The GKR method

Ariel Gabizon
Zeta Function Technologies

Overview

- Mutlilinear functions and sumcheck basics
- GKR - motivation and example.

Multilinear polynomials

Polynomials that are linear in each variable:

Multilinear polynomials

Polynomials that are linear in each variable:

Example: equality function

Multilinear polynomials

Polynomials that are linear in each variable:

Example: equality function

$$eq(x, y) = \prod_{i \in [n]} (x_i y_i + (1 - x_i)(1 - y_i))$$

Multilinear polynomials

Polynomials that are linear in each variable:

Example: equality function

$$eq(x, y) = \prod_{i \in [n]} (x_i y_i + (1 - x_i)(1 - y_i))$$

For $x, y \in \{0, 1\}^n$, $eq(x, y) = 1$ iff $x = y$.

Multilinear polynomials

Polynomials that are linear in each variable:

Example: equality function

$$eq(x, y) = \prod_{i \in [n]} (x_i y_i + (1 - x_i)(1 - y_i))$$

For $x, y \in \{0, 1\}^n$, $eq(x, y) = 1$ iff $x = y$.

"Multilinear Lagranges": $L_x(Y) = eq(x, Y)$ for some $x \in \{0, 1\}^n$.

Multilinear polynomials

Polynomials that are linear in each variable:

Example: equality function

$$eq(x, y) = \prod_{i \in [n]} (x_i y_i + (1 - x_i)(1 - y_i))$$

For $x, y \in \{0, 1\}^n$, $eq(x, y) = 1$ iff $x = y$.

"Multilinear Lagranges": $L_x(Y) = eq(x, Y)$ for some $x \in \{0, 1\}^n$.

We have $L_x(x) = 1$ and $L_x(y) = 0$ for any $y \neq x$ in $\{0, 1\}^n$.

Sumcheck basics

$\mathcal{P}$ has n-variate poly f of degree ≤ 3 in each variable.

Sumcheck basics

$\mathcal{P}$ has n-variate poly f of degree ≤ 3 in each variable.

Wants to prove to $\mathcal{V}$

$$\sum_{x \in \{0,1\}^n} f(x) = 0$$

Sumcheck basics

$\mathcal{P}$ has n-variate poly f of degree ≤ 3 in each variable.

Wants to prove to $\mathcal{V}$

$$\sum_{x \in \{0,1\}^n} f(x) = 0$$

The [LFKN] **sumcheck protocol** between $\mathcal{P}$ and $\mathcal{V}$ reduces this claim to claim of form $f(r) = v$ for random $r \in \mathbb{F}^n$.

Sumcheck basics

$\mathcal{P}$ has n-variate poly f of degree ≤ 3 in each variable.

Wants to prove to $\mathcal{V}$

$$\sum_{x\in\{0,1\}^n} f(x) = 0$$

The [LFKN] **sumcheck protocol** between $\mathcal{P}$ and $\mathcal{V}$ reduces this claim to claim of form $f(r) = v$ for random $r \in \mathbb{F}^n$.

Reduction doesn't require $\mathcal{P}$ to do FFT's or commit to other polynomials

Main application: Zero Testing

- $\mathcal{P}$ wants to prove to $\mathcal{V}$ that $\mathbf{f}(x) = \mathbf{0}$, $\forall x \in \{\mathbf{0}, \mathbf{1}\}^n$.

Main application: Zero Testing

- $\mathcal{P}$ wants to prove to $\mathcal{V}$ that $f(x) = 0$, $\forall x \in \{0, 1\}^n$.
- $\mathcal{V}$ chooses random $\beta \in \mathbb{F}^n$.
- Define $f'(X) := eq(\beta, X) f(X)$.

Main application: Zero Testing

- $\mathcal{P}$ wants to prove to $\mathcal{V}$ that $f(x) = 0$, $\forall x \in \{0, 1\}^n$.
- $\mathcal{V}$ chooses random $\beta \in \mathbb{F}^n$.
- Define $f'(X) := eq(\beta, X) f(X)$.
- $\mathcal{P}$ shows using sumcheck protocol that $\sum_{x \in \{0,1\}^n} f'(x) = 0$. This implies desired claim on f w.h.p.

Polynomials commitment schemes[KZG] (for mutlilinear polynomials)

- $\mathcal{P}$ sends short commitment $\mathbf{cm}(\mathbf{h})$ to $\mathfrak{n}$-variate mutlilinear polynomial $\mathbf{h}$.

Polynomials commitment schemes[KZG] (for mutlilinear polynomials)

- $\mathcal{P}$ sends short commitment $\mathbf{cm}(\mathbf{h})$ to $\mathbf{n}$-variate mutlilinear polynomial $\mathbf{h}$.
- Later $\mathcal{V}$ chooses $\mathbf{r} \in \mathbb{F}^{\mathbf{n}}$.

Polynomials commitment schemes[KZG] (for mutlilinear polynomials)

- $\mathcal{P}$ sends short commitment $\mathbf{cm}(h)$ to n-variate mutlilinear polynomial h.
- Later $\mathcal{V}$ chooses $r \in \mathbb{F}^n$.
- $\mathcal{P}$ sends back $z = f(r)$; together with *short* proof $\mathbf{open}(f, r)$ that z is correct.

Polynomials commitment schemes[KZG] (for mutlilinear polynomials)

- $\mathcal{P}$ sends short commitment $\mathbf{cm}(\mathbf{h})$ to $\mathbf{n}$-variate mutlilinear polynomial $\mathbf{h}$.
- Later $\mathcal{V}$ chooses $\mathbf{r} \in \mathbb{F}^{\mathbf{n}}$.
- $\mathcal{P}$ sends back $z = \mathbf{f}(\mathbf{r})$; together with *short* proof $\mathbf{open}(\mathbf{f}, \mathbf{r})$ that z is correct.

State of the art: Basefold, Binius, Brakedown, Gemini,Zeromorph,...

Zero Testing - typical example

$\mathcal{P}$ has multilinears $\mathbf{f_1, f_2, f_3}$. $\mathcal{V}$ has $\mathbf{cm}(\mathbf{f_1}), \mathbf{cm}(\mathbf{f_2}), \mathbf{cm}(\mathbf{f_3})$.

Zero Testing - typical example

$\mathcal{P}$ has multilinears $\mathbf{f_1}, \mathbf{f_2}, \mathbf{f_3}$. $\mathcal{V}$ has $\mathbf{cm}(\mathbf{f_1}), \mathbf{cm}(\mathbf{f_2}), \mathbf{cm}(\mathbf{f_3})$.

$\mathcal{P}$ wants to prove to $\mathcal{V}$ that

$$\forall x \in \{\mathbf{0}, \mathbf{1}\}^n : \mathbf{f_1}(x)\mathbf{f_2}(x) - \mathbf{f_3}(x) = \mathbf{0}.$$

GKR Motivation

GKR = "Delegating Computation: Interactive Proofs for Muggles" by Goldwasser, Kalai and Rothblum.

GKR Motivation

GKR="Delegating Computation: Interactive Proofs for Muggles" by Goldwasser, Kalai and Rothblum.

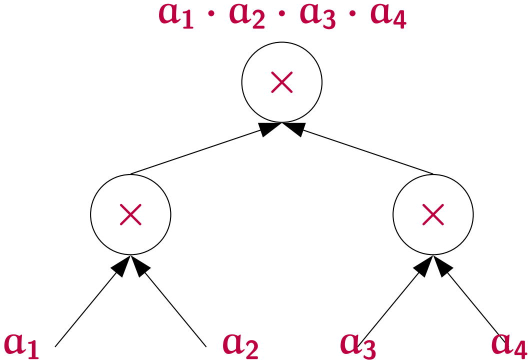
Committing to polynomials is expensive. Can we use sumcheck for polynomials we **don't** *have a commitment to?*

GKR idea - iterative sumcheck

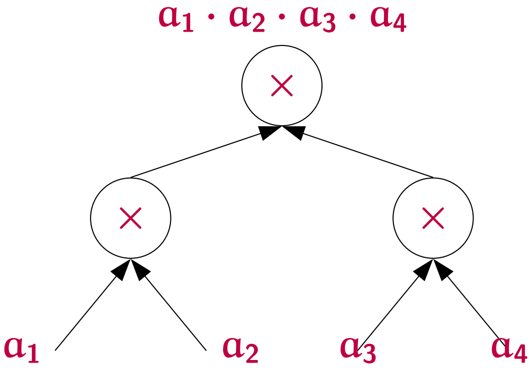
When we don't have a commitment to the polynomial we're summing, reduce the random evaluation at the end to *another* sumcheck over a different polynomial

$$\textbf{sum} \xrightarrow{\text{sumcheck}} \textbf{randeval} \xrightarrow{\text{reduction}} \textbf{sum} \xrightarrow{\text{sumcheck}} \dots$$

Example from [Thaler13]

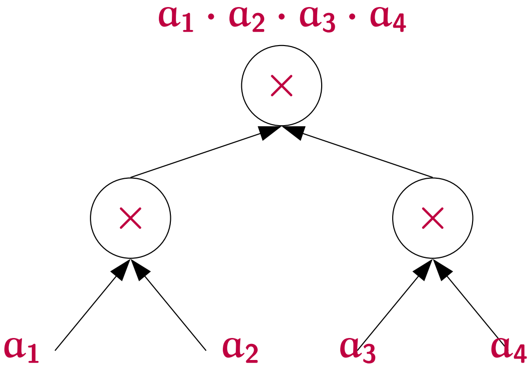


Example from [Thaler13]



$\mathcal{P}$ has $\mathbf{f}(\mathbf{Y_1}, \mathbf{Y_2})$. $\mathcal{V}$ has $\mathbf{cm}(\mathbf{f})$

Example from [Thaler13]



$\mathcal{P}$ has $f(Y_1, Y_2)$. $\mathcal{V}$ has $\mathbf{cm}(f)$

Wants to prove to $\mathcal{V}$ correctness of
$u := f(0,0) \cdot f(0,1) \cdot f(1,0) \cdot f(1,1)$.

Define multilinear "Intermediate layer function" $\mathbf{g}$:

$\mathbf{g}(\mathbf{0}) := \mathbf{f}(\mathbf{0}, \mathbf{0}) \cdot \mathbf{f}(\mathbf{0}, \mathbf{1})$

$\mathbf{g}(\mathbf{1}) := \mathbf{f}(\mathbf{1}, \mathbf{0}) \cdot \mathbf{f}(\mathbf{1}, \mathbf{1})$.

Define multilinear "Intermediate layer function" $\mathbf{g}$:
$\mathbf{g}(\mathbf{0}) := \mathbf{f}(\mathbf{0}, \mathbf{0}) \cdot \mathbf{f}(\mathbf{0}, \mathbf{1})$
$\mathbf{g}(\mathbf{1}) := \mathbf{f}(\mathbf{1}, \mathbf{0}) \cdot \mathbf{f}(\mathbf{1}, \mathbf{1})$.

Our claim is $\mathbf{g}(\mathbf{0}) \cdot \mathbf{g}(\mathbf{1}) = \mathbf{u}$.

Define multilinear “Intermediate layer function” $\mathbf{g}$:
$\mathbf{g}(\mathbf{0}) := \mathbf{f}(\mathbf{0}, \mathbf{0}) \cdot \mathbf{f}(\mathbf{0}, \mathbf{1})$
$\mathbf{g}(\mathbf{1}) := \mathbf{f}(\mathbf{1}, \mathbf{0}) \cdot \mathbf{f}(\mathbf{1}, \mathbf{1})$.

Our claim is $\mathbf{g}(\mathbf{0}) \cdot \mathbf{g}(\mathbf{1}) = \mathbf{u}$.

Exercise: Can reduce this to evaluating $\mathbf{g}(\mathbf{r})$ for one random $\mathbf{r} \in \mathbb{F}$.

Define multilinear "Intermediate layer function" $\mathbf{g}$:
$\mathbf{g}(\mathbf{0}) := \mathbf{f}(\mathbf{0}, \mathbf{0}) \cdot \mathbf{f}(\mathbf{0}, \mathbf{1})$
$\mathbf{g}(\mathbf{1}) := \mathbf{f}(\mathbf{1}, \mathbf{0}) \cdot \mathbf{f}(\mathbf{1}, \mathbf{1})$.

Our claim is $\mathbf{g}(\mathbf{0}) \cdot \mathbf{g}(\mathbf{1}) = \mathbf{u}$.

Exercise: Can reduce this to evaluating $\mathbf{g}(\mathbf{r})$ for one random $\mathbf{r} \in \mathbb{F}$.

Main goal: Avoid needing to compute $\mathbf{cm}(\mathbf{g})$ as "traditional SNARKs" would do!

Interlude: Representing mutlilinear functions via $\mathbf{eq}$

Recall

$$\mathbf{eq}(x, y) = \prod_{i \in [n]} (x_i y_i + (1 - x_i)(1 - y_i))$$

Interlude: Representing mutlilinear functions via eq

Recall

$$eq(x, y) = \prod_{i \in [n]} (x_i y_i + (1 - x_i)(1 - y_i))$$

Claim: When h is multilinear, we have for any r

$$h(r) = \sum_{x \in \{0,1\}^n} eq(r, x) h(x)$$

Heart of GKR - representing $g(r)$ as sum over f

$$g(r) = eq(r,0)f(0,0)f(0,1) + eq(r,1)f(1,0)f(1,1$$

Heart of GKR - representing $\mathbf{g}(\mathbf{r})$ as sum over $\mathbf{f}$

$$\mathbf{g}(\mathbf{r}) = \mathbf{eq}(\mathbf{r},\mathbf{0})\mathbf{f}(\mathbf{0},\mathbf{0})\mathbf{f}(\mathbf{0},\mathbf{1}) + \mathbf{eq}(\mathbf{r},\mathbf{1})\mathbf{f}(\mathbf{1},\mathbf{0})\mathbf{f}(\mathbf{1},\mathbf{1}$$

$$= \sum_{x \in \{\mathbf{0},\mathbf{1}\}} \mathbf{f}'(x)$$

where $\mathbf{f}'(\mathbf{X}) := \mathbf{eq}(\mathbf{r},\mathbf{X})\mathbf{f}(\mathbf{X},\mathbf{0})\mathbf{f}(\mathbf{X},\mathbf{1})$.

Heart of GKR - representing $\mathbf{g}(\mathbf{r})$ as sum over $\mathbf{f}$

$$\mathbf{g}(\mathbf{r}) = \mathbf{eq}(\mathbf{r}, \mathbf{0})\mathbf{f}(\mathbf{0}, \mathbf{0})\mathbf{f}(\mathbf{0}, \mathbf{1}) + \mathbf{eq}(\mathbf{r}, \mathbf{1})\mathbf{f}(\mathbf{1}, \mathbf{0})\mathbf{f}(\mathbf{1}, \mathbf{1}$$

$$= \sum_{x \in \{\mathbf{0}, \mathbf{1}\}} \mathbf{f}'(x)$$

where $\mathbf{f}'(\mathbf{X}) := \mathbf{eq}(\mathbf{r}, \mathbf{X})\mathbf{f}(\mathbf{X}, \mathbf{0})\mathbf{f}(\mathbf{X}, \mathbf{1})$.

Thus, using SCP can reduce evaluating $\mathbf{g}(\mathbf{r})$ to evaluating $\mathbf{f}'(\mathbf{r_2})$ for a random $\mathbf{r_2} \in \mathbb{F}$.

Evaluating $\mathbf{f'(r_2)}$

$$\mathbf{f'(r_2) = eq(r, r_2) f(r_2, 0) f(r_2, 1)}$$

$\mathcal{V}$ can evaluate $\mathbf{eq(r, r_2)}$ itself.

Since it has $\mathbf{cm(f)}$ it can ask $\mathcal{P}$ for $\mathbf{f(r_2, 0), f(r_2, 1)}$ with proofs of correctness.