STIR (blueprint)

LeastAuthority

April 24, 2025

 /- Copyright (c) 2025 ZKLib Contributors. All rights reserved. Released under Apache 2.0 license as described in the file LICENSE. Authors: Least Authority -/

Chapter 1

Preliminaries

Definition 1.1 (Interactive Oracle Proofs of Proximity (IOPP)). *A* k-round public-coin interactive-oracle proof of proximity (IOPP) *for a ternary relation* $\mathcal{R} = \{(x, y, w)\}$ *is an interactive protocol between a prover* P *and a verifier* V *defined as follows.*

- *The prover receives* (x, y, w)*, while the verifier receives* x *and oracle access to* y*.*
- *For each round* $i \in [k]$ *the verifier sends a uniformly random message* α_i *to the prover, who responds with a proof string* π_i*.*
- *After* k *rounds, the verifier may query* y *and the proof strings* $\pi_1, \dots, \pi_k$ *and finally outputs a decision bit.*

Formally, let $\mathsf{IOP} = (\mathsf{P}, \mathsf{V})$ *where* P *is an interactive algorithm and* V *is an interactive-oracle algorithm. The protocol has* ***perfect completeness*** *and* ***soundness error*** β *if the following conditions hold.*

Perfect completeness. *For every* $(x, y, w) \in \mathcal{R}$*,*

$$\Pr_{\alpha_1,\dots,\alpha_k}\Big[\mathsf{V}^{y,\pi_1,\dots,\pi_k}(x, \alpha_1, \dots, \alpha_k) = 1 \;\Big|\; \pi_1 \leftarrow \mathsf{P}(x, y, w),\ \dots,\ \pi_k \leftarrow \mathsf{P}(x, y, w, \alpha_1, \dots, \alpha_k)\Big] = 1.$$

Soundness. *For every* $(x, y) \notin L(\mathcal{R})$ *and every (unbounded) malicious prover* $\tilde{\mathsf{P}}$*,*

$$\Pr_{\alpha_1,\dots,\alpha_k}\Big[\mathsf{V}^{y,\pi_1,\dots,\pi_k}(x, \alpha_1, \dots, \alpha_k) = 1 \;\Big|\; \pi_1 \leftarrow \tilde{\mathsf{P}}(\alpha_1),\ \dots,\ \pi_k \leftarrow \tilde{\mathsf{P}}(x, y, \alpha_1, \dots, \alpha_k)\Big] \le \beta(x, y).$$

When the soundness error depends only on the input lengths and on the proximity δ *of* y *to the language*

$$L_x \;:=\; \{\, y' \mid \exists w,\ (x, y', w) \in \mathcal{R}\},$$

we write $\beta(|x|, |y|, \delta)$*, or simply* $\beta(\delta)$ *when* $|x|$ *and* $|y|$ *are clear from context.*

Definition 1.2. *Let* $k \in \mathbb{N}$ *be an integer,* $\mathbb{F}$ *be a finite field and* $\mathcal{L} \subset \mathbb{F}$ *be a subset of* $\mathbb{F}$*. Then*

$$\mathcal{L}^k := \{x^k \ s.t.\ x \in \mathcal{L}\}$$

Definition 1.3 (Reed-Solomon Code)**.** *The* Reed-Solomon code *over finite field* $\mathbb{F}$*, evaluation domain* $\mathcal{L} \subseteq \mathbb{F}$ *and degree* $d \in \mathbb{N}$ *is the set of evaluations (over* $\mathcal{L}$*) of univariate polynomials (over* $\mathbb{F}$*) of degree less than* d*:*

$$\mathrm{RS}[\mathbb{F}, \mathcal{L}, d] := \left\{ f : \mathcal{L} \to \mathbb{F} \;\middle|\; \exists \hat{f} \in \mathbb{F}^{<d}[X] \text{ such that } \forall x \in \mathcal{L},\ f(x) = \hat{f}(x) \right\}.$$

The rate of $\mathrm{RS}[\mathbb{F}, \mathcal{L}, d]$ *is* $\rho := \frac{d}{|\mathcal{L}|}$.

Given a code $\mathcal{C} := \mathrm{RS}[\mathbb{F}, \mathcal{L}, d]$ *and a function* $f : \mathcal{L} \to \mathbb{F}$*, we sometimes use* $\hat{f} \in \mathbb{F}^{<d}[X]$ *to denote a nearest polynomial to* f *on* $\mathcal{L}$ *(breaking ties arbitrarily).*

Remark 1.4. *Note that the evaluation domain* $\mathcal{L} \subseteq \mathbb{F}$ *is a non-empty set.*

Definition 1.5. *For a Reed-Solomon code* $\mathcal{C} := \mathrm{RS}[\mathbb{F}, \mathcal{L}, d]$*, parameter* $\delta \in [0, 1]$*, and a function* $f : \mathcal{L} \to \mathbb{F}$*, let* $\mathsf{List}(f, d, \delta)$ *denote the list of codewords in* $\mathcal{C}$ *whose relative Hamming distance from* f *is at most* δ*. We say that* $\mathcal{C}$ *is* (δ, l)-list decodable *if*

$$\left|\mathsf{List}(f, d, \delta)\right| \leq l \quad \text{for every function } f.$$

The Johnson bound provides an upper bound on the list size of this Reed-Solomon code:

Theorem 1.6 (Johnson bound)**.** *The Reed-Solomon code* $\mathrm{RS}[\mathbb{F}, \mathcal{L}, d]$ *is* $(1 - \sqrt{\rho} - \eta, \frac{1}{2\eta\rho})$*-list-decodable for every* $\eta \in (0, 1 - \sqrt{\rho})$*, where* $\rho := \frac{d}{|\mathcal{L}|}$ *is the rate of the code.*

Chapter 2

Tools for Reed-Solomon codes

2.1 Random linear combination as a proximity generator

Theorem 2.1. *Let $\mathcal{C} := \mathrm{RS}[\mathbb{F}, \mathcal{L}, d]$ be a Reed-Solomon code with rate $\rho := \frac{d}{|\mathcal{L}|}$ and let $B'(\rho) := \sqrt{\rho}$. For every $\delta \in (0, 1 - B'(\rho))$ and functions $f_1, \dots, f_m : \mathcal{L} \to \mathbb{F}$, if*

$$\Pr_{r \leftarrow \mathbb{F}}\Big[\Delta\Big(\sum_{j=1}^{m} r^{j-1} \cdot f_j, \mathrm{RS}[\mathbb{F}, \mathcal{L}, d]\Big) \leq \delta\Big] > \mathsf{err}'(d, \rho, \delta, m),$$

then there exists a subset $S \subseteq \mathcal{L}$ with $|S| \geq (1 - \delta) \cdot |L|$, and for every $i \in [m]$, there exists $u \in \mathrm{RS}[\mathbb{F}, \mathcal{L}, d]$ such that $f_i(S) = u(S)$.

Above, $\mathsf{err}'(d, \rho, \delta, m)$ is defined as follows:

- *if $\delta \in \left(0, \frac{1-\rho}{2}\right]$ then*

$$\mathsf{err}'(d, \rho, \delta, m) = \frac{(m-1) \cdot d}{\rho \cdot |\mathbb{F}|}$$

- *if $\delta \in \left(\frac{1-\rho}{2}, 1 - \sqrt{\rho}\right)$ then*

$$\mathsf{err}'(d, \rho, \delta, m) = \frac{(m-1) \cdot d^2}{|\mathbb{F}| \cdot \left(2 \cdot \min 1 - \sqrt{\rho} - \delta, \frac{\sqrt{\rho}}{20}\right)^7}$$

2.2 Univariate Function Quotienting

In the following, we start by defining the *quotient* of a univariate function.

Definition 2.2. *Let $f : \mathcal{L} \to \mathbb{F}$ be a function, $S \subseteq \mathbb{F}$ be a set, and $\mathsf{Ans}, \mathsf{Fill} : S \to \mathbb{F}$ be functions. Let $\widehat{\mathsf{Ans}} \in \mathbb{F}^{<|S|}[X]$ be the (unique) polynomial with $\widehat{\mathsf{Ans}}(x) = \mathsf{Ans}(x)$ for every $x \in S$, and let*

$\hat{V}_S \in \mathbb{F}^{<|S|+1}[X]$ *be the unique non-zero polynomial with* $\hat{V}_S(x) = 0$ *for every* $x \in S$. *The* quotient function $\mathsf{Quotient}(f, S, \mathsf{Ans}, \mathsf{Fill}) : \mathcal{L} \to \mathbb{F}$ *is defined as follows:*

$$\forall x \in \mathcal{L}, \quad \mathsf{Quotient}(f, S, \mathsf{Ans}, \mathsf{Fill})(x) := \begin{cases} \mathsf{Fill}(x) & \text{if } x \in S \\ \dfrac{f(x) - \hat{\mathsf{Ans}}(x)}{\hat{V}_S(x)} & \text{otherwise} \end{cases}$$

Next we define the polynomial quotient operator, which quotients a polynomial relative to its output on evaluation points. The polynomial quotient is a polynomial of lower degree.

Definition 2.3. *Let* $\hat{f} \in \mathbb{F}^{<d}[X]$ *be a polynomial and* $S \subseteq \mathbb{F}$ *be a set, let* $\hat{V}_S \in \mathbb{F}^{<|S|+1}[X]$ *be the unique non-zero polynomial with* $\hat{V}_S(x) = 0$ *for every* $x \in S$. *The* polynomial quotient $\mathsf{PolyQuotient}(\hat{f}, S) \in \mathbb{F}^{<d-|S|}[X]$ *is defined as follows:*

$$\mathsf{PolyQuotient}(\hat{f}, S)(X) := \frac{\hat{f}(X) - \hat{\mathsf{Ans}}(X)}{\hat{V}_S(X)}$$

The following lemma, implicit in prior works, shows that if the function is "quotiented by the wrong value", then its quotient is far from low-degree.

Lemma 2.4. *Let* $f : \mathcal{L} \to \mathbb{F}$ *be a function,* $d \in \mathbb{N}$ *be the degree parameter,* $\delta \in (0, 1)$ *be a distance parameter,* $S \subseteq \mathbb{F}$ *be a set with* $|S| < d$, *and* $\mathsf{Ans}, \mathsf{Fill} : S \to \mathbb{F}$ *are functions. Suppose that for every* $u \in \mathsf{List}(f, d, \delta)$ *there exists* $x \in S$ *with* $\hat{u}(x) \neq \mathsf{Ans}(x)$. *Then*

$$\Delta(\mathsf{Quotient}(f, S, \mathsf{Ans}, \mathsf{Fill}), \mathrm{RS}[\mathbb{F}, \mathcal{L}, d - |S|]) + \frac{|T|}{|\mathcal{L}|} > \delta,$$

where $T := \{x \in \mathcal{L} \cap S : \hat{\mathsf{Ans}}(x) \neq f(x)\}$.

2.3 Out of domain sampling

Lemma 2.5. *Let* $f : \mathcal{L} \to \mathbb{F}$ *be a function,* $d \in \mathbb{N}$ *be a degree parameter,* $s \in \mathbb{N}$ *be a repetition parameter, and* $\delta \in [0, 1]$ *be a distance parameter. If* $\mathrm{RS}[\mathbb{F}, \mathcal{L}, d]$ *be* (d, l)*-list decodable then*

$$\Pr_{r_1,\ldots,r_s \leftarrow \mathbb{F}\setminus\mathcal{L}}[\exists \text{ distinct } u, u' \in \mathsf{List}(f, d, \delta) : \forall i \in [s], \hat{u}(r_i) = \hat{u}'(r_i)] \leq \binom{l}{2} \cdot \left(\frac{d-1}{|\mathbb{F}| - |\mathcal{L}|}\right)^s$$
$$\leq \left(\frac{l^2}{2}\right) \cdot \left(\frac{d}{|\mathbb{F}| - |\mathcal{L}|}\right)^s$$

2.4 Folding univariate functions

STIR relies on k-wise folding of functions and polynomials - this is similar to prior works, although presented in a slightly different form. As shown below, folding a function preserves proximity from the Reed-Solomon code with high probability.

The folding operator is based on the following fact, decomposing univariate polynomials into bivariate ones.

Fact 2.6. *Given a polynomial $\hat{q} \in \mathbb{F}[X]$:*

- *For every univariate polynomial $\hat{f} \in \mathbb{F}[X]$, there exists a unique bivariate polynomial $\hat{Q} \in \mathbb{F}[X, Y]$ with $\mathsf{deg}_X(\hat{Q}) := \lfloor \mathsf{deg}(\hat{f})/\mathsf{deg}(\hat{q}) \rfloor$ and $\mathsf{deg}_Y(\hat{Q}) < \mathsf{deg}(\hat{q})$ such that $\hat{f}(Z) = \hat{Q}(\hat{q}(Z), Z)$. Moreover $\hat{Q}$ can be computed efficiently given $\hat{f}$ and $\hat{q}$. Observe that if $\mathsf{deg}(\hat{f}) < t \cdot \mathsf{deg}(\hat{q})$ then $\mathsf{deg}(\hat{Q}) < t$.*

- *For every $\hat{Q}[X, Y]$ with $\mathsf{deg}_X(\hat{Q}) < t$ and $\mathsf{deg}_Y(\hat{Q}) < \mathsf{deg}(\hat{q})$, the polynomial $\hat{f}(Z) = \hat{Q}(\hat{q}(Z), Z)$ has degree $\mathsf{deg}(\hat{f}) < t \cdot \mathsf{deg}(\hat{q})$.*

Below, we define folding of a polynomial followed by folding of a function.

Definition 2.7. *Given a polynomial $\hat{f} \in \mathbb{F}^{<d}[X]$, a folding parameter $k \in \mathbb{N}$ and $r \in \mathbb{F}$, we define a polynomial $\mathsf{PolyFold}(\hat{f}, k, r) \in \mathbb{F}^{d/k}[X]$ as follows. Let $\hat{Q}[X, Y]$ be the bivariate polynomial derived from $\hat{f}$ using Fact 2.6 with $\hat{q}(X) := X^k$. Then $\mathsf{PolyFold}(\hat{f}, k, r)(X) := \hat{Q}(X, r)$.*

Definition 2.8. *Let $f : \mathcal{L} \to \mathbb{F}$ be a function, $k \in \mathbb{N}$ a folding parameter and $\alpha \in \mathbb{F}$. For every $x \in \mathcal{L}^k$, let $\hat{p}_x \in \mathbb{F}^{<k}[X]$ be the polynomial where $\hat{p}_x(y) = f(y)$ for every $y \in \mathcal{L}$ such that $y^k = x$. We define $\mathsf{Fold}(f, k, \alpha) : \mathcal{L} \to \mathbb{F}$ as follows.*

$$\mathsf{Fold}(f, k, \alpha) := \hat{p}_x(\alpha).$$

In order to compute $\mathsf{Fold}(f, k, \alpha)(x)$ it suffices to interpolate the k values $\{f(y) : y \in \mathcal{L} \text{ s.t. } y^k = x\}$ into the polynomial $\hat{p}_x$ and evaluate this polynomial at α.

The following lemma shows that the distance of a function is preserved under folding. If a functions f has distance δ to a Reed-Solomon code then, with high probability over the choice of folding randomness, its folding also has a distance of δ to the "k-wise folded" Reed-Solomon code.

Lemma 2.9. *For every function $f : \mathcal{L} \to \mathbb{F}$, degree parameter $d \in \mathbb{N}$, folding parameter $k \in \mathbb{N}$, distance parameter $\delta \in (0, \min\{\Delta(\mathsf{Fold}[f, k, r^{\mathsf{fold}}], \mathrm{RS}[\mathbb{F}, \mathcal{L}^k, d/k]), 1 - \mathsf{B}^*(\rho)\})$, letting $\rho := \frac{d}{|\mathcal{L}|}$,*

$$\Pr_{r^{\mathsf{fold}} \leftarrow \mathbb{F}}[\Delta(\mathsf{Fold}[f, k, r^{\mathsf{fold}}], \mathrm{RS}[\mathbb{F}, \mathcal{L}^k, d/k]) < \delta] > \mathsf{err}^*(d/k, \rho, \delta, k).$$

Above, B^* and err^* are the proximity bound and error (respectively) described in Section 2.1.

2.5 Combine functions of varying degrees

We show a new method for combining functions of varying degrees with minimal proximity require- ments using geometric sums. We begin by recalling a fact about geometric sums.

Fact 2.10. *Let $\mathbb{F}$ be a field, $r \in \mathbb{F}$ be a field element, $a \in \mathbb{N}$ be a natural number. Then*

$$\sum_{i=0}^{a} r^i := \begin{cases} \left(\frac{1 - r^{a+1}}{1 - r}\right) & r \neq 1 \\ a + 1 & r = 1 \end{cases}$$

Definition 2.11. *Given target degree $d^* \in \mathbb{N}$, shifting parameter $r \in \mathbb{F}$, functions $f_1, \dots, f_m : \mathcal{L} \to \mathbb{F}$, and degrees $0 \le d_1, \dots, d_m \le d^*$, we define* $\mathsf{Combine}(d^*, r, (f_1, d_1), \dots, (f_m, d_m)) : \mathcal{L} \to \mathbb{F}$ *as follows:*

$$\mathsf{Combine}(d^*, r, (f_1, d_1), \dots, (f_m, d_m))(x) := \sum_{i=1}^{m} r_i \cdot f_i(x) \cdot \Big(\sum_{l=0}^{d^*-d_i} (r \cdot x)^l \Big)$$
$$= \begin{cases} \sum_{i=1}^{m} r_i \cdot f_i(x) \cdot \left(\frac{1-(xr)^{d^*-d_i+1}}{1-xr} \right) & x \cdot r \neq 1 \\ \sum_{i=1}^{m} r_i \cdot f_i(x) \cdot (d^* - d_i + 1) & x \cdot r = 1 \end{cases}$$

Above, $r_1 := 1$, $r_i := r^{i-1+\sum_{j<i}(d^*-d_i)}$ for $i > 1$.

Definition 2.12. *Given target degree $d^* \in \mathbb{N}$, shifting parameter $r \in \mathbb{F}$, function $f : \mathcal{L} \to \mathbb{F}$, and degree $0 \le d \le d^*$, we define* $\mathsf{DegCor}(d^*, r, f, d)$ *as follows.*

$$\mathsf{DegCor}(d^*, r, f, d)(x) := f(x) \cdot \left(\sum_{i=0}^{m} (r \cdot x)^l \right) = \begin{cases} f(x) \cdot \frac{1-(xr)^{d^*-d_i+1}}{1-xr} & x \cdot r \neq 1 \\ f(x) \cdot (d^* - d_i + 1) & x \cdot r = 1 \end{cases}$$

(Observe that $\mathsf{DegCor}(d^*, r, f, d) = \mathsf{Combine}(d^*, r, (f, d))$*.)*

Below it is shown that combining multiple polynomials of varying degrees can be done as long as the proximity error is bounded by $(\min \{1 - \mathsf{B}^*(\rho), 1 - \rho - 1/|\mathcal{L}|\})$.

Lemma 2.13. *Let d^* be a target degree, $f_1, \dots, f_m : \mathcal{L} \to \mathbb{F}$ be functions, $0 \le d_1, \dots, d_m \le d^*$ be degrees, $\delta \in \min \{1 - \mathsf{B}^*(\rho), 1 - \rho - 1/|\mathcal{L}|\}$ be a distance parameter, where $\rho = d^*/|\mathcal{L}|$. If*

$$\Pr_{r \leftarrow \mathbb{F}}[\Delta(\mathsf{Combine}(d^*, r, (f_1, d_1), \dots, (f_m, d_m)), \mathrm{RS}[\mathbb{F}, \mathcal{L}, d^*])] > \mathsf{err}^*(d^*, \rho, \delta, m \cdot (d^* + 1) - \sum_{i=1}^{m} d_i),$$

then there exists $S \subseteq \mathcal{L}$ with $|S| \ge (1 - \delta) \cdot |\mathcal{L}|$, and

$$\forall i \in [m], \exists u \in \mathrm{RS}[\mathbb{F}, \mathcal{L}, d_i], f_i(S) = u(S).$$

Note that this implies $\Delta(f_i, \mathrm{RS}[\mathbb{F}, \mathcal{L}, d_i]) < \delta$ for every i. Above, B^* and err^* are the proximity bound and error (respectively) described in Section 2.1.

Chapter 3

STIR

3.1 STIR Main Theorem

Theorem 3.1 (STIR Main Theorem)**.** *Consider the following ingrediants:*

- *A security parameter* $\lambda \in \mathbb{N}$*.*
- *A Reed-Solomon code* $\mathrm{RS}[\mathbb{F}, \mathcal{L}, d]$ *with* $\rho := \frac{d}{|\mathcal{L}|}$ *where* d *is a power of* 2*, and* $\mathcal{L}$ *is a smooth domain.*
- *A proximity parameter* $\delta \in (0, 1 - 1.05 \cdot \sqrt{\rho})$*.*
- *A folding parameter* $k \in \mathbb{N}$ *that is power of* 2 *with* $k \geq 4$*.*

If $|\mathbb{F}| = \Omega(\frac{\lambda \cdot 2^{\lambda} \cdot d^2 \cdot |\mathcal{L}|^2}{\log(1/\rho)})$*, there is a public-coin IOPP for* $\mathrm{RS}[\mathbb{F}, \mathcal{L}, d]$ *with the following parameters:*

- *Round-by-round soundness error* $2^{-\lambda}$*.*
- *Round complexity:* $M := O(\log_k d)$*.*
- *Proof length:* $|\mathcal{L}| + O_k(\log d)$*.*
- *Query complexity to the input:* $\frac{\lambda}{-\log{(1-\delta)}}$*.*
- *Query complexity to the proof strings:* $O_k(\log d + \lambda \cdot \log\left(\frac{\log d}{\log 1/\rho}\right))$*.*

3.2 The STIR Construction

Consider the following ingrediants:

- a field $\mathbb{F}$,
- an iteration count $M \in \mathbb{N}$,
- an initial degree parameter $d \in \mathbb{N}$ that is a power of 2,
- a folding parameters $k_0, \ldots, k_M \in \mathbb{N}$ that are powers of 2 with $d \geq \prod_i k_i$,
- evaluation domains $\mathcal{L}_0, \ldots, \mathcal{L}_M \subseteq \mathbb{F}$ where $\mathcal{L}_i$ is a smooth coset of $\mathbb{F}^*$ with $|\mathcal{L}_i| > \frac{d}{\prod_{j<i} k^j}$

- repetition parameters $t_0, \dots, t_M \in \mathbb{N}$ where $t_i + 1 \leq \frac{d}{\prod_{j \leq i} k^j}$ for every $i \in \{0, \dots, M-1\}$,
- out of domain repetition parameter $s \in \mathbb{N}$.

For every $i \in \{0, \dots, M\}$, set $d_i := \frac{d}{\prod_{j<i} k^j}$. The protocol proceeds as follows.

- **Initial function:** Let $f_0 : \mathcal{L} \to \mathbb{F}$ be an oracle function. In the honest case, $f_0 = \mathrm{RS}[\mathbb{F}, \mathcal{L}_0, d_0]$ and the prover has access to the polynomial $\hat{f} \in \mathbb{F}^{<d_0}[X]$ whose restriction to $\mathcal{L}_0$ is f_0.
- **Initial folding:** The verifier sends $r^{\mathsf{Fold}} \leftarrow \mathbb{F}$
- **Interaction phase loop:** For $i \in \{1, \dots, M\}$:
  1. **Send folded function:** The prover sends a function $g_i : \mathcal{L}_i \to \mathbb{F}$. In the honest case g_i is the evaluation of the polynomial $\hat{g}_i := \mathsf{PolyFold}(\hat{f}_{i-1}, k_{i-1}, r^{\mathsf{fold}}_{i-1})$ over $\mathcal{L}_i$.
  2. **Out-of-domain samples:** The verifier sends $r^{\mathsf{out}}_{i,1}, \dots, r^{\mathsf{out}}_{i,s} \in \mathbb{F} \setminus \mathcal{L}_i$
  3. **Out-of-domain reply:** The prover sends field elements $\beta_{i,1}, \dots, \beta_{i,s} \in \mathbb{F}$. In the honest case, $\beta_{i,j} := \hat{g}_i(r^{\mathsf{out}}_{i,j})$.
  4. **STIR message:** The verifier sends $r^{\mathsf{fold}}_i, r^{\mathsf{shift}}_i \in \mathbb{F}$ and $r^{\mathsf{shift}}_{i,1}, \dots, r^{\mathsf{shift}}_{i,t_{i-1}} \leftarrow \mathcal{L}^{k_i - 1}_{i-1}$
  5. **Define next polynomial and send hole fills:** The prover sends the oracle message $\mathsf{Fill}_i := (r^{\mathsf{shift}}_{i,1}, \dots, r^{\mathsf{shift}}_{i,t_{i-1}}) \cap \mathcal{L}_i \to \mathbb{F}$. In the honest case, the prover defines $\mathcal{G}_i = \{r^{\mathsf{out}}_{i,1}, \dots, r^{\mathsf{out}}_{i,s}, r^{\mathsf{shift}}_{i,1}, \dots, r^{\mathsf{shift}}_{i,t_{i-1}}\}$, $\hat{g}'_i := \mathsf{PolyQuotient}(\hat{g}_i, \mathcal{G}_i)$ and $\mathsf{Fill}_i(r^{\mathsf{shift}}_{i,j}) := \hat{g}'_i(r^{\mathsf{shift}}_{i,j})$ (If $r^{\mathsf{shift}}_{i,j} \in \mathcal{L}_i$)

     Additionally, the honest prover defines the degree-corrected polynomial $\hat{f}_i \in \mathbb{F}^{<d}[X]$ as follows:
     $$\hat{f}_i := \mathsf{DegCor}(d_i, r^{\mathsf{comb}}_i, \hat{g}'_i, d_i - |\mathcal{G}_i|)$$
     The protocol proceeds to the next iteration with $\hat{f}_i$.
- **Final round:** The prover sends d_M coefficients of a polynomial $\hat{p} \in \mathbb{F}^{<d_M}[X]$. In the honest case, $\hat{p} := \mathsf{Fold}(\hat{f}_M, k_M, r^{\mathsf{fold}_M})$.
- **Verifier decision phase:**
  1. **Main loop:** For $i = 1, \dots, M$:
     (a) For every $j \in [t_{i-1}]$, query $\mathsf{Fold}(f_{i-1}, k_{i-1}, r^{\mathsf{fold}}_{i-1})$ at $r^{\mathsf{shift}}_{i,j}$. This involves querying f_{i-1} at all k_{i-1} points $x \in \mathcal{L}_{i-1}$ with $x^{k_i - 1} = r^{\mathsf{shift}}_{i,j}$.
     (b) Define $\mathcal{G}_i = \{r^{\mathsf{out}}_{i,1}, \dots, r^{\mathsf{out}}_{i,s}, r^{\mathsf{shift}}_{i,1}, \dots, r^{\mathsf{shift}}_{i,t_{i-1}}\}$ and let $\mathsf{Ans}_i : \mathcal{G}_i \to \mathbb{F}$ be the function where $\mathsf{Ans}_i(r^{\mathsf{out}}_{i,j}) = \beta_{i,j}$ and $\mathsf{Ans}_i(r^{\mathsf{shift}}_{i,j}) = \mathsf{Fold}(f_{i-1}, k_{i-1}, r^{\mathsf{fold}}_{i-1})(r^{\mathsf{shift}}_{i,j})$. Finally, (virtually) set $g'_i := \mathsf{Quotient}(g_i, \mathcal{G}_i, \mathsf{Ans}_i, \mathsf{Fill}_i)$.
     (c) Define the virtual oracle $f_i : \mathcal{L}_i \to \mathbb{F}$ as follows:
     $$f_i := \mathsf{DegCor}(d_i, r^{\mathsf{comb}}_i, g'_i, d_i - |\mathcal{G}_i|).$$
     Observe that a query x to f_i translates to a single query either to g_i (if($x \notin \mathcal{G}_i$)) or to Fill_i (If ($x \in \mathcal{G}_i$)).
  2. **Consistency with final polynomial:**

(a) Sample random points $r_1^{\mathsf{fin}}, \dots, r_{t_M}^{\mathsf{fin}} \to \mathcal{L}_M^{k_M}$.

(b) Check that $\hat{p}(r_j^{\mathsf{fin}}) = \mathsf{Fold}(f_M, k_M, r_M^{\mathsf{fold}})(r_j^{\mathsf{fin}})$ for every $j \in [t_M]$.

3. **Consistency with Ans:** For every $i \in \{i, \dots, M\}$ and every $x_i \in \mathcal{G}_i \cap \mathcal{L}_i$ query $g_i(x)$ and check that $g_i(x) = \mathsf{Ans}_i(x)$.

3.3 Round-by-round soundness

Lemma 3.2. *Consider $(\mathbb{F}, M, d, k_0, \dots, k_M, \mathcal{L}_0, \dots, \mathcal{L}_M, t_0, \dots, t_M)$ and $d_0, \dots, d_M$ as in Construction 3.2, and for every $0 \le i \le M$ let $\rho_i := d_i/|\mathcal{L}_i|$. For every $f \notin \mathrm{RS}[\mathbb{F}, \mathcal{L}_0, d_0]$ and every $\delta_0, \dots, \delta_M$ where*

- $\delta_0 \in (0, \Delta(f, \mathrm{RS}[\mathbb{F}, \mathcal{L}_0, d_0])] \cap (0, 1 - \mathsf{B}^*(\rho_0))$
- *for every $0 < i \le M$: $\delta_i \in (0, \min\{1 - \rho_i - \frac{1}{|\mathcal{L}_i|}, 1 - \mathsf{B}^*(\rho_i)\})$, and*
- *for every $0 < i \le M$: $\mathrm{RS}[\mathbb{F}, \mathcal{L}_i, d_i]$ is (δ_i, l_i)-list decodable,*

STIR (Construction 3.2) has round-by-round soundness error $(\epsilon^{\mathsf{fold}}, \epsilon_1^{\mathsf{out}}, \epsilon_1^{\mathsf{shift}}, \dots, \epsilon_M^{\mathsf{out}}, \epsilon_M^{\mathsf{shift}}, \epsilon^{\mathsf{fin}})$ where:

- $\epsilon^{\mathsf{fold}} \le \mathsf{err}^*(d_0/k_0, \rho_0, \delta_0, k_0)$.
- $\epsilon_i^{\mathsf{out}} \le \frac{l_i^2}{2} \cdot \left(\frac{d_i}{|\mathbb{F}| - |\mathcal{L}_i|}\right)^s$
- $\epsilon_i^{\mathsf{shift}} \le (1 - \delta_{i-1})^{t_{i-1}} + \mathsf{err}^*(d_i, \rho_i, \delta_i, t_{i-1} + s) + \mathsf{err}^*(d_i/k_i, \rho_i, \delta_i, k_i)$.
- $\epsilon^{\mathsf{fin}} \le (1 - \delta_M)^{t_M}$.

Above, B^* and err^* are the proximity bound and error (respectively) described in Section 2.1.