Overview of GKR

Ilia Zavidnyi & Svetlana Ivanova

Mentored by *Marshall Ball*

Parameters

Fix circuit $C : \{0,1\}^n \to \{0,1\}$ with size S and depth $d \leq S$.

Let $\mathbb{H}$ be extension field of $\mathbb{GF}[2]$ s.t. :

$$max\{d, \log(S)\} \leq |\mathbb{H}| \leq \mathrm{poly}(d, \log(S)) \quad (1)$$

Let $\mathbb{F}$ be an extension field s.t.:

$$|\mathbb{F}| \leq \mathrm{poly}(|\mathbb{H}|) \quad (2)$$

Parameters

Let m be an integer s.t.:

$$S \leq |\mathbb{H}|^m \leq \text{poly}(S) \quad (3)$$

Let $m' \leq m$ be an integer s.t.:

$$n \leq |\mathbb{H}|^{m'} \leq n \cdot \text{poly}(d, \log(S)) \quad (4)$$

Finally, let $\delta \in \mathbb{N}$ be a degree parameter s.t.:

$$|\mathbb{H}| - 1 \leq \delta \leq |\mathbb{F}| \quad (5)$$

Assumptions and Notations

W.L.O.G. we work with *layered* arithmetic circuit $C : \mathbb{F}^n \to \mathbb{F}$ with *fan-in 2*

Layers are numbered from 0 to d where 0 is the output layer

We assume that all layers have the same size S.

! Any circuit can be made layered with at most quadratic increase in *size*

In particular, we add dummy gates to output layer to get circuit $C' : \mathbb{F}^S \to \mathbb{F}^S$ s.t.:

$$C'(x_1, \ldots, x_S) = (C(x_1, \ldots, x_S), 0, \ldots, 0)$$

Assumptions and Notations

We denote S gates of the i-th layer as $(g_{i,0}, g_{i,1}, \dots, g_{i,S-1})$

Now we define $\mathrm{add}_i, \mathrm{mult}_i : \{0, 1, \dots, S-1\}^3 \to \{0, 1\}$ as follows:

$$\mathrm{add}(i_1, i_2, i_3) = \begin{cases} 1 & \text{if } g_{i-1,j_1} = g_{i,j_2} + g_{i,j_3} \\ 0 & \text{otherwise} \end{cases}$$

$$\mathrm{mult}(i_1, i_2, i_3) = \begin{cases} 1 & \text{if } g_{i-1,j_1} = g_{i,j_2} \cdot g_{i,j_3} \\ 0 & \text{otherwise} \end{cases}$$

Assumptions and Notations

Let $\widetilde{\mathrm{add}_i}, \widetilde{\mathrm{mult}_i} : \mathbb{F}^{3m} \to \mathbb{F}$ be extensions of $\mathrm{add}_i, \mathrm{mult}_i$ with degree $\leq \delta$ in each variable

During run of the protocol prover and verifier have access to the following oracle:

$$\mathcal{F} = \{\widetilde{\mathrm{add}_i}, \widetilde{\mathrm{mult}_i}\}_{\forall i \in [d]}$$

! Note, that $\{\mathrm{add}_i, \mathrm{mult}_i\}_{\forall i \in [d]}$ uniquly defines C unlike $\{\widetilde{\mathrm{add}_i}, \widetilde{\mathrm{mult}_i}\}_{\forall i \in [d]}$

Assumptions and Notations

Now consider vector $v_i = (v_{i,0}, v_{i,1}, \ldots, v_{i,S-1})$ corresponding to values of the gates on layer i, which (with fixed $\alpha : \mathbb{H}^m \to [|\mathbb{H}^m| - 1]$) can be views as:

$$V_i(j) = \begin{cases} v_{i,j} & \text{if } \alpha(j) \leq S - 1 \\ 0 & \text{otherwise} \end{cases}$$

With extension $\widetilde{V}_i : \mathbb{F}^m \to \mathbb{F}$ with degree $\leq |\mathbb{H}| - 1$ and computable in time $\leq |\mathbb{H}^m| \cdot \mathrm{poly}(|\mathbb{H}|, m) = \mathrm{poly(S)}$.

GKR protocol

Prover $\mathcal{P}$ and verifier $\mathcal{V}$ given oracle access to $\mathcal{F}$, verifier sends an input $x \in \{0,1\}^S$

$\mathcal{P}$ wants to prove $C(x) = 0$ or equivalently $\widetilde{V_i}(0,\ldots,0) = 0$

On each iteration $0 \le i \le d$ protocol does the following:

$$\begin{array}{c} \widetilde{V}_{i-1}(z_{i-1)}{=}r_{i-1} \\ \text{reduce}\Big\downarrow \\ \widetilde{V}_i(z_{i)}{=}r_i \end{array}$$

Where the fact that $\widetilde{V}_d(z_d) = r_d$ verifier computes on his own in quasi-linear time

GKR protocol

By definition of LDE, we have that for every $z \in \mathbb{F}^m$:

$$\widetilde{V}_{i-1}(z) = \sum_{p \in \mathbb{H}^m} \tilde{\beta}(z,p) \cdot \widetilde{V}_{i-1}(p)$$

While $\forall p \in \mathbb{H}^m$:

$$V_{i-1}(p) = \sum_{\omega_1,\omega_2 \in \mathbb{H}^m} \widetilde{\mathrm{add}}_i(p,\omega_1,\omega_2) \cdot \left(\widetilde{V}_i(\omega_1) + \widetilde{V}_i(\omega_2)\right) + \widetilde{\mathrm{mult}}_i(p,\omega_1,\omega_2) \cdot \widetilde{V}_i(\omega_1) \cdot \widetilde{V}_i(\omega_2)$$

Now, $\forall z \in \mathbb{F}^m$ let $f_z : \mathbb{F}^{3m} \to \mathbb{F}$ be defined as:

$$f_z(p,\omega_1,\omega_2) = \tilde{\beta}(z,p) \cdot \left(\widetilde{\mathrm{add}}_i(p,\omega_1,\omega_2) \cdot \left(\widetilde{V}_i(\omega_1) + \widetilde{V}_i(\omega_2)\right) + \widetilde{\mathrm{mult}}_i(p,\omega_1,\omega_2) \cdot \widetilde{V}_i(\omega_1) \cdot \widetilde{V}_i(\omega_2)\right)$$

GKR protocol

$$\widetilde{\beta}(z,p) \cdot \left(\widetilde{\mathrm{add}}_i(p,\omega_1,\omega_2) \cdot \left(\widetilde{V}_i(\omega_1) + \widetilde{V}_i(\omega_2)\right) + \widetilde{\mathrm{mult}}_i(p,\omega_1,\omega_2) \cdot \widetilde{V}_i(\omega_1) \cdot \widetilde{V}_i(\omega_2)\right)$$

$deg(\widetilde{\beta}) \leq |\mathbb{H}| - 1$, computation time $\leq$ poly$(|\mathbb{H}|, m)$;
$deg(\widetilde{V}_i) \leq |\mathbb{H}| - 1$, computation time $\leq |\mathbb{H}|^m \cdot$poly$(|\mathbb{H}|, m) =$ poly(S) [3];
$deg(\widetilde{\mathrm{add}}_i, \widetilde{\mathrm{mult}}_i) \leq \delta$ in each its variable [5], computation time $\leq$ poly(δ, m);
Altogether resulting at $deg(f_z) \leq \delta + |\mathbb{H}| - 1 \leq 2\delta$, computation time $\leq$ poly(S).

GKR protocol

Now for every $z \in \mathbb{F}^m$,

$$\widetilde{V}_{i-1}(z) = \sum_{p,\omega_1,\omega_2 \in \mathbb{H}^m} f_{z-1}(p, \omega_1, \omega_2).$$

Thus, proving that $\widetilde{V}_{i-1}(z_{i-1}) = r_{i-1}$ is equivalent to proving that

$$r_{i-1} = \sum_{p,\omega_1,\omega_2 \in \mathbb{H}^m} f_{z-1}(p, \omega_1, \omega_2)$$

This is done by running the interactive sum-check protocol.

GKR protocol

At the end of sum-check verifier needs to compute $f_{z-1}(p, \omega_1, \omega_2)$ which is $\mathrm{poly}(S)$ hard due to $\widetilde{V_i}$.

Since $\mathcal{V}$ can't handle such computations we require $\mathcal{P}$ to compute $\widetilde{V_i}(\omega_1), \widetilde{V_i}(\omega_2)$ for us, which we also need to verify.

This is done via the following interactive process.

GKR protocol

1. Based on the fixed elements $t_1, t_2 \in \mathbb{F}$ known to the $\mathcal{P}$ and $\mathcal{V}$ they construct the linear function $\gamma : \mathbb{F} \to \mathbb{F}^m$, such that for every $i \in \{1, 2\}$:

$$\gamma(t_i) = \omega_i$$

2. The prover $\mathcal{P}$ sends the function $f = \tilde{V}_i \circ \gamma : \mathbb{F} \to \mathbb{F}$ to the verifier $\mathcal{V}$.
3. Upon receiving a function $f : \mathbb{F} \to \mathbb{F}$ from the prover, the verifier $\mathcal{V}$ checks that f is a polynomial of degree at most $m \cdot (|\mathbb{H}| - 1)$, and that $f(t_i) = v_i$ for $i \in \{1, 2\}$. If these tests pass, then $\mathcal{V}$ chooses a random element $t \in \mathbb{F}$ and sends it to $\mathcal{P}$.
4. The prover and verifier continue to Phase $i + 1$ with $z_i = \gamma(t)$ and $r_i = f(t)$.

GKR protocol. Final verification

After the d'th phase, the verifier $\mathcal{V}$ needs to verify on his own that $\tilde{V}_d(z_d) = r_d$. This amounts to computing a single point in the low-degree extension of the input x (with respect to $\mathbb{F}, \mathbb{H}, m'$). The verifier runs this computation on its own.