Circle STARK

A (kind of) brief overview: bottom-up approach

Kurt Pan

ZKPunk.pro

November 10, 2024

Table of Contents

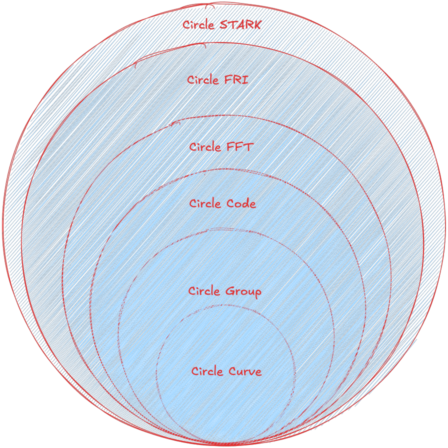
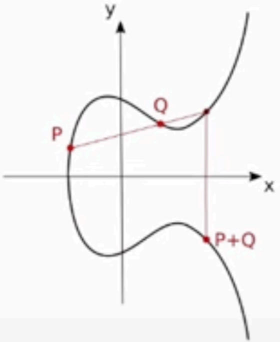
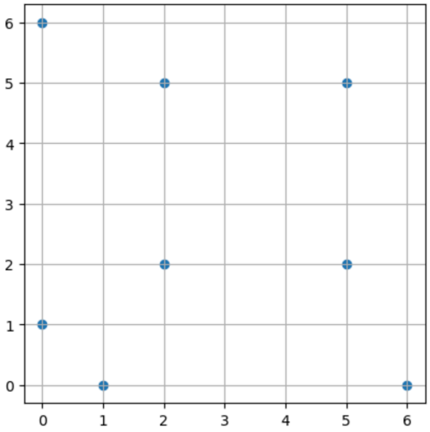


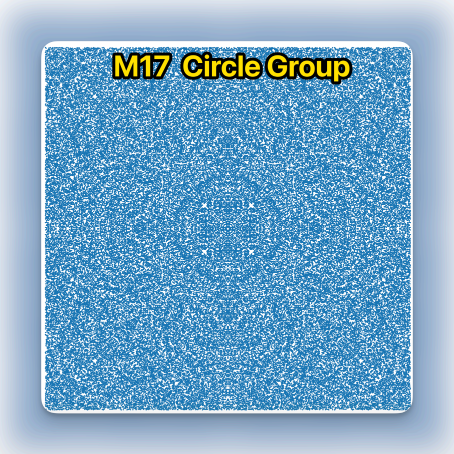
Table of Contents

regular STARK	EC STARK (BCKL21/22)
($p-1$ smooth)	(any p)
	
$x \neq 0$	$y^2 = x^3 + a \cdot x + b$
$p(x) \in \mathbb{F}_p[x]$	$\frac{p(x,y)}{v(x,y)} \in \mathbb{F}_p(x,y)$

M3/M17 Circle Group

Applies to M31 $p = 2^{31} - 1$





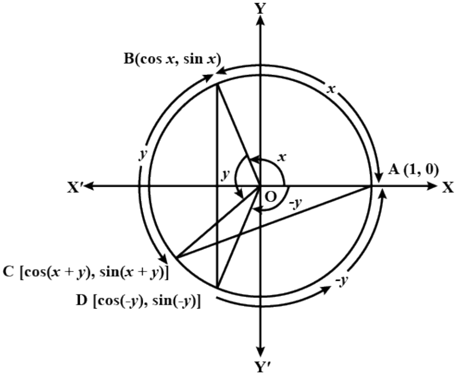
```c
#define MASK (0x7FFFFFFF) // 2^31 - 1 = 1111...1111

uint64_t reduce(uint64_t x) {
    return (x & MASK) + (x >> 31);
}
```

The circle curve

simpler construction for $p + 1$ smooth

$$C : x^2 + y^2 = 1$$



The $p+1$ points of $C(\mathbb{F}_p)$

Group Operation

$$(x_0, y_0) \cdot (x_1, y_1) := (x_0 \cdot x_1 - y_0 \cdot y_1, x_0 \cdot y_1 + y_0 \cdot x_1)$$

The group has $(1, 0)$ as its neutral element, and for any $P = (P_x, P_y)$ in $C(\mathbb{F}_p)$ we shall call

$$T_P(x, y) := P \cdot (x, y) = (P_x \cdot x - P_y \cdot y, P_x \cdot y + P_y \cdot x)$$

the *rotation*, or *translation* by P.
The *squaring map* with respect to the group operation is the quadratic map

$$\pi(x, y) := (x, y) \cdot (x, y) = \left(x^2 - y^2, 2 \cdot x \cdot y\right) = \left(2 \cdot x^2 - 1, 2 \cdot x \cdot y\right)$$

Group inverses are given by the degree-one map

$$J(x, y) := (x, -y)$$

Definition (twin-coset)

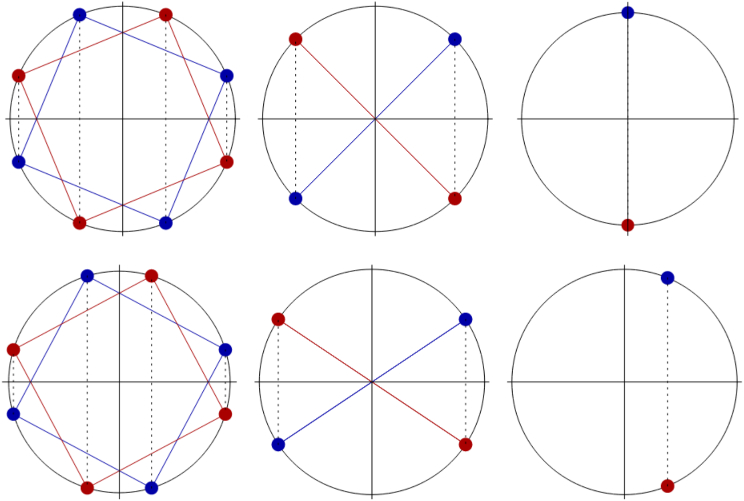
Let G_{n-1} be a (cyclic) subgroup of $C(\mathbb{F}_p)$ of size $|G_{n-1}| = 2^{n-1}, n \geq 1$.
Any *disjoint union* $D = Q \cdot G_{n-1} \cup Q^{-1} \cdot G_{n-1}$ with $Q \cdot G_{n-1} \cap Q^{-1} \cdot G_{n-1} = \varnothing$, is called a twin-coset of size $N = 2^n$.

Definition (standard position coset)

A twin-coset D is again a *coset of the subgroup* G_n.

Lemma

If D is a twin-coset of size $N = 2^n, n \geq 2$, then its image $\pi(D)$ under the squaring map π is a twin-coset of size $N/2$.



Lemma

There is an isomorphism between the circle curve $C(\mathbb{F}_p)$ and the projective line $P^1(\mathbb{F}_p)$.
For any finite extension F of $\mathbb{F}_p$, the number of points of the circle curve over F is $|F| + 1$.

Proof.

The isomorphism is the stereographic projection onto the y-axis with center $(-1, 0)$, defined by the equations

$$t = \frac{y}{x+1}, \quad (x, y) = \left(\frac{1-t^2}{1+t^2}, \frac{2 \cdot t}{1+t^2}\right)$$

Under this isomorphism $(-1, 0)$ is identified with infinity of the projective line, and the two curve points $\infty = (1 : +i : 0), \bar{\infty} = (1 : -i : 0)$ correspond to $t = \pm i$. $\square$

The space of polynomials and Circle Codes

Definition (The space of polynomials)

Let F be an extension field of $\mathbb{F}_p$. Over the circle curve, for any even integer $N \geq 0$ we define $\mathcal{L}_N(F)$ as the space of all bivariate polynomials with coefficients in F, and of total degree at most $N/2$,

$$\mathcal{L}_N(F) = \left\{ p(x, y) \in F[x, y] / \left(x^2 + y^2 - 1\right) : \deg p \leq \frac{N}{2} \right\}$$

Definition (Circle Code)

The circle code with values in F and evaluation domain D, is linear code with code words from the space

$$\mathcal{C}_N(F, D) = \left\{ f(P)|_{P \in D} : f \in \mathcal{L}_N(F) \right\}$$

Vanishing polynomials

Let D be a subset of $C(\mathbb{F}_p)$ of even size N, where $2 \leq N < p+1$.

Definition (vanishing polynomial)

Any non-zero polynomial from $\mathcal{L}_N = \mathcal{L}_N(\mathbb{F}_p)$, which evaluates to zero over D a *vanishing polynomial* of the set D.
$\mathcal{V}(D) = \{v \in \mathcal{L}_N : v|_D = 0\}$

Decomposing D into pairs of points $\{P_k, Q_k\}, 1 \leq k \leq N/2$, and taking the product of linear functions going through these pairs,

$$v_D(x,y) = \prod_{k=1}^{N/2} \left((x - P_{k,x}) \cdot (Q_{k,y} - P_{k,y}) - (y - P_{k,y}) \cdot (Q_{k,x} - P_{k,x})\right)$$

Vanishing polynomials of FFT domains

Given a twin-coset $D = Q \cdot G_{n-1} \cup Q^{-1} \cdot G_{n-1}$, where $n \geq 1$, its image under the power map π^{n-1} is a twin-coset of size two $\pi^{n-1}(D) = \{(x_D, \pm y_D)\}$.

Definition (vanishing polynomial of D)

$$v_D(x, y) := v_n(x, y) - x_D$$

with

$$v_n(x, y) := \pi_x \circ \pi^{n-1}(x, y)$$

where π_x is the projection onto the x-axis

Notice that whenever D is a standard position coset, its image $\pi^{n-1}(D)$ is again a standard position coset and thus $x_D = 0$. In this case the vanishing polynomial v_D is v_n itself.

$$
\begin{aligned}
v_1(x) &= x \\
v_2(x) &= 2 \cdot x^2 - 1 \\
v_3(x) &= 2 \cdot \left(2 \cdot x^2 - 1\right)^2 - 1 = 8 \cdot x^4 - 8 \cdot x^2 + 1 \\
v_4(x) &= 8 \cdot \left(2 \cdot x^2 - 1\right)^4 - 8 \cdot \left(2 \cdot x^2 - 1\right)^2 + 1 \\
&= 128 \cdot x^8 - 256 \cdot x^6 + 160 \cdot x^4 - 32 \cdot x^2 + 1
\end{aligned}
$$

Table of Contents

The sequence of domains

Chain of 2-to-1 maps

$$\begin{tikzcd}
D_n \arrow[r, "\pi_n"] \arrow[d, "t_n"] & D_{n-1} \arrow[r, "\pi_{n-1}"] \arrow[d, "t_{n-1}"] & \dots \arrow[r, "\pi_2"] & D_1 \arrow[r, "\pi_1"] \arrow[d, "t_1"] & D_0 \\
\mathbb{F} & \mathbb{F} & & \mathbb{F} &
\end{tikzcd}$$

$$\begin{tikzcd}
D_n \arrow[r, "\pi"] \arrow[d, "\phi_J"] & D_{n-1} \arrow[r, "\pi"] \arrow[d, "\phi_J"] & D_{n-2} \arrow[r, "\pi"] \arrow[d, "\phi_J"] & \cdots\cdots\cdots \arrow[r, "\pi"] & D_1 \arrow[d, "\phi_J"] \\
D_n/J \arrow[r, "\pi"] & D_{n-1}/J \arrow[r, "\pi"] & D_{n-2}/J \arrow[r, "\pi"] & \cdots\cdots\cdots \arrow[r, "\pi"] & D_1/J
\end{tikzcd}$$

Circle FFT

$$
\begin{tikzcd}
D_n \arrow[r, "\pi"] \arrow[d, "\pi_x"] & D_{n-1} \arrow[r, "\pi"] \arrow[d, "\pi_x"] & \cdots \arrow[r, "\pi"] & D_1 \arrow[d, "\pi_x"] \\
S_n \arrow[r, "\pi"] & S_{n-1} \arrow[r, "\pi"] & \cdots \arrow[r, "\pi"] & S_1
\end{tikzcd}
$$

In the first step,

$$f(x, y) = f_0(x) + y \cdot f_1(x)$$

$$f_0(x) = \frac{f(x, y) + f(x, -y)}{2}$$

$$f_1(x) = \frac{f(x, y) - f(x, -y)}{2 \cdot y}$$

$$
\begin{tikzcd}
D_n \arrow[r, "\pi"] \arrow[d, "\pi_x"] & D_{n-1} \arrow[r, "\pi"] \arrow[d, "\pi_x"] & \cdots\cdots \arrow[r, "\pi"] & D_1 \arrow[d, "\pi_x"] \\
S_n \arrow[r, "\pi"] & S_{n-1} \arrow[r, "\pi"] & \cdots\cdots \arrow[r, "\pi"] & S_1
\end{tikzcd}
$$

In the other steps, we receive a function $f_{k_0,\ldots,k_{n-j}} \in F^{S_j}$ from a previous step, where $2 \leq j \leq n$.

$$f(x) = f_0(\pi(x)) + x \cdot f_1(\pi(x))$$

$$f_0(\pi(x)) = \frac{f(x) + f(-x)}{2}$$
$$f_1(\pi(x)) = \frac{f(x) - f(-x)}{2 \cdot x}$$

$\pi(x) = 2 \cdot x^2 - 1.$

Theorem

Given $f \in F^D$ a function over D with values in an extension field F of $\mathbb{F}_p$, the above described algorithm outputs the coefficients $c_k \in F, 0 \leq k \leq 2^n - 1$, with respect to the FFT basis, so that $\sum_{k=0}^{2^n-1} c_k \cdot b_k$ evaluates to f over D.

FFT-basis

Definition (FFT-basis)

For any integer j from the interval $0 \le j \le 2^n - 1$, let $(j_0, \dots, j_{n-1}) \in \{0,1\}^n$ denote its bit representation, satisfying $j = j_0 + j_1 \cdot 2 + \dots + j_{n-1} \cdot 2^{n-1}$.

The *FFT-basis* of order n is the family $\mathcal{B}_n$ of polynomials

$$b_j^{(n)}(x,y) := y^{j_0} \cdot v_1(x)^{j_1} \cdot \ldots \cdot v_{n-1}^{j_{n-1}}(x), \quad 0 \le j \le 2^n - 1$$

where $v_k(x), 1 \le k \le n-1$, is the vanishing polynomial of the standard position coset of size 2^k

n=2: basis: $[1, Y, X, X * Y]$

n=3: basis:

$[1, Y, X, X * Y, 2 * X^2 - 1, 2 * X^2 * Y - Y, 2 * X^3 - X, 2 * X^3 * Y - X * Y]$

CFFT in Python

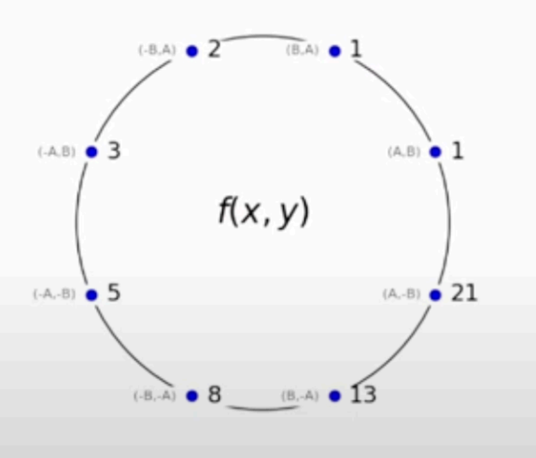
```python
def fft(vals, domain=None):
    if len(vals) == 1:
        return vals
    if domain is None:
        domain = get_initial_domain_of_size(vals[0].__class__, len(vals))
    half_domain = halve_domain(domain)
    if isinstance(domain[0], tuple):
        left = vals[:len(domain)//2]
        right = vals[len(domain)//2:][::-1]
        f0 = [(L+R)/2 for L,R in zip(left, right)]
        f1 = [(L-R)/(2*y) for L,R,(x,y) in zip(left, right, domain)]
    else:
        left = vals[:len(domain)//2]
        right = vals[len(domain)//2:][::-1]
        f0 = [(L+R)/2 for L,R in zip(left, right)]
        f1 = [(L-R)/(2*x) for L,R,x in zip(left, right, domain)]
    o = [0] * len(domain)
    o[::2] = fft(f0, half_domain)
    o[1::2] = fft(f1, half_domain)
    return o
```

ICFFT in Python

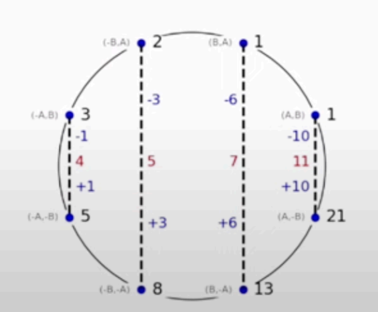
```python
def inv_fft(vals, domain=None):
    if len(vals) == 1:
        #print('o', vals)
        return vals
    if domain is None:
        domain = get_initial_domain_of_size(vals[0].__class__, len(vals))
    half_domain = halve_domain(domain)
    f0 = inv_fft(vals[::2], half_domain)
    f1 = inv_fft(vals[1::2], half_domain)
    if isinstance(domain[0], tuple):
        left = [L+y*R for L,R,(x,y) in zip(f0, f1, domain)]
        right = [L-y*R for L,R,(x,y) in zip(f0, f1, domain)]
    else:
        left = [L+x*R for L,R,x in zip(f0, f1, domain)]
        right = [L-x*R for L,R,x in zip(f0, f1, domain)]
    return left+right[::-1]
```

Example

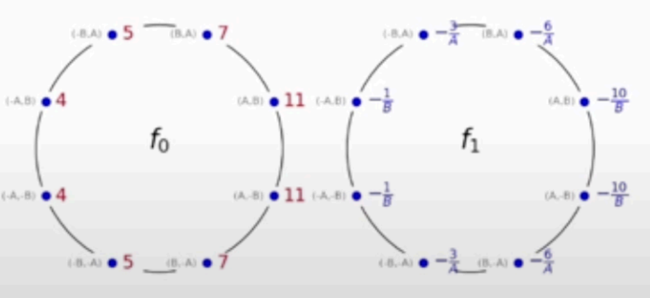
Let $G = (A, B)$ be a circle generator of order 16 . The circle domain of size 8. Interpolate a set of values by a circle polynomial from $\mathbb{F}_p \mid x, y]^{\leq 4}$



Example

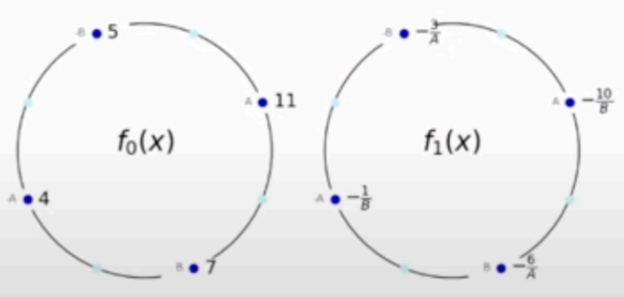


(-B,A) 2
(B,A) 1
-3
-6
(-A,B) 3
(A,B) 1
-1
-10
4
5
7
11
+1
+10
(-A,-B) 5
(A,-B) 21
+3
+6
(-B,-A) 8
(B,-A) 13



f_0
(-B,A) 5
(B,A) 7
(-A,B) 4
(A,B) 11
(-A,-B) 4
(A,-B) 11
(-B,-A) 5
(B,-A) 7
f_1
(-B,A) $-\frac{3}{A}$
(B,A) $-\frac{6}{A}$
(-A,B) $-\frac{1}{B}$
(A,B) $-\frac{10}{B}$
(-A,-B) $-\frac{1}{B}$
(A,-B) $-\frac{10}{B}$
(-B,-A) $-\frac{3}{A}$
(B,-A) $-\frac{6}{A}$

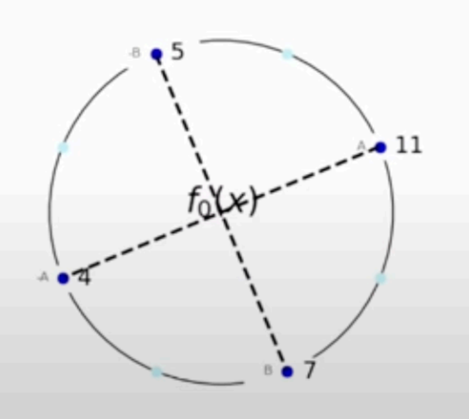
Example

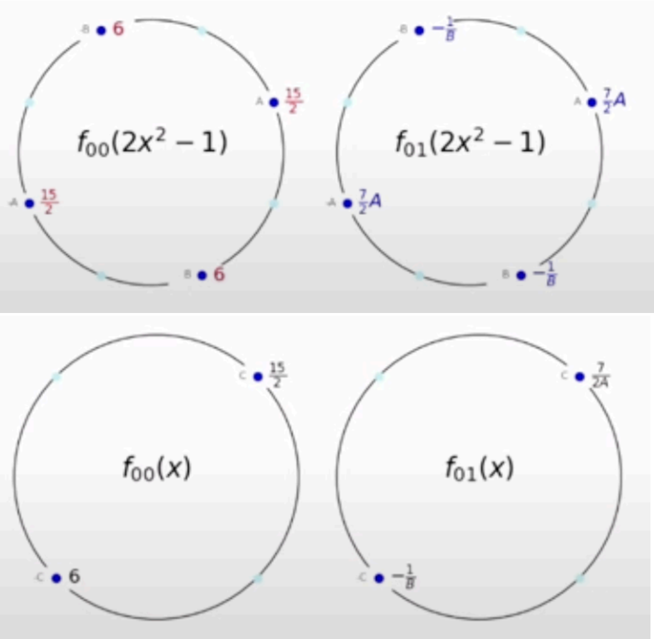


Second decomposition

$$f_0(x) = f_{00}\left(2x^2 - 1\right) + x \cdot f_{01}\left(2x^2 - 1\right)$$

$$f_{00}\left(2x^2 - 1\right) = \frac{f_0(x) + f_0(-x)}{2} \quad f_{01}\left(2x^2 - 1\right) = \frac{f_0(x) - f_0(-x)}{2x}$$





$f_{00}(2x^2 - 1)$
6
$\frac{15}{2}$
$f_{01}(2x^2 - 1)$
$-\frac{1}{B}$
$\frac{7}{2}A$
$f_{00}(x)$
$\frac{15}{2}$
6
$f_{01}(x)$
$\frac{7}{2A}$
$-\frac{1}{B}$

Third decomposition

$$f_{000}\left(2x^2-1\right)=\frac{f_{00}(x)+f_{00}(-x)}{2} \quad f_{001}\left(2x^2-1\right)=\frac{f_{00}(x)-f_{00}(-x)}{2x}$$

$$f_{000}=\frac{27}{4} \quad f_{001}=\frac{3}{4C}$$

Full decomposition

$f(x,y) =$

// First decomposition

$f_0(x) + y f_1(x) =$

// Second decomposition

$\left[f_{00}(2x^2-1) + x f_{01}(2x^2-1)\right] + y\left[f_{10}(2x^2-1) + x f_{11}(2x^2-1)\right]$

// Third decomposition

$\left[(f_{000} + (2x^2-1)f_{001}) + x(f_{010} + (2x^2-1)f_{011})\right] +$

$y\left[(f_{100} + (2x^2-1)f_{101}) + x(f_{110} + (2x^2-1)f_{111})\right] =$

// Expand the terms

$f_{000} + f_{001}(2x^2-1) + f_{010}x + f_{011}(2x^2-1)x + f_{100}y +$

$f_{101}y(2x^2-1) + f_{110}yx + f_{111}y(2x^2-1)x =$

// Substitute our values

$\frac{27}{4} + \frac{3}{4C}(2x^2-1) - \frac{11A+9C}{4AB}x + \frac{-11A+9C}{4ABC}x(x^2-1) + \cdots$

Table of Contents

The circle FRI

1. Commit phase:
   - (Decomposition.) The prover compute the decomposition of $f \in \mathcal{L}_N(F)$ into $f = g + \lambda \cdot v_n$ with $g \in \mathcal{L}'_N(F)$ and $\lambda \in F$. It sends λ to the verifier.
   - (Folding.) For each $j = 1, \ldots, r$, the prover holds a function $g_{j-1} \in F^{S_{j-1}}$ from the previous round. (In the first round $g_0 = g$.) It receives a random challenge $\lambda_j \leftarrow \$F$ from the verifier, and uses it to build the random linear combination

$$g_j = g_{j-1,0} + \lambda_j \cdot g_{j-1,1}$$

$$g_{j-1,0} \circ \pi_j = \frac{g_{j-1} + (g_{j-1} \circ T_j)}{2}$$

$$g_{j-1,1} \circ \pi_j = \frac{g_{j-1} - (g_{j-1} \circ T_j)}{2 \cdot t_j}$$

     The prover sets up the oracle for g_j, and sends it to the verifier. (In the last round, the prover sends $g_{r+1} \in \mathcal{L}^{(r)}$, in plain.)

2. Query phase:
    - The verifier samples $s \geq 1$ queries uniformly from D. For each query Q, we write $Q_0 = Q$ and consider its trace $Q_j \in S_j, j = 1 \ldots, n$, under the chain of projections $\pi_j : S_{j-1} \longrightarrow S_j$.
    - The verifier asks the oracle for the values of f at Q_0 and $T_1(Q_0)$, and of g_j at Q_j and $T_j(Q_j)$, for $j = 1, \ldots, r$.
    - It takes the answers to check whether each g_j was properly formed from g_{j-1} via the folding , using $g_0 = f - \lambda \cdot v_n$ and the equations above for $j = 1, \ldots, r$.

If the oracle answers satisfy these checks for each of the s queries, then the verifier accepts. (Otherwise, it rejects.)

Batch Circle FRI

The IOP for a batch of functions $f_1, \ldots, f_L \in F^D$ having correlated $(1-\theta)$-agreement to a codeword from $\mathcal{C}_N(F, D)$, is as follows.
In the first step, given a random challenge $\lambda_0 \leftarrow \$F$ from the verifier, the prover computes the values of the linear combination

$$f = \sum_{k=1}^{L} \lambda_0^{k-1} \cdot f_i$$

over D.
Now, both prover and verifier run Protocol Circle FRI on f, with its query phase extended by a check of the new equation at each of the s queries Q.

Table of Contents

Trace domain H (coset of) a subgroup of C

Definition

The *trace domain* is the standard position coset

$$H \subset C(\mathbb{F}_p)$$

of a cyclic and proper subgroup $G = G_n$ of the circle curve $C(\mathbb{F}_p)$, of size $N = 2^n$, with $n \geq 1$,

The trace is organised column-wise $t_1, \ldots, t_w \in \mathbb{F}_p^N$, each placed over the domain H in the usual manner, using the group translation T by a generator of G for the timeline.
The trace columns (witness data) are interpolated by polynomials

$$p_1, \ldots, p_w \in \mathbb{F}_p[x, y] / \left(x^2 + y^2 - 1\right)$$

of total degree at most $N/2$, meaning that $p_i \in \mathcal{L}_N(\mathbb{F}_p)$ (actually, they are from the FFT-space $\mathcal{L}'_N(\mathbb{F}_p)$), These polynomials are subject to a set of constraints, say

$$P_i\left(s_i, p_1, \ldots, p_w, p_1 \circ T, \ldots, p_w \circ T\right)=0$$

for $i=1, \ldots, C$, holding over the entire domain H, and where $s_i \in \mathcal{L}_N\left(\mathbb{F}_p\right)$ is a predefined *selector polynomial*.
Each constraint is a polynomial

$$P_i \in \mathbb{F}_p\left[S, X_1, \ldots, X_w, Y_1, \ldots, Y_w\right]$$

The polynomials $p_1, \ldots, p_w$, as well as further ones provided in the course of the protocol, are committed by their values over a larger *evaluation domain* $D \subseteq C(\mathbb{F}_p)$, a standard position coset of at least double the size of H.
In other words, the prover commits to code words of the circle code $\mathcal{C}_N(D)$ with values in the prime field $\mathbb{F}_p$, or some finite extension F of it. Being again in standard position, D is disjoint from H, and we assume that

$$\frac{|D|}{|H|} = 2^B$$

for some $B \geq 1$.

Circle IOP for AIR

0. The prover sets up the domain evaluation oracles $[p_1], \ldots, [p_w]$ for the values of $p_1(X, Y), \ldots, p_w(X, Y)$ over D, and sends them to the verifier.
1. Upon receiving a randomness $\beta \leftarrow \$F$ from the verifier, the prover computes the domain quotient $q_\beta(X, Y) \in F[X, Y]$ of degree $\leq (d-1) \cdot |H|/2$ satisfying the identity

$$\sum_{i=1}^{C} \beta^{i-1} \cdot P_i\left(s_i, p_1, \ldots, p_w, (p_1 \circ T), \ldots, (p_w \circ T)\right) = q_\beta \cdot v_H$$

and decomposes it into segment polynomials $q_{\beta,j}(X, Y) \in F[X, Y], j = 1, \ldots, d-1$, each of degree $\leq |H|/2$, and the dimension-gap scalar $\lambda \in F$, with respect to a union of twin-cosets $\bar{H} = \bigcup_{j=1}^{d-1} H_j$ having overall size $(d-1) \cdot |H|$. It sends the oracles

$$[q_{\beta,1}], \ldots, [q_{\beta,d-1}]$$

for their values over D, and λ to the verifier. The overall identity to be proven is therefore

$$\sum_{i=1}^{C} \beta^{i-1} \cdot P_i \left(s_i, p_1, \ldots, p_w, (p_1 \circ T), \ldots, (p_w \circ T)\right)$$
$$= v_H \cdot \left(\lambda \cdot v_{\bar{H}} + \sum_{j=1}^{d-1} \frac{v_{\bar{H}}}{v_{H_j}} \cdot q_{\beta,j}\right)$$

2. The verifier samples a random DEEP query, i.e. a random point $\gamma \longleftarrow C(F) \backslash (D \cup H)$ drawn uniformly from the circle curve over the extension field F, and sends it to the prover.
   The prover answers with the evaluation claims $(\gamma, v_{i,1}), (T(\gamma), v_{i,2}), i = 1, \ldots, w$, for the witness polynomials $p_i(X, Y)$, and (γ, v_j), $j = 1, \ldots, d-1$, for the segment polynomials $q_{\beta,j}(X, Y)$

3. Both prover and verifier engage in the batch circle FRI for the real and imaginary parts of the DEEP quotients,

$$\mathrm{Re}\,/\,\mathrm{Im}\left(\frac{p_i - v_{i,0}}{v_\gamma}\right), \mathrm{Re}\,/\,\mathrm{Im}\left(\frac{p_i - v_{i,1}}{v_{T(\gamma)}}\right)$$

for each $i = 1, \ldots, w$, and

$$\mathrm{Re}\,/\,\mathrm{Im}\left(\frac{q_{\beta,1} - v_1}{v_\gamma}\right), \ldots, \mathrm{Re}\,/\,\mathrm{Im}\left(\frac{q_{\beta,d-1} - v_{d-1}}{v_\gamma}\right)$$

which is a joint proximity test to the circle code $\mathcal{C}_N(F, D)$, where $N = |H|$.

If circle FRI passes, and if the evaluation claims satisfy the overall identity at $(X, Y) = \gamma$, the verifier accepts. (Otherwise, it rejects.)

Thanks! Kob-khun kub!