Math Companion to Soundcalc

December 4, 2025

1 Notation and Preliminaries

1.1 Fields

Fields of size q are denoted as $\mathbb{F}_q$ or simply $\mathbb{F}$.

1.2 Reed-Solomon codes

We use the following notation:

- $RS[\mathbb{F}, S, \rho]$: Reed-Solomon code over the field $\mathbb{F}$ with evaluation domain S and rate ρ.
- $\deg(f)$: degree of the polynomial f.

2 FRI

This section contains the soundness formula for the FRI protocol.

2.1 FRI parameters

Global parameters used in the FRI analysis:

- m_J — Johnson parameter.
- r_{FRI} — number of FRI rounds.
- Folding factors $\widehat{\text{folds}} = [k_0, k_1, \dots, k_{r_{FRI}-1}]$;
- t — number of queries.
- θ.
- δ.
- ρ — rate of the Reed-Solomon code.
- l_t — trace length.
- L — list size.
- $b_{\text{grind},Q}$ — grinding parameter for the query phase.
- n — witness size.
- b_{hash} — number of bits in the hash function output.
- b_{proof} — proof size in bits.
- s_{btch} — batch size.

Notation specific to the Johnson bound:

-

2.2 Fixed constants

We fix the following constants for the soundness calculator:

- $m_J = 16$. Set in

  `fri.py/get_johnson_parameter_m()`

2.3 Security level for a FRI-based VM

The security level is calculated in

`zkvms/fri_based_vm.py/get_security_levels()`

.

It is done separately for two different regimes: UDR and JBR — using the same procedure:

1. Calculate the FRI round-by-round soundness errors $\epsilon_{\text{FRI},U}, \epsilon_{\text{FRI},J}$ using the formula from the next section.
2. Obtain optimal δ_U, δ_J parameters.
3. Obtain the list sizes L_U, L_J.

2.3.1 RBR soundness in UDR

2.3.2 RBR soundness in UDR

2.4 Soundness formula

This is calculated in

`fri.py/get_FRI_query_phase_error()`

.

Query phase error:

$$\epsilon_{\text{query}} = (1-\theta)^t \cdot 2^{-b_{\text{grind},Q}} \tag{1}$$

The query phase error without grinding is computed as per [?][1]

2.5 Proof size

This calculation is performed in

`fri.py/get_FRI_proof_size_bits()`

. The FRI proof contains two parts: Merkle roots, and one "openings" per query, where an "opening" is a Merkle path for each folding layer. For each layer we count the size that this layer contributes, which includes the root and all Merkle paths.

Initial round: one root and one path per query. We assume that for the initial functions, there is only one Merkle root, and each leaf i for that root contains symbols i for all initial functions.

Folding rounds: we assume that "siblings" for the following layers are grouped together in one leaf. This is natural as they always need to be opened together.

[1] Code refers to (7) and Th2 of [Hab22]

The proof size is calculated as follows:

$$
\begin{aligned}
b_{\text{proof}} = {} & \underbrace{b_{\text{hash}} + t \cdot MP(\frac{n}{\widehat{\text{folds}}[0]}, s_{\text{btch}}, |\mathbb{F}|, b_{\text{hash}})}_{\text{Initial round}} + \\
& + \underbrace{\sum_{1 \le i \le r_{FRI}-2} \left(b_{\text{hash}} + t \cdot MP(\frac{n}{\prod_{1 \le j \le i} \widehat{\text{folds}}[j]}, s_{\text{btch}}, |\mathbb{F}|, b_{\text{hash}}) \right)}_{\text{Folding rounds but last}} + \\
& + \underbrace{\left(b_{\text{hash}} + t \cdot MP(\frac{n}{\widehat{\text{folds}}[r_{FRI}-1] \prod_{1 \le j \le r_{FRI}-1} \widehat{\text{folds}}[j]}, s_{\text{btch}}, |\mathbb{F}|, b_{\text{hash}}) \right)}_{\text{Last folding round}}
\end{aligned}
\tag{2}
$$

where $MP(n, s, q, b)$ is the Merkle path size calculated as

$$
MP(n, s, q, b) = \underbrace{sq}_{\text{leaf size}} + \underbrace{sq}_{\text{sibling}} + \underbrace{\lceil \log_2 n \rceil \cdot b}_{\text{co-path}} \tag{3}
$$