GR1CS applied to R1CS

Definition 1 (R1CS): An *R1CS*-relation is a triplet of $\mathbb{F}^{m \times n}$ matrices $(\mathrm{A}, \mathrm{B}, \mathrm{C})$ such that a witness $\boldsymbol{w} \in \mathbb{F}^n$ satisfies

$$(\mathrm{A} \cdot \boldsymbol{w}) \odot (\mathrm{B} \cdot \boldsymbol{w}) = (\mathrm{C} \cdot \boldsymbol{w}).$$

Definition 1: R1CS

Algorithm 1: Client side prover algorithm

Given Noir circuit $\mathcal{C}$ and satisfying arguments p_i.

1 **Prover** sends $\mathsf{hash}(\mathcal{C})$.

2 **Prover** derives R1CS matrices $\mathrm{A}, \mathrm{B}, \mathrm{C}$ from $\mathcal{C}$.

3 **Prover** sends the public inputs $\{(\boldsymbol{x_i}, p_i)\}_i$.

4 **Prover** computes satisfying witness vector as MLE $w(\boldsymbol{x})$ and derives

$$a(\boldsymbol{x}) = \sum_{\boldsymbol{y} \in \{0,1\}^n} \mathrm{A}(\boldsymbol{x}, \boldsymbol{y}) \cdot w(\boldsymbol{y})$$

$$b(\boldsymbol{x}) = \sum_{\boldsymbol{y} \in \{0,1\}^n} \mathrm{B}(\boldsymbol{x}, \boldsymbol{y}) \cdot w(\boldsymbol{y})$$

$$c(\boldsymbol{x}) = \sum_{\boldsymbol{y} \in \{0,1\}^n} \mathrm{C}(\boldsymbol{x}, \boldsymbol{y}) \cdot w(\boldsymbol{y}).$$

5 **Prover** sends $\mathsf{commit}_{\mathsf{WHIR}}(w)$.

6 **Prover** and **verifier** use zero-check on the R1CS relation:

7 **Verifier** send challenge $\boldsymbol{r}_0 \xleftarrow{\$} \mathbb{F}^k$.

8 **Prover** and **verifier** run sumcheck on

$$0 = \sum_{\boldsymbol{x} \in \{0,1\}^m} \mathrm{eq}(\boldsymbol{r}_0, \boldsymbol{x}) \cdot (a(\boldsymbol{x}) \cdot b(\boldsymbol{x}) - c(\boldsymbol{x}))$$

to reduces it to a claim

$$h = \mathrm{eq}(\boldsymbol{r}_0, \boldsymbol{r}_1) \cdot (a(\boldsymbol{r}_1) \cdot b(\boldsymbol{r}_1) - c(\boldsymbol{r}_1)).$$

9 **Prover** sends a_r, b_r, c_r.

10 **Verifier** checks $h = \mathrm{eq}(\boldsymbol{r}_0, \boldsymbol{r}_1) \cdot (a_r \cdot b_r - c_r)$.

11 **Verifier** sends challenge $r_2 \xleftarrow{\$} \mathbb{F}$.

12 **Prover** and **verifier** use WHIR to proof

$$p_i = w(\boldsymbol{x_i}) \text{ for each public input}$$

$$a_r = \sum_{\boldsymbol{y} \in \{0,1\}^k} \mathrm{A}(\boldsymbol{r}_1, \boldsymbol{y}) \cdot w(\boldsymbol{y})$$

$$b_r = \sum_{\boldsymbol{y} \in \{0,1\}^k} \mathrm{B}(\boldsymbol{r}_1, \boldsymbol{y}) \cdot w(\boldsymbol{y})$$

$$c_r = \sum_{\boldsymbol{y} \in \{0,1\}^k} \mathrm{C}(\boldsymbol{r}_1, \boldsymbol{y}) \cdot w(\boldsymbol{y}).$$

$$\sum_{\boldsymbol{y} \in \{0,1\}^k} \left(\mathrm{A}(\boldsymbol{r}_1, \boldsymbol{y}) + r_2 \cdot \mathrm{B}(\boldsymbol{r}_1, \boldsymbol{y}) + r_2^2 \cdot \mathrm{C}(\boldsymbol{r}_1, \boldsymbol{y}) + \sum_i r_2^{3+i} \cdot \mathrm{eq}(\boldsymbol{x}_i, \boldsymbol{y}) \right) \cdot w(\boldsymbol{y})$$

Note that up to Line 3 the prover work is only circuit specific and can be cached between instances.

Prover needs vectors $w, a, b, c, A + r \cdot B + r^2 \cdot C$. The first four can be computed in a single pass as part of witness generation. The last three require $\boldsymbol{r}_1$ and need to be computed in a second pass.