Notes on HyperNova

arnaucube

May 2023

Abstract

Notes taken while reading about Spartan [1], [2].

Usually while reading papers I take handwritten notes, this document contains some of them re-written to $LaTeX$.

The notes are not complete, don't include all the steps neither all the proofs.

Contents

1 CCS

1.1 R1CS to CCS overview

R1CS instance: $S_{R1CS} = (m, n, N, l, A, B, C)$

CCS instance: $S_{CCS} = (m, n, N, l, t, q, d, M, S, c)$

R1CS-to-CCS parameters:
$n = n,\ m = m,\ N = N,\ l = l,\ t = 3,\ q = 2,\ d = 2$
$M = \{A, B, C\},\ S = \{\{0,\ 1\},\ \{2\}\},\ c = \{1, -1\}$

Then, we can see that the CCS relation:

$$\sum_{i=0}^{q-1} c_i \cdot \bigcirc_{j \in S_i} M_j \cdot z == 0$$

where $z = (w, 1, x) \in \mathbb{F}^n$.
In our R1CS-to-CCS parameters is equivalent to

$$
\begin{aligned}
&c_0 \cdot ((M_0 z) \circ (M_1 z)) + c_1 \cdot (M_2 z) == 0\\
\Longrightarrow &1 \cdot ((Az) \circ (Bz)) + (-1) \cdot (Cz) == 0\\
\Longrightarrow &((Az) \circ (Bz)) - (Cz) == 0
\end{aligned}
$$

which is equivalent to the R1CS relation: $Az \circ Bz == Cz$

An example of the conversion from R1CS to CCS implemented in SageMath can be found at
https://github.com/arnaucube/math/blob/master/r1cs-ccs.sage.

1.2 Committed CCS

R_{CCCS} instance: (C, x), where C is a commitment to a multilinear polynomial in $s' - 1$ variables.

Sat if:

i. $\text{Commit}(pp, \widetilde{w}) = C$

ii. $\sum_{i=1}^{q} c_i \cdot \left(\prod_{j \in S_i} \left(\sum_{y \in \{0,1\}^{\log m}} \widetilde{M}_j(x, y) \cdot \widetilde{z}(y) \right) \right)$
where $\widetilde{z}(y) = \widetilde{(w, 1, \mathsf{x})}(x)\ \forall x \in \{0,1\}^{s'}$

1.3 Linearized Committed CCS

R_{LCCCS} instance: $(C, u, \mathsf{x}, r, v_1, \ldots, v_t)$, where C is a commitment to a multilinear polynomial in $s' - 1$ variables, and $u \in \mathbb{F},\ \mathsf{x} \in \mathbb{F}^l,\ r \in \mathbb{F}^s,\ v_i \in \mathbb{F}\ \forall i \in [t]$.

Sat if:

i. $\text{Commit}(pp, \widetilde{w}) = C$

ii. $\forall i \in [t],\ v_i = \sum_{y \in \{0,1\}^{s'}} \widetilde{M}_i(r, y) \cdot \widetilde{z}(y)$
where $\widetilde{z}(y) = \widetilde{(w, u, \mathsf{x})}(x)\ \forall x \in \{0,1\}^{s'}$

2 Multifolding Scheme for CCS

Recall sum-check protocol:
$\underline{C \leftarrow <P, V(r)> (g, l, d, T)}$:
$T = \sum_{x_1 \in \{0,1\}} \sum_{x_2 \in \{0,1\}} \cdots \sum_{x_l \in \{0,1\}} g(x_1, x_2, \ldots, x_l)$ l-variate polynomial g, degree $\leq d$ in each variable.

let $s = \log m,\ s' = \log n$.

1. $V \to P : \gamma \in^R \mathbb{F},\ \beta \in^R \mathbb{F}^s$
2. $V : r'_x \in^R \mathbb{F}^s$

3. $V \leftrightarrow P$: sum-check protocol:

$$c \leftarrow < P, V(r'_x) > (g, s, d+1, \sum_{j \in [t]} \gamma^j \cdot v_j)$$

where:

$$g(x) := \left(\sum_{j \in [t]} \gamma^j \cdot L_j(x) \right) + \gamma^{t+1} \cdot Q(x)$$

$$L_j(x) := \widetilde{eq}(r_x, x) \cdot \left(\sum_{y \in \{0,1\}^{s'}} \widetilde{M}_j(x, y) \cdot \widetilde{z}_1(y) \right)$$

$$Q(x) := \widetilde{eq}(\beta, x) \cdot \left(\sum_{i=1}^{q} c_i \cdot \prod_{j \in S_i} \left(\sum_{y \in \{0,1\}^{s'}} \widetilde{M}_j(x, y) \cdot \widetilde{z}_2(y) \right) \right)$$

4. $P \to V$: $((\sigma_1, \ldots, \sigma_t), (\theta_1, \ldots, \theta_t))$ where

$$\sigma_j = \sum_{y \in \{0,1\}^{s'}} \widetilde{M}_j(x, y) \cdot \widetilde{z}_1(y)$$

$$\theta_j = \sum_{y \in \{0,1\}^{s'}} \widetilde{M}_j(x, y) \cdot \widetilde{z}_2(y)$$

5. V: $e_1 \leftarrow \widetilde{eq}(r_x, r'_x)$, $e_2 \leftarrow \widetilde{eq}(\beta, r'_x)$
   check:

$$c = \left(\sum_{j \in [t]} \gamma^j e_1 \sigma_j + \gamma^{t+1} e_2 \left(\sum_{i=1}^{q} c_i \cdot \prod_{j \in S_i} \sigma \right) \right)$$

6. $V \to P : \rho \in^R \mathbb{F}$

7. V, P: output the folded LCCCS instance $(C', u', \mathsf{x}', r'_x, v'_1, \ldots, v'_t)$, where $\forall i \in [t]$:

$$\begin{aligned} C' &\leftarrow C_1 + \rho \cdot C_2 \\ u' &\leftarrow u + \rho \cdot 1 \\ \mathsf{x}' &\leftarrow \mathsf{x}_1 + \rho \cdot \mathsf{x}_2 \\ v'_i &\leftarrow \sigma_i + \rho \cdot \theta_i \end{aligned}$$

8. P: output folded witness: $\widetilde{w}' \leftarrow \widetilde{w}_1 + \rho \cdot \widetilde{w}_2$.

References

[1] Abhiram Kothapalli and Srinath Setty. Hypernova: Recursive arguments for customizable constraint systems. Cryptology ePrint Archive, Paper 2023/573, 2023. https://eprint.iacr.org/2023/573.

[2] Srinath Setty, Justin Thaler, and Riad Wahby. Customizable constraint systems for succinct arguments. Cryptology ePrint Archive, Paper 2023/552, 2023. https://eprint.iacr.org/2023/552.