Flowering graphs

Interactive proximity test to codes on graphs by flowering

Hugo Delavenne, Tanguy Medevielle, Élina Roussel

LIX, École Polytechnique, Institut Polytechnique de Paris
Inria

Thursday 10th April 2025
@ Paris 8, Saint-Denis

Roadmap

1 Interactive Oracle Proofs of Proximity

2 Flowering protocol

3 Flowering graphs

- SNARKs
- Interactive Oracle Proof of Proximity
- Fast Reed-Solomon IOPP

Why verifiable computing: a sad story

- $\mathcal{P}$ has executed a complex algorithm $A : x \mapsto y$

- $\mathcal{P}$ has executed a complex algorithm $A : x \mapsto y$
- very proud, it wants to share y to $\mathcal{V}$

- $\mathcal{P}$ has executed a complex algorithm $A : x \mapsto y$
- very proud, it wants to share y to $\mathcal{V}$
- $\mathcal{V}$ only trusts what it sees

- $\mathcal{P}$ has executed a complex algorithm $A : x \mapsto y$
- very proud, it wants to share y to $\mathcal{V}$

- $\mathcal{V}$ only trusts what it sees
- it can't accept $y = A(x)$ and doesn't want to compute it itself

- $\mathcal{P}$ has executed a complex algorithm $A : x \mapsto y$
- very proud, it wants to share y to $\mathcal{V}$

- $\mathcal{V}$ only trusts what it sees
- it can't accept $y = A(x)$ and doesn't want to compute it itself

- $\mathcal{P}$ is very sad

Why verifiable computing: a sad story

- $\mathcal{P}$ has executed a complex algorithm $A : x \mapsto y$
- very proud, it wants to share y to $\mathcal{V}$
- $\mathcal{V}$ only trusts what it sees
- it can't accept $y = A(x)$ and doesn't want to compute it itself
- $\mathcal{P}$ is very sad
- #emotion

SNARK means **S**uccinct **N**on-interactive **AR**gument of **K**nowledge.

It turns $\boldsymbol{A : x \mapsto y}$ (that runs in $\tau(|x|)$) into

$$\underbrace{A'}\ : \underbrace{x}_{\text{input}}, \underbrace{\alpha}_{\text{randomness}} \mapsto \underbrace{y}_{\text{output}}, \underbrace{\pi}_{\text{SNARK proof that "}A(x) = y\text{"}}$$

SNARK means **S**uccinct **N**on-interactive **AR**gument of **K**nowledge.

It turns $A : x \mapsto y$ (that runs in $\tau(|x|)$) into

$$A' : x, \alpha \mapsto y, \pi$$

input, randomness, output, SNARK proof that "$A(x) = y$"

satisfying:

- $|\pi| \ll \tau$
- A' runs in $\tilde{O}(\tau)$
- there is a verifier $\mathcal{V}$ in $\mathrm{poly}(|\pi|)$ such that
  - **Completeness:** if $A(x) = y$ then $\underset{\alpha}{\mathbb{P}}(\mathcal{V}(\pi)$ accepts$) = 1$
  - **Soundness:** if $A(x) \neq y$ then $\underset{\alpha}{\mathbb{P}}(\mathcal{V}(\pi)$ accepts$) \leqslant s$.

- Proof of **Transaction** (Blockchains)
- Proof of **Authenticity** (Signature preservation)
- Proof of **Emulation** (Speedrun)
- Proof of **Training** (AI regulation)

Definition **Relative Hamming distance**

Let $u, v \in \mathbb{F}^N$.

$$\Delta(u, v) := \frac{1}{N} |\{i \in [N] \mid u_i \neq v_i\}|$$

A linear **error correcting code** is a linear subspace of $\mathbb{F}^N$.

Definition **Reed-Solomon codes**

Let $\mathcal{L}_N \subseteq \mathbb{F}$, $|\mathcal{L}_N| = N$ and $K < N$.

$$\mathsf{RS}[\mathcal{L}_N, K] := \{f : \mathcal{L}_N \to \mathbb{F} \mid f \in \mathbb{F}[X]_{\leqslant K-1}\}$$

$\mathsf{RS}[\mathcal{L}_N, K]$ has length N, dimension K and minimal distance $1 - \frac{K+1}{N}$.

Reduction from **checking computation** to **testing proximity** to $\mathsf{RS}[\mathcal{L}_N, K]$

$$\begin{aligned} y = A(x) \quad &\Longrightarrow \quad \mathsf{Arithmetization}(A, x, y) \in \mathsf{RS}[\mathcal{L}_N, K] \\ y \neq A(x) \quad &\Longrightarrow \quad \Delta(\mathsf{Arithmetization}(A, x, y), \mathsf{RS}[\mathcal{L}_N, K]) > \delta \end{aligned}$$

Idea:
Computation = arithmetic circuit = composed polynomials $\to$ Reed-Solomon code

Definition **Locally-testable code**

A code C is **(ℓ, δ, s)-locally-testable** if there is $\mathcal{V}$ **only ℓ accesses** to u such that

- **Completeness:** if $u \in C$ then $\mathbb{P}(\mathcal{V}^u \text{ accepts}) = 1$
- **Soundness:** if $\Delta(u, C) > \delta$ then $\mathbb{P}(\mathcal{V}^u \text{ accepts}) \leqslant s$.

C has **locality ℓ** if C is (ℓ, δ, s)-l.-t. for $s < 1$.

Example **Reed-Solomon codes are not locally-testable**

RS$[\mathcal{L}_N, K]$ doesn't have locality $\ell < K + 1$.

Proof.
Any K values correspond to a degree $\leqslant K - 1$ polynomial by interpolation.

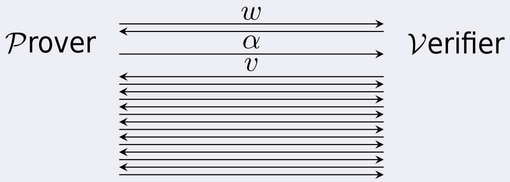
Definition **Non-interactive proof**

$$\mathcal{P}\text{rover} \xrightarrow{\quad w \quad} \mathcal{V}\text{erifier}$$

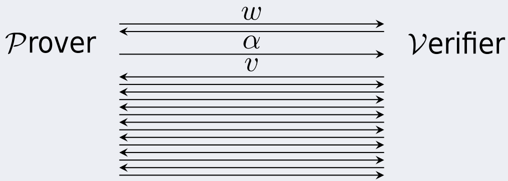
Definition **Sigma protocol**

$$\mathcal{P}\text{rover} \quad \begin{array}{c} \xrightarrow{\quad\quad w \quad\quad} \\ \xleftarrow{\quad\quad\quad\quad} \\ \alpha \\ \xrightarrow[\quad\quad v \quad\quad]{} \end{array} \quad \mathcal{V}\text{erifier}$$

Definition **Interactive Proof**



Definition **Interactive Proof**



Euh ??? **We are doing SNARKs right?**

This is **not** Succinct Non-interactive.

Definition **Cryptographic hash function**

A cryptographic **hash function** $H : \{0,1\}^* \to \{0,1\}^{|H|}$

- looks **injective**: $\mathcal{P}$ cannot find collisions
- looks **random**: $\mathcal{P}$ cannot find input to get desired output

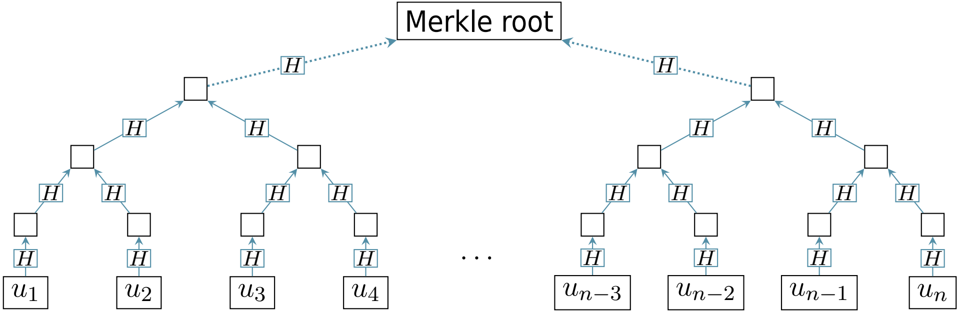
Fiat-Shamir replaces $\mathcal{V}$'s randomness by hash of previous messages:

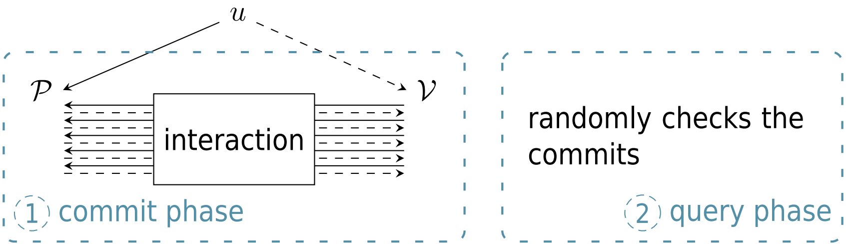
$$\mathcal{P} \begin{array}{c} \xrightarrow{\quad w \quad} \\ \xleftarrow{\quad \alpha \quad} \\ \xrightarrow[\quad v \quad]{} \end{array} \mathcal{V} \qquad \text{becomes} \qquad \alpha := H(w) \circlearrowleft \mathcal{P} \begin{array}{c} \xrightarrow{\quad w \quad} \\ \xrightarrow[\quad v \quad]{} \end{array} \mathcal{V}$$

Definition **Oracle access**

$\mathcal{P}$ provides to $\mathcal{V}$ an **oracle access** to $u \in \mathbb{F}^n$ by giving **black-box** access to u.

In practice, $\mathcal{P}$ provides the root of a **Merkle tree**.





- **Completeness:** if $u \in C$ then $\mathbb{P}(\mathcal{V}^{u,\leftrightarrow\mathcal{P}}$ accepts$) = 1$
- **Soundness:** if $\Delta(u, C) > \delta$ then for any $\mathcal{P}$, $\mathbb{P}(\mathcal{V}^{u,\leftrightarrow\mathcal{P}}$ accepts$) \leqslant s$

[BCS16] Eli Ben-Sasson, Alessandro Chiesa, and Nicholas Spooner. Interactive Oracle Proofs.
In *Theory of Cryptography: 14th International Conference, TCC 2016-B, Beijing, China, October 31-November 3, 2016, Proceedings, Part II 14*, pages 31–60. Springer, 2016

Idea: Test even and odd parts $f(X) =: f_{\text{even}}(X^2) + X f_{\text{odd}}(X^2)$.

Definition **Folding**

Let $f : \mathcal{L}_N \to \mathbb{F}$ and $\alpha \in \mathbb{F}$.

$$\text{Fold}[f, \alpha](X^2) := f_{\text{even}}(X^2) + \alpha f_{\text{odd}}(X^2) = \frac{f(X) + f(-X)}{2} + \alpha \frac{f(X) - f(-X)}{2X}$$

[BBHR18] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Fast Reed-Solomon Interactive Oracle Proofs of Proximity. **In *45th international colloquium on automata, languages, and programming (ICALP 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018**

Idea: Test even and odd parts $f(X) =: f_{\text{even}}(X^2) + X f_{\text{odd}}(X^2)$.

Definition **Folding**

Let $f : \mathcal{L}_N \to \mathbb{F}$ and $\alpha \in \mathbb{F}$.

$$\mathsf{Fold}[f,\alpha](X^2) := f_{\text{even}}(X^2) + \alpha f_{\text{odd}}(X^2) = \frac{f(X)+f(-X)}{2} + \alpha\frac{f(X)-f(-X)}{2X}$$

- **Field restriction:** $\mathcal{L}_{N/2} := \{x^2 \mid x, -x \in \mathcal{L}_N\}$ so $\mathbb{F}$ must have 2^N roots of unity

[BBHR18] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Fast Reed-Solomon Interactive Oracle Proofs of Proximity. **In *45th international colloquium on automata, languages, and programming (ICALP 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018**

Idea: Test even and odd parts $f(X) =: f_{\text{even}}(X^2) + X f_{\text{odd}}(X^2)$.

Definition **Folding**

Let $f : \mathcal{L}_N \to \mathbb{F}$ and $\alpha \in \mathbb{F}$.

$$\text{Fold}[f,\alpha](X^2) := f_{\text{even}}(X^2) + \alpha f_{\text{odd}}(X^2) = \frac{f(X)+f(-X)}{2} + \alpha\frac{f(X)-f(-X)}{2X}$$

- **Field restriction:** $\mathcal{L}_{N/2} := \{x^2 \mid x, -x \in \mathcal{L}_N\}$ so $\mathbb{F}$ must have 2^N roots of unity
- **Validity preservation:** $f \in \text{RS}[\mathcal{L}_N, K] \iff \mathbb{P}_{\alpha}(\text{Fold}[f,\alpha] \in \text{RS}[\mathcal{L}_{N/2}, K/2]) > \frac{1}{|\mathbb{F}|}$
- **Local check:** $\mathcal{V}$ computes $\text{Fold}[f,\alpha](x^2)$ with 2 queries to f

[BBHR18] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Fast Reed-Solomon Interactive Oracle Proofs of Proximity. **In *45th international colloquium on automata, languages, and programming (ICALP 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018**

FRI protocol

$$f_0 : \mathcal{L}_N \to \mathbb{F}$$

$\mathcal{P}$ $\mathcal{V}$

$$\alpha_0 \xleftarrow{\$} \mathbb{F}$$

$$f_1 : \mathcal{L}_{N/2} \to \mathbb{F}$$

$$f_1 = \mathsf{Fold}[f_0, \alpha_0]$$

1 commit phase

2 query phase

[BBHR18] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Fast Reed-Solomon Interactive Oracle Proofs of Proximity. **In *45th international colloquium on automata, languages, and programming (ICALP 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018**

$$f_0 : \mathcal{L}_N \to \mathbb{F}$$

$\mathcal{P}$ $\mathcal{V}$

$$\alpha_0 \xleftarrow{\$} \mathbb{F}$$

$$f_1 : \mathcal{L}_{N/2} \to \mathbb{F}$$

$$f_1 = \mathsf{Fold}[f_0, \alpha_0]$$

1 commit phase

$$x_0 \xleftarrow{\$} \mathcal{L}_N,\ x_1 := x_0^2$$

$$f_1(x_1) \stackrel{?}{=} \mathsf{Fold}[f_0, \alpha_0](x_1)$$

2 query phase

[BBHR18] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Fast Reed-Solomon Interactive Oracle Proofs of Proximity. **In *45th international colloquium on automata, languages, and programming (ICALP 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018**

FRI protocol

$$f_0 : \mathcal{L}_N \to \mathbb{F}$$

$\mathcal{P}$ $\mathcal{V}$

$$\alpha_0 \xleftarrow{\$} \mathbb{F}$$

$$f_1 : \mathcal{L}_{N/2} \to \mathbb{F}$$

$$f_1 = \mathsf{Fold}[f_0, \alpha_0]$$

$$\vdots$$

$$\alpha_{r-1} \xleftarrow{\$} \mathbb{F}$$

$$f_r : \mathcal{L}_{N/2^r} \to \mathbb{F}$$

$$f_r = \mathsf{Fold}[f_{r-1}, \alpha_{r-1}]$$

1 commit phase

$$x_0 \xleftarrow{\$} \mathcal{L}_N,\ x_1 := x_0^2$$

$$f_1(x_1) \stackrel{?}{=} \mathsf{Fold}[f_0, \alpha_0](x_1)$$

2 query phase

[BBHR18] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Fast Reed-Solomon Interactive Oracle Proofs of Proximity. **In *45th international colloquium on automata, languages, and programming (ICALP 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018**

$$f_0 : \mathcal{L}_N \to \mathbb{F}$$

$\mathcal{P}$ $\mathcal{V}$

$$\alpha_0 \xleftarrow{\$} \mathbb{F}$$

$$f_1 : \mathcal{L}_{N/2} \to \mathbb{F}$$

$$f_1 = \mathsf{Fold}[f_0, \alpha_0]$$

$$\vdots$$

$$\alpha_{r-1} \xleftarrow{\$} \mathbb{F}$$

$$f_r : \mathcal{L}_{N/2^r} \to \mathbb{F}$$

$$f_r = \mathsf{Fold}[f_{r-1}, \alpha_{r-1}]$$

1 commit phase

$$x_0 \xleftarrow{\$} \mathcal{L}_N,\ x_1 := x_0^2$$

$$f_1(x_1) \stackrel{?}{=} \mathsf{Fold}[f_0, \alpha_0](x_1)$$

$$\vdots$$

$$x_r := x_{r-1}^2$$

$$f_r(x_r) \stackrel{?}{=} \mathsf{Fold}[f_{r-1}, \alpha_{r-1}](x_r)$$

2 query phase

[BBHR18] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Fast Reed-Solomon Interactive Oracle Proofs of Proximity. **In *45th international colloquium on automata, languages, and programming (ICALP 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018**

$$f_0 : \mathcal{L}_N \to \mathbb{F}$$

$\mathcal{P}$ $\mathcal{V}$

$$\alpha_0 \xleftarrow{\$} \mathbb{F}$$

$$f_1 : \mathcal{L}_{N/2} \to \mathbb{F}$$

$$f_1 = \mathsf{Fold}[f_0, \alpha_0]$$

$$\vdots$$

$$\alpha_{r-1} \xleftarrow{\$} \mathbb{F}$$

$$f_r : \mathcal{L}_{N/2^r} \to \mathbb{F}$$

$$f_r = \mathsf{Fold}[f_{r-1}, \alpha_{r-1}]$$

1 commit phase

$$x_0 \xleftarrow{\$} \mathcal{L}_N,\ x_1 := x_0^2$$

$$f_1(x_1) \stackrel{?}{=} \mathsf{Fold}[f_0, \alpha_0](x_1)$$

$$\vdots$$

$$x_r := x_{r-1}^2$$

$$f_r(x_r) \stackrel{?}{=} \mathsf{Fold}[f_{r-1}, \alpha_{r-1}](x_r)$$

repeat m times

2 query phase

[BBHR18] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Fast Reed-Solomon Interactive Oracle Proofs of Proximity. **In *45th international colloquium on automata, languages, and programming (ICALP 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018**

$$f_0 : \mathcal{L}_N \to \mathbb{F}$$

$\mathcal{P}$ $\mathcal{V}$

$$\alpha_0 \xleftarrow{\$} \mathbb{F}$$

$$f_1 : \mathcal{L}_{N/2} \to \mathbb{F}$$

$$f_1 = \text{Fold}[f_0, \alpha_0]$$

$$\vdots$$

$$\alpha_{r-1} \xleftarrow{\$} \mathbb{F}$$

$$f_r : \mathcal{L}_{N/2^r} \to \mathbb{F}$$

$$f_r = \text{Fold}[f_{r-1}, \alpha_{r-1}]$$

1 commit phase

$$x_0 \xleftarrow{\$} \mathcal{L}_N, \; x_1 := x_0^2$$

$$f_1(x_1) \stackrel{?}{=} \text{Fold}[f_0, \alpha_0](x_1)$$

$$\vdots$$

$$x_r := x_{r-1}^2$$

$$f_r(x_r) \stackrel{?}{=} \text{Fold}[f_{r-1}, \alpha_{r-1}](x_r)$$

$$f_r \stackrel{?}{\in} \text{RS}[\mathcal{L}_{N/2^r}, K/2^r]$$

repeat m times

2 query phase

[BBHR18] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Fast Reed-Solomon Interactive Oracle Proofs of Proximity. **In *45th international colloquium on automata, languages, and programming (ICALP 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018**

Proposition **FRI complexities [BBHR18]**

FRI protocol for RS$[\mathcal{L}_N, K]$ with m repetitions has following complexity:

- Prover complexity: $< 8N$ (after encoding)
- Verifier complexity: $< 2m \log K$
- Number of queries: $2m \log K$
- Number of rounds: $\log K$

[BBHR18] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Fast Reed-Solomon Interactive Oracle Proofs of Proximity. **In *45th international colloquium on automata, languages, and programming (ICALP 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018**

Proposition **FRI completeness**

If $f_0 \in \mathsf{RS}[\mathcal{L}_N, K]$ then $\mathcal{V}$ accepts with probability 1.

Theorem **FRI soundness [BCI+23]**

If $\Delta(f_0, \mathsf{RS}[\mathcal{L}_N, K]) > \delta$ then for any $\tilde{\mathcal{P}}$ and $\varepsilon > 0$, $\mathcal{V}$ accepts with probability

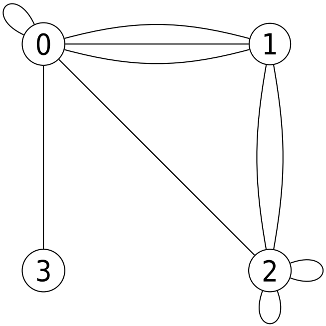
$$\leqslant \frac{K^2 \log K}{(2\varepsilon)^7 |\mathbb{F}|} + \left(1 - \min\left(\delta, 1 - \sqrt{K/N} - \varepsilon\right)\right)^m.$$

[BCI+23] Eli Ben-Sasson, Dan Carmon, Yuval Ishai, Swastik Kopparty, and Shubhangi Saraf. Proximity Gaps for Reed–Solomon Codes. *J. ACM*, 70(5), October 2023

- Graphs
- Folding graphs
- Flowering

$\Gamma = (V, E)$ is a n-RIM:

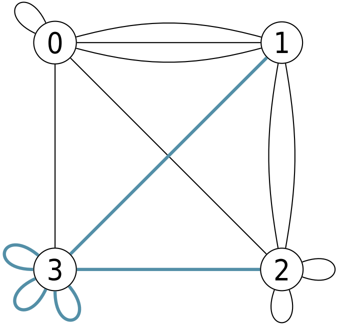
- **Multigraph:** multiple edges and loops



Regular Indexed Multigraphs (RIM)

$\Gamma = (V, E)$ is a n-RIM:

- ▷ **Multigraph:** multiple edges and loops
- ▶ **Regular:** same number n of edges

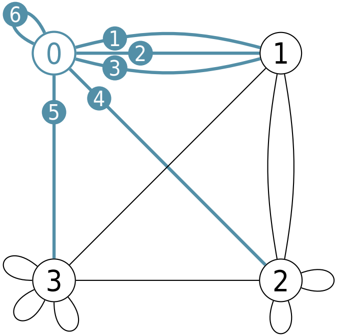


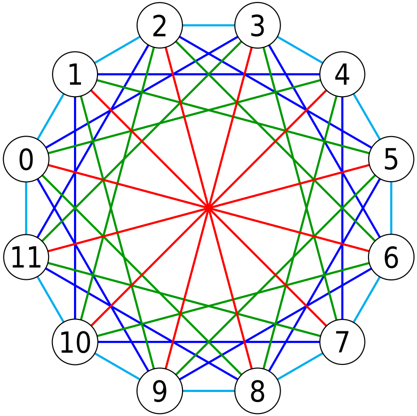
$\Gamma = (V, E)$ is a n-RIM:

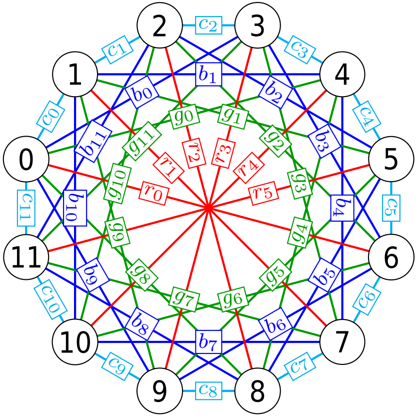
▷ **Multigraph:** multiple edges and loops

▷ **Regular:** same number n of edges

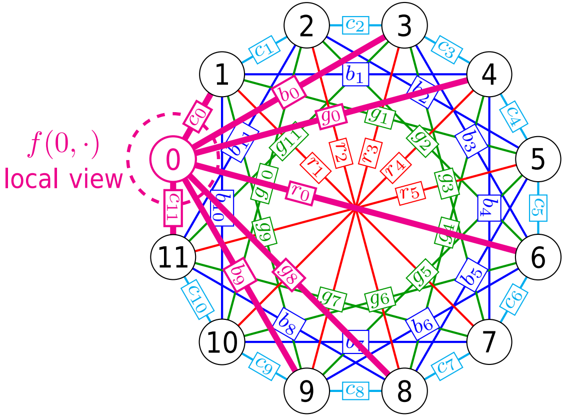
▶ **Indexed:** edge is $(v, \ell) \in V \times [n]$
Write $\boldsymbol{E(v, \ell)} \in V$ the neighbor of v by ℓ




2
3
1
4
0
5
11
6
10
7
9
8



Word $f : V \times [n] \to \mathbb{F}$ on a graph Γ



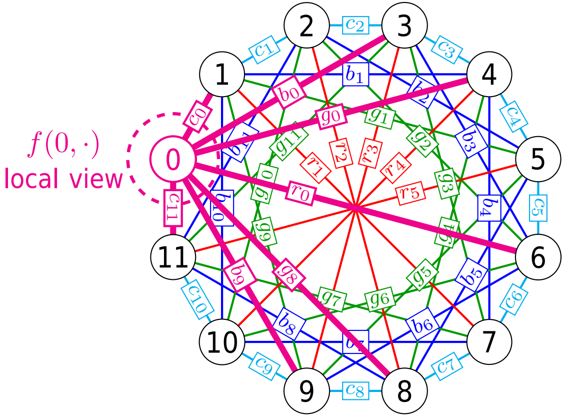
Word $f : V \times [n] \to \mathbb{F}$ on a graph Γ

Definition **Code $\mathcal{C}[\Gamma, C_0]$**

Given Γ a n-RIM and $C_0 \subseteq \mathbb{F}^n$,

$$f \in \mathcal{C}[\Gamma, C_0] \iff \forall v,\ f(v,\cdot) \in C_0.$$

We'll only use $C_0 = \mathrm{RS}[n,k]$.



Word $f : V \times [n] \to \mathbb{F}$ on a graph Γ

Definition **Code $\mathcal{C}[\Gamma, C_0]$**

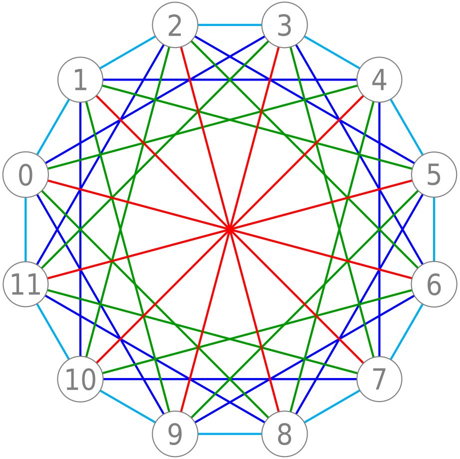
Given Γ a n-RIM and $C_0 \subseteq \mathbb{F}^n$,

$$f \in \mathcal{C}[\Gamma, C_0] \iff \forall v,\ f(v,\cdot) \in C_0.$$

We'll only use $C_0 = \mathsf{RS}[n,k]$.

For 0, we must have
$(c_0, b_0, g_0, r_0, g_8, b_9, c_{11}) \in \mathsf{RS}[7,k]$.

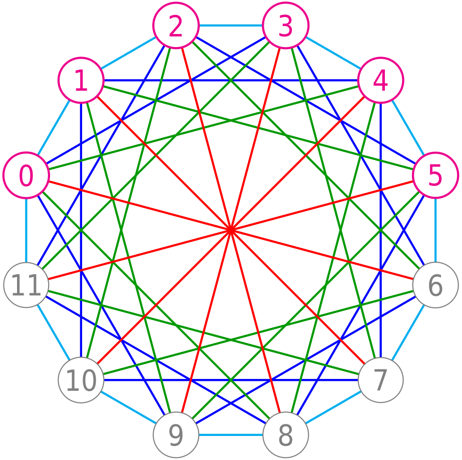
Cut-graph $\Gamma' = \mathsf{Cut}[\Gamma, V']$:



[DMR25] Hugo Delavenne, Tanguy Medevielle, and Élina Roussel. Interactive Oracle Proofs of Proximity to Codes on Graphs, 2025.
Submitted

Cut-graph $\Gamma' = \mathsf{Cut}[\Gamma, V']$:

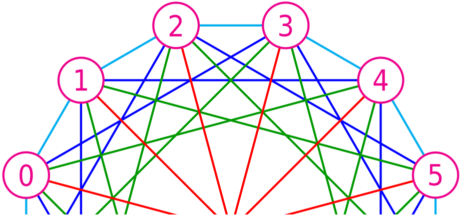
- Choose vertices $V' \subseteq V$



[DMR25] Hugo Delavenne, Tanguy Medevielle, and Élina Roussel. Interactive Oracle Proofs of Proximity to Codes on Graphs, 2025. **Submitted**

Cut-graph $\Gamma' = \text{Cut}[\Gamma, V']$:

- Choose vertices $V' \subseteq V$
- Cut the rest

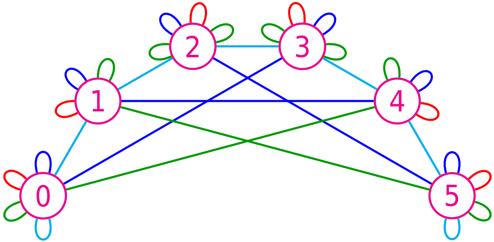


[DMR25] Hugo Delavenne, Tanguy Medevielle, and Élina Roussel. Interactive Oracle Proofs of Proximity to Codes on Graphs, 2025. **Submitted**

Cut-graph $\Gamma' = \text{Cut}[\Gamma, V']$:

- ▷ Choose vertices $V' \subseteq V$
- ▷ Cut the rest
- ▶ Enjoy your new graph

$$E_{V'}(v,\ell) = \begin{cases} E(v,\ell) & \text{if } E(v,\ell) \in V' \\ v & \text{otherwise} \end{cases}$$

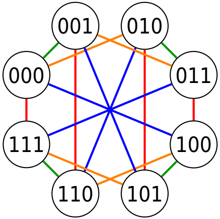


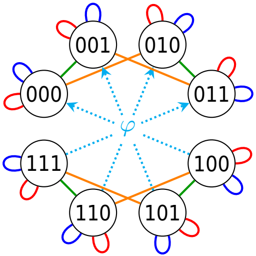
[DMR25] Hugo Delavenne, Tanguy Medevielle, and Élina Roussel. Interactive Oracle Proofs of Proximity to Codes on Graphs, 2025. **Submitted**

Definition **Graph isomorphism**

A bijection $\varphi : V' \to V''$ is an **isomorphism** $\Gamma' \to \Gamma''$ if

$$\forall (v', \ell) \in V' \times [n], \quad \varphi(E'(v', \ell)) = E''(\varphi(v'), \ell).$$





[DMR25] Hugo Delavenne, Tanguy Medevielle, and Élina Roussel. Interactive Oracle Proofs of Proximity to Codes on Graphs, 2025. **Submitted**

Definition **Graph isomorphism**

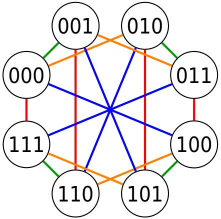
A bijection $\varphi : V' \to V''$ is an **isomorphism** $\Gamma' \to \Gamma''$ if

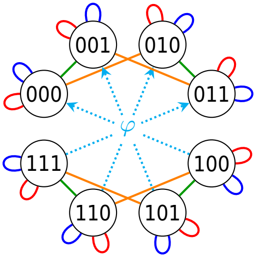
$$\forall (v', \ell) \in V' \times [n], \quad \varphi(E'(v', \ell)) = E''(\varphi(v'), \ell).$$

Definition **Flowering cut**

With $V'' = V \setminus V'$, if $\mathsf{Cut}[\Gamma, V'] \sim \mathsf{Cut}[\Gamma, V'']$, the cut is **flowering**.

The cut-word $\mathsf{Cut}[f, V']$ is the restriction $f_{|V' \times [n]}$.



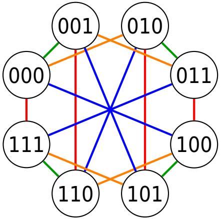


[DMR25] Hugo Delavenne, Tanguy Medevielle, and Élina Roussel. Interactive Oracle Proofs of Proximity to Codes on Graphs, 2025. **Submitted**

Definition **Graph isomorphism**

A bijection $\varphi : V' \to V''$ is an **isomorphism** $\Gamma' \to \Gamma''$ if

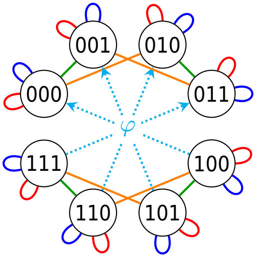
$$\forall (v', \ell) \in V' \times [n], \quad \varphi(E'(v', \ell)) = E''(\varphi(v'), \ell).$$



Definition **Flowering cut**

With $V'' = V \setminus V'$, if Cut$[\Gamma, V'] \sim$ Cut$[\Gamma, V'']$, the cut is **flowering**.

The cut-word Cut$[f, V']$ is the restriction $f_{|V' \times [n]}$.

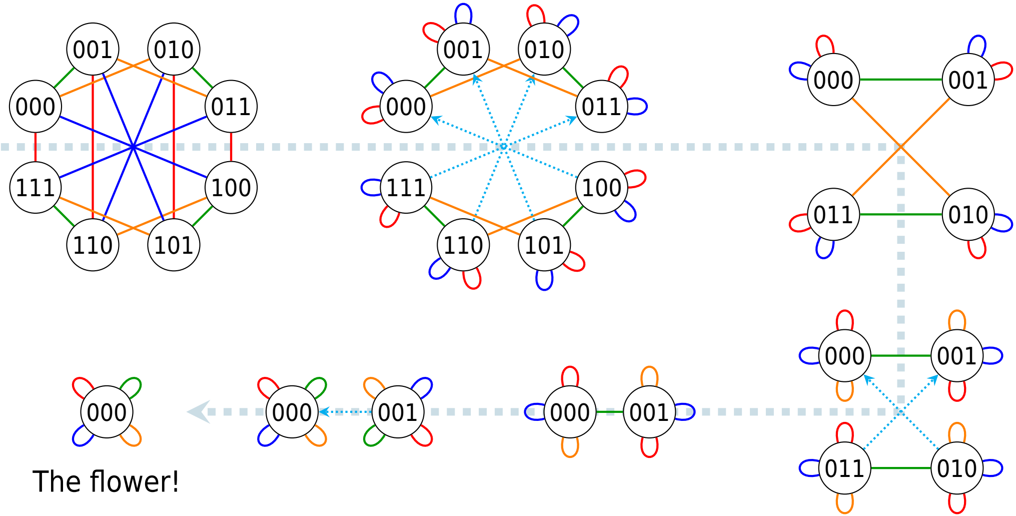


Definition **Folding**

For $\alpha \in \mathbb{F}$, $(v', \ell) \in V' \times [n]$,

$$\text{Fold}[f, \alpha](v', \ell) := \text{Cut}[f, V'](v, \ell) + \alpha \text{Cut}[f, V''](\varphi^{-1}(v'), \ell).$$

[DMR25] Hugo Delavenne, Tanguy Medevielle, and Élina Roussel. Interactive Oracle Proofs of Proximity to Codes on Graphs, 2025.
Submitted


001
010
000
011
111
100
110
101
001
010
000
011
111
100
110
101
000
001
011
010
000
001
011
010
000
000
001
000
001

The flower!

Flowering protocol

$$f_0 : V_0 \times [n] \to \mathbb{F}$$

$\mathcal{P}$ $\mathcal{V}$

$$\alpha_0 \xleftarrow{\$} \mathbb{F}$$

$$f_1 : V_1 \times [n] \to \mathbb{F}$$

$$f_1 = \mathsf{Fold}[f_0, \alpha_0]$$

1 commit phase

2 query phase

[DMR25] Hugo Delavenne, Tanguy Medevielle, and Élina Roussel. Interactive Oracle Proofs of Proximity to Codes on Graphs, 2025. **Submitted**

Flowering protocol

$$f_0 : V_0 \times [n] \to \mathbb{F}$$

$\mathcal{P}$ $\qquad\qquad$ $\mathcal{V}$

$$\alpha_0 \xleftarrow{\$} \mathbb{F}$$

$$f_1 : V_1 \times [n] \to \mathbb{F}$$

$$f_1 = \mathsf{Fold}[f_0, \alpha_0]$$

① commit phase

$$v_0 \xleftarrow{\$} V_0,\ v_1 := \pi_1(v_0)$$

$$f_1(v_1, \cdot) \stackrel{?}{=} \mathsf{Fold}[f_0, \alpha_0](v_1, \cdot)$$

② query phase

[DMR25] Hugo Delavenne, Tanguy Medevielle, and Élina Roussel. Interactive Oracle Proofs of Proximity to Codes on Graphs, 2025. **Submitted**

Flowering protocol

$$f_0 : V_0 \times [n] \to \mathbb{F}$$

$\mathcal{P}$ $\mathcal{V}$

$$\alpha_0 \xleftarrow{\$} \mathbb{F}$$

$$f_1 : V_1 \times [n] \to \mathbb{F}$$

$$f_1 = \mathsf{Fold}[f_0, \alpha_0]$$

$$\vdots$$

$$\alpha_{r-1} \xleftarrow{\$} \mathbb{F}$$

$$f_r : V_r \times [n] \to \mathbb{F}$$

$$f_r = \mathsf{Fold}[f_{r-1}, \alpha_{r-1}]$$

1 commit phase

$$v_0 \xleftarrow{\$} V_0,\ v_1 := \pi_1(v_0)$$

$$f_1(v_1, \cdot) \stackrel{?}{=} \mathsf{Fold}[f_0, \alpha_0](v_1, \cdot)$$

2 query phase

[DMR25] Hugo Delavenne, Tanguy Medevielle, and Élina Roussel. Interactive Oracle Proofs of Proximity to Codes on Graphs, 2025. **Submitted**

$$f_0 : V_0 \times [n] \to \mathbb{F}$$

$\mathcal{P}$ $\mathcal{V}$

$$\alpha_0 \xleftarrow{\$} \mathbb{F}$$

$$f_1 : V_1 \times [n] \to \mathbb{F}$$

$$f_1 = \mathsf{Fold}[f_0, \alpha_0]$$

$$\vdots$$

$$\alpha_{r-1} \xleftarrow{\$} \mathbb{F}$$

$$f_r : V_r \times [n] \to \mathbb{F}$$

$$f_r = \mathsf{Fold}[f_{r-1}, \alpha_{r-1}]$$

1 commit phase

$$v_0 \xleftarrow{\$} V_0,\ v_1 := \pi_1(v_0)$$

$$f_1(v_1, \cdot) \stackrel{?}{=} \mathsf{Fold}[f_0, \alpha_0](v_1, \cdot)$$

$$\vdots$$

$$v_r := \pi_r(v_{r-1})$$

$$f_r(v_r, \cdot) \stackrel{?}{=} \mathsf{Fold}[f_{r-1}, \alpha_{r-1}](v_r, \cdot)$$

2 query phase

[DMR25] Hugo Delavenne, Tanguy Medevielle, and Élina Roussel. Interactive Oracle Proofs of Proximity to Codes on Graphs, 2025. **Submitted**

$$f_0 : V_0 \times [n] \to \mathbb{F}$$

$\mathcal{P}$ $\mathcal{V}$

$$\alpha_0 \xleftarrow{\$} \mathbb{F}$$

$$f_1 : V_1 \times [n] \to \mathbb{F}$$

$$f_1 = \mathsf{Fold}[f_0, \alpha_0]$$

$$\vdots$$

$$\alpha_{r-1} \xleftarrow{\$} \mathbb{F}$$

$$f_r : V_r \times [n] \to \mathbb{F}$$

$$f_r = \mathsf{Fold}[f_{r-1}, \alpha_{r-1}]$$

1 commit phase

$$v_0 \xleftarrow{\$} V_0,\ v_1 := \pi_1(v_0)$$

$$f_1(v_1, \cdot) \stackrel{?}{=} \mathsf{Fold}[f_0, \alpha_0](v_1, \cdot)$$

$$\vdots$$

$$v_r := \pi_r(v_{r-1})$$

$$f_r(v_r, \cdot) \stackrel{?}{=} \mathsf{Fold}[f_{r-1}, \alpha_{r-1}](v_r, \cdot)$$

repeat m times

2 query phase

[DMR25] Hugo Delavenne, Tanguy Medevielle, and Élina Roussel. Interactive Oracle Proofs of Proximity to Codes on Graphs, 2025. **Submitted**

Flowering protocol

$$f_0 : V_0 \times [n] \to \mathbb{F}$$

$\mathcal{P}$ $\mathcal{V}$

$$\alpha_0 \xleftarrow{\$} \mathbb{F}$$

$$f_1 : V_1 \times [n] \to \mathbb{F}$$

$$f_1 = \mathsf{Fold}[f_0, \alpha_0]$$

$$\vdots$$

$$\alpha_{r-1} \xleftarrow{\$} \mathbb{F}$$

$$f_r : V_r \times [n] \to \mathbb{F}$$

$$f_r = \mathsf{Fold}[f_{r-1}, \alpha_{r-1}]$$

1 commit phase

$$v_0 \xleftarrow{\$} V_0,\ v_1 := \pi_1(v_0)$$

$$f_1(v_1, \cdot) \stackrel{?}{=} \mathsf{Fold}[f_0, \alpha_0](v_1, \cdot)$$

$$\vdots$$

$$v_r := \pi_r(v_{r-1})$$

$$f_r(v_r, \cdot) \stackrel{?}{=} \mathsf{Fold}[f_{r-1}, \alpha_{r-1}](v_r, \cdot)$$

$$f_r \stackrel{?}{\in} \mathsf{RS}[n, k]$$

repeat m times

2 query phase

[DMR25] Hugo Delavenne, Tanguy Medevielle, and Élina Roussel. Interactive Oracle Proofs of Proximity to Codes on Graphs, 2025. **Submitted**

Proposition **Flowering complexities**

Flowering with m repetitions has complexities: (recall FRI)

- Prover complexity: $< 3N$ — $< 8N$
- Verifier complexity: $4mnr$ — $< 2m \log K$
- Number of queries: $\sim 2mnr$ — $2m \log K$
- Number of rounds: r — $\log K$

[BBHR18] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Fast Reed-Solomon Interactive Oracle Proofs of Proximity. **In *45th international colloquium on automata, languages, and programming (ICALP 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018**

[DMR25] Hugo Delavenne, Tanguy Medevielle, and Élina Roussel. Interactive Oracle Proofs of Proximity to Codes on Graphs, 2025. **Submitted**

Complexity comparison

Proposition **Flowering complexities**

Flowering with m repetitions has complexities: (with our first graphs) (recall FRI)

- Prover complexity: $< 3N$ — $< 8N$
- Verifier complexity: $4mnr$ ($< 4m\log^2 N$) — $< 2m\log K$
- Number of queries: $\sim 2mnr$ ($< 2m\log^2 N$) — $2m\log K$
- Number of rounds: r ($< \log N$) — $\log K$

[BBHR18] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Fast Reed-Solomon Interactive Oracle Proofs of Proximity. **In *45th international colloquium on automata, languages, and programming (ICALP 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018**

[DMR25] Hugo Delavenne, Tanguy Medevielle, and Élina Roussel. Interactive Oracle Proofs of Proximity to Codes on Graphs, 2025. **Submitted**

Completeness and soundness

Proposition **Flowering completeness**

If $f \in \mathcal{C}[\Gamma, \mathsf{RS}[n,k]]$ then $\mathcal{V}$ accepts with probability 1.

Theorem **Flowering soundness**

If $\Delta(f, \mathcal{C}[\Gamma, \mathsf{RS}[n,k]]) > \delta$ then for any $\tilde{\mathcal{P}}$ and $\varepsilon > 0$, $\mathcal{V}$ accepts with probability

$$\leqslant \frac{r}{\varepsilon|\mathbb{F}|} + (1 - \delta + \varepsilon r)^m .$$

Recall FRI: $$\frac{K^2 \log K}{(2\varepsilon)^7 q|\mathbb{F}|} + \left(1 - \min\left(\delta, 1 - \sqrt{K/N} - \varepsilon\right)\right)^m$$

[BCI+23] Eli Ben-Sasson, Dan Carmon, Yuval Ishai, Swastik Kopparty, and Shubhangi Saraf. Proximity Gaps for Reed–Solomon Codes. ***J. ACM*, 70(5), October 2023**

[DMR25] Hugo Delavenne, Tanguy Medevielle, and Élina Roussel. Interactive Oracle Proofs of Proximity to Codes on Graphs, 2025. **Submitted**

- First graphs
- Expander graphs
- New cuts (*current work*)

Cayley graphs

Definition **Cayley graph Cay(G, S)**

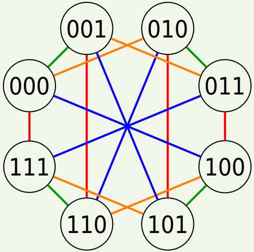
Fix $(G, \cdot)$. Let $S \subseteq G$ symmetric generating. Define $\Gamma = (G, E)$ where $E(g, s) = g \cdot s$.

We take

- $G = (\mathbb{F}_2^r, +)$
- $S \subseteq G$ of size n
- $\Gamma = \mathsf{Cay}[G, S]$
- $C = \mathcal{C}[\Gamma, \mathsf{RS}[n, k]]$

Example

With $G = (\mathbb{F}_2^3, +)$ and $S = \{100, 010, 001, 111\}$,



[Cay78] Arthur Cayley. Desiderata and Suggestions: No. 2. The Theory of Groups: Graphical Representation. ***American Journal of Mathematics*, 1(2):174–176, 1878**

Using S the columns of a parity check matrix of a $[n, n-r, d]_2$ binary code

Proposition **Parameters of the code**

- $N = n2^{r-1}$
- $\operatorname{rate} C \geqslant \frac{2k}{n} - 1$
- $\frac{1}{2}\delta \leqslant \Delta(C) \leqslant \delta$, with $\delta = \frac{1}{2^{r-d+1}}\left(1 - \frac{k-1}{n}\right) = \frac{n2^{d-2}}{N}\left(1 - \frac{k-1}{n}\right)$

[DMR25] Hugo Delavenne, Tanguy Medevielle, and Élina Roussel. Interactive Oracle Proofs of Proximity to Codes on Graphs, 2025.
Submitted

Using S the columns of a parity check matrix of a $[n, n-r, d]_2$ binary code

Proposition **Parameters of the code**

- $N = n2^{r-1}$
- $\text{rate}\, C \geqslant \frac{2k}{n} - 1$
- $\frac{1}{2}\delta \leqslant \Delta(C) \leqslant \delta$, with $\delta = \frac{1}{2^{r-d+1}}\left(1 - \frac{k-1}{n}\right) = \frac{n2^{d-2}}{N}\left(1 - \frac{k-1}{n}\right)$

If $d \ll r$ (i.e. far from MDS), δ is **terrible** ($O(1/N)$).

[DMR25] Hugo Delavenne, Tanguy Medevielle, and Élina Roussel. Interactive Oracle Proofs of Proximity to Codes on Graphs, 2025. **Submitted**

Expander graphs

Definition **Graph expansion**

Let $\Gamma = (V, E)$ a n-regular graph and $A \in \{0,1\}^{|V|\times|V|}$ its adjacency matrix.
Let $\Lambda_1 \geqslant \Lambda_2 \geqslant \dots \geqslant \Lambda_n \in \mathbb{R}$ be the eigenvalues of A.

Then Γ is **λ-expander** if $|\Lambda_i| \leqslant n\lambda$ for $i \geqslant 2$.

Expander graphs

Definition **Graph expansion**

Let $\Gamma = (V, E)$ a n-regular graph and $A \in \{0,1\}^{|V|\times|V|}$ its adjacency matrix.
Let $\Lambda_1 \geqslant \Lambda_2 \geqslant \ldots \geqslant \Lambda_n \in \mathbb{R}$ be the eigenvalues of A.

Then Γ is **λ-expander** if $|\Lambda_i| \leqslant n\lambda$ for $i \geqslant 2$.

$\lambda \in [0, 1]$ characterizes random walk propagation in Γ.

Small λ means good expansion.

Definition Graph expansion

Let $\Gamma = (V, E)$ a n-regular graph and $A \in \{0,1\}^{|V|\times|V|}$ its adjacency matrix.
Let $\Lambda_1 \geqslant \Lambda_2 \geqslant \ldots \geqslant \Lambda_n \in \mathbb{R}$ be the eigenvalues of A.

Then Γ is **λ-expander** if $|\Lambda_i| \leqslant n\lambda$ for $i \geqslant 2$.

$\lambda \in [0,1]$ characterizes random walk propagation in Γ.

Small λ means good expansion.

Lemma Minimal distance expansion lower bound [AC88]

If Γ is λ-expander, with $\delta = 1 - \frac{k+1}{n}$, $\mathcal{C}[\Gamma, \mathsf{RS}[n,k]]$ has minimal distance $\geqslant \delta(\delta - \lambda)$.

Thus if $(\Gamma_i)_{i\in\mathbb{N}}$ has **constant expansion**,
then $(\mathcal{C}[\Gamma_i, \mathsf{RS}[n_i, \gamma n_i]])_{i\in\mathbb{N}}$ has **constant minimal distance**.

[AC88] Noga Alon and Fan Chung. Explicit construction of linear sized tolerant networks. ***Discrete Mathematics*, 72(1-3):15–19, 1988**

Expander Cayley graph examples

Let

- $G_p = \mathrm{SL}_3(\mathbb{F}_p)$
- S_p symmetric generating
- $\Gamma_p = \mathsf{Cay}(G_p, S_p)$

then $(\Gamma_p)_p$ has **constant expansion**.

Expander Cayley graph examples

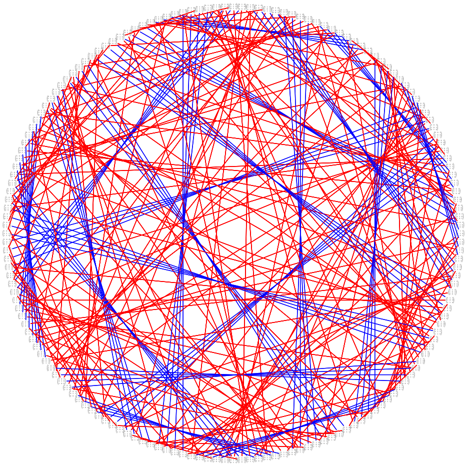
Let

- $G_p = \mathrm{SL}_3(\mathbb{F}_p)$
- S_p symmetric generating
- $\Gamma_p = \mathsf{Cay}(G_p, S_p)$

then $(\Gamma_p)_p$ has **constant expansion**.

We obtain Γ_2 with

$$S_2 = \left\{ \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}^{\pm 1} \right\}:$$



Expander Cayley graph examples

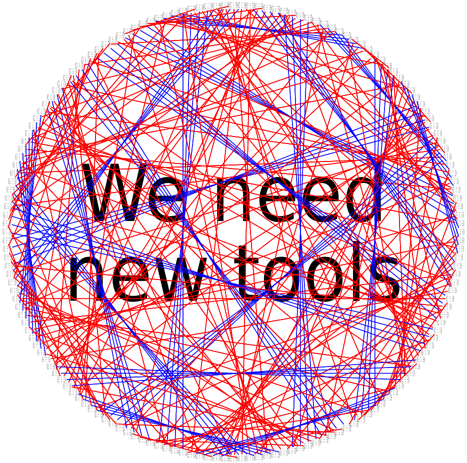
Let

- $G_p = \mathrm{SL}_3(\mathbb{F}_p)$
- S_p symmetric generating
- $\Gamma_p = \mathsf{Cay}(G_p, S_p)$

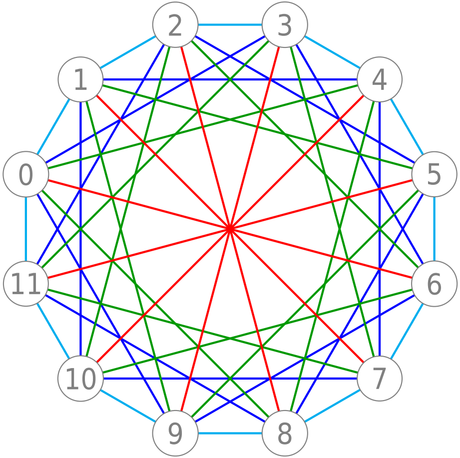
then $(\Gamma_p)_p$ has **constant expansion**.

We obtain Γ_2 with

$$S_2 = \left\{ \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}^{\pm 1} \right\}:$$

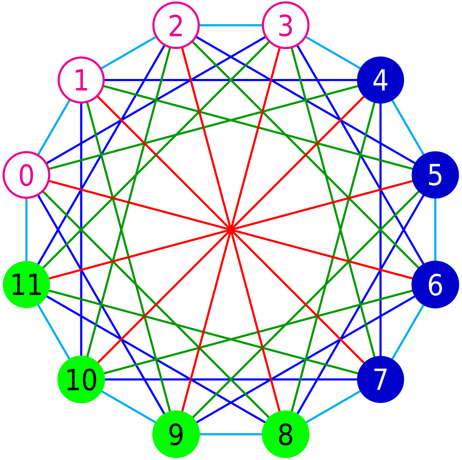


We can cut into $m > 2$ cuts



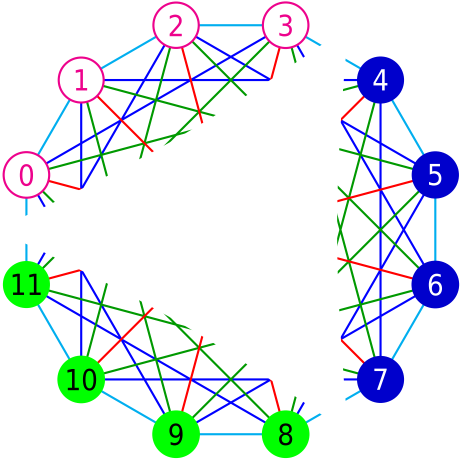
We can cut into $m > 2$ cuts

- Choose m sets of vertices $V_0, ..., V_{m-1} \subseteq V$



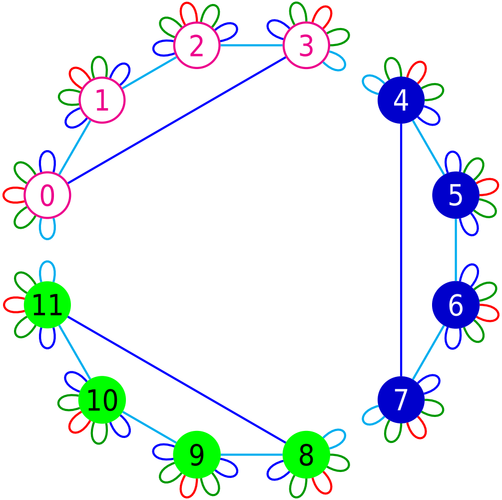
We can cut into $m > 2$ cuts

- Choose m sets of vertices $V_0, ..., V_{m-1} \subseteq V$
- Cut outgoing edges



We can cut into $m > 2$ cuts

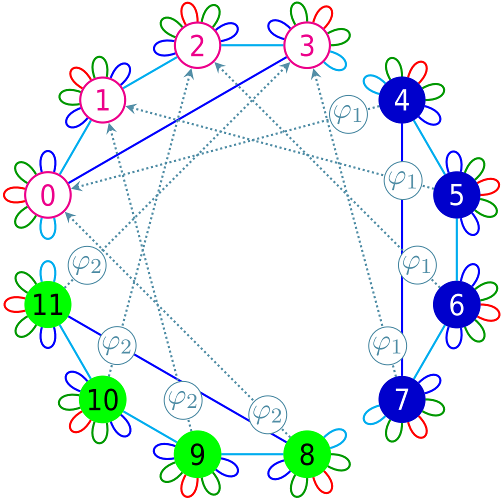
- ▷ Choose m sets of vertices $V_0, ..., V_{m-1} \subseteq V$
- ▷ Cut outgoing edges
- ▶ Get new graphs with petals



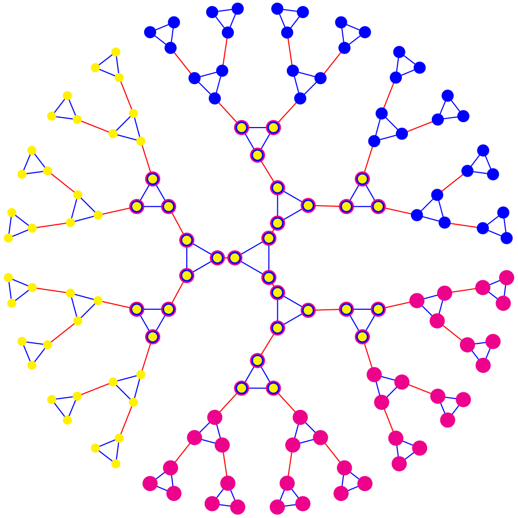
We can cut into $m > 2$ cuts

- ▷ Choose m sets of vertices $V_0, \ldots, V_{m-1} \subseteq V$
- ▷ Cut outgoing edges
- ▷ Get new graphs with petals
- ▶ Define the new Fold with the isomorphisms

$$\mathsf{Fold}[f, \alpha](v, \ell) := \sum_{i=0}^{m-1} \alpha^i f(\varphi_i^{-1}(v), \ell)$$



Cuts $V_1, ..., V_m$ may not be disjoint:



Let $S = \{s_0^{\pm 1}, s_2^{\pm 1}, \ldots, s_{\tilde{n}-1}^{\pm 1}\}$ and $G = \langle S \rangle$ be finite. Write $\tilde{s}_i := s_{i \bmod \tilde{n}}$. Let

$$r_g := \min\{k \in \mathbb{N} \mid g = \tilde{s}_0^{j_1} \cdots \tilde{s}_{r_g}^{j_{r_g}}\}$$
$$r := \max_{g \in G} r_g \leqslant \tilde{n} \cdot \operatorname{diam} \mathsf{Cay}(G, S) = O(\tilde{n} \log |G|)$$

The cuts are $\quad V_{i,j} := \left\{\tilde{s}_i^{j} \cdot \tilde{s}_{i+1}^{j_{i+1}} \cdots \tilde{s}_r^{j_r} \mid j_{i+1}, \ldots, j_k \in \mathbb{N}\right\}$.

- $\Gamma_{i,j} \sim \Gamma_{i,0}$ with $\varphi_{i,j}(g) = \tilde{s}_i^{-j} g \quad \rightarrow \mathbf{order}(\tilde{s}_i)$ **cuts**
- $\Gamma_{r,0}$ is a flower $\quad \rightarrow O(\tilde{n} \log |G|)$ **rounds**

Competing parameters with FRI

- We have a better soundness
- Our complexity could be improved

New cuts, new graphs

- 2 disjoint cuts V', V'' $\rightarrow$ m covering cuts $V_1, V_2, ..., V_m$
- Compute complexity for general Cayley graphs

Make this actually useful

- Arithmetize circuits to graphs: colored De Bruijn
- Encode words on graphs into bigger flowering graphs