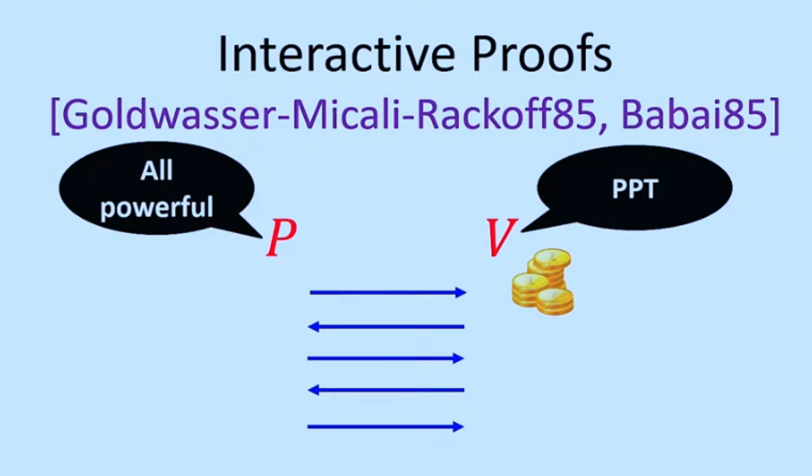
Delegating Computation: Interactive Proofs for Muggles

Ziyang Jin[1]

[1]Theory Group
Department of Computer Science
University of Toronto



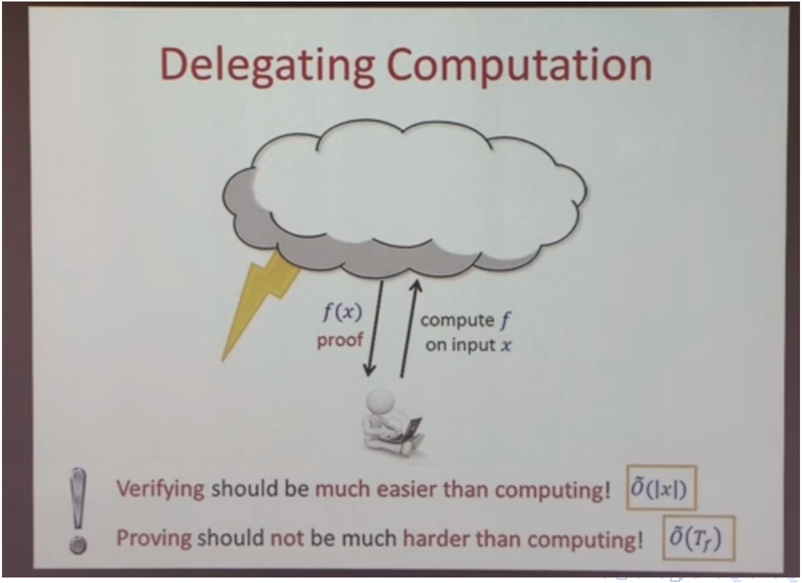
(*Images in the slides are stolen from Kalai's talks on YouTube)

The problem

We have seen interactive protocols where the prover has unbounded computational resources, and the verifier runs in probabilistic polynomial time.

Today we will see interactive proofs for muggles: The prover runs in probabilistic polynomial time (in other words, a "muggle"). The verifier runs in nearly-linear time (e.g. $O(n \log^k n)$).

Why is it useful?

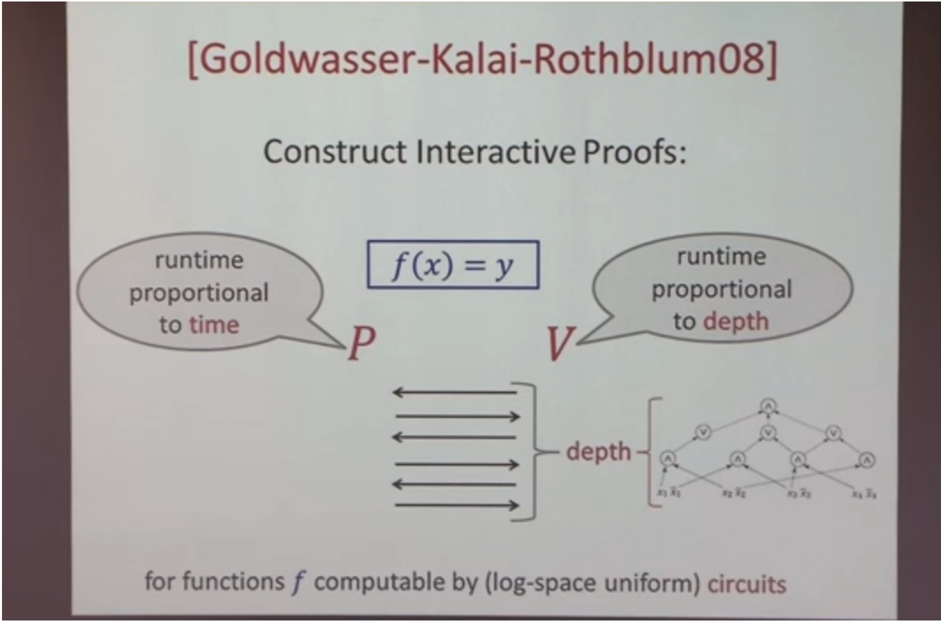


Main theorem

Theorem (Goldwasser-Kalai-Rothblum '08)

Let L be a language that can be computed by a family of $O(\log(S))$-space uniform boolean circuits of size S and depth d. L has an interactive proof where:

1. *The prover runs in time $\text{poly}(S)$. The verifier runs in time $n \cdot \text{poly}(d, \log S)$ and space $O(\log(S))$.*
2. *The protocol has perfect completeness and soundness $1/100$.*
3. *The protocol is public-coin, with communication complexity $d \cdot \text{polylog}(S)$.*


[Goldwasser-Kalai-Rothblum08]
Construct Interactive Proofs:
$f(x) = y$
runtime proportional to time
runtime proportional to depth
P
V
depth
for functions f computable by (log-space uniform) circuits

Circuit

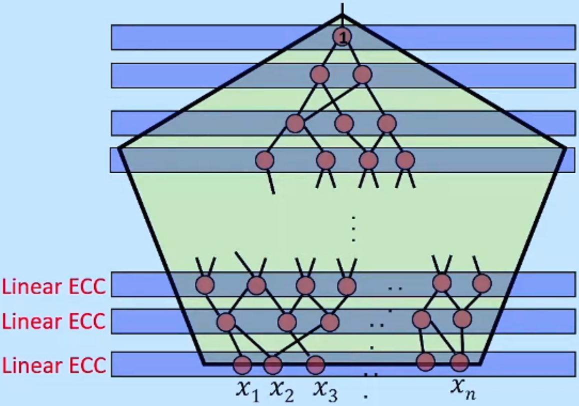
Circuit C:

- NAND gates only
- fan-in 2
- layered structure

Input $x \in \{0,1\}^n$
Circuit size is $S = \text{poly}(n)$.
Circuit depth is $d = \text{polylog}(n)$ typically.

UNIVERSITY OF TORONTO

[GKR08] Blueprint



Naive approach

[board work]

The GKR solution

[board work]

Padding each layer

You can think of the circuit evaluation as a table of width S and depth $d+1$.

$$
\begin{array}{rcccccccc}
\text{layer } 0: & 1 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\
\text{layer } 1: & 1 & 1 & 0 & 0 & 0 & 0 & \dots & 0 \\
\text{layer } 2: & 1 & 0 & 1 & 1 & 0 & 0 & \dots & 0 \\
\vdots & \vdots & & & & & & & \\
\text{layer } i: & 0 & 0 & 1 & 0 & 0 & 0 & \dots & 0 \\
\text{layer } i+1: & 1 & 1 & 0 & 1 & 0 & 0 & \dots & 0 \\
\vdots & \vdots & & & & & & & \\
\text{layer } d: & 0 & 1 & 1 & 1 & 1 & 1 & \dots & 1
\end{array}
$$

Every row i can be described by a function $\alpha_i : [S] \to \{0,1\}$.

Low Degree Extension 1

We have

$$\alpha_i(g) : [S] \to \{0,1\}$$

where $g \in [S]$ is the name of a gate.

Now we one-to-one map each element $g \in [S]$ to an element $z \in \mathbb{H}^m$.
$|\mathbb{H}|^m = S$. $\mathbb{H}$ is an extension field of $\mathbb{GF}[2]$.

$$\alpha'_i(z) : \mathbb{H}^m \to \{0,1\}$$

where $z \in \mathbb{H}^m$ is the name of a gate.

Low Degree Extension 2

We have

$$\alpha_i'(z) : \mathbb{H}^m \to \{0, 1\}$$

where $z \in \mathbb{H}^m$ is the name of a gate.

This is the low-degree extension:

$$\tilde{\alpha}_i(z) : \mathbb{F}^m \to \mathbb{F}$$

where $\mathbb{F}$ is an extension field of $\mathbb{H}$, so $\mathbb{H} \subseteq \mathbb{F}$. $\tilde{\alpha}_i(z)$ is a low-degree polynomial, and $\tilde{\alpha}_i(z)$ agrees with $\alpha_i'(z)$ on $\mathbb{H}^m$.

Standard Low Degree Extension

You have a binary array (or truth table, or a layer of gate outputs) $\vec{w} \in \{0,1\}^S$.

$$\vec{w} \quad = \quad 1 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad \dots \quad 0$$

Suppose $\vec{w} = w_1, w_2, ..., w_S$. Equivalently, $\vec{w} = w_1, w_2, ..., w_{|\mathbb{H}|^m}$.

The low-degree extension $\tilde{\alpha}(z) : \mathbb{F}^m \to \mathbb{F}$ is

$$\tilde{\alpha}(z) = \sum_{t \in \mathbb{H}^m} \tilde{\beta}(z,t) \cdot w_t$$

where $\tilde{\beta}(z,t) = 1$ if $z = t$;
and $\tilde{\beta}(z,t) = 0$ if $z \neq t$ and $z \in \mathbb{H}^m$;
and $\tilde{\beta}(z,t)$ can "go crazy" when $z \in \mathbb{F}^m \setminus \mathbb{H}^m$.

Important Claim

Claim

When $\tilde{\alpha}(z)$ has individual degree at most $|\mathbb{H}| - 1$, thus total degree $m \cdot (|\mathbb{H}| - 1)$, the low-degree extension is unique.

!!!

We will use a different low-degree extension for gate output layers.

!!!

We will only apply the standard low-degree extension to the input x.

Circuit structure function

Define the circuit structure function $\phi_i(z, w_1, w_2) : \mathbb{H}^{3m} \to \{0, 1\}$

$$\phi_i(z, w_1, w_2) = \begin{cases} 1, & z \text{ is the parent gate of } w_1, w_2 \\ 0, & \text{otherwise.} \end{cases}$$

Low-degree extend it, we get $\tilde{\phi}_i(z, w_1, w_2) : \mathbb{F}^{3m} \to \mathbb{F}$.

$$\tilde{\phi}_i(z, w_1, w_2) = \begin{cases} 1, & z \text{ is the parent gate of } w_1, w_2 \\ 0, & \text{not parent-children but } z, w_1, w_2 \in \mathbb{H}^m, \\ \text{“go crazy”}, & \text{otherwise.} \end{cases}$$

The GKR construction of $\tilde{\phi}_i$ has individual degree δ slightly bigger than $|\mathbb{H}| - 1$.

The GKR low-degree extension

Define $\tilde{\alpha}_{i-1} : \mathbb{F}^m \to \mathbb{F}$

$$\tilde{\alpha}_{i-1}(z_{i-1}) = \sum_{w_1, w_2 \in \mathbb{H}^m} \tilde{\phi}_i(z_{i-1}, w_1, w_2) \cdot \widetilde{\mathrm{NAND}}(\tilde{\alpha}_i(w_1), \tilde{\alpha}_i(w_2)).$$

- i means ith layer
- $z_{i-1} \in \mathbb{F}^m$ is a *virtual gate*
- $\tilde{\phi}_i$ is the low-degree extension of the circuit structure
- $\widetilde{\mathrm{NAND}}$ is the arithmetization of NAND gate

Oracle

Define

$$\mathcal{F} = \{\tilde{\phi}_i : 1 \leq i \leq d\}$$

and $\mathcal{F}$ is given to both the prover and the verifier as an oracle.

The Bare-Bones Protocol

Bare-Bones Protocol $(\mathcal{P}^{\mathcal{F}}(x), \mathcal{V}^{\mathcal{F}}(x))$:

- Input $x \in \{0,1\}^n$.
- Circuit $C : \{0,1\}^n \to \{0,1\}$ with fan-in 2 of size S and depth d.
- An oracle $\mathcal{F}$ computing (an extension of) the function specifying C.
- Prover $\mathcal{P}$ wants to convince the verifier $\mathcal{V}$ that $C(x) = 0$.
- Both prover and verifier have access to oracle $\mathcal{F}$.

The Parameters

- x: the input, and $|x| = n$
- S: circuit size, $S = poly(n)$
- d: circuit depth, typically, we take $d = \text{polylog}(S) = \text{polylog}(n)$
- $\mathbb{H}$: extension field of $\mathbb{GF}[2]$, $|\mathbb{H}| = n^{0.01}$, and $|\mathbb{H}|^m = S$
- m : since $|\mathbb{H}|^m = S$, m is a big constant
- $\mathbb{F}$: extension field of $\mathbb{H}$, $|\mathbb{F}| = \text{poly}(|\mathbb{H}|)$, but we cannot let $|\mathbb{F}|$ to be bigger than n, so for example, $|\mathbb{F}| = n^{0.3}$

In Phase i

z_0 is the output gate; set $r_0 = 0$ since we assume $C(x) = 0$.

Goal

Reduce proving $\tilde{\alpha}_{i-1}(z_{i-1}) = r_{i-1}$ to proving that $\tilde{\alpha}_i(z_i) = r_i$.

Within each phase, we do:

1. sum-check protocol
2. 2-to-1 trick

Before Sum-check

Goal 1

Reduce from proving $\tilde{\alpha}_{i-1}(z_{i-1}) = r_{i-1}$ to proving two points $\tilde{\alpha}_i(w_1) = v_1$ and $\tilde{\alpha}_i(w_2) = v_2$.

Goal 2

Make sure the verifier has a good runtime, i.e. $\mathrm{poly}(|\mathbb{H}|)$.

Sum-check

Recall

$$\tilde{\alpha}_{i-1}(z_{i-1}) = \sum_{w_1, w_2 \in \mathbb{H}^m} \tilde{\phi}_i(z_{i-1}, w_1, w_2) \cdot \widetilde{\mathrm{NAND}}(\tilde{\alpha}_i(w_1), \tilde{\alpha}_i(w_2))$$

Define $f_z : \mathbb{F}^{2m} \to \mathbb{F}$

$$f_z(w_1, w_2) := \tilde{\phi}_i(z_{i-1}, w_1, w_2) \cdot \widetilde{\mathrm{NAND}}(\tilde{\alpha}_i(w_1), \tilde{\alpha}_i(w_2)).$$

Thus, we have

$$\tilde{\alpha}_{i-1}(z_{i-1}) = \sum_{w_1, w_2 \in \mathbb{H}^m} f_z(w_1, w_2).$$

So we want to check

$$r_{i-1} = \sum_{w_1, w_2 \in \mathbb{H}^m} f_z(w_1, w_2)$$

and define $r_{i-1,0} := r_{i-1}$.

Sum-check continued

$$\tilde{\alpha}_{i-1,0} = \sum_{\vec{w}_1, \vec{w}_2 \in \mathbb{H}^m} f_z(\vec{w}_1, \vec{w}_2)$$

$$\tilde{\alpha}_{i-1,1}(x) = \sum_{w_{1,2},..,w_{1,m} \in \mathbb{H}, \vec{w}_2 \in \mathbb{H}^m} f_z(x, w_{1,2}, \ldots, w_{1,m}, \vec{w}_2)$$

$$\tilde{\alpha}_{i-1,2}(x) = \sum_{w_{1,3},..,w_{1,m} \in \mathbb{H}, \vec{w}_2 \in \mathbb{H}^m} f_z(w_{1,1}, x, w_{1,3}, \ldots, w_{1,m}, \vec{w}_2)$$

$$\tilde{\alpha}_{i-1,3}(x) = \sum_{w_{1,4},..,w_{1,m} \in \mathbb{H}, \vec{w}_2 \in \mathbb{H}^m} f_z(w_{1,1}, w_{1,2}, x, w_{1,4}, \ldots, w_{1,m}, \vec{w}_2)$$

$$\cdots$$

$$\tilde{\alpha}_{i-1,2m}(x) = f_z(w_{1,1}, \ldots, w_{1,m}, w_{2,1}, \ldots, w_{2,m-1}, x)$$

Sum-check final

Finally, the verifier wants to check

$$f_z(\vec{w}_1, \vec{w}_2) = r_{i-1,2m}$$

Replace f_z with its definition:

$$\tilde{\phi}_i(\vec{z}_{i-1}, \vec{w}_1, \vec{w}_2) \cdot \widetilde{\mathrm{NAND}}(\tilde{\alpha}_i(\vec{w}_1), \tilde{\alpha}_i(\vec{w}_2)) = r_{i-1,2m}$$

Prover sends $v_1 = \tilde{\alpha}_i(\vec{w}_1)$ and $v_2 = \tilde{\alpha}_i(\vec{w}_2)$.

2-to-1 Trick

Goal

Reduce from proving $\tilde{\alpha}_i(w_1) = v_1$ and $\tilde{\alpha}_i(w_2) = v_2$ to proving a single point $\tilde{\alpha}_i(z_i) = r_i$.

1. Fix $t_1, t_2 \in \mathbb{F}$. Think $t_1 = 0, t_2 = 1$.
2. Interpolate line $\gamma : \mathbb{F} \to \mathbb{F}^m$ s.t. $\gamma(t_1) = w_1, \gamma(t_2) = w_2$.
3. Prover sends $f := \tilde{\alpha}_i \circ \gamma : \mathbb{F} \to \mathbb{F}$. (or fake $\tilde{g}_i \circ \gamma$).
4. Verifier test $f(t_1) = v_1, f(t_2) = v_2$.
5. Verifier pick a random $t \in \mathbb{F}$, thus $z_i = \gamma(t)$ and $r_i = f(t)$.

Properties

- **Completeness:** If $C(x) = 0$, then
$$\Pr[(\mathcal{P}^{\mathcal{F}}(x), \mathcal{V}^{\mathcal{F}}(x)) = 1] = 1.$$
- **Soundness:** If $C(x) \neq 0$, then for every (unbounded) prover $\mathcal{P}^*$,
$$\Pr[(\mathcal{P}^{*\mathcal{F}}(x), \mathcal{V}^{\mathcal{F}}(x)) = 1] \leq \frac{1}{100}.$$

Proof of Soundness

Suppose that $C(x) = 1$ and there exists a cheating prover $\mathcal{P}^*$ such that

$$\Pr[(\mathcal{P}^{*\mathcal{F}}, \mathcal{V}^{\mathcal{F}}) = 1] = s$$

for some $0 \leq s \leq 1$. We would like to show $s \leq \frac{1}{100}$ as claimed in the main theorem.

- Let A denote the event $(\mathcal{P}^{*\mathcal{F}}, \mathcal{V}^{\mathcal{F}}) = 1$, i.e., the verifier eventually accepts.
- Let T_i denote the event that indeed $\tilde{\alpha}_i(z_i) = r_i$, where $0 \leq i \leq d$. Thus, $C(x) \neq 0$ means $\neg T_0$. Note that $\tilde{\alpha}_i(z_i)$ means the true polynomial for layer i computed by an honest prover. The cheating prover will give the verifier a fake polynomial $\tilde{g}_i$ (actually $\tilde{g}_{i,0}, \dots, \tilde{g}_{i,2m}$ in the sum-check, and $\tilde{g}_i \circ \gamma$ in the 2-to-1 trick).
- Let E_i denote the event that indeed $\tilde{\alpha}_i(w_1) = v_1$ and $\tilde{\alpha}_i(w_2) = v_2$, for $i \in [d]$. A cheating prover can send v_1, v_2 such that it matches $\tilde{g}_i(w_1) = v_1$ and $\tilde{g}_i(w_2) = v_2$.

$$s = \Pr[A \wedge \neg T_0 \wedge T_d] \leq \Pr[\exists i \in [d], A \wedge \neg T_{i-1} \wedge T_i] \leq \sum_{i=1}^{d} \Pr[A \wedge \neg T_{i-1} \wedge T_i]$$

$$\Pr[A \wedge \neg T_{i-1} \wedge T_i] = {\color{red}\Pr[A \wedge \neg T_{i-1} \wedge T_i \wedge E_i]} + {\color{blue}\Pr[A \wedge \neg T_{i-1} \wedge T_i \wedge \neg E_i]}$$

$$\Pr[A \wedge \neg T_{i-1} \wedge T_i \wedge E_i] \le \Pr[A \wedge \neg T_{i-1} \wedge E_i] \le \frac{4m\delta}{|\mathbb{F}|}$$

Note that $A \wedge \neg T_{i-1} \wedge E_i$ is the event that the cheating prover successfully survive the sum-check protocol.

$$\Pr[A \wedge \neg T_{i-1} \wedge T_i \wedge \neg E_i] \le \Pr[A \wedge T_i \wedge \neg E_i] \le \frac{m\delta}{|\mathbb{F}|}$$

Note that $A \wedge T_i \wedge \neg E_i$ is the event that fake polynomial $\tilde{g}_i$ agrees with the true $\tilde{\alpha}_i$ on input z_i.

Therefore,

$$\Pr[A \wedge \neg T_{i-1} \wedge T_i] \leq \frac{4m\delta}{|\mathbb{F}|} + \frac{m\delta}{|\mathbb{F}|} \leq \frac{5m\delta}{|\mathbb{F}|}$$

By union bound on d phases,

$$s = \Pr[A \wedge \neg T_0 \wedge T_d] \leq \frac{5md\delta}{|\mathbb{F}|}$$

Taking $\mathbb{F}$ such that $|\mathbb{F}| \geq 500md\delta = \mathrm{poly}(|\mathbb{H}|)$, we get $s \leq \frac{1}{100}$ as desired.

Q & A

Questions?

References

[1] Yael Kalai. Talk on GKR based Zero-Knowledge Proofs workshop. https://www.youtube.com/watch?v=x8pUxFptfb0
[2] Yael Kalai. Delegating Computation I. Talk on Cryptography Boot Camp. https://www.youtube.com/watch?v=ExuEEZOBjL8
[3] Shafi Goldwasser; Yael Tauman Kalai; Guy N. Rothblum. Delegating computation: interactive proofs for muggles. https://eccc.weizmann.ac.il/report/2017/108/
[4] Roei Tell. Mutilinear and Low-Degree Extensions. Unpublished manuscript: https://sites.google.com/site/roeitell/Expositions