18.408 Topics in Theoretical Computer Science Fall 2022
Lecture 12

Dor Minzer

In this lecture, we will present a transformation on PCPs called aggregation of queries. This technique enables one to achieve the block property which we used in the last few lectures in order to perform composition of PCPs.

1 Aggregation of Queries

Suppose that we have a PCP construction with s queries, namely that we know that gap-CSG$[1, 1-\varepsilon]$ is NP-hard on instances with s queries and alphabet $\Sigma = \{0,1\}$ (you should think of s as $\mathsf{poly}(\log n)$). We recall that an instance of this problem is an s-uniform hypergraph $H = (V, E)$ along with a collection of constraints on the edges $\{C_e\}_{e\in E}$. A constraint C_e is a collection of tuples $(a_1, \ldots, a_s) \in \Sigma^s$ that are considered satisfactory, and the goal is to find an assignment $A\colon V \to \Sigma$ that satisfies as many of the constraints as possible, i.e. that maximizes

$$|\{e = (v_1, \ldots, v_s) \in E \mid (A(v_1), \ldots, A(v_s)) \in C_e\}|.$$

It is easy to see that all of the PCP constructions and tests we constructed in this course can be formalized in this way, and in particular the $\mathsf{poly}(\log n)$ query PCP we constructed.

The idea of aggregation of queries is to enlarge the set of vertices of the graph, so that for each edge $e \in E$, there will be a vertex v_e in the graph whose label will encode together the labels of all of the vertices in e. Thus, verifying the constraint on e will only require us to read the label of v_e.

One simple way to do that is to consider the following construction: define the bi-partite graph G whose vertices are $(V \cup V', E')$, where V is the original set of vertices of H and in V' we have a vertex v_e for each edge $e \in E$. We connected v and v_e by an edge if v is a vertex in the edge e in H. The alphabets of the CSG defined on G are Σ on V, and Σ_2 on V' where for each v_e we interpret a symbol from the alphabet as some tuple from C_e (i.e., as an assignment that satisfies e). The constraints on G are Φ_{v,v_e}, and given the label σ of v and c of c_e, we put (σ, c) in Φ_{v,v_e} if $c(v) = \sigma$, i.e. of the label assigned to v_e gives the variable v the value σ. This construction works with limited success, and you will see in the problem set that if H is satisfiable then G is satisfies, and if H is at most $(1-\varepsilon)$-satisfiable, then G is at most $(1-\varepsilon/s)$-satisfiable. Thus, if s is large (as in our case), this is not very good (but note that if s was constant, such transformation reduce the number of quires from s queries to 2 while maintaining the soundness bounded away from 1).

The issue with this approach is that we need to more effectively enforce that the new witness locations packing the values assigned to all the vertices in v are globally consistent, in the sense that a vertex v appearing in two different edges would be assigned the same value by them.

We resolve this issue by taking utilizing the idea of low-degree extensions and low-degree testing, again. In particular, we will think of the assignments to v_e as projections of low-degree polynomials over a large space and perform low-degree extensions to ensure global consistency of all of these polynomials.

1.1 Packing Into Curves

Let $n = |V|$, and identify V with $\mathbb{H}^m$ where $\mathbb{H}$ is a subset of a field $\mathbb{F}_q$ of size $h = |\mathbb{H}| = \log n$, $m = \frac{\log n}{\log\log n}$, and the field size q is $\log^{100} n$. Thus, we can identify an assignment $A\colon V \to \{0,1\}$ with an assignment $A\colon \mathbb{H}^m \to \{0,1\}$, and then consider its low-degree extension $A_{\mathsf{extension}}\colon \mathbb{F}_q^m \to \{0,1\}$ which is multi-variate polynomial of individual degree at most h.

Our new CSG will contain, as part of it, locations for each entry of $A_{\mathsf{extension}}$, and in particular we will want to design a test that ensures that a given table f of values is close to a truth table of a low-degree function. We have already seen how to solve this issue using the plane versus point encoding and the plane versus point test, so we may assume that f is close to a function of total degree at most mh.

Next, we will want to pack all of the values of the assignment A that are given to an edge $e \in E$ into a single table. Fix an edge $e \in E$, and let $v_1, \ldots, v_s \in \mathbb{H}^m$ be all of vertices of H that are in e. We intend to pack $v_1, \ldots, v_q$ into a single curve, defined as follows:

Definition 1.1. *A curve $\gamma\colon \mathbb{F}_q \to \mathbb{F}_q^m$ is a tuple of univariate polynomials, i.e. $\gamma(t) = (\gamma_1(t), \ldots, \gamma_m(t))$. The degree of a curve γ is $\mathsf{deg}(\gamma) = \max_i \mathsf{deg}(\gamma_i)$.*

We have the following basic interpolation claim.

Claim 1.2. *Let $a_1, \ldots, a_s \in \mathbb{F}_q$ be distinct, and $v_1, \ldots, v_s \in \mathbb{F}_q^m$. Then there is a curve $\gamma\colon \mathbb{F}_q \to \mathbb{F}_q^s$ of degree at most $s-1$ such that $\gamma(a_i) = v_i$ for $i = 1, \ldots, s$.*

Proof. By interpolation, for each $j = 1, \ldots, m$ we may find a univariate polynomial $\gamma_j\colon \mathbb{F}_q \to \mathbb{F}_q$ of degree at most $s-1$ such that $\gamma_j(a_i) = (v_i)_j$ for all $i = 1, \ldots, s$. The proof is concluded by taking $\gamma(t) = (\gamma_1(t), \ldots, \gamma_m(t))$. □

For each edge $e \in E$ given as $e = (v_1, \ldots, v_s)$ and an additional point $x \in \mathbb{F}_q^m$, by Claim 1.2 we may pick a curve $\gamma_{e,x}$ of degree at most s such that $\gamma_e(i) = v_i$ for $i = 1, \ldots, s$ and $x = \gamma_e(s+1)$. The idea is that the univariate function $A_{\mathsf{extension}} \circ \gamma_{e,x}$ then is a polynomial of degree at most $mhs = \mathsf{poly}(\log n)$, so to give all of the values of $A_{\mathsf{extension}}$ concerning the edge e at once we may simply give the restriction of $A_{\mathsf{extension}}$ to $\gamma_{e,x}$. As mhs is much smaller than q, our hope is that the properties of low-degree polynomials will enable us to ensure the global consistency.

We next describe the "aggregation of queries" transformation more precisely. Our CSG will have nodes for each entry in the points table A_0 and each entry in the planes table A_2, which are supposed to encode $A_{\mathsf{extension}}$. Also, for each edge $e \in H$ and point $x \in \mathbb{F}_q^m$ our CSG will have mhs nodes specifying a univariate polynomial $p_{e,x}$ of degree at most mhs, which is supposed to be $A_{\mathsf{extension}} \circ \gamma_{e,x}$. We next describe the test:

1. Perform the Plane versus Point test on A_0 and A_2. I.e. choose a point $x \in \mathbb{F}_q^m$ and a plane P containing it, and check that $A_0(x) = A_2[P](x)$.

2. Choose $e \in E$ an edge in H uniformly.

3. Sample a point $z \in \mathbb{F}_q^m$ and read off the coefficients of $p_{e,z}$ to construct a univariate polynomial of degree at most mhs.

4. Take $i \in \mathbb{F}_q \setminus \{1, \ldots, s+1\}$ randomly, compute $y = \gamma_{e,z}(i)$ and check that $p_{e,z}(i) = A_0(y)$.

5. Compute $\sigma_i = p_{e,z}(i)$ for each $i = 1, \ldots, s$, and check that $(\sigma_1, \ldots, \sigma_s)$ satisfy the constraint of e.

It is clear that the new PCP construction has size which is polynomial in the size of H, and that the run-time of the reduction is also polynomial. The following lemma addresses the completeness and soundness of the construction.

Lemma 1.3. *Denote by Ψ the CSG instance constructed above from H.*

1. *If H is satisfiable, then Ψ is satisfiable.*
2. *For all $\varepsilon > 0$, there is $\delta > 0$ such that if H is at most $(1-\varepsilon)$-satisfiable, then Ψ is at most $(1-\delta)$-satisfiable.*

Proof. The first item is clear, since we can take a satisfying assignment A of H and assign the tables A_0, A_2 truthfully according to the low-degree extension of A, and then assign the rest of the witness according to the coefficients of $A \circ \gamma_{e,x}$ for each $e \in E$ and $x \in \mathbb{F}_q^m$.

For the second item, we prove counter-positively that if there are tables A_0, A_2 and a table of coefficients that satisfy at least $1-\delta$ fraction of the constraints of Ψ, then there is an assignment to A satisfying more than $1-\varepsilon$ of the constraints of H.

To see that, first note that by the analysis of the Plane versus Point test that as $A_0(x) = A_2[P](x)$ with probability at least $1-\delta$, it follows that there is a polynomial $f\colon \mathbb{F}_q^m \to \mathbb{F}_q$ of degree at most mhs such that $\Pr_{x\in\mathbb{F}_q^m}[f(x) = A_0(x)] \geqslant 1-\delta-\frac{mhs}{q^{1/10}} \geqslant 1-2\delta$, and we fix f henceforth.

By an averaging argument, for at least $1-\sqrt{\delta}$ of the edges $e \in E$, the probability the test passes conditioned on choosing e is at least $1-\sqrt{\delta}$, and we show that A satisfies each such e. This would finish the proof as $1-\sqrt{\delta} > 1-\varepsilon$.

Fix e, and note that over the randomness of the choice of z, the distribution of $\gamma_{e,z}(i)$ for each $i \in \mathbb{F}_q \setminus \{1,\ldots,s+1\}$ is uniform in $\mathbb{F}_q^m$, so we get that $A_0(y) = f(y)$ with probability $1-2\delta$. Thus, we get that

$$\Pr_{z,i}[f \circ \gamma_{e,z}(i) = p_{e,z}(i) \wedge \text{rest of the test succeeds}] \geqslant 1-3\delta,$$

so there is some z such that $\Pr_i[f \circ \gamma_{e,z}(i) = p_{e,z}(i)] \geqslant 1-3\delta$. As $f \circ \gamma_{e,z}$, and $p_{e,z}$ are univariate polynomials of degree at most mhs, it follows from the Schwarz-Zippel lemma that $f \circ \gamma_{e,z} \equiv p_{e,z}$, and from the test of the test we get that the values $\sigma_i = p_{e,x}(i) = f(\gamma_{e,z}(i)) = f(v_i)$ for $i = 1,\ldots,s$ satisfy the constraint e.

Therefore, defining $A\colon \mathbb{H}^m \to \{0,1\}$ by taking $A(v) = f(v)$ if $f(v) \in \{0,1\}$ and arbitrarily otherwise, we get that A satisfies at least $1-\sqrt{\delta}$ of the constraints, and we are done. □

1.2 The Block Property

We finish the lecture by observing that the above transformation gave us the block property. Indeed, each symbol of the tables A_0 and A_2 will be its own block, and for each $e \in E$ and $x \in \mathbb{F}_q^m$ we have a single block containing all of the coefficients of $p_{e,x}$. Note that these blocks are disjoint, and our tester looked at 4 blocks. We also note that the total number of queries made is $O(mhs)$, therefore we only incurred a polynomial blow-up in the query complexity while keeping the soundness bounded away from 1 and achieving the block property. Summarizing, in the main features of the aggregation of queries that we used are:

1. Soundness and completeness: the transformation keeps perfect completeness, and if the original soundness was bounded away from 1, then the soundness after the transformation is still bounded away from 1.

2. Query complexity: if the original query complexity was s, then the new query complexity will be $s' = s\mathsf{poly}(\log n)$. Thus, in the case that s was poly-logarithmic to begin with, we only incur a polynomial blow-up in the query complexity.

3. The block property: the new CSG instance has the (k, s') block property for $k = 4$.

2 Some More Words on Composition

One way to think of the aggregation of queries technique is that it reduces us to checking that some univariate polynomial $p_{e,x}$ satisfies some constraint (in the case above, that the values $p_{e,x}(1), \ldots, p_{e,x}(s)$ satisfy some constraint C_e), and that we managed to ensure global consistency using the low-degree test and the tables A_0, A_2.

Thus, we have effectively reduced our original problem to a similar looking problem of smaller scale: we want to verify that the values of assignment $g\colon X' \to \{0,1\}$ (which you can think of as encoding the coefficients of $p_{e,x}$) satisfies some constraint C_e. The main differences are (1) the domain X' is of much smaller size, and more specifically $n' = \mathsf{poly}(\log n)$, and (2) we need to check that the values of the assignment g are consistent with A_0. In light of (1), one may expect that we should be able to run the algebraic PCP construction restricted to the domain X', namely re-interpreting that as quadratic equations, running the sum-check protocol and using the low-degree test again to further reduce the number of queries from n' to $\mathsf{poly}(\log n') = \mathsf{poly}(\log\log n)$ time.

This is indeed possible, thankfully to the block property, in a similar manner to the composition step we did with the Hadamard code. Drawing further analogies, point (2) above is analogous to the fact we needed to check that our Hadamard encodings satisfy some quadratic equation, and indeed it can be achieved by our algebraic PCP. The details of this construction though get rather hairy and hence are omitted, but by now you have all of the tools and ideas necessary to prove the PCP theorem from scratch.