COSC 544 Probabilistic Proof Systems 10/31/17

GGPR: A Linear PCP Of Size $|\mathbb{F}|^{O(S)}$

Lecturer: Justin Thaler

1 A Linear PCP Of Size $O(|\mathbb{F}|^S)$ for Arithmetic Circuit-SAT

In a breakthrough result, Gennaro, Gentry, Parno, and Raykova [GGPR13] gave a linear PCP for non-deterministic circuit evaluation of size $O(|\mathbb{F}|^S)$, referring to their linear PCP as a *Quadratic Arithmetic Program* (QAP).[1,2] The QAPs of [GGPR13] have been highly influential, and form the foundation of many of the implementations of argument systems.

QAPs use the same constraint-based formalism as the linear PCP described in the previous lecture. Recall that there are $\ell = S + |y| - |w|$ constraints $Q_i(W) = 0$, where Q_i is a polynomial in the variables of W that always takes one of three forms. The three forms are: (1) $W_i - c_i = 0$ for some constant c_i depending on the input x or outputs y, (2) $W_i - (W_j \cdot W_k) = 0$, or (3) $W_i - (W_j + W_k) = 0$. A crucial observation is that in all three cases, Q_i can always be written in the form $f_{1,i}(W) \cdot f_{2,i}(W) - f_{3,i}(W) = 0$, for some linear functions $f_{1,i}$, $f_{2,i}$, and $f_{3,i}$. This is a stronger notion of structure than was exploited in the previous lecture (the previous lecture only exploited that each constraint is a polynomial in W of total degree at most 2).

Let $H := \{\sigma_1, \dots, \sigma_\ell\}$ be arbitrary distinct values in $\mathbb{F}$.

For each gate i in $\mathcal{C}$, define three univariate polynomials A_i, B_i, and C_i, each of degree $\ell - 1$, via interpolation as follows.

$$A_i(\sigma_j) = \text{the coefficient of } W_i \text{ in } f_{1,j}.$$

$$B_i(\sigma_j) = \text{the coefficient of } W_i \text{ in } f_{2,j}.$$

$$C_i(\sigma_j) = \text{the coefficient of } W_i \text{ in } f_{3,j}.$$

Finally, define via interpolation 3 univariate polynomials of degree $S - 1$ via interpolation as follows.

$$A'(\sigma_j) = \text{the constant term in } f_{1,j}.$$

$$B'(\sigma_j) = \text{the constant term in } f_{2,j}.$$

$$C'(\sigma_j) = \text{the constant term in } f_{3,j}.$$

Let $g_{x,y,W}(t)$ denote the univariate polynomial

$$g_{x,y,W}(t) = \left(\left(\sum_{\text{gates } i \text{ in } \mathcal{C}} W_i \cdot A_i(t) \right) + A'(t) \right) \cdot \left(\left(\sum_{\text{gates } i \text{ in } \mathcal{C}} W_i \cdot B_i(t) \right) + B'_i(t) \right) - \left(\left(\sum_{\text{gates } i \text{ in } \mathcal{C}} W_i \cdot C_i(t) \right) + C'_i(t) \right).$$

By design, $g_{x,y,W}$ vanishes on H if and only all constraints are satisfied, i.e., if and only if W is a correct transcript for $\{\mathcal{C}, x, y\}$.

[1] The argument system of Gennaro et al. can be understood in multiple ways, and [GGPR13] did not present it within the framework of linear PCPs. Subsequent work [SBV$^+$13, BCI$^+$13] identified QAPs as an example of a linear PCP.

[2] The focus of Gennaro et al. [GGPR13] was on the development of non-interactive argument systems satisfying various additional properties, such as zero-knowledge. We will describe such non-interactive arguments in the next lecture. The QAP-based interactive argument from this section was described and implemented by Setty et al. [SBV$^+$13].

Note that checking whether checking whether $g_{x,y,W}$ vanishes on H is very similar to the core statement checked in our MIP from Lecture 14. There, we checked that a *multivariate* polynomial derived from x,y, and W vanished on all Boolean inputs. Here, we are checking whether a univariate polynomial $g_{x,y,W}$ vanishes on all inputs in a pre-specified set H. We will rely on the following key lemma.

Lemma 1.1 ([BS08]). *Let $h_H(t) = \prod_{j=1}^{\ell}(t-\sigma_j)$. A degree d univariate polynomial $g_{x,y,W}(z)$ vanishes on H if and only if the polynomial $h_H(t)$ divides $g_{x,y,W}(z)$, i.e., if and only if there exists a polynomial h^* with* $\deg(h^*) \leq d - |H|$ *such that $g_{x,y,W}(z) = h_H(z) \cdot h^*(z)$.*

Proof. If $g_{x,y,W}(z) = h_H(z) \cdot h^*(z)$, then for any $\alpha \in H$, it holds that $g_{x,y,W}(\alpha) = h_H(\alpha) \cdot h^*(\alpha) = 0 \cdot \alpha = 0$, so $g_{x,y,W}$ indeed vanishes on H.

For the other direction, observe that if $g_{x,y,W}(\alpha) = 0$, then the polynomial $(z-\alpha)$ divides $g_{x,y,W}(z)$. It follows immediately that if $g_{x,y,W}$ vanishes on H, then $g_{x,y,W}$ is divisible by h_H. □

By inspection, the degree of the polynomial $g_{x,y,W}$ is at most $d = 2(\ell-1)$, where $\ell = |S| + |y| - |w|$ is the number of constraints. By Lemma 1.1, to convince $\mathcal{V}$ that $g_{x,y,Z}$ vanishes on H, the proof merely needs to convince $\mathcal{V}$ that $g_{x,y,Z}(z) = h_H(z) \cdot h^*(z)$ for some polynomial h^* of degree $d - |H| = \ell - 1$. To be convinced of this, $\mathcal{V}$ can pick a random point $r \in \mathbb{F}$ and check that

$$g_{x,y,Z}(r) = h_H(r) \cdot h^*(r). \tag{1}$$

Indeed, because any two distinct degree $(\ell-1)$ polynomials can agree on at most $d+1$ points, if $g_{x,y,Z} \neq h_H \cdot h^*$, then this equality will fail with probability at least $1-(\ell-1)/|\mathbb{F}|$.

To this end, a correct proof represents two linear functions. The first is $f_{\text{coeff}(h^*)}$, where $\text{coeff}(h^*)$ denotes the vector of coefficients of h^* The second is f_W. Note that $f_{\text{coeff}(h^*)}(1, r, r^2, \ldots, r^S) = h^*(r)$, so $\mathcal{V}$ can evaluate $h^*(r)$ with a single query to the proof. Similarly, $\mathcal{V}$ can evaluate $g_{x,y,W}$ at r by evaluating f_W at the three vectors $(A_1(r), \ldots, A_S(r))$, $(B_1(r), \ldots, B_S(r))$, and $(C_1(r), \ldots, C_S(r))$.

Just as in the linear PCP of the previous section, the verifier also has to perform linearity testing on $f_{\text{coeff}(h^*)}$ and f_W. The verifier must also replace the four queries described above with two queries each to ensure that all queries are uniformly distributed.

Protocol Costs. The costs of the argument system obtained by combining QAPs with the commitment protocol are summarized in Table 1. The honest prover $\mathcal{P}$ needs to perform the following steps, assuming $\mathcal{P}$ knows a witness w for $\mathcal{C}$. First, evaluate $\mathcal{C}$ gate-by-gate to find a correct transcript W. Second, compute the polynomial $g_{x,y,W}(t)$. Third, divide $g_{x,y,W}$ by h_H to find the quotient polynomial h^*. Fourth run the linear commitment/reveal protocol described in Lecture 16, to commit to $f_{\text{coeff}(h^*)}$ and f_W and answer the verifier's queries.

The first and fourth steps can clearly be done in time $O(S)$. The second step can be done in time $O(S \log^2 S)$ using standard FFT-based multipoint interpolation algorithms. The third step can be done in time $O(S \log S)$ using FFT-based polynomial division algorithms.

$\mathcal{V}$'s time and $\mathcal{P}$'s time are both $\tilde{\Theta}(S)$, but if $\mathcal{V}$ is simultaneously verifying $\mathcal{C}$'s execution over a large batch of inputs, then the $\Theta(S)$ cost for $\mathcal{V}$ can be amortized over the entire batch. Total communication from $\mathcal{V}$ to $\mathcal{P}$ is $\Theta(S)$ as well (this cost can also be amortized), but the communication in the reverse direction is just a constant number of field elements per input.

$\mathcal{V} \to \mathcal{P}$ Communication	$\mathcal{P} \to \mathcal{V}$ Communication	Queries	$\mathcal{V}$ time	$\mathcal{P}$ time
$O(S)$ field elements	$O(1)$ field elements	$O(1)$	$\tilde{O}(S)$	$\tilde{O}(S)$

Table 1: Costs of the argument system from Section 1 when run on a non-deterministic circuit $\mathcal{C}$ of size S. The $\tilde{O}$ notation hides polylogarithmic factors in S. Note that the verifier's cost and the communication cost can be amortized when outsourcing $\mathcal{C}$'s execution on a *batch* of inputs. The stated bound on $\mathcal{P}$'s time assumes $\mathcal{P}$ knows a witness w for $\mathcal{C}$.

References

[BCI+13] Nir Bitansky, Alessandro Chiesa, Yuval Ishai, Rafail Ostrovsky, and Omer Paneth. Succinct non-interactive arguments via linear interactive proofs. In *TCC*, pages 315–333, 2013.

[BS08] Eli Ben-Sasson and Madhu Sudan. Short PCPs with polylog query complexity. *SIAM J. Comput.*, 38(2):551–607, 2008.

[GGPR13] Rosario Gennaro, Craig Gentry, Bryan Parno, and Mariana Raykova. Quadratic span programs and succinct NIZKs without PCPs. In *EUROCRYPT*, pages 626–645, 2013.

[SBV+13] Srinath T. V. Setty, Benjamin Braun, Victor Vu, Andrew J. Blumberg, Bryan Parno, and Michael Walfish. Resolving the conflict between generality and plausibility in verified computation. In Zdenek Hanzálek, Hermann Härtig, Miguel Castro, and M. Frans Kaashoek, editors, *EuroSys*, pages 71–84. ACM, 2013.