Math Building Blocks of PLONK

rkm0959

Open Source Contributon: Advanced Track

July 22nd

Arithmetic Circuits $\rightarrow$ Vectors

Focus: How do we *encode* an arithmetic circuit?

In PLONK, we encode an arithmetic circuit as follows -

$$q_{M,i}x_{a_i}x_{b_i} + q_{L,i}x_{a_i} + q_{R,i}x_{b_i} + q_{O_i}x_{c_i} + q_{C,i} = 0$$

By using *switches* $q_{M,i}, q_{L,i}, q_{R,i}, q_{O,i}, q_{C,i}$, we may write

- addition, subtraction, multiplication gates
- boolean constraints, public input constraints

Arithmetic Circuits $\rightarrow$ Vectors

Looking back at the equation

$$q_{M,i}x_{a_i}x_{b_i} + q_{L,i}x_{a_i} + q_{R,i}x_{b_i} + q_{O_i}x_{c_i} + q_{C,i} = 0$$

The vectors of interest are

- selectors $q_M = (q_{M,0}, \cdots, q_{M,n-1})$ and similar for q_L, q_R, q_O, q_C
- The left values $x_L = (x_{a_0}, x_{a_1}, \cdots, x_{a_{n-1}})$ and similar for x_R, x_O

Vectors $\rightarrow$ Polynomials

Focus: The isomorphism, known as *Lagrange Interpolation*.

Work in $\mathbb{F}_q$ with $q \equiv 1 \pmod{n}$. Select $\omega \in \mathbb{F}_q$ as a nth root of unity.
Denote $Ann_S(x) = \prod_{w \in S}(x - w)$. We map the vector $(v_0, v_1, \cdots, v_{n-1})$ to

$$P(x) = \sum_{i=0}^{n-1} v_i \cdot \frac{Ann_{S \setminus \{\omega^i\}}(x)}{Ann_{S \setminus \{\omega^i\}}(\omega^i)}$$

where $S = \{1, \omega, \omega^2, \cdots, \omega^{n-1}\}$ is the subgroup generated by ω.

Note that $Ann_S(x)$ annhilates S and hence $P(\omega^i) = v_i$.
We convert our vectors into polynomials and abuse the notation.

Polynomials are Very Useful

Focus: High School Math Strikes Again. $(x - a)|(P(x) - P(a))$.

Now we aim to prove

$$q_M f_L f_R + q_L f_L + q_R f_R + q_O f_O + f_C = 0$$

holds over $S = \{1, \omega, \cdots, \omega^{n-1}\}$.

This is equivalent to showing

$$q_M f_L f_R + q_L f_L + q_R f_R + q_O f_O + f_C \equiv 0 \pmod{Ann_S(x)}$$

Knowledge $\rightarrow$ Polynomial Identity

Focus: In the end, we have a goal: a polynomial identity.

So we want to show

$$q_M f_L f_R + q_L f_L + q_R f_R + q_O f_O + f_C \equiv 0 \pmod{Ann_S}$$

which is simply equivalent to the existence of t such that

$$q_M f_L f_R + q_L f_L + q_R f_R + q_O f_O + f_C = Ann_S \cdot t$$

which is a polynomial identity!

Polynomial Identity $\rightarrow$ Single Point Evaluation

Focus: Polynomial Identity is Hard to Test. Schwartz-Zippel Lemma helps.

Schwartz-Zippel Lemma

Let $P \in F[x_1, \cdots, x_n]$ be a non-zero polynomial with total degree d over finite field F. Then, with r_i are selected i.i.d. uniformly over F,

$$Pr[P(r_1, \cdots, r_n) = 0] \leq \frac{d}{|F|}$$

Therefore, testing on a random point (Fiat-Shamir in practice) is enough.

This is extremely useful, and is used in many other areas in TCS.

Polynomial Commitments: [KZG10]

Focus: Prover should "commit" f_L, f_R, f_O in advance. We have [KZG10].

We want a *polynomial commitment scheme* that does the following

- It should be hiding and binding.
- It should be easy to do evaluation proofs.
- Extra: degree conditions, batchable, updateable, etc..

Homomorphic Hiding

Focus: We use the isomorphism $\mathbb{F}_p \to G$ & DLP & pairings.

From DLP, $a \to ag$ is an one-way homomorphism.

In certain groups, *pairings* exist -

Bilinear Pairings

Let q be a prime and G_1, G_2, G_T be groups of order q.
A pairing is a map $e : G_1 \times G_2 \to G_T$ such that

- $\forall a, b \in \mathbb{F}_q$ and $P \in G_1,\ Q \in G_2,\ e(aP, bQ) = ab \cdot e(P, Q)$
- e is non-trivial, but still efficiently computable

Polynomial Commitments: [KZG10]

Start with pairing $e : G_1 \times G_2 \to G_T$.
Denote $[x]_1 = xg_1$ and $[x]_2 = xg_2$ to be hidings of x on G_1, G_2.

For some secret s, start with SRS $[1]_1, [s]_1, \cdots, [s^n]_1, [1]_2, [s]_2$.

- **Commit**: Commitment of $f(x)$ is $[f(s)]_1$. Note $\deg f \leq n$.
- **Evaluate**: To prove $v = f(t)$, reveal $[(f(s) - v)/(s - t)]_1$.

$$e\left(\left[\frac{f(s) - v}{s - t}\right]_1, [s - t]_2\right) = e(\mathcal{C} \cdot [-v]_1, [1]_2)$$

Permutation Checks: Yet Another Schwartz-Zippel Lemma

Focus: Idea is simple. Proof itself is just Schwartz-Zippel.
To prove that $g(\omega^i) = f(\omega^{\sigma(i)})$, we choose β, γ randomly and

$$\prod_{i=0}^{n-1} \left(f(\omega^i) + \beta \cdot i + \gamma\right) = \prod_{i=0}^{n-1} \left(g(\omega^i) + \beta \cdot \sigma(i) + \gamma\right)$$

The check is very intuitive, and the proof is really just Schwartz-Zippel.

Hiding: Don't Forget Zero Knowledge!

Focus: This is ZK. You have to hide f_L, f_R, f_O via *blinding*.

It can be proved that for a polynomial $p(x)$ opened k times, doing

$$p(x) \leftarrow p(x) + Ann_S(x) \cdot bl(x)$$

blinds p where $bl(x)$ is a random polynomial of degree k.

This is under DLP's hardness. For proof see PLONKUP paper.

More KZG: Single Polynomial, Multiple Points

Focus: Polynomial Division still works.

To prove evaluations of f over S, write

$$f(x) = Ann_S(x)q(x) + r(x)$$

the evaluations are encoded in $r(x)$, and $Ann_S(x)$ is public.

The proof is simply $[q(s)]_1$, and we may check

$$e([q(s)]_1, [Ann_S(s)]_2) = e(\mathcal{C} \cdot [-r(s)]_1, [1]_2)$$

More KZG: Multiple Polynomials, Single Point

Focus: Random Linear Combinations, but with single variable.

Due to the bilinearity of pairings, we may batch the proofs into one.
We check multiple equations at once with random linear combinations.

However, we may choose our combination as $1, r, r^2, \cdots, r^k$ for random r.
The proof is more or less Schwartz-Zippel style, check [Claim 4.6].

More KZG: Maller's Optimization

Focus: Exploit KZG commitments' linearity to its fullest.

To prove the evaluation of $f_1 f_2 - f_3$ on t, we do not need to send a proof for each f_1, f_2, f_3 - we may simply send one for f_1 and $f_1(t)f_2 - f_3$.

This is because the KZG commitments are *linear*.

Thank You! Any Questions?