

(Completeness) For every φ and every satisfying assignment w of φ, there exists a proof π such that

$$\Pr_{i,r}[f(\pi_J) = w_i] = 1$$

[2]The robust PCP constructed in the earlier did *not* satisfy these regularity conditions. We will need to some standard regularizing transformations, which we will not go into in this lecture, to make the PCP regular.

I.e., for every satisfying assignment there is a faithful encoding that can be decoded as well

(Soundness) For every φ and and π, there is at most one satisfying assignment w such that

$$\Pr_{i,r}[f(\pi_J) \notin \{\perp, w_i\}] \leq \delta$$

Or in other words for any proof, there is at most one satisfying assignment w that it can possible decode to (when it does not reject).

We could even try to extend it to a "robust dPCP" with the following stronger condition.

(Robust Soundness) Let $\mathsf{BAD}(f,i) \stackrel{\text{def}}{=} \{z \ : \ f(z) \notin \{\perp, w_i\}\}$. Then,

$$\mathop{\mathbb{E}}_{i,r}[\mathsf{agr}(\pi_J, \mathsf{BAD}(f,i))] \leq \delta$$

Let us first see how this definition would be useful. We are given a proof Π to show that $\Phi \in \mathsf{3SAT}$. The outer verifier chooses a window I and a predicate φ and wishes to check if Π_I is a satisfying assignment of φ. To do this, it picks a random index $i \in I$ and runs the dPCP dor φ on index i. The verifier then checks if $(\Pi_I)_i$ matches with the output of the dPCP.

Why would this work? Suppose $\Phi \notin \mathsf{3SAT}$. Notice that by the robustness of the outer verifier, on average, Π_I is far from any satisfying assignment. Therefore, no matter what proof the prover gives to the inner dPCP the outer verifier's view Π_I can agree with w in no more than δ fraction of the places. Therefore, this new verifier would reject the proof.

Unfortunately the above definition of dPCP is not realisable. The main reason being that we want a unique w for any proof π in the soundness condition. Unique decoding is possible if δ were large, close to 1. However, we are interested in the case when δ is very small, in fact, even sub-constant. To get around this problem, we will let the dPCP decode consistently with a short list of proofs instead of just one polynomial! list is inevitable!

Similar to the low-degree test discussed in the earlier lectures, we shall modify the above definition to include not one satisfying assignment but a "small list of satisfying assignments" that the dPCP can decode to.

10.2.2 Modified definition

Definition 10.2.1 (dPCPs)**.** *Let φ be a predicate. φ is said to have a decodable PCP (dPCP) with list size L if there exists a verifier (or decoder) that behaves as follows. On input a index i and oracle access to a proof π, the verifier/decoder uses randomness r to pick a window $J = J(i,r)$ and a function $f = f(i,r)$ with the following properties:*

(Completeness) *If w is a satisfying assignment of φ, then there exists a π such that*

$$\Pr_{i,r}[f(\pi_J) = w_i] = 1.$$

(Soundness) *For every π, there is a short list of satisfying assignments list$(\varphi) = \{w^1, \cdots, w^L\}$ of φ (i.e., $\varphi(w^k) = 1$ for all k) such that*

$$\Pr_{i,r}\left[f(\pi_J) \notin \{\perp, w_i^1, w_i^2, \cdots, w_i^L\}\right] \leq \delta.$$

Further the dPCP *is said to be* robust *if the following stronger soundness condition holds:*

(Robust Soundness) *Given proof π, there is a short list of satisfying assignments list$(\varphi) = \{w^1, \cdots, w^L\}$ of φ such that*

$$\mathop{\mathbb{E}}_{i,J}\left[\mathsf{agr}\left(\pi_J, \mathsf{BAD}(f,i)\right)\right] \leq \delta$$

where $\mathsf{BAD}(f,i) \stackrel{\text{def}}{=} \{z \ : \ f(z) \notin \{\perp, w_i^1, w_i^2, \cdots, w_i^L\}\}$.

This is the definition we would be working with. Do we actually have such decodable PCPs? Yes, we do — the Zero-On-Subcube test can be changed slightly to give us a decodable PCP.

10.2.3 dPCPs from the Zero-On-Subcube test

The robust PCP for NP from last lecture proceeded as follows: The prover claimed that a certain function $P_{\Phi,A}$ was zero on a subcube H^m and provided the evaluation of $P_{\Phi,A}$ and several other auxillary functions on all points in $\mathbb{F}^m$. The verifier V would pick a random "object" $\Omega \subseteq \mathbb{F}^m$ and query the function value on all points in Ω and checks if there is indeed a "good" polynomial on the restriction. Recall that the objects Ω were sampled by picking points z, z', y, y' at random and setting $\Omega = \text{span}(z, \rho(z), \rho^2(z), z', y, y)$.

Lets see how we can construct a dPCP from this. Now suppose we are given an index i and, if the verifier was going to accept the proof, we want the i-th bit of A. Recall that i was identified by a point $p_i \in H^m$ and what we are asking for is the evaluation of A at this point.

The most natural thing to do is to enlarge the Ω's that contain the point p_i. We can modify the Zero-On-Subcube test to use objects of the form

$$\Omega = \text{span}(z, \rho(z), \rho^2(z), z', y, y, p_i)$$

And since the point p_i is in each Ω that we sample, we also have the evaluation of A at i. The verifier/decoder of this dPCP returns this $A(i)$ if it were to accept and returns $\perp$ if it were to reject. Let us call this function f. The list-decoding statement we had was the following:

> For every "objects oracle" $\mathcal{A}$, there exists a small list of polynomials $Q^1, \cdots, Q^\ell$ (where $\ell \leq \text{poly}(1/\delta)$) constructed from satisfying assignments such that
>
> $$\Pr_{\Omega}\left[\text{Zero-On-Subcube test fails} \quad \text{or} \quad \exists i : Q^i \mathop{\equiv}_{\Omega} \mathcal{A}(\Omega)\right] \leq \delta$$

For any object Ω, let $\text{eval}(\Omega)$ be the set of evaluations on Ω. Define the set $\mathsf{BAD}(f,i) = \left\{\omega \in \text{eval}(\Omega) \ : \ f(\omega) \notin \left\{\perp, Q^1(p_i), Q^2(p_i), \cdots, Q^\ell(p_i)\right\}\right\}$. By the robustness of the PCP, we have

$$\mathop{\mathbb{E}}_{i,\Omega}\left[\mathsf{agr}\left(\omega, \mathsf{BAD}(f,i)\right)\right] \leq \delta$$

which is exactly what we would want for a robust dPCP.

Remark 10.2.2. *Notice that if we want to run a* dPCP *as an "inner verifier" to check if* Π_I *is a satisfying assigment to* φ *or not, the alphabet size for the* dPCP *must be larger than the alphabet size of the outer proof. Therefore it is essential that we reduce the alphabet size. But for now, let us assume that the alphabet size can be brought down to polylog*(n) *without altering the other parameters too much. The details of alphabet reduction are discussed in Section A.*

10.3 Composition

Now we have a robust outer PCP and a robust inner dPCP and we'll see how we can compose them to reduce the query complexity. Here is one possibility that we discussed earlier.

> Given a formula Φ, the prover provides a proof Π. The old verifier, using his random coins R, would have picked a pair (I_R, φ_R) and accepted if and only if $\varphi_R(\Pi_{I_R})$ is true.
>
> In the composed PCP, the prover not only provides the proof Π but also provides a proof $\pi^{(R)}$ for every random coin R of the old verifier. $\pi^{(R)}$ is supposed to be the dPCP for φ_R. The composed verifier, first runs the outer verifier, and uses random coins R to choose a pair (I_R, φ_R) as before. Then the verifier picks a random index $i \in I_R$ and now runs the inner dPCP decoder on input φ_R and index i on proof $\pi^{(R)}$. It accepts if the value decoded by the dPCP decoder equals $(\Pi_I)_i$.

The following are the parameters of the composed PCP.

	Randomness	Query Complexity	Robust Soundness Error
Outer PCP	R	Q	Δ
Inner PCP	r	q	δ, ℓ
Composed PCP	$R + r + \log Q$	$q + 1$(outer)	?

The query complexity has dropped from Q to around q, which is what we wanted to achieve. What can we say about the robust soundness error of the composed PCP? Defining it is a bit dicey since all queries of the composed verifier aren't over the same alphabet – it involves one big query from the outer proof, and several smaller queries in the inner proof. But suppose we still define agreement by an appropriate weight put on the outer and inner queries, can we argue that the robust soundness error is small? Unfortunately, no; we can't say anything better than $1/2$ for the robust soundness error and here is why.

The views of this composed verifier consists of a probe in the outer proof Π_i and smaller probes in the inner proof $\bar{\pi} = \pi_1, \cdots, \pi_q$. Satisfying views are of the form $\bar{z} = (z_0, z_1, \cdots, z_q)$ where $z_0 = f(z_1, \cdots, z_q)$. To define agreement, if more than half the weight was put on $\pi'_j s$, then $\mathsf{agr}\left((\Pi_i, \pi_1, \cdots, \pi_d), (f(\bar{\pi}), \pi_1, \cdots, \pi_d)\right)$ is at least $1/2$. Similarly, if there was more than $1/2$ weight put on the Π_i, then we can choose $z_1, \cdots, z_d$ appropriately so that $f(z_1, \cdots, z_d) = \Pi_i$ and a similar thing would hold.

Hence this method of composition would not work if we want to retain the robustness of the PCP. How can we modify this composition so that query complexity is reduced without compromising the robustness. Note that any composition technique in which the proof is split logically across two parts – an outer and an inner proof, is going to run into the same problem. Can we do away with one of the proofs? What was the role of the outer proof anyway? What prevented the prover from giving satisfying assignments for each of the inner proofs? The only role that the outer proof plays is to enforce some sort of consistency across various windows chosen by the verifier. This is what prevents the prover from just choosing fresh satisfying assignments for each window; the assignments that are being encoded must be consistent across the intersections.

How about getting rid of the outer proof completely and just checking if the inner proofs are consistent amongst each other? This would be our modified composition procedure [MR08, DH09] and we will show that it surprisingly preserves the robustness of the composed PCPs.

Modified Composition

1. The prover provides an inner-proof $\pi^{(R)}$ for every possible random coin R of the outer PCP. Thus, the proof for the composed verifier is $\Upsilon = \{\pi^{(R)} | R \text{ outer random coins } \}$.

2. The verifier choose an index $i \in [M]$ in the outer proof and let $I_1, \cdots, I_D$ be all possible windows that contain i. We will refer to the quantity D as the proof degree of the outer PCP. Let $R_1, \cdots, R_D$ be the outer random coins that led to the windows $I_1, \cdots, I_D$ respectively and $\varphi_1, \cdots, \varphi_D$ be the corresponding consistency predicates.

3. For $k = 1$ to D, run the (inner) dPCP decoder on input φ_k and index i on proof π_{R_k} using randomness r.

4. Accept if and only if all the D "dPCP decoders" do not reject and return the *same* value. In other words, accept only if

$$f_1\left((\pi_{R_1})_{J_1}\right) = f_2\left((\pi_{R_2})_{J_2}\right) = \cdots = f_D\left((\pi_{R_D})_{J_D}\right) \neq \perp,$$

where (J_k, f_k) refers to the window-decoding function pair output by the dPCP decoder on input φ_k and index i.

The parameters of this composition are as follows:

	Proof Size	Randomness	# Queries	Soundness Error	Proof degree
Outer PCP	M	R	Q	Δ	D
Inner dPCP	m	r	q	δ, ℓ	d
Composed PCP	$2^R \cdot m$	$\log M + r$[3]	qD	$\Delta\ell + \delta$	d
Sampler based- -Composed PCP	$2^R \cdot m$	$\log M + r + \log D$	$q \cdot \text{poly}\left(\frac{1}{\varepsilon}\right)$	$(\Delta + \varepsilon)\ell + \delta$	d

We will show below that the composed verifier has robust soundness error at most $\Delta\ell + \delta$, which is small as long as $\Delta \ll 1/\ell$. Thus this robust composition reduces query complexity from Q to qD without compromising on the robustness. However, this is a query reduction only if $qD < Q$. This might not always be the case, as the proof degree D could be very large. To get around this issue of large D, we could do the following: in step 2, instead of querying all the windows $I_1, \ldots, I_D$ that the index i participated in, query only a pseudorandom sample of the windows. Assuming the composed PCP has parameters as mentioned in the table above, it is easy to see that this sampler based composed PCP has parameters as indicated in the bottom row of the above table (the details of this are defered to the Appendix B.). For now, we will assume that D is small and proceed with the robustness analysis.

10.3.1 Robustness Analysis

As mentioned before, we assume that the PCPs in context are regular in the sense that all windows are of equal sizes and every index i participates in the same number of windows. Proposition 10.3.1 shows that the above composition is sound and maintains robustness.

Let us first understand what are the views of the composed verifier and the corresponding accepting configurations. The prover provides a proof $\pi^{(R)}$ for every possible outer random coins R of the outer verifier. The composed verifier chooses an index i into the outer proof and let $I_1, \cdots, I_D$ be all possible windows that involve i. Let $R_1, \cdots, R_D$ be the outer randoms coins that generated these windows. Let $\pi^{(1)} = \pi^{(R_1)}, \cdots, \pi^{(D)} = \pi^{(R_D)}$ be the corresponding proofs provided by the prover. On input i, using the randomness r, the D inner dPCP verifier/decoder generate $(J_1, f_1), \cdots, (J_D, f_D)$ window-function pairs. Let $\sigma_k = \pi^{(k)}_{J_k}$. The composed verifier V then checks if the decodings across the f_i's match (i.e., $f_1(\sigma_1) = \cdots = f_D(\sigma_D) \neq \perp$. Thus, satisfying views for the composed verifier are the following

$$\mathsf{SAT}(i, r) = \{z_1 \cdots z_D \ : \ f_1(z_1) = f_2(z_2) = \cdots = f_D(z_D) \neq \perp\}$$

Recall that for every $\pi^{(R)}$, there is an associated $\text{list}(\pi^{(R)})$ of at most ℓ satisfying assignments $w_1^R, \cdots, w_\ell^R$ of φ_R. The robustness of the dPCP stated that for every proof $\pi^{(R)}$,

$$\mathop{\mathbb{E}}_{i,r}\left[\mathsf{agr}\left(\pi^{(R)}_J, \mathsf{BAD}(f, i)\right)\right] \leq \delta,$$

where

$$\mathsf{BAD}(f, i) = \left\{z \ : \ f(z) \notin \{\perp\} \cup (\text{list}(\pi^{(R)}))_i\right\}$$

Recall that proof Υ for the composed PCP verifier is the concatenation of all $\pi^{(R)}$'s over the different random coins R of the outer proof. Let $S_{i,r} = J_1 \cup J_2 \cup \cdots J_D$, the

[3] can use the same random seed r for all the parallel runs

window corresponding to i and r. $S_{i,r}$ denotes the local window of the composed verifier on randomness (i,r). The following proposition shows that the composed PCP is robust.

Proposition 10.3.1. *If $\Phi \notin \mathsf{3SAT}$, then for every proof Υ for the composed PCP, we have*

$$\mathop{\mathbb{E}}_{i,r}\left[\mathsf{agr}\left(\Upsilon_{S_{i,r}}, \mathsf{SAT}(i,r)\right)\right] \leq \Delta\ell + \delta.$$

Proof. For any values that i and r take, let $z_1 \cdots z_D$ be the string in $\mathsf{SAT}(i,r)$ that is closest to $\Upsilon_{S_{i,r}}$ and let k be a random index in $1, \cdots, D$. By the regularity assumption on the PCP,

$$\mathsf{agr}\left(\Upsilon_{S_{i,r}}, \mathsf{SAT}(i,r)\right) = \mathop{\mathbb{E}}_{k}\left[\mathsf{agr}\left(\sigma_k, z_k\right)\right]$$

Let $\perp \neq \alpha = \alpha(i,r) = f_1(z_1) = f_2(z_2) = \cdots = f_D(z_D)$. Let $c_k(i,r)$ be the indicator random variables defined as follows:

$$c_k = \begin{cases} 1 & \text{if } \alpha \in (\mathrm{list}(\pi^{(k)}))_i \\ 0 & \text{otherwise} \end{cases}$$

Therefore,

$$\begin{aligned} \mathop{\mathbb{E}}_{i,r}\left[\mathsf{agr}\left(\Upsilon_{S_{i,r}}, \mathsf{SAT}(i,r)\right)\right] &= \mathop{\mathbb{E}}_{i,r,k}\left[\mathsf{agr}\left(\sigma_k, z_k\right)\right] \\ &= \mathop{\mathbb{E}}_{i,r,k}\left[c_k \cdot \mathsf{agr}\left(\sigma_k, z_k\right)\right] + \mathop{\mathbb{E}}_{i,r,k}\left[(1-c_k) \cdot \mathsf{agr}\left(\sigma_k, z_k\right)\right] \end{aligned}$$

Lets focus on the second term. For every fixed k, if $c_k = 0$, then $f_k(z_k) = \alpha$ which is not in $\{\perp\} \cup (\mathrm{list}(f_k))_i$. Therefore, $z_k \in \mathsf{BAD}(f_k, i)$ and the second term is upper bounded by $\mathbb{E}_{i,r,k}\left[\mathsf{agr}\left(\pi^{(k)}, \mathsf{BAD}(f_k,i)\right)\right]$ which is upper bounded by δ by the robustness of the dPCP. Hence,

$$\begin{aligned} \mathop{\mathbb{E}}_{i,r}\left[\mathsf{agr}\left(\Upsilon_{S_{i,r}}, \mathsf{SAT}(i,r)\right)\right] &\leq \mathop{\mathbb{E}}_{i,r,k}\left[c_k \cdot \mathsf{agr}\left(\sigma_k, z_k\right)\right] + \delta \\ &\leq \mathop{\mathbb{E}}_{i,r,k}\left[c_k\right] + \delta \end{aligned}$$

Suppose we assumed that $\mathbb{E}_{i,r}\left[\mathsf{agr}\left(\Upsilon_{S_{i,r}}, \mathsf{SAT}(i,r)\right)\right] > \Delta\ell + \delta$. Then the above equation shows that $\mathbb{E}_{i,r,k}\left[c_k\right] > \Delta\ell$. To get the contradiction, we'll use this to construct an a proof Π for outer verifier that violates the robustness of the outer verifier. The proof Π is constructed in a randomized fashion as follows:

- For every outer random coins R, there is a small list of satisfying assignments $\mathrm{list}(\pi^{(R)})$. Define $y(R)$ to be one satisfying assignment randomly chosen from $\mathrm{list}(\pi^{(R)})$. Note that $y(R)$ is a satisfying assignment for φ_R.
- For every $i \in [M]$, set Π_i to be the most popular choice of $y(R)_i$ amongst all random coins R's that generate windows I_R that contain i.

$$\begin{aligned} \text{If} \quad \mathop{\mathbb{E}}_{i,r}\left[\Pr_{k}[\alpha \in (\mathrm{list}(\pi^{(R_k)}))_i]\right] &= \mathop{\mathbb{E}}_{i,r,k}\left[c_k\right] > \Delta\ell \\ \text{then} \quad \mathop{\mathbb{E}}_{i,r}\left[\Pr_{k}[\alpha = y(R_k)_i]\right] &> \Delta \end{aligned}$$

By regularity, picking an i and then an I_k is the same as picking a window I and an index $i \in I$. Also, observe that for every fixing of i and r, the Π_i was chosen to be $\text{argmax}_\alpha \Pr_{R:i\in I_R}[\alpha = y(R)_i]$. Therefore,

$$\begin{aligned}\Delta &< \mathop{\mathbb{E}}_{i,r}\left[\Pr_k[\Pi_i = y(R_k)_i]\right] \\ &= \mathop{\mathbb{E}}_{R}\left[\Pr_{i\in I_R}[\Pi_i = y(R)_i]\right] \\ &= \mathop{\mathbb{E}}_{R}[\mathsf{agr}(\Pi_{I_R}, y(R))] \leq \mathop{\mathbb{E}}_{R}[\mathsf{agr}(\Pi_{I_R}, \mathsf{SAT}(\varphi_R))]\end{aligned}$$

which is a contradiction to the robustness of the outer PCP. □

Therefore the composition is indeed robust.

10.3.2 Fixing Parameters

The following parameters describes the parameters that we get after composing the outer PCP with the inner dPCP using the sampler-based composition.

	Outer PCP	Inner dPCP	Composed PCP
Randomness	$O(\log n)$	$O(\log\log n)$	$O(\log n)$
Query Complexity	$\text{polylog}(n)$	$\text{polyloglog}(n)$	$\text{polylog}\log(n)$
Alphabet Size [4]	$\text{polylog}(n)$	$\text{polylog}(n)$	$\text{polyloglog}(n)$
Robust soundness error	$\frac{1}{\text{polylog}(n)}$	$\frac{1}{\text{polyloglog}(n)}$, $\text{polyloglog}(n)$	$\frac{1}{\text{polyloglog}(n)}$

Thus, one round of composition reduces the window size (ie., query complexity) from $\text{polylog}(n)$ to $\text{polyloglog}(n)$ and in process the robust soundness error has only increased from $1/\text{polylog}(n)$ to $1/\text{polyloglog}(n)$. We could keep repeating this process till we obtain PCPs with the desired query complexity (or robust soundness error). For instance, to obtain PCPs with constant query complexity, we perform $\log^* n$ rounds of composition. We, thus, have the following *Robust* PCP *Theorem for* NP.

Theorem 10.3.2 (Robust PCP Theorem for NP)**.** *For every language* $\mathsf{L} \in \mathsf{NP}$ *and for every* $\delta : \mathbb{Z}^{\geq 0} \to (0,1)$*, there exists a* robustPCP *for* L *with randomness* $O(\log n)$*, query complexity* $\text{poly}\left(\frac{1}{\delta}\right)$ *over an alphabet of size* $\text{poly}(1/\delta)$ *with perfect completeness and robust-soundness error at most* δ*.* □

References

[AS98] Sanjeev Arora and Shmuel Safra. *Probabilistic checking of proofs: A new characterization of NP.* J. ACM, 45(1):70–122, January 1998. (Preliminary Version in *33rd FOCS*, 1992). doi:10.1145/273865.273901.

[DH09] Irit Dinur and Prahladh Harsha. *Composition of low-error 2-query PCPs using decodable PCPs.* In *Proc. 50th IEEE Symp. on Foundations of Comp. Science (FOCS)*, pages 472–481. 2009. eccc:TR09-042, doi:10.1109/FOCS.2009.8.

[4] The alphabet size has been reduced using alphabet reduction (see Theorem A.6 in Appendix A).

[Gol97] Oded Goldreich. *A sample of samplers – a computational perspective on sampling.* Technical Report TR97-020, Electronic Colloquium on Computational Complexity, 1997. eccc:TR97-020.

[MR08] Dana Moshkovitz and Ran Raz. *Two query PCP with sub-constant error.* In *Proc. 49th IEEE Symp. on Foundations of Comp. Science (FOCS)*, pages 314–323. 2008. eccc:TR08-071, doi:10.1109/FOCS.2008.60.

Appendix

We didn't quite discuss alphabet reduction and degree reduction in detail during the class. However, here is a complete (more-or-less) description of the procedure.

A Alphabet Reduction

It is convenient to work with the label cover formulation instead of the robust PCP formulation for the purpose of alphabet reduction. Suppose we are given an instance of a δ-gap label cover $(U, V, \Sigma', \Sigma, \mathcal{F})$. We would like to reduce the alphabet Σ to something smaller. The idea is to use an error correcting code and encode the labelling using it. This would increase the query complexity a little but would still remain polylog(n). Before we go into the actual construction, we need a few facts about error correcting codes.

Definition A.1 (Error correcting codes)**.** *A function $\mathcal{C} : \Sigma \longrightarrow \sigma^k$ is said to be an error-correcting code of distance $1-\delta$ if $\mathsf{agr}(\mathcal{C}(a), \mathcal{C}(b)) \leq \delta$ for every $a \neq b$.*

There are standard, well-known constructions of error correcting codes with excellent parameters. The following would be sufficient for our purposes.

Fact A.2. *For every $\delta > 0$ and alphabet Σ there exists a code $\mathcal{C} : \Sigma \longrightarrow \sigma^k$ with relative distance $1-\delta$ where $|\sigma| = O(1/\delta^2)$ and $k = O(\log|\Sigma|/\delta^2)$.*

The following claim shows that there aren't too many codewords close to a given message.

Fact A.3. *For any set of words $w, \beta_1, \cdots, \beta_\ell \in \sigma^k$,*

$$\sum_i \mathsf{agr}(w, \beta_i) \leq 1 + \sum_{i \neq j} \mathsf{agr}(\beta_i, \beta_j)$$

Proof. Just use inclusion exclusion. □

Claim A.4. *Let $\mathcal{C}$ be a code of distance $1-\delta$ and let $\eta > 2\sqrt{\delta}$. Then for any $w \in \sigma^k$,*

$$|\{a \in \Sigma \ : \ \mathsf{agr}(\mathcal{C}(a), w) \geq \eta\}| \quad \leq \quad \frac{2}{\eta}$$

Proof. Plug in $\mathcal{C}(a)$'s for the β_i's in the above fact and the bound follows. □

We are now well-equipped to do alphabet reduction. Let $\mathcal{C} : \Sigma \longrightarrow \sigma^k$ be an error-correcting code with distance $1 - \eta^3$ for some $\eta \leq 1/4$. Given an instance $(U, V, \Sigma', \Sigma, \mathcal{F})$ of label-cover, it shall be reduced to an instance $(U, V \times [k], \Sigma', \sigma, \mathcal{F}')$. The projections functions on the edge $e = (u, (v, i))$ is a function $f'_e : \Sigma' \longrightarrow \sigma$ is just the i-th block of the encoded version:

$$f'_e(\alpha) = \mathcal{C}(f_{(u,v)}(\alpha))_i$$

It is clear that the reduction has completeness 1, (i.e) maps satisfiable instance to satisfiable instances. The following theorem shows that it maintains soundness as well.

Theorem A.5. *If the instance $I = (U, V, \Sigma', \Sigma, \mathcal{F})$ is at most δ-satisfiable, then the reduced instance $I' = (U, V \times [k], \Sigma', \sigma, \mathcal{F}')$ is at most $(\delta + 3\eta)$-satisfiable.*

Proof. Suppose $\Pi' = (\pi_1 : U \to \Sigma' \ , \ \pi'_2 : V \times [k] \to \sigma)$ was a labelling to the reduced instance. For every vertex $v \in V$, and let us look at the fraction of edges a label β on v would have satisfied:

$$\delta_v(\beta) = \Pr_{u \in \Gamma(v)} \left[f_{(u,v)}(\pi_1(u)) = \beta \right]$$

The fraction of edges incident on $\{(v, i)\}_i$ that are satisfied is given by:

$$\delta'_v = \sum_\beta \delta_v(\beta) \cdot \Pr_i \left[\pi'_2(v, i) = \mathcal{C}(\beta)_i \right]$$

Assuming that the label-cover instance is regular, the fraction of edges that satisfied is $\mathbb{E}_v[\delta'_v]$. From Π', construct a labelling Π for the instance I as follows:

> The labelling on the left vertices U is given by π_1. As for the right vertices, do the most natural thing – choose $\pi_2(v)$ to be the β that maximizes $\delta_v(\beta)$.

If $\delta_v = \max \delta_v(\beta)$, the fraction of edges that are satisfied is $\mathbb{E}_v[\delta_v]$. We'll show that $\delta'_v \leq \delta_v + 3\eta$ and that would prove theorem. For every $v \in V$, let $\pi'_2(v) = \pi'_2(v, 1) \cdots \pi'_2(v, k)$ and $\text{list}(v) = \{\beta \ : \ \mathsf{agr}\left(\pi'_2(v), \mathcal{C}(\beta)\right) \geq \eta\}$.

$$\begin{aligned}
\delta'_v &= \sum_\beta \delta_v(\beta) \cdot \Pr_i \left[\pi'_2(v, i) = \mathcal{C}(\beta)_i \right] \\
&= \sum_{\beta \in \text{list}(v)} \delta_v(\beta) \cdot \mathsf{agr}\left(\pi'_2(v), \mathcal{C}(\beta)\right) + \sum_{\beta \notin \text{list}(v)} \delta_v(\beta) \cdot \mathsf{agr}\left(\pi'_2(v), \mathcal{C}(\beta)\right) \\
&\leq \sum_{\beta \in \text{list}(v)} \delta_v(\beta) \cdot \mathsf{agr}\left(\pi'_2(v), \mathcal{C}(\beta)\right) \quad + \quad \eta \\
&\leq \left(\max_\beta \delta_v(\beta) \right) \sum_{\beta \in \text{list}(v)} \mathsf{agr}\left(\pi'_2(v), \mathcal{C}(\beta)\right) \quad + \quad \eta \\
&\leq \delta_v \left(1 + \sum_{\beta_1 \neq \beta_2 \in \text{list}(v)} \mathsf{agr}\left(\mathcal{C}(\beta_1), \mathcal{C}(\beta_2)\right) \right) + \eta \quad \text{(by Fact A.3)} \\
&\leq \delta_v \left(1 + \binom{|\text{list}(v)|}{2} \cdot \eta^3 \right) + \eta \quad \leq \quad \delta_v + 3\eta \qquad \text{(by Claim A.4)}
\end{aligned}$$

□

Now, if we translate the above theorem to the language of robust PCPs, we have the following. Let $\mathcal{C} : \Sigma \to \sigma^k$ be a code with distance $1 - \eta^3$. Suppose L has a robust PCP with randomness r, query complexity q and robust soundness error δ over the alphabet Σ, then L has a robust PCP with randomness r, query complexity qk and robust soundness error $\delta + 3\eta$ over the alphabet σ. Now, setting $\eta = \delta/3$ and choosing a code given by Fact A.2, we have the following.

Theorem A.6 (alphabet reduction)**.** *Suppose L has a robust PCP with randomness r, query complexity q and robust soundness error δ over the alphabet Σ, then L has a robust PCP with randomness r, query complexity $O\left(q \log |\Sigma|/\delta^6\right)$ and robust soundness error 2δ over an alphabet of size at most $O\left(1/\delta^6\right)$.*

In other words, if the robust soundness error is δ, we might as well reduce the alphabet size to poly$(1/\delta)$ without affecting other parameters too much.

B Proof Degree Reduction via samplers

Suppose we are given a label-cover instance and suppose the average degree of vertices on the right is large. We wish to reduce this instance to another instance with small right-degree and not too much loss in the other parameters. The reduction would use objects called *samplers* which are defined as follows.

Definition B.1 (Sampler)**.** *A bipartite graph $H = (A, B, E)$ is said to be an (ε, δ)-sampler if for every subset $S \subseteq A$,*

$$\Pr_{b \in B}\left[\frac{|\Gamma(b) \cap S|}{|\Gamma(b)|} > \frac{|S|}{|A|} + \varepsilon\right] \leq \delta$$

In other words, neighbourhoods of most vertices b behave like a random sample of A, in the sense that their density within any fixed S is close to what is expected. Construction of such graphs is well-known and the following theorem is from a technical report by Oded Goldreich titled 'A Sample of Samplers'.

Theorem B.2. *[Gol97]There exists a polynomial time algorithm that, given an integer n and a parameter $\varepsilon > 0$, output an $(\varepsilon, \varepsilon^2)$-sampler with $|A| = |B| = n$ and the right degree $4/\varepsilon^4$.* □

With these in our arsenal, we shall go ahead with the degree reduction procedure. Suppose we are given a label cover instance $I = (U, V, \Sigma_1, \Sigma_2, \mathcal{F})$ such that the average right degree is D. We'll reduce it to an instance $I' = (U, V', \Sigma_1, \Sigma_2, \mathcal{F}')$ as follows:

- Vertices: For every vertex $v \in V$, let the degree of v be D_v. In V', each v is going to be expanded by $C_v = \{v\} \times [D_v]$, a cloud of D_v many vertices.
- Edges: For every v, place an (μ, μ^2)-sampler between $\Gamma(v)$ and C_v. Let d be the degree of the sampler.

- Constraints: The constraint on an edge between u an (v, i) is same as the constraint on (u, v) in the original graph.

All that is left to do is to show that completeness and soundness are more or less maintained. Of course, it is easy to observe that completeness is maintained — just use the honest labelling. The following lemma shows that soundness is maintained as well.

Lemma B.3. *If the instance I was at most δ-satisfiable, then the instance I' is at most $(4\mu + \delta)$-satisfiable.*

Proof. Let $L' = (L'_1, L'_2)$ be a labelling of I' that satisfies the largest fraction of constraints. From this define a randomized labelling of I in the most natural way:

> For every vertex $u \in U$, the label $L_1(u)$ is just $L'_1(u)$. For a vertex $v \in V$, pick a random element $(v, i) \in C_v$ and set $L_2(v) = L'_2(v, i)$.

For each $v \in V$, let δ_v denote the fraction of edges adjacent on v that are satisfied by this assignment.

$$\delta_v = \sum_{\sigma \in \Sigma_2} \Pr[L_2(v) = \sigma] \Pr_{u \in \Gamma(v)} [f(L_1(u)) = \sigma]$$

And since I is at most δ-satisfiable, we have that $\sum_v \delta_v D_v \leq \delta |E|$.

Claim B.4. *For each v, the fraction of edges in I' between C_v and $\Gamma(v)$ that are satisfied is at most $4\mu + \delta_v$.*

Before we prove this claim, let us see how this implies the lemma. Since the number of between C_v and $\Gamma(v)$ is dD_v, the number of edges between them that are satisfied is at most $dD_v(4\mu + \delta_v)$. Summing over all v's, we have

$$\begin{aligned} \#\text{ satisfied edges in } I' &\leq \sum_{v \in V} dD_v(4\mu + \delta_v) \\ &= 4\mu d \sum_{v \in V} D_v + d \sum_{v \in V} \delta_v D_v \\ &\leq 4\mu d|E| + d\delta|E| \\ &= d|E|(4\mu + \delta) = |E'|(4\mu + \delta) \end{aligned}$$

which proves the Lemma (modulo the proof of the claim). □

Proof of Claim B.4. Let $A = \Gamma(v)$ and let $B = C_v$. For any subset $S \subseteq \Sigma_2$, let $B(S)$ be the set of vertices in B that have been assigned a label in S. And let $A(S)$ be the set of vertices in A that have been assigned a label "compatible with S". That is,

$$A(S) = \left\{ u \in A \ : \ f_{(u,v)}(L_1(u)) \in S \right\} \quad , \quad B(S) = \{ v \in B \ : \ L_2(v) \in S \}$$

It is clear that if $\{S_1, \cdots, S_k\}$ is a partition of Σ_2, then the $\{A(S_1), \cdots, A(S_n)\}$ and $\{B(S_1), \cdots, B(S_n)\}$ are partitions of A and B respectively. We'll create such a partition with some additional properties.

Start with $\{A(\sigma)\}_{\sigma\in\Sigma_2}$. If $|A(\sigma)| > \mu|A|$ for any σ, keep $\{\sigma\}$ as a singleton in the partition. As for smaller sets, greedily keep grouping σ's into a sets S_i until $|A(S_i)| \geq \mu|A|$. Thus we have a partition of Σ_2 into sets $S_1, \cdots, S_k$ with the following properties:

- The parts are either singletons or larger sets. For every singleton $\{\sigma\}$ in the partition, the corresponding $A(\sigma)$ is at least $\mu|A|$.
- For every non-singleton S_i in the partition, $|A(S_i)| \leq 2\mu|A|$.
- The number of sets in the partition is at most $1/\mu$.

For brevity, let $A_i = A(S_i)$ and $B_i = B(S_i)$. Observe that the satisfied edges between A and B are all contained in $\bigcup E(A_i, B_i)$. We'll estimate $E(A_i, B_i)$ for each i to prove the claim. Since we put a sampler between A and B, it must be the case that almost every $b \in B$, the number of neighbours of b in A_i must be roughly as expected. However, there might be a few "bad" b's that have way too many neighbours in A_i, but thankfully these are few in number. Formally, for each i define the set B_i^* to be those b such that

$$\frac{|\Gamma(b) \cap A_i|}{|\Gamma(b)|} > \frac{|A_i|}{|A|} + \mu$$

By the property of the sampler, $|B_i^*| \leq \mu^2|B|$ and hence $\sum_i |B_i^*| \leq \frac{1}{\mu} \cdot \mu^2|B| = \mu|B|$. Therefore,

$$\begin{aligned}
|E(A_i, B_i)| &\leq |B_i^*|d \quad + \quad |B_i| \cdot d\left(\frac{|A_i|}{D_v} + \mu\right) \\
\implies \sum_i |E(A_i, B_i)| &\leq d\sum_i |B_i^*| \quad + \quad d\mu \sum_i |B_i| + \sum_i \frac{d \cdot |A_i||B_i|}{D_v} \\
&= d\mu|B| \quad + \quad d\mu|B| + \sum_i \frac{d \cdot |A_i||B_i|}{D_v} \\
&= dD_v\left(2\mu + \sum_i \frac{|A_i||B_i|}{D_v^2}\right) \\
&= dD_v\left(2\mu + \sum_{i:|S_i|=1} \frac{|A_i||B_i|}{D_v^2} + \sum_{i:|S_i|>1} \frac{|A_i||B_i|}{D_v^2}\right)
\end{aligned}$$

Lets focus on the singleton sets first. That sum is clearly bounded by $\sum_\sigma \frac{|A(\sigma)||B(\sigma)|}{D_v^2}$ which is precisely δ_v, the fraction of edges between A and B that are satisfied. Hence,

$$\begin{aligned}
\sum_i |E(A_i, B_i)| &\leq dD_v\left(2\mu + \delta_v + \sum_{i:|S_i|>1} \frac{|A_i||B_i|}{D_v^2}\right) \\
&\leq dD_v\left(2\mu + \delta_v + \sum_{i:|S_i|>1} \frac{2\mu D_v \cdot |B_i|}{D_v^2}\right) \\
&= dD_v\left(2\mu + \delta_v + 2\mu\right) = dD_v(4\mu + \delta_v)
\end{aligned}$$

And as mentioned earlier, $\bigcup E(A_i, B_i)$ contains all the edges between A and B that are satisfied and hence we are done. □

Thus, the degree reduction procedure can be summarized as the following theorem.

Theorem B.5. *For every $\mu > 0$, there is a polynomial time algorithm to reduce δ-gap label-cover instances $I = (U, V, \Sigma_1, \Sigma_2, \mathcal{F})$ of average right-degree D to $(4\mu + \delta)$-gap label-cover instances $I' = (U, V', \Sigma_1, \Sigma_2, \mathcal{F}')$ with right-degree $4/\mu^4$.* □