CS355 Lecture #12:

SNARGs from polynomial commitments & Interactive Oracle Proofs (IOPs)

Last time

- SNARGs from PCPs
- polynomial commitments:

  Protocol between $\mathcal{P}$ & $\mathcal{V}$:

  $\mathrm{Setup}(d) \to pp$ public parameters

  degree bound

  $\mathrm{Commit}(pp, f) \to c$ comm. to $f(\cdot)$

  polynomial, degree $< d$

  $\mathrm{Open}(pp, f, x) \to \pi$

  $\mathrm{Check}(pp, c, x, y, \pi) \to \{\mathrm{True}, \mathrm{False}\}$

For today: $|\pi| \in O(\log d)$

$|c| \in O(1)$

Today:

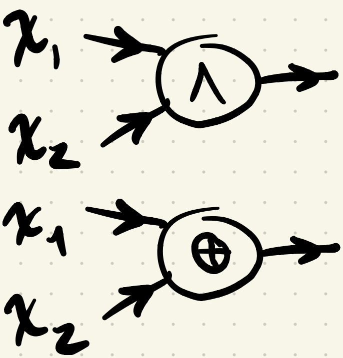
1. Arithmetic circuits & constraints
2. From polynomial commitments to SNARGs via Interactive Oracle Proofs
3. Fun with polynomials:
   - polynomial equality testing
   - proving a polynomial vanishes on a subgroup of $\mathbb{F}$
   - "univariate sum-check" [BCRSVW'19]
4. "Marlin-Lite" IOP [CHMMVW'20]
5. Putting it all together.

$\boxed{1}$ Arithmetic circuits & constraints

A boolean circuit is a DAG where:

- nodes are "gates" — $\wedge$ or $\oplus$
- nodes have in-degree 2
- edges are "wires" labeled 0 or 1.

• We say circuit C is <u>satisfied</u> if all wires are labeled such that:



$$y = x_1 \wedge x_2$$

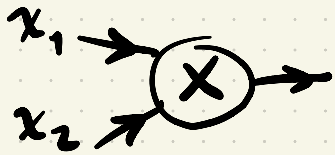
$$y = x_1 \oplus x_2$$

How can we generalize this?

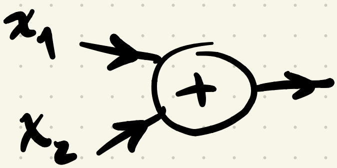
Arithmetic circuit:

- gates are $\times$ or $+$ over $\mathbb{F}$
- wires take values from $\mathbb{F}$

Arithmetic circuit $\mathcal{C}$ is satisfied if all wires are labeled such that:



$y = x_1 \times x_2 \in \mathbb{F}$



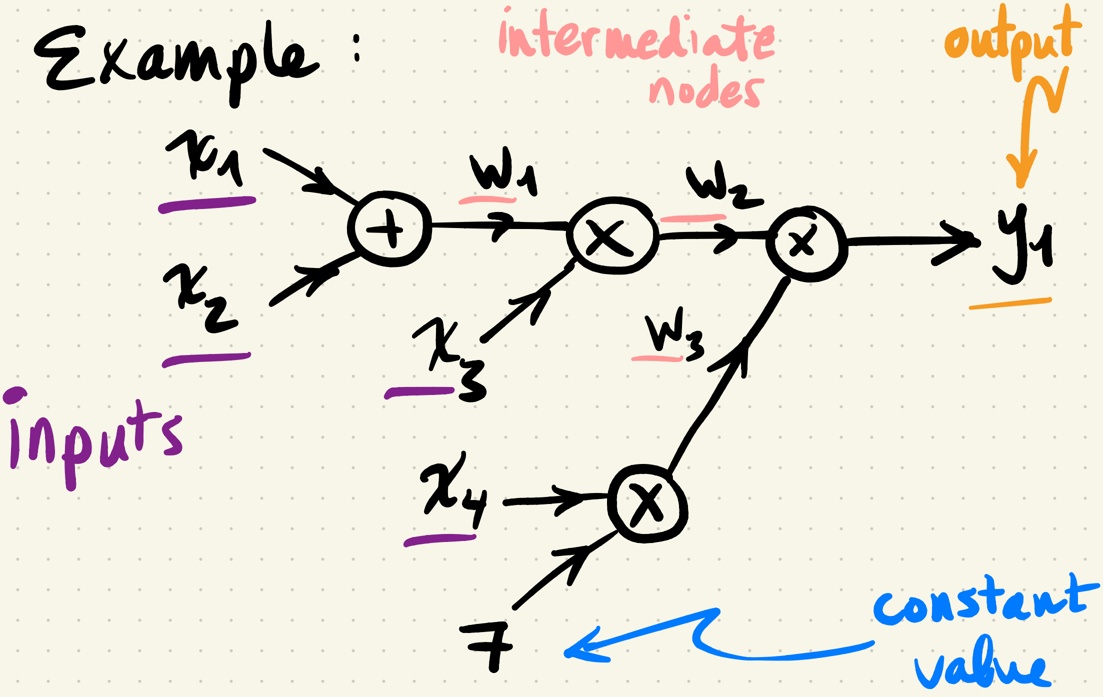
$y = x_1 + x_2 \in \mathbb{F}$

$\Rightarrow$ A Boolean circuit is an A.C. where $\mathbb{F} = \mathbb{F}_2$.

$\Longrightarrow$ $\times$ in $\mathbb{F}_2$ is $\wedge$

$+$ in $\mathbb{F}_2$ is $\oplus$

Example :



Equivalently, as constraints:

$$w_1 = x_1 + x_2$$
$$w_2 = w_1 \times x_3$$
$$w_3 = 7 \times x_4$$
$$y_1 = w_2 \times w_3$$

But: these can be $+$ or $\times$.

Can we rewrite using one kind of constraint?

Rank-1 constraints:

Define $z \triangleq (\vec{x}, \vec{y}, \vec{w}, 1) \in \mathbb{F}^n$

For $a, b, c \in \mathbb{F}^n$

vectors of length n

a rank-1 constraint is:

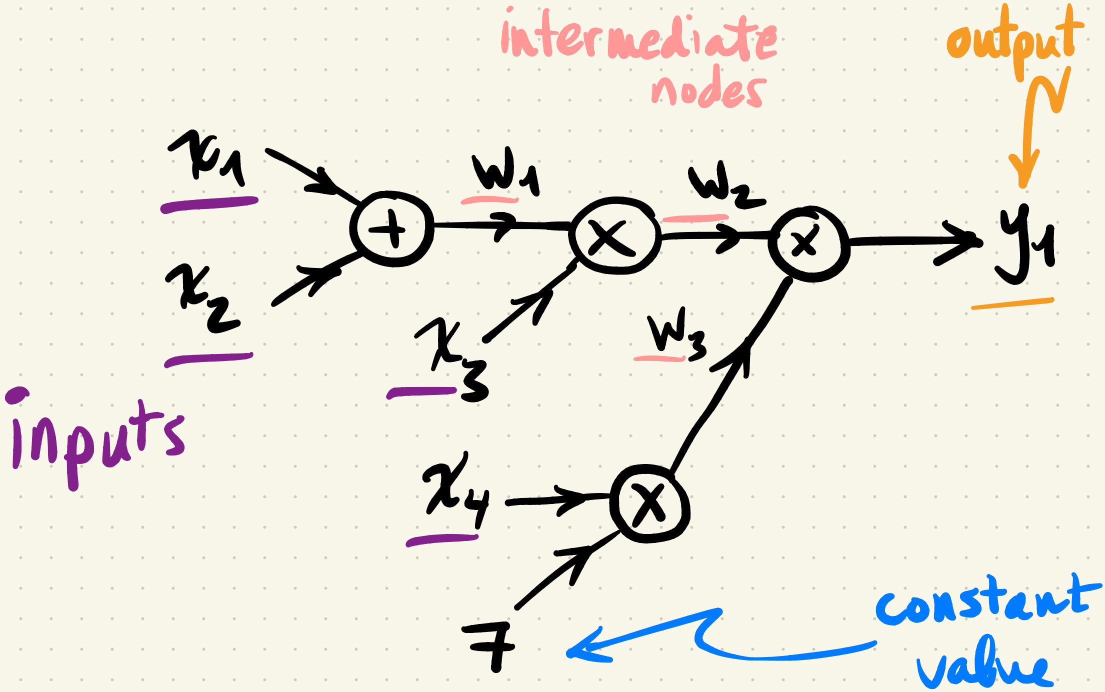
$$\langle a, z \rangle \times \langle b, z \rangle = \langle c, z \rangle$$

$\langle \cdot, \cdot \rangle$ is dot (inner product)

$$\Rightarrow \langle a, z \rangle = \sum_j a[j] \times z[j]$$

$a[j]$ is j^{th} entry of a
$z[j]$ is j^{th} entry of z.

$\Rightarrow$ constraints on linear combinations of wire values.



As Rank-1 constraints:

① write down outputs of × gates.

① $w_2 = (x_1 + x_2) \times x_3$

② $w_3 = 7 \times x_4$

③ $y_1 = w_2 \times w_3$

② define z:

$$z \triangleq (x_1, x_2, x_3, x_4, y_1, w_2, w_3, 1)$$

③ for each constraint, read out a, b, & c vectors

$$c_1 = (0,0,0,0,0,1,0,0) \qquad w_2$$
$$a_1 = (1,1,0,0,0,0,0,0) \qquad x_1 + x_2$$
$$b_1 = (0,0,1,0,0,0,0,0) \qquad x_3$$

$$c_2 = (0,0,0,0,0,0,1,0) \qquad w_3$$
$$a_2 = (0,0,0,0,0,0,0,7) \qquad 7$$
$$b_2 = (0,0,0,1,0,0,0,0) \qquad x_4$$

$$c_3 = (0,0,0,0,1,0,0,0) \qquad y_1$$
$$a_3 = (0,0,0,0,0,1,0,0) \qquad w_2$$
$$b_3 = (0,0,0,0,0,0,1,0) \qquad w_3$$

A Rank-1 constraint system (R1CS) is given by three $m \times n$ matrices $A, B, C \in \mathbb{F}^{m \times n}$

$$A \triangleq \begin{bmatrix} a_1 \in \mathbb{F}^n \\ a_2 \\ \vdots \\ a_m \end{bmatrix} \qquad B \triangleq \begin{bmatrix} b_1 \in \mathbb{F}^n \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

$$C \triangleq \begin{bmatrix} c_1 \in \mathbb{F}^n \\ c_2 \\ \vdots \\ c_m \end{bmatrix}$$

In other words, each row is a rank-1 constraint

For $z \in \mathbb{F}^n$, R1CS is satisfied if

$$Az \circ Bz = Cz$$

element-wise product.

For R1CS $A, B, C \in \mathbb{F}^{m \times n}$ and input/output vectors x, y, we say that an instance is <u>satisfiable</u> if

$$\exists w: \quad z \triangleq (x, y, w, 1) \in \mathbb{F}^n$$

$$Az \circ Bz = Cz$$

<u>Theorem</u>: R1CS-SAT is NP complete.

⇒ We saw AC-SAT-to-R1CS-SAT transformation above.

(R1CS-SAT-to-AC-SAT is likewise mechanical)

AC-SAT is NP-complete ⇒ R1CS-SAT is, too.

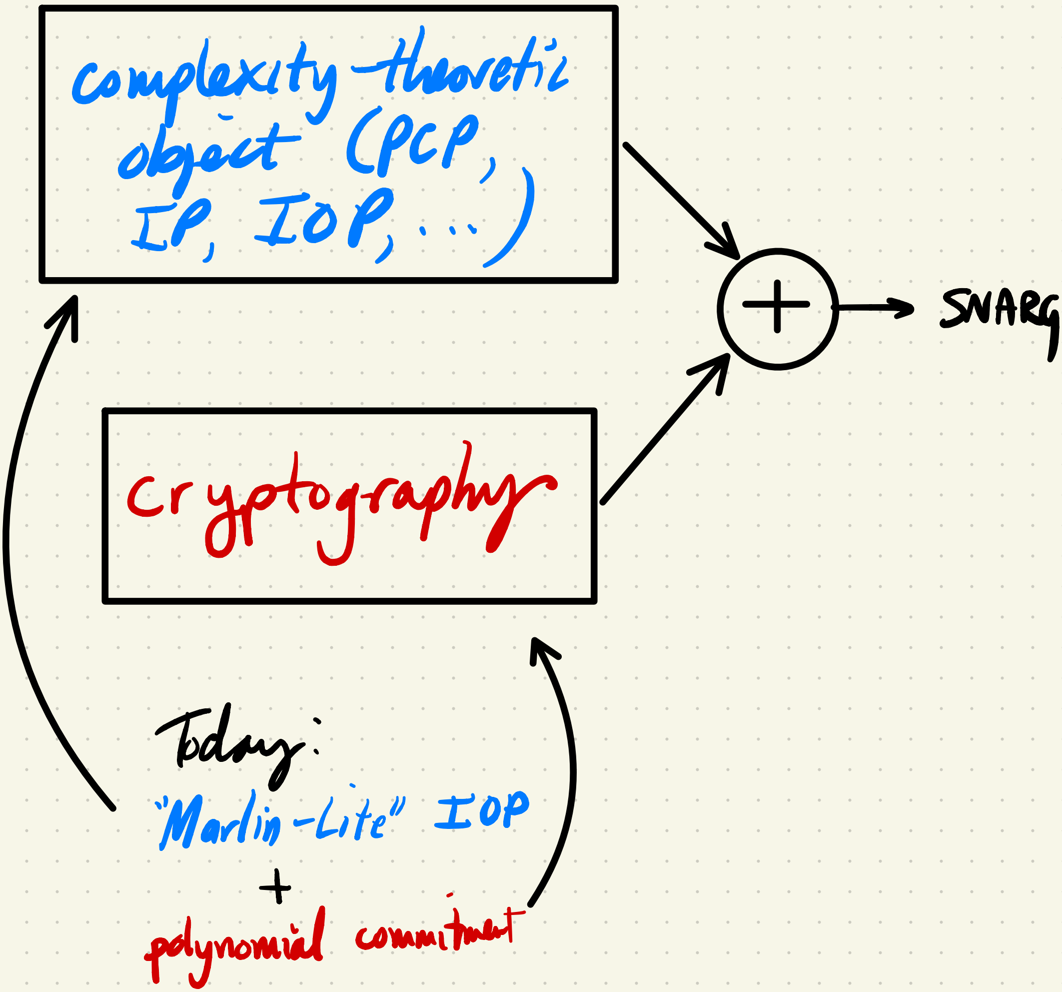
So: why R1CS-SAT?

⇒ Good structure for polynomial IOPs.

⇒ good compilation target (maybe...)

2 From polynomial commitments to SNARGs via Interactive Oracle Proofs.

High-level idea:



Interactive Oracle Proofs (IOPs)

IOPs [RRR'16, BCS'16] generalize IPs & PCPs.

Informally: in each round, $\mathcal{P}$ gives $\mathcal{V}$ **oracle access** to its message

$\Rightarrow$ $\mathcal{V}$ queries $\mathcal{P}$'s messages

Today: our focus is on public coin IOPs where $\mathcal{P}^*$ is limited to sending polynomials of bounded degree, which $\mathcal{V}$ queries.

cheating prover

$\Rightarrow$ Spoiler: implemented w/ polynomial commitments.

$\underline{\mathcal{P}}$ $\qquad\qquad\qquad\qquad$ $\underline{\mathcal{V}}$

$\xrightarrow{\text{polynomial}}$

$\xleftarrow[\text{random challenge}]{}$

repeat

Then: make evaluation queries to polynomials, accept/reject.

To "compile" this IOP to a SNARG:

- $\mathcal{P}$ commits to polynomials, opens at $\mathcal{V}$'s query points
- $\mathcal{V}$ checks $\mathcal{P}$'s opening proofs
- non-interactivity via Fiat-Shamir

$\Rightarrow$ First, some polynomial sub-protocols.

$\boxed{3}$ Fun with polynomials

⇒ sub-protocols we need for Marlin-Lite.

Polynomial protocol #1 $\boxed{\text{PP\#1}}$: polynomial equality IOP.

Fact #1: two distinct polynomials of degree $\leq d$ agree on $\leq d$ points.

⇒ Univariate case: for Q, R each of degree at most d, $S = Q - R$ is a polynomial of degree $\leq d$. By fundamental theorem of algebra, S has $\leq d$ roots ⇒ Q & R agree on at most d points.

⇒ multivariate: Schwarz-Zippel lemma.

PROTOCOL:

(1) $\mathcal{P}$ sends Q, R oracles to $\mathcal{V}$

(2) $\mathcal{V}$ selects $\gamma \xleftarrow{\$} \mathbb{F}$ and queries $Q(\gamma)$, $R(\gamma)$

(3) $Q(\gamma) = R(\gamma) \Longleftrightarrow$ ACCEPT

Completeness: immediate

Soundness: error $\leq \dfrac{d}{|\mathbb{F}|}$ by Fact#1

$\boxed{\text{PP\#2}}$: polynomial vanishing on a multiplicative subgroup $H \subseteq \mathbb{F}$

Let $H \subseteq \mathbb{F}$, $|H| = n$, h a generator.

Define

$$Z_H(X) = \prod_{i=1}^{n} (X - h^i)$$

"vanishing polynomial on H"

Fact #2 A degree $\leq d$ polynomial $g(X)$ vanishes on a set H iff there exists a polynomial R of degree $\leq d - |H|$ s.t.

$$g(X) = Z_H(X) R(X)$$

PROTOCOL:

(1) $\mathcal{P}$ sends g, R oracles to $\mathcal{V}$

(2) $\mathcal{V}$ selects $\gamma \xleftarrow{\$} \mathbb{F}$, queries $g(\gamma)$ & $R(\gamma)$, evaluates $Z_H(\gamma)$

(3) $g(\gamma) = Z_H(\gamma) R(\gamma) \Leftrightarrow$ ACCEPT

(easy to evaluate)

Completeness: by Fact #2

Soundness: as $\boxed{PP\#1}$

PP#3 : "univariate sum-check"

[BCRSVW'19]

Goal : for $H \subseteq \mathbb{F}$ a multiplicative subgroup and polynomial Q of degree d, P convinces V that

(Eqn 1) $$\sum_{\alpha \in H} Q(\alpha) = 0$$

[Byott & Chapman 1999]

Fact #3 (Eqn 1) holds iff there exist polynomials $\begin{cases} R \text{ of degree } \leq d-n \\ S \text{ of degree } < n-1 \end{cases}$

such that

$$Q(X) = Z_H(X)R(X) + X S(X)$$

PROTOCOL:

(1) $\mathcal{P}$ sends α, R, S oracles to $\mathcal{V}$

(2) $\mathcal{V}$ selects $\gamma \xleftarrow{\$} \mathbb{F}$, queries $\alpha(\gamma)$, $R(\gamma)$, & $S(\gamma)$, and evaluates $Z_H(\gamma)$ (easy)

(3) $\alpha(\gamma) = Z_H(\gamma) R(\gamma) + \gamma S(\gamma)$

$\Leftrightarrow$ ACCEPT

Completeness: by Fact #3

Soundness: as $\boxed{\text{PP\#1}}$

$\Rightarrow$ now we're ready to build an IOP for R1CS!

$\boxed{4}$ "Marlin-Lite": an IOP for R1CS-SAT. [CHMMVW'20]

Fix $A, B, C \in \mathbb{F}^{n \times n}$ simplification

Goal: prove $\exists z \in \mathbb{F}^n : Az \circ Bz = Cz$

Idea #1: encode $z, Az, Bz, Cz \in \mathbb{F}^n$ as polynomials of degree $n-1$.

As before, $H \subseteq \mathbb{F}$ is a multiplicative subgroup of $\mathbb{F}$ with $|H| = n$.

Fix generator $h \Rightarrow H = \{h, h^2, h^3, \dots\}$

Define $\hat{z}(X)$ to be the (unique) polynomial w/degree $<n$ such that

$$\hat{z}(h^i) \triangleq z[i], \quad i \in \{1, \dots, n\}$$

Likewise,

$$\hat{z}_A(h^i) \triangleq (Az)[i]$$

$$\hat{z}_B(h^i) \triangleq (Bz)[i]$$

$$\hat{z}_C(h^i) \triangleq \underbrace{(Cz)}_{\text{length-}n\text{ vector}}[i]$$

$\Rightarrow$ $\mathcal{P}$ sends $\mathcal{V}$ $\hat{z}, \hat{z}_A, \hat{z}_B, \hat{z}_C$ oracles

Then

$$Az \circ Bz = Cz \iff$$

$$\hat{z}_A(h^i) \cdot \hat{z}_B(h^i) = \hat{z}_C(h^i) \quad {}_{i \in \{1, \dots, n\}}$$

$Az \circ Bz = Cz$ iff

$\hat{z}_A(X)\,\hat{z}_B(X) - \hat{z}_C(X)$ is a polynomial that vanishes on H

$\Rightarrow$ use $\boxed{\text{PP\#2}}$!

(1) $\mathcal{P}$ sends $\mathcal{V}$ R oracle s.t.

$$\hat{z}_A(X)\,\hat{z}_B(X) - \hat{z}_C(X) = R(X)\,z_H(X)$$

(2) $\mathcal{V}$ picks $\gamma \xleftarrow{\$} \mathbb{F}$, checks that

$$\hat{z}_A(\gamma)\,\hat{z}_B(\gamma) - \hat{z}_C(\gamma) = z_H(\gamma)\,R(\gamma)$$

$\Rightarrow$ Soundness error $\leq \dfrac{2n}{|\mathbb{F}|}$

But: how do we know that $\hat{z}_A$ is consistent w/ the vector Az?

Idea #2 Encode $A \in \mathbb{F}^{n \times n}$ as a bi-variate polynomial as before: unique $\hat{A}$ s.t.

$$\hat{A}(h^i, h^j) = A_{i,j}$$

By def'n of matrix-vector product.

Then

(Eqn 2) $$\hat{z}_A(X) = \sum_{i=1}^{n} \hat{A}(X, h^i)\,\hat{z}(h^i)$$

and likewise for

$\hat{z}_B, \hat{B}$ and $\hat{z}_C, \hat{C}$

How do we check this?

• To start, apply $\boxed{\text{PP\#1}}$:

$\Rightarrow V$ picks $\beta_A \xleftarrow{\$} \mathbb{F}$, sends to P

Now, if

(Eqn 3) $$\hat{z}_A(\beta_A) = \sum_{i=1}^{n} \hat{A}(\beta_A, h^i)\,\hat{z}(h^i)$$

then (Eqn 2) holds except w/probability $\frac{n}{|\mathbb{F}|}$.

• To check (Eqn 3), first define

$$Q_A(X) \triangleq \hat{A}(\beta_A, X)\,\hat{z}(X) - \frac{\hat{z}_A(\beta_A)}{\underbrace{|H|}_{=n}}$$

Now, (Eqn 3) holds iff

$$\sum_{i=1}^{n} Q_A(h^i) = 0$$ sum over H.

$\Rightarrow$ use $\boxed{\text{PP\#3}}$!

(1) $\mathcal{P}$ sends R_A, S_A oracles

(2) $\mathcal{V}$ selects $\gamma_A \xleftarrow{\$} \mathbb{F}$, evaluates $\hat{A}(\beta_A, \gamma_A)$, $Z_H(\gamma_A)$, queries $\hat{z}_A(\beta_A)$, $\hat{z}(\gamma_A)$, $R_A(\gamma_A)$, $S_A(\gamma_A)$, and checks that

$$\hat{A}(\beta_A, \gamma_A)\hat{z}(\gamma_A) - \frac{z_A(\beta_A)}{n} \stackrel{?}{=} R_A(\gamma_A)Z_H(\gamma_A) + \gamma_A S(\gamma_A)$$

if so, $\hat{z}_A$ is correct up to soundness.

$\Rightarrow$ repeat for $\hat{z}_B, B$ & $\hat{z}_C, C$.

$\boxed{5}$ Putting it all together.

- $\mathcal{P}$ knows witness w s.t., $z \triangleq (x, y, w, 1) \in \mathbb{F}^n$

$$Az \circ Bz = Cz$$

for R1CS matrices $A, B, C \in \mathbb{F}^{n \times n}$.

- $\mathcal{V}$ knows A, B, C, x, y.

⇒ Checking for R1CS-SAT w/ $\hat{z}_A, \hat{z}_B, \hat{z}_C$

(1) $\mathcal{P}$ sends $\hat{z}, \hat{z}_A, \hat{z}_B, \hat{z}_C$, R oracles

(2) V samples $\gamma \xleftarrow{\$} \mathbb{F}$, checks

$$\hat{z}_A(\gamma)\hat{z}_B(\gamma) - \hat{z}_C(\gamma) = R(\gamma) Z_H(\gamma)$$

⇒ Checking $\hat{z}_A, \hat{z}_B, \hat{z}_C$ vs $A, B, C, \hat{z}$

(3) V samples $\beta_A, \beta_B, \beta_C \xleftarrow{\$} \mathbb{F}$

(4) $\mathcal{P}$ sends $R_A, S_A, R_B, S_B, R_C, S_C$

(5) $\mathcal{V}$ evaluates & checks per previous page.

Some details we ignored:

(1) $\mathcal{P}$ sends $\hat{z}$, not $\hat{w}$ —
how can $\mathcal{V}$ ensure that

$$z = (\underbrace{x, y}_{\mathcal{V} \text{ supplies these!}}, w, 1)$$

is ok?

$\Rightarrow$ use polycommit
homomorphism tricks.

(2) Does an H exist?

$\Rightarrow$ pick $\mathbb{F}$ carefully, pad
n to a "good" size

(3) Evaluating $\hat{A}(\cdot,\cdot)$ is expensive for $\mathcal{V}$.

$\Rightarrow$ could be OK.

<u>OR</u>: "structured" computation makes evaluating $\hat{A}(\cdot,\cdot)$ cheap.

<u>OR</u>: "outsource" evaluation

<u>Idea</u>: $\mathcal{V}$ commits to $\hat{A}$, then $\mathcal{P}$ evaluates, convinces $\mathcal{V}$ that claimed eval is correct.

$\Rightarrow$ See §9 of Justin Thaler's book:

https://people.cs.georgetown.edu/jthaler/ProofsArgsAndZK.pdf