Introduction to STARK

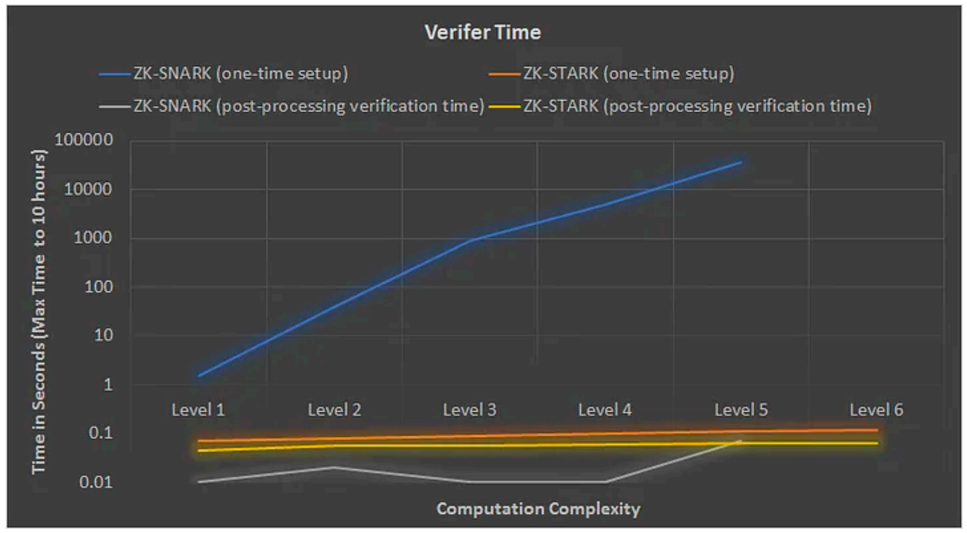
STARK vs SNARK

Pros

- **Quantum resistant**
- **Transparent (don’t use trusted setup)**
- **Prover complexity:** ZK-STARKs are ~10x faster than ZK-SNARKs as computation size increases
- **Verifier complexity:** As computation size grows, ZK-STARKs grow only slightly vs. ZK-SNARKS

Cons

- **Proof size**



STARKWARE STARK 101

Stark 101: Part 1

Statement, LDE and Commitment

FibonacciSq

(Fibonacci Square)

FibonacciSq (Fibonacci Square)

FibonacciSq: $a_{n+2} = a_{n+1}^2 + a_n^2$

- Represented as: $a_0, a_1, a_2, a_3, \ldots$
- Determined by first two elements
- Example:
  - 1, 3, 10, 109, 11981, 143556242,...

Tiny Problem

$$a_{10} = 1058538448149133154544343598019533016808522710856082409891927825821583978969754411443713008055652428916885458657978238751812992228283226160560814552379774771482746584257000514878526588336710877240208661850336931934256166336593387070293738452872952783090264176685$$

FibonacciSq Mod Prime

FibonacciSq mod prime: $a_{n+2} = a_{n+1}^2 + a_n^2$ mod *prime*

Example:

- 1, 3, 10, 109, 11981, 143556242,...

mod 7:

- 1, 3, 3, 4, 4, 4, ...

FibonacciSq Mod Prime

FibonacciSq mod prime: $a_{n+2} = a_{n+1}^2 + a_n^2 \mod prime$

- Example - mod 7:
  - 1, 3, 3, 4, 4, 4, ...

We use $prime = 3 \cdot 2^{30} + 1 = 322122547$

Finite field F

Statement

Statement to Prove

There is a number x such that:

For the FibonacciSq mod 3221225473 with

- $a_0 = 1$
- $a_1 = x$

we have $a_{1022} = 2338775057$

X = 3141592

STARK Protocol - Part I

- LDE - Low Degree Extension
- Commitment

Low Degree Extension (LDE)

LDE in 3 Steps

1. Generate input
2. Interpolate
3. Extend

LDE - General

LDE Step 1 - Generate Input

Input: $y_0, y_1, y_2, y_3, y_4, \ldots$

Choose: $x_0, x_1, x_2, x_3, x_4, \ldots$

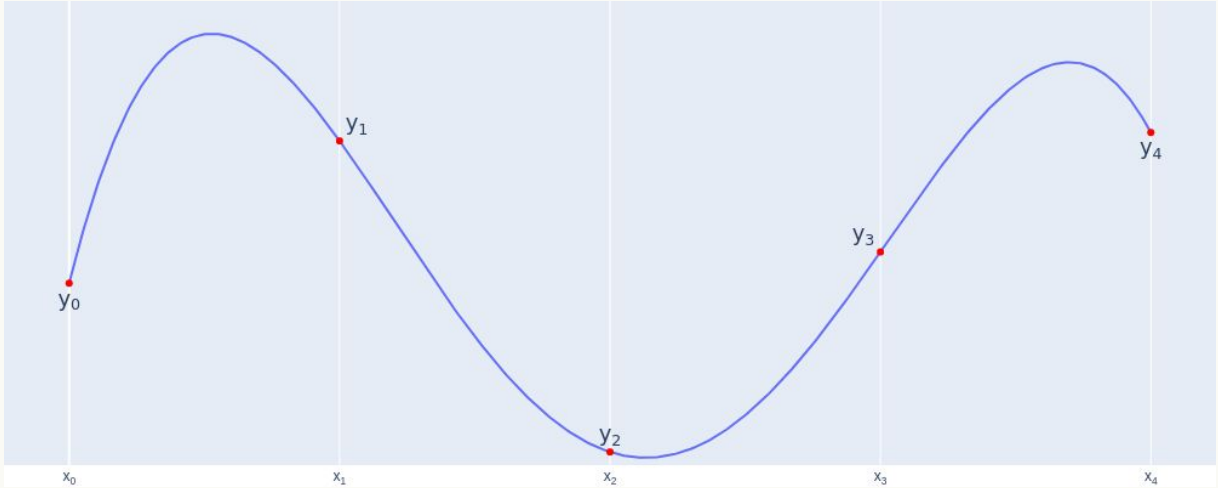
x	*y*
x_0	y_0
x_1	y_1
x_2	y_2
x_3	y_3
x_4	y_4

LDE Step 2 - Interpolate Polynomial

Interpolate a polynomial *f*:

$$For\ each\ i : f(x_i)=y_i$$

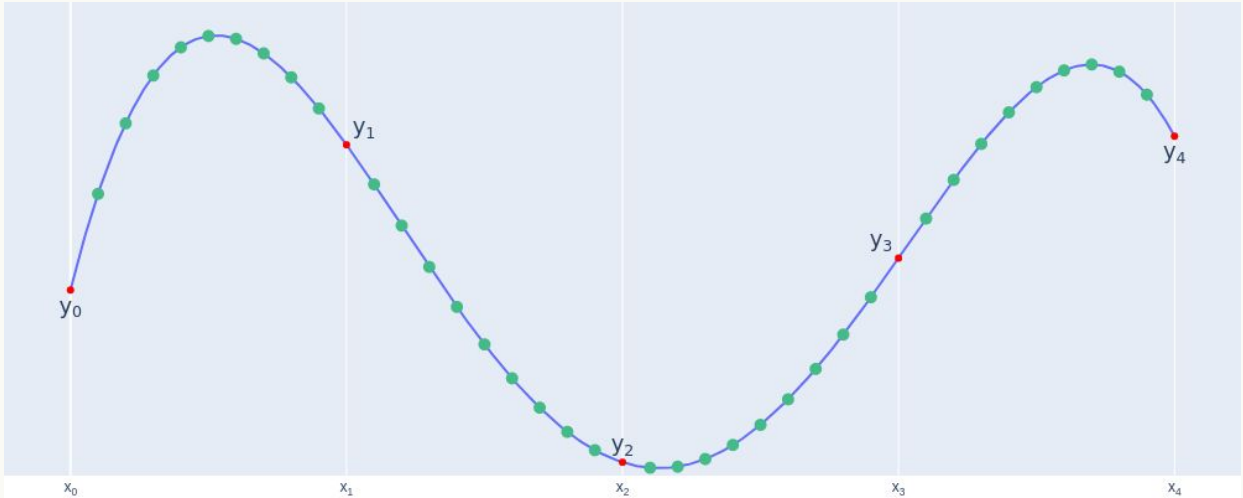
x	$f(x)$
x_0	y_0
x_1	y_1
x_2	y_2
x_3	y_3
x_4	y_4



LDE Step 3 - Extend

- Pick a larger evaluation domain $\{x_j`\}$
- Output: $\{f(x_j`)\}$

$x`$	$f(x`)$
$x`_0$	$f(x`_0)$
$x`_1$	$f(x`_1)$
$x`_2$	$f(x`_2)$
$x`_3$	$f(x`_3)$
...	...



LDE in STARK

LDE for STARK Step 1 - Generate Input

Input: $a_0, a_1, a_2, \ldots, a_{1022}$

The **Trace**

We choose: $1, g, g^2, g^3, \ldots, g^{1022}$

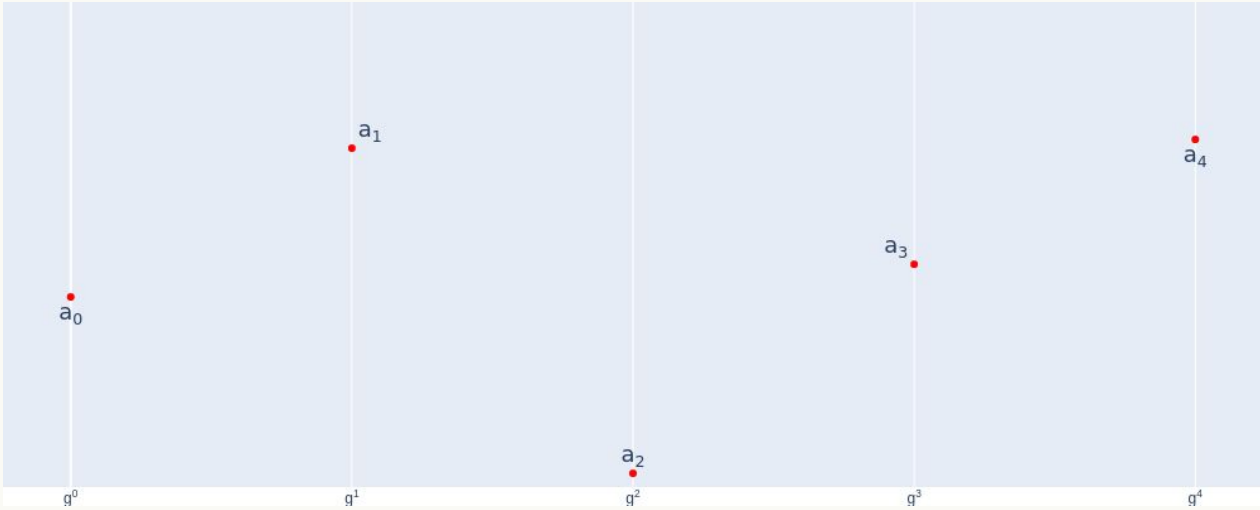
g - element from F

LDE for STARK Step 1 - Generate Input

Input: $a_0, a_1, a_2, \ldots, a_{1022}$

We choose: $1, g, g^2, g^3, \ldots, g^{1022}$

x	*f(x)*
g^0	a_0
g^1	a_1
g^2	a_2
...	...
g^{1022}	a_{1022}

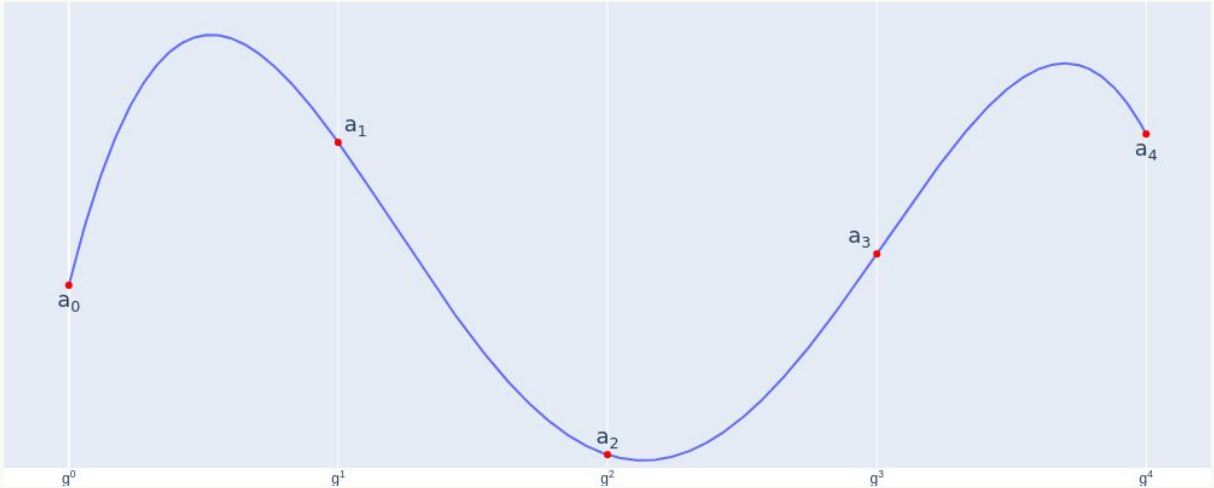


LDE for STARK Step 2 - Interpolate Poly

Interpolate a polynomial f:

$$for\ each\ i : f(g^i)=a_i$$

x	$f(x)$
g^0	a_0
g^1	a_1
g^2	a_2
...	...
g^{1022}	a_{1022}



LDE for STARK Step 3 - Extend

- Pick a larger evaluation domain (8k)
- $\{x_i`\} = w, w \cdot h, w \cdot h^2, \ldots, w \cdot h^{8191}$

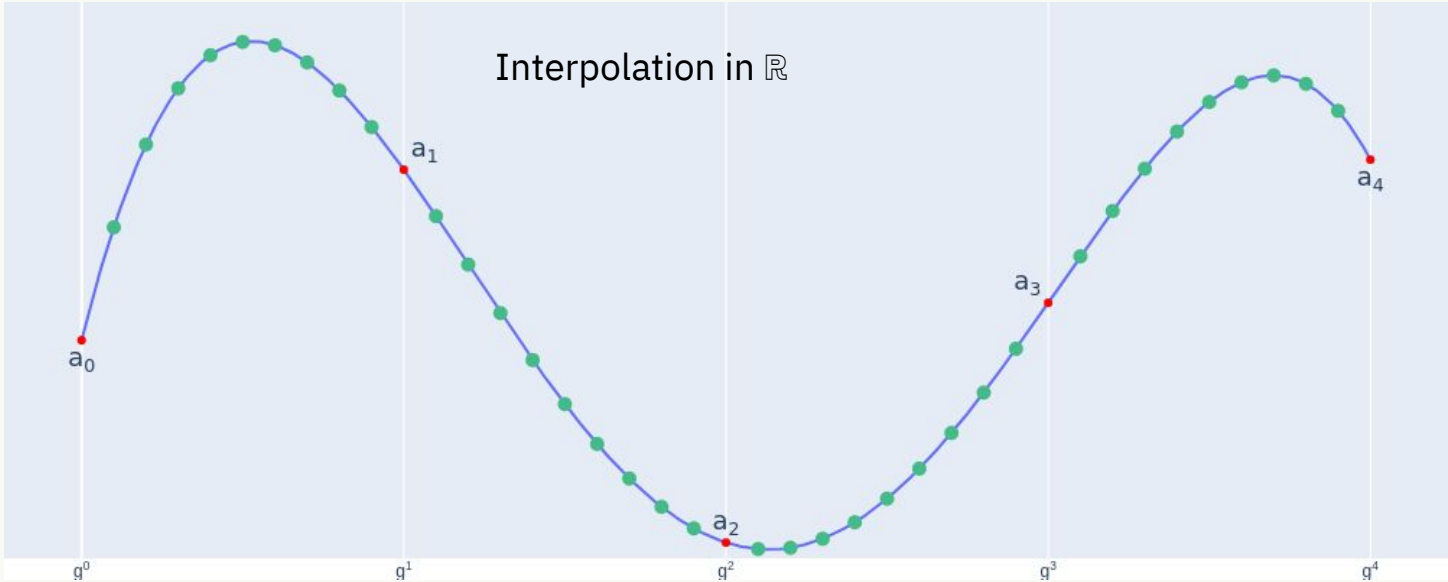
w, h - elements from *F*

- Result: $f(w), f(w \cdot h), f(w \cdot h^2), \ldots$

Reed-Solomon codeword

LDE for STARK Step 3 - Extend

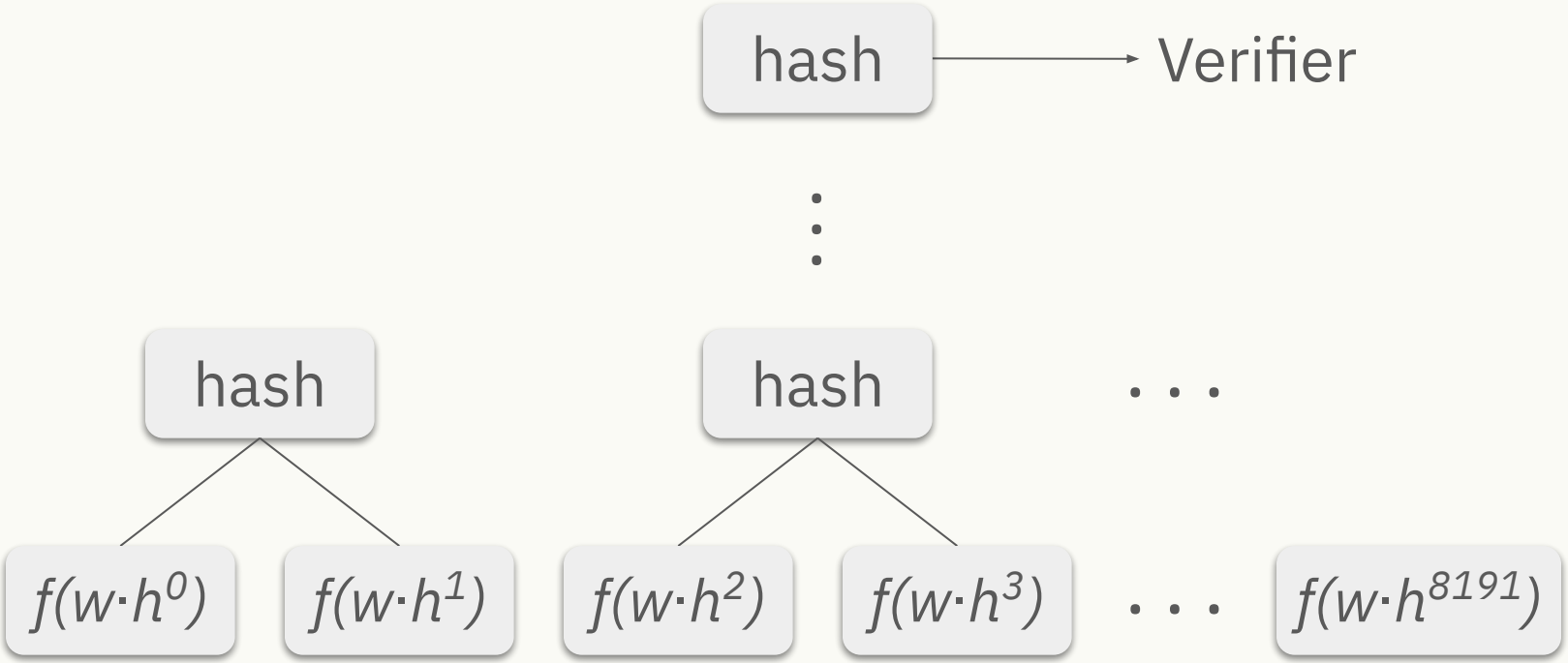
x	$f(x)$
$w \cdot h^0$	$f(w \cdot h^0)$
$w \cdot h^1$	$f(w \cdot h^1)$
$w \cdot h^2$	$f(w \cdot h^2)$
...	...
$w \cdot h^{8191}$	$f(w \cdot h^{8191})$



Commitment

Commit on LDE

Merkle Tree



Summary

- Statement
  - There is x s.t. a_{1022}= 2338775057 in FibonacciSq mod prime
- STARK protocol - part I:
  - LDE - Low Degree Extension
  - Commitment - Merkle Tree

STARKWARE STARK 101

Stark 101: Part 2

Polynomial Constraints

What Do We Want to Prove?

There is a number x such that:

$a_0 = 1$

$a_1 = x$

$a_{1022} = 2338775057$

For $\{a_n\}$ FibonacciSq: $a_{n+2} = a_{n+1}^2 + a_n^2$ mod prime, for any n

We will use part I:

Trace - a

Generator of G - g

Trace Polynomial - $f(x)$

Constraints on $\{a_n\}$

We need:

$a_0 = 1$

$a_{1022} = 2338775057$

$a_{n+2} = a_{n+1}^2 + a_n^2$

If $\{a_n\}$ satisfies constraints ⟶ Original statement is true!

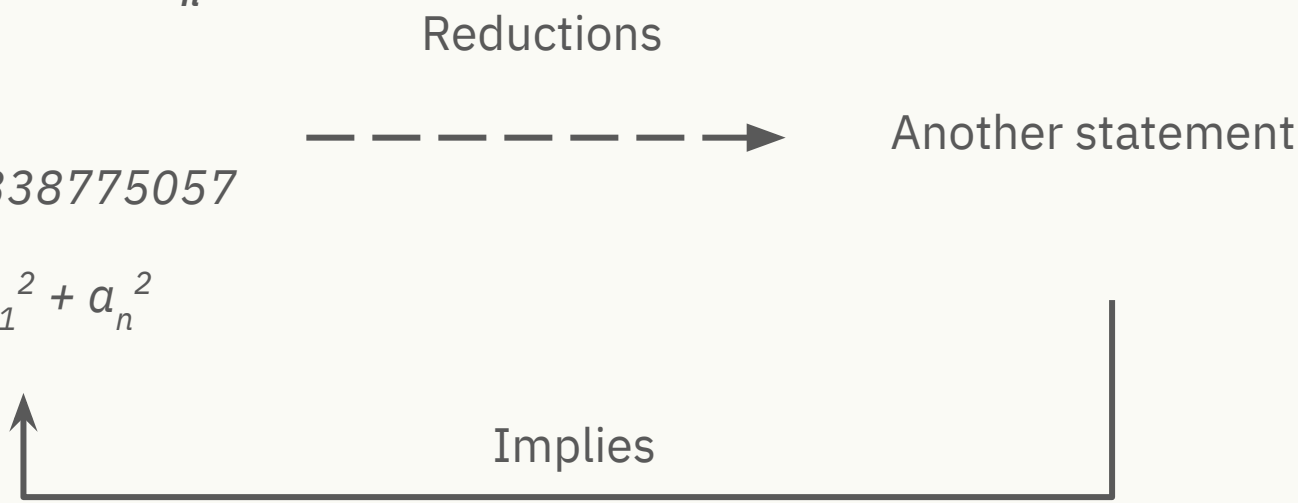
Where are We Heading?

Constraints on $\{a_n\}$:

$a_0 = 1$

$a_{1022} = 2338775057$

$a_{n+2} = a_{n+1}^2 + a_n^2$



Where are We Heading?

Constraints on $\{a_n\}$:

$a_0 = 1$

$a_{1022} = 2338775057$

$a_{n+2} = a_{n+1}^2 + a_n^2$

Reductions

Trace polynomial

Exists a polynomial $f(x)$

such that:

3 rational functions

$p_0(x)$, $p_1(x)$, $p_2(x)$ are **polynomials**

$$\frac{a(x)}{b(x)}$$

Step I - From $\{a_n\}$ to $f(x)$

Trace polynomial

3 constraints on $\{a_n\}$ ------► 3 constraints on $f(x)$

$a_0 = 1$ ------► $f(x) = 1$, for $x = g^0$

$a_{1022} = 2338775057$ ------► $f(x) = 2338775057$, for $x = g^{1022}$

$a_{n+2} = a_{n+1}^2 + a_n^2$

x	$f(x)$
g^0	a_0
g^1	a_1
g^2	a_2
...	...
g^{1022}	a_{1022}

Step I - From $\{a_n\}$ to $f(x)$

$a_{n+2} = a_{n+1}^2 + a_n^2$ ------► $f(g^2x) = f(gx)^2 + f(x)^2$,

for $x = g^i$, $0 \le i \le 1020$

Example: for $x = g^5$:

$$f(\underbrace{g^2 \cdot g^5}_{g^7}) = f(\underbrace{g \cdot g^5}_{g^6})^2 + f(\underbrace{g^5}_{g^5})^2$$

x	f(x)
g^0	a_0
g^1	a_1
g^2	a_2
...	...
g^5	a_5
g^6	a_6
g^7	a_7
...	...
g^{1022}	a_{1022}

Step I - From $\{a_n\}$ to $f(x)$

3 constraints on $\{a_n\}$ ⟶ 3 constraints on $f(x)$

$a_0 = 1$ ⟶ $f(x) = 1$, for $x = g^0$

$a_{1022} = 2338775057$ ⟶ $f(x) = 2338775057$, for $x = g^{1022}$

$a_{n+2} = a_{n+1}^2 + a_n^2$ ⟶ $f(g^2x) = f(gx)^2 + f(x)^2$,

for $x = g^i$, $0 \le i \le 1020$

Step I - From $\{a_n\}$ to $f(x)$

3 constraints on $\{a_n\}$		3 constraints on $f(x)$
$a_0 = 1$	⟶	$f(x) = 1$, for $x = g^0$
$a_{1022} = 2338775057$	⟶	$f(x) = 2338775057$, for $x = g^{1022}$
$a_{n+2} = a_{n+1}^2 + a_n^2$	⟶	$f(g^2x) = f(gx)^2 + f(x)^2$, for $x = g^i$, $0 \le i \le 1020$

If $f(x)$ satisfies constraints ⟶ Original statement is true

Step II - From Constraints to Roots

z is a root of $p(x)$ if $p(z)=0$

$f(x) - 1 = 0$, for $x = g^0$ ⟶ root: g^0

($f(x) = 1$, for $x = g^0$)

Step II - From Constraints to Roots

$f(x) - 1 = 0$, for $x = g^0$ ⟶ root: g^0

$f(x) - 2338775057 = 0$, for $x = g^{1022}$ ⟶ root: g^{1022}

Step II - From Constraints to Roots

$f(x) - 1 = 0$, for $x = g^0$ ⟶ root: g^0

$f(x) - 2338775057 = 0$, for $x = g^{1022}$ ⟶ root: g^{1022}

$f(g^2x) - f(gx)^2 - f(x)^2 = 0$, for $x = g^i$, $0 \le i \le 1020$ ⟶ roots: $\{g^i \mid 0 \le i \le 1020\}$

Step II - From Constraints to Roots

$f(x) - 1 = 0$, for $x = g^0$ ⟶ root: g^0

$f(x) - 2338775057 = 0$, for $x = g^{1022}$ ⟶ root: g^{1022}

$f(g^2x) - f(gx)^2 - f(x)^2 = 0$, for $x = g^i$, $0 \le i \le 1020$ ⟶ roots: $\{g^i \mid 0 \le i \le 1020\}$

g^0 *is a root of* $f(x) - 1$

g^{1022} *is a root of* $f(x) - 2338775057$

$\{g^i \mid 0 \le i \le 1020\}$ *are roots of* $f(g^2x) - f(gx)^2 - f(x)^2$

⟶ Original statement is true

Step III - From Roots to Rational Functions

Thm: z is a root of $p(x)$ ⇔ $(x - z)$ divides $p(x)$

Def: $(x - z)$ divides $p(x)$ if $p(x) / (x - z)$ is a polynomial

Polynomial

$$\frac{x^2 - 3x + 2}{x - 2} = \frac{(x - 2)(x - 1)}{x - 2} = x - 1$$

2 is a root

Not polynomial

$$\frac{x^2 - 7x + 6}{x - 2} = \frac{(x - 1)(x - 6)}{x - 2}$$

2 is NOT a root

Step III - From Roots to Rational Functions

Thm: z is a root of $p(x)$ ⇔ $(x - z)$ divides $p(x)$

Def: $(x - z)$ divides $p(x)$ if $p(x) / (x - z)$ is a polynomial

g^0 is a root of $f(x) - 1$ ⟶ $\frac{f(x) - 1}{x - g^0}$ is a polynomial

g^{1022} is a root of $f(x) - 2338775057$ ⟶ $\frac{f(x) - 2338775057}{x - g^{1022}}$ is a polynomial

Step III - From Roots to Rational Functions

$\{g^i \mid 0 \le i \le 1020\}$ are roots of $f(g^2x) - f(gx)^2 - f(x)^2$ ----►

$$\frac{f(g^2x) - f(gx)^2 - f(x)^2}{\prod_{i=0}^{1020}(x - g^i)}$$

is a polynomial

$$\prod_{i=0}^{1023}(x - g^i) = x^{1024} - 1$$

fix:

$$\frac{f(g^2x) - f(gx)^2 - f(x)^2}{(x^{1024} - 1)/\left[(x - g^{1021})(x - g^{1022})(x - g^{1023})\right]}$$

3 Rational Functions

$$p_0(x) = \frac{f(x) - 1}{x - g^0}$$

$$p_1(x) = \frac{f(x) - 2338775057}{x - g^{1022}}$$

$$p_2(x) = \frac{f(g^2x) - f(gx)^2 - f(x)^2}{(x^{1024} - 1)/\left[(x - g^{1021})(x - g^{1022})(x - g^{1023})\right]}$$

If $p_0(x)$, $p_1(x)$, $p_2(x)$ are polynomials ⟶ Original statement is true!

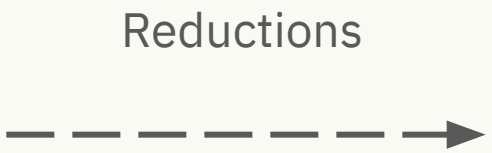
Where are We Heading?

Constraints on $\{a_n\}$:

$a_0 = 1$

$a_{1022} = 2338775057$

$a_{n+2} = a_{n+1}^2 + a_n^2$



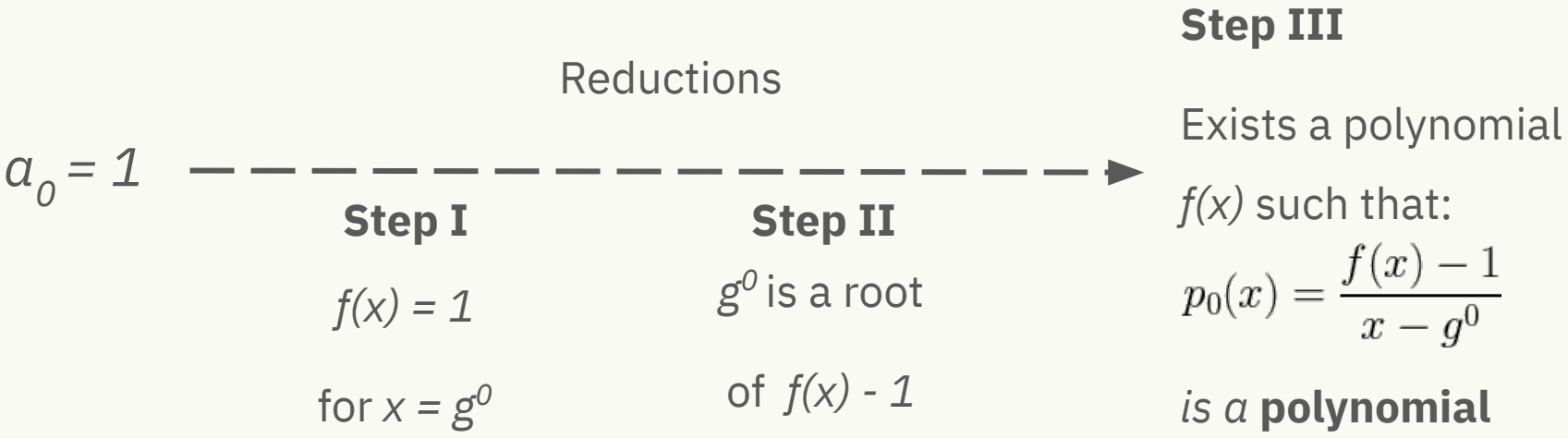
Exists a polynomial $f(x)$

such that:

3 rational functions

$p_0(x)$, $p_1(x)$, $p_2(x)$ are **polynomials**

Reduction Overview - First Constraint



3 Rational Functions

$$p_0(x) = \frac{f(x) - 1}{x - g^0}$$

$$p_1(x) = \frac{f(x) - 2338775057}{x - g^{1022}}$$

$$p_2(x) = \frac{f(g^2x) - f(gx)^2 - f(x)^2}{(x^{1024} - 1)/\left[(x - g^{1021})(x - g^{1022})(x - g^{1023})\right]}$$

If $p_0(x)$, $p_1(x)$, $p_2(x)$ are polynomials ⟶ Original statement is true!

Combining $p_i(x)$'s

Random linear combination:

Composition Polynomial

$$CP = \alpha_0 \cdot p_0(x) + \alpha_1 \cdot p_1(x) + \alpha_2 \cdot p_2(x)$$

With high probability:

CP is a polynomial ⇔ all p_i's are polynomials

Commiting on *CP* with Merkle Tree

STARKWARE STARK 101

Stark 101: Part 3

FRI Commitment

Recap

Goal : prove a statement on FibonacciSq

- Trace in 1023 points
- Create *Trace* polynomial (Lagrange interpolation)
- Evaluate and commit on a larger domain

Recap

- 3 constraints on *f(x):*

$$f(x) - 1 = 0 \text{ , for } x = 1$$

...

- 3 rational functions from the constraints:

$$p_0(x) = \frac{f(x) - 1}{x - g^0}$$

...

Recap

- **C**omposition **P**olynomial:

$$CP(x) = \alpha_0 \cdot p_0(x) + \alpha_1 \cdot p_1(x) + \alpha_2 \cdot p_2(x)$$

- Prover commits on CP
- Goal - show that CP is a **polynomial**
- CP is a **polynomial** $\longrightarrow$ All constraints satisfied

What Will We Do?

Goal:

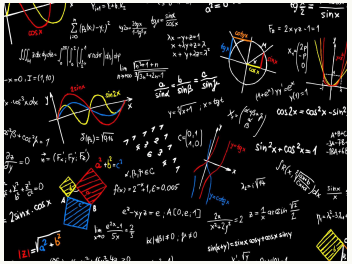
Prove that CP is a **polynomial** ✗

Instead:

Prove that CP is **close** to a **polynomial** of **low degree**

What is close?

What is low degree?

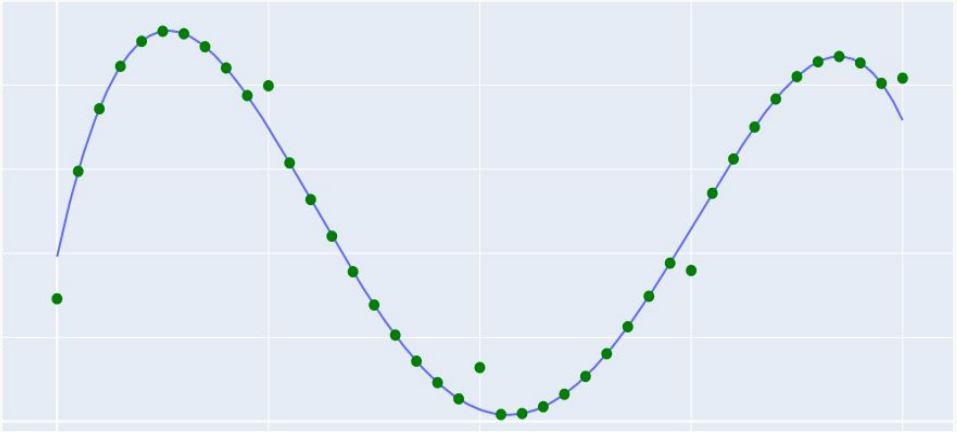


Proximity to Polynomials

Distance (def):

Distance between a function ***f**: D → F to a polynomial **p**:*

D(f,p) := # points *x∈D* such that *f(x)≠p(x)*



*D(**f**, **p**) = 5*

Proximity to Polynomials

Distance (def):

Distance between a function ***f**: $D \to F$ to a polynomial **p**:*

$D(f,p) := \#$ points $x \in D$ such that $f(x) \neq p(x)$

Proximity

A function $f: D \to F$ is ***close** to a* polynomial p if: $D(f,p)$ is **small**

What Will We Do? - Reminder

Goal:

Prove that CP is **close** to a **polynomial** of **low degree**

How?


Trust me,
the commitment is close to
a low degree polynomial
gifs.com

FRI

Fast Reed-Solomon Interactive Oracle Proofs of Proximity

By Ben-Sasson, E., Bentov, I., Horesh, Y., & Riabzev, M.

https://eccc.weizmann.ac.il/report/2017/134/

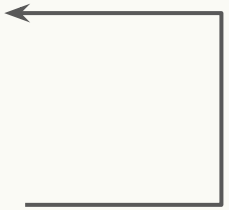
FRI - Goal

Prover convinces verifier:

> **“The commitment is close to a low degree polynomial“**

FRI - The Protocol

- Receive random β
- Apply the FRI operator
- Commit
- Lastly the prover sends the result



Do it repeatedly

FRI

- FRI operator - motivation
- FRI steps overview
- Deep into the FRI operator

FRI Operator

FRI Operator

Goal:

Prove that a function is close to a polynomial of a degree < ***D***

Applying the FRI operator

New Goal:

Prove that a **new** function is close to a **new** polynomial

Half of the domain size

Degree < ***D/2***

FRI Operator - Example
Before applying FRI operator

- Prove:

  A function is close to a polynomial of a degree < **1024**

  where domain size = **8192**

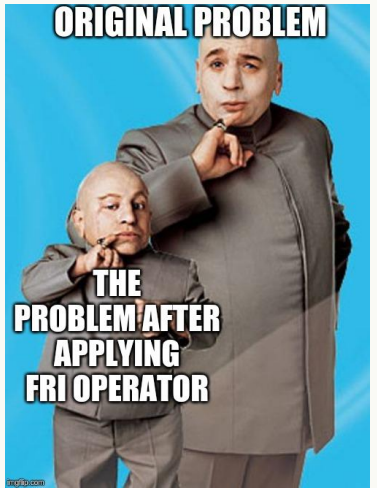
FRI Operator - Example

~~Before~~ After applying FRI operator

- Prove:

  A function is close to a polynomial of a degree < ~~1024~~ 512

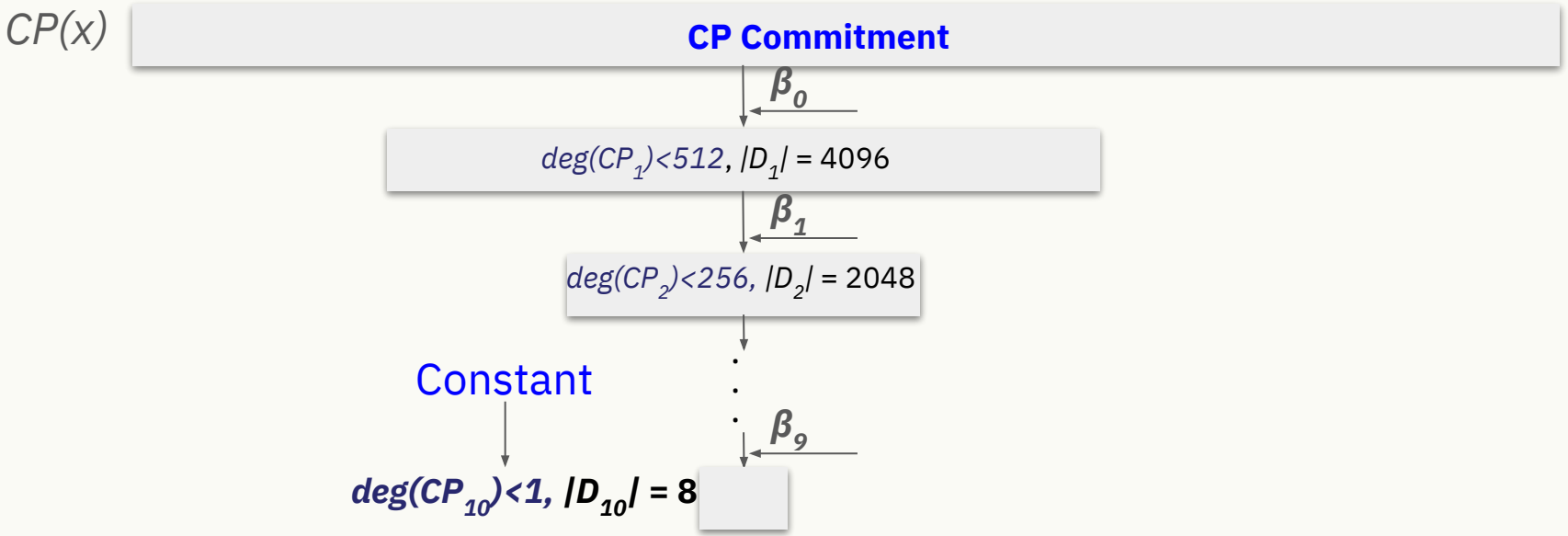
  where domain size = ~~8192~~ 4096



FRI Steps Overview

FRI Steps Overview

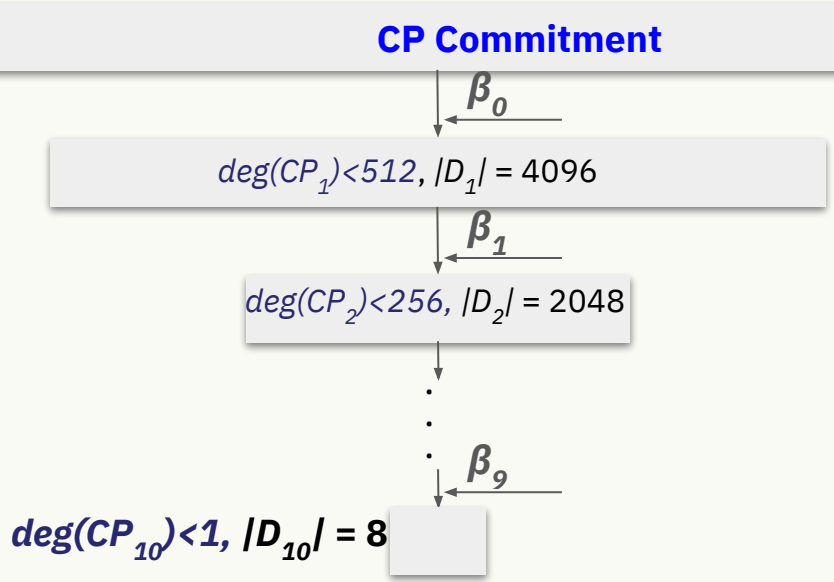
Showing that *deg(CP)<1024, |D|=8192*



FRI Steps Overview

Showing that *deg(CP)<1024, |D|=8192*

CP(x)



Deep Into the FRI Operator

STARKWARE

FRI Operator - How Does it Work?

- Split to even and odd powers

$$P_0(x) = g(x^2) + xh(x^2)$$

- Get a random β

- Consider the new function:

$$P_1(y) = g(y) + \beta h(y)$$

- Example:

$$P_0(x) = 5x^5+3x^4+7x^3+2x^2+x+3$$

$g(x^2)$ $3x^4$ $2x^2$ 3

$xh(x^2)$ $5x^5$ $7x^3$ x

FRI Operator - How Does it Work?

- Split to even and odd powers

$$P_0(x) = g(x^2) + xh(x^2)$$

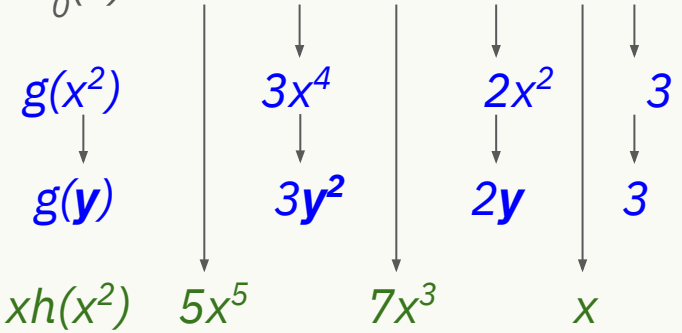
- Get a random β

- Consider the new function:

$$P_1(y) = \mathbf{g(y)} + \beta h(y)$$

- Example:

$$P_0(x) = 5x^5+3x^4+7x^3+2x^2+x+3$$



FRI Operator - How Does it Work?

- Split to even and odd powers

$$P_0(x) = g(x^2) + xh(x^2)$$

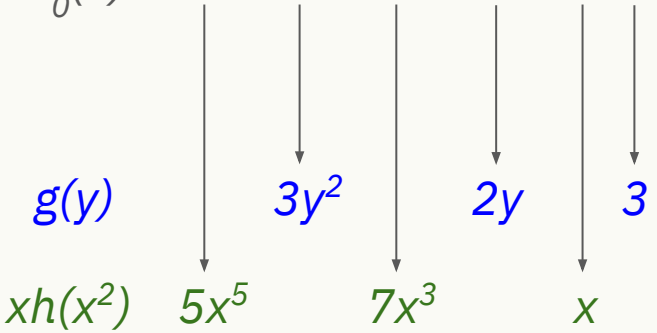
- Get a random β

- Consider the new function:

$$P_1(y) = g(y) + \beta h(y)$$

- Example:

$$P_0(x) = 5x^5+3x^4+7x^3+2x^2+x+3$$



FRI Operator - How Does it Work?

- Split to even and odd powers

$$P_0(x) = g(x^2) + xh(x^2)$$

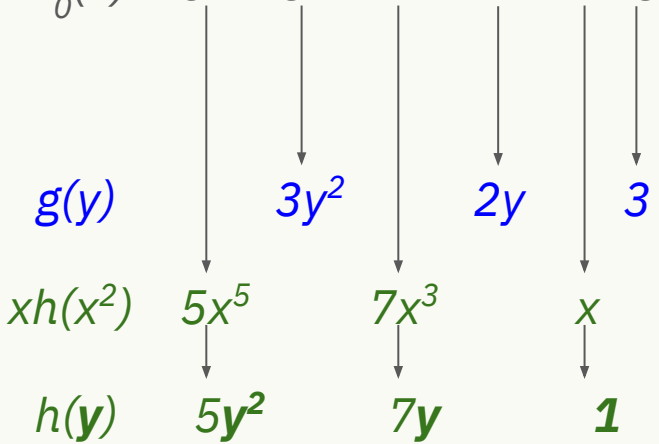
- Get a random β

- Consider the new function:

$$P_1(y) = g(y) + \beta \mathbf{h(y)}$$

- Example:

$P_0(x) = 5x^5+3x^4+7x^3+2x^2+x+3$



FRI Operator - How Does it Work?

- Split to even and odd powers

$$P_0(x) = g(x^2) + xh(x^2)$$

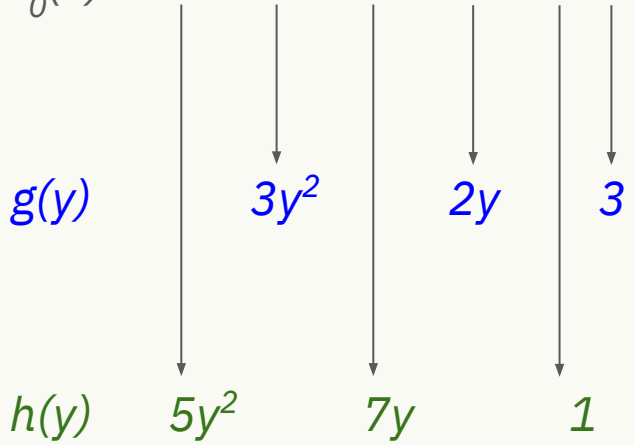
- Get a random β

- Consider the new function:

$$P_1(y) = g(y) + \beta h(y)$$

- Example:

$$P_0(x) = 5x^5+3x^4+7x^3+2x^2+x+3$$



FRI Operator - How Does it Work?

- Split to even and odd powers

$$P_0(x) = g(x^2) + xh(x^2)$$

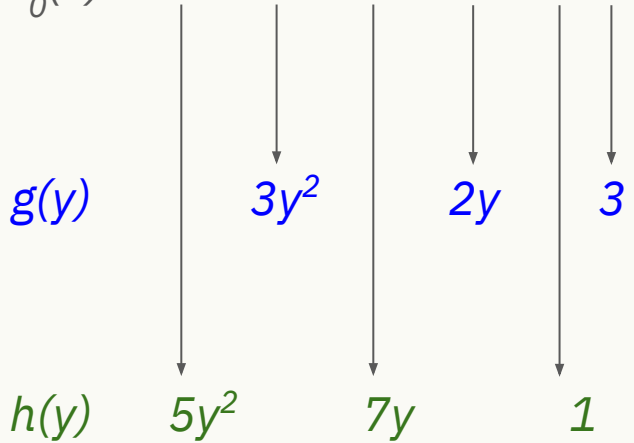
- Get a random β

- Consider the new function:

$$P_1(y) = g(y) + \beta h(y)$$

- Example:

$$P_0(x) = 5x^5+3x^4+7x^3+2x^2+x+3$$

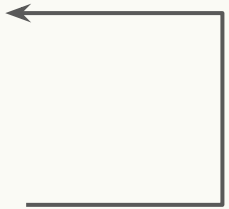


- $P_1(y) = 3y^2 +2y+3+\beta(5y^2+7y +1)$

$$= (3+5\beta)y^2+(2+7\beta)y+3+\beta$$

FRI - The Protocol - Reminder

- Receive random β
- Apply the FRI operator
- Commit
- Lastly the prover sends the result

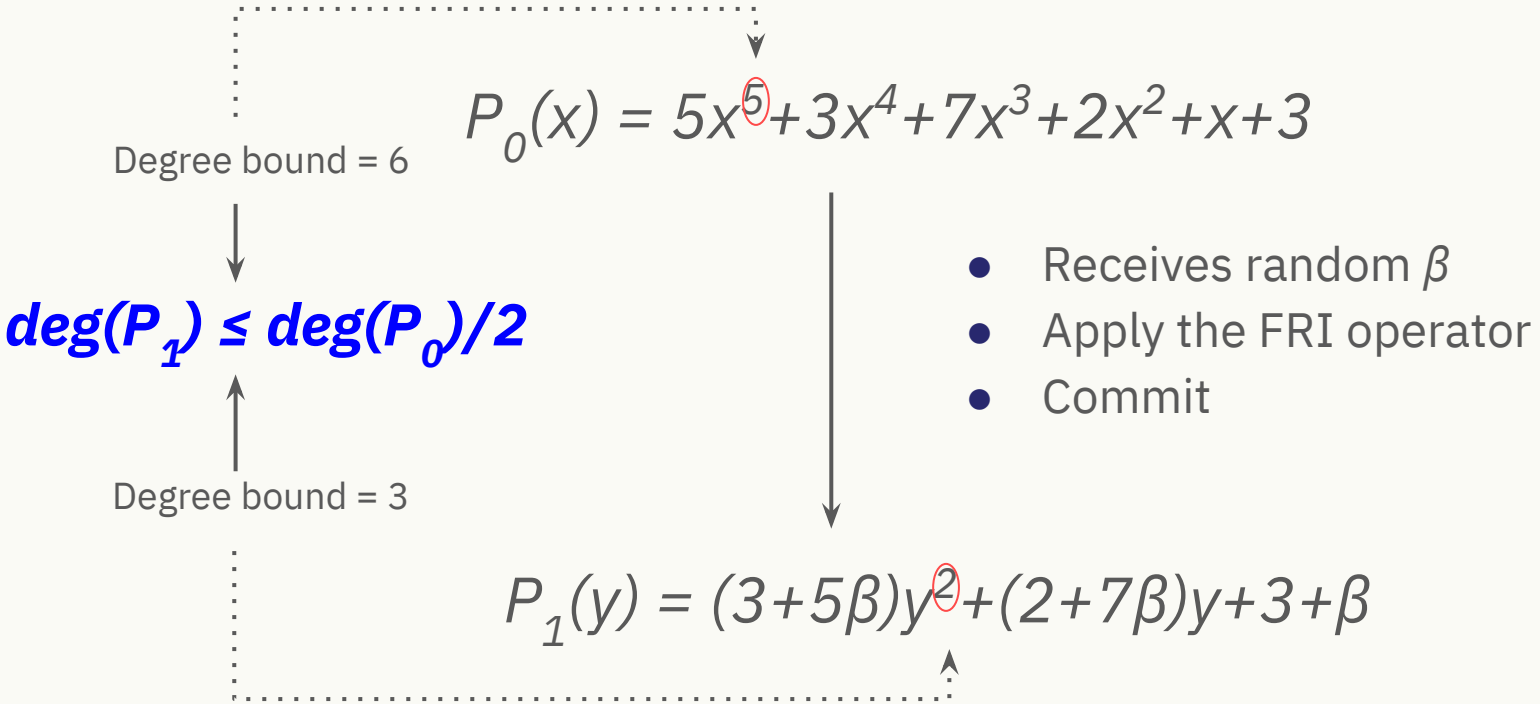


Do it repeatedly

deg(poly) < 1
where
domain size is 8

constant

FRI - The Protocol - A Single Step



STARKWARE STARK 101

Stark 101: Part 4

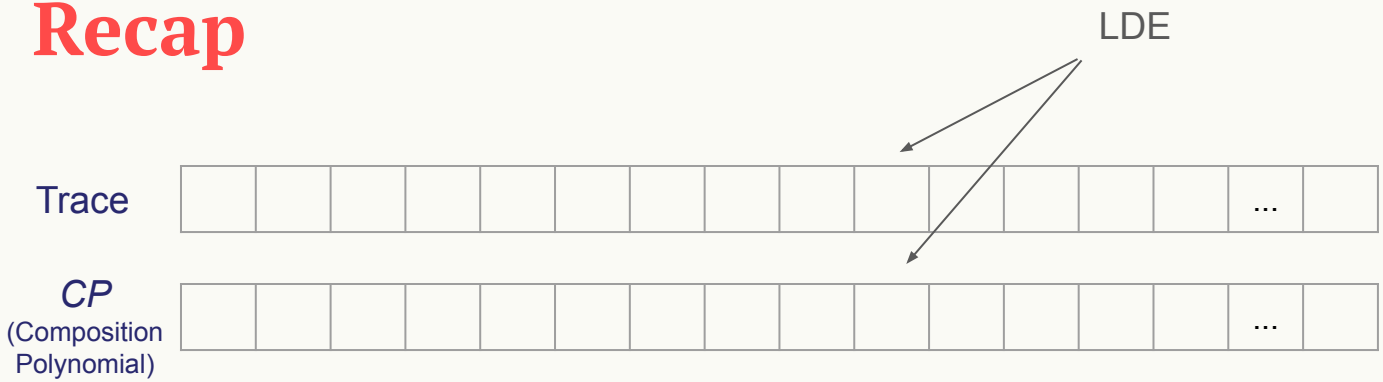
The Proof

Recap

Commitment

Trace Root

Recap



Commitment

Trace Root

CP Root

Recap

Commitment

Trace Root

CP Root

CP_1 Root

CP_2 Root

:

CP_{10} Root

The Entire Proof

Commitment

- Decommitment (Convincing)

How Can the Prover Convince the Verifier?

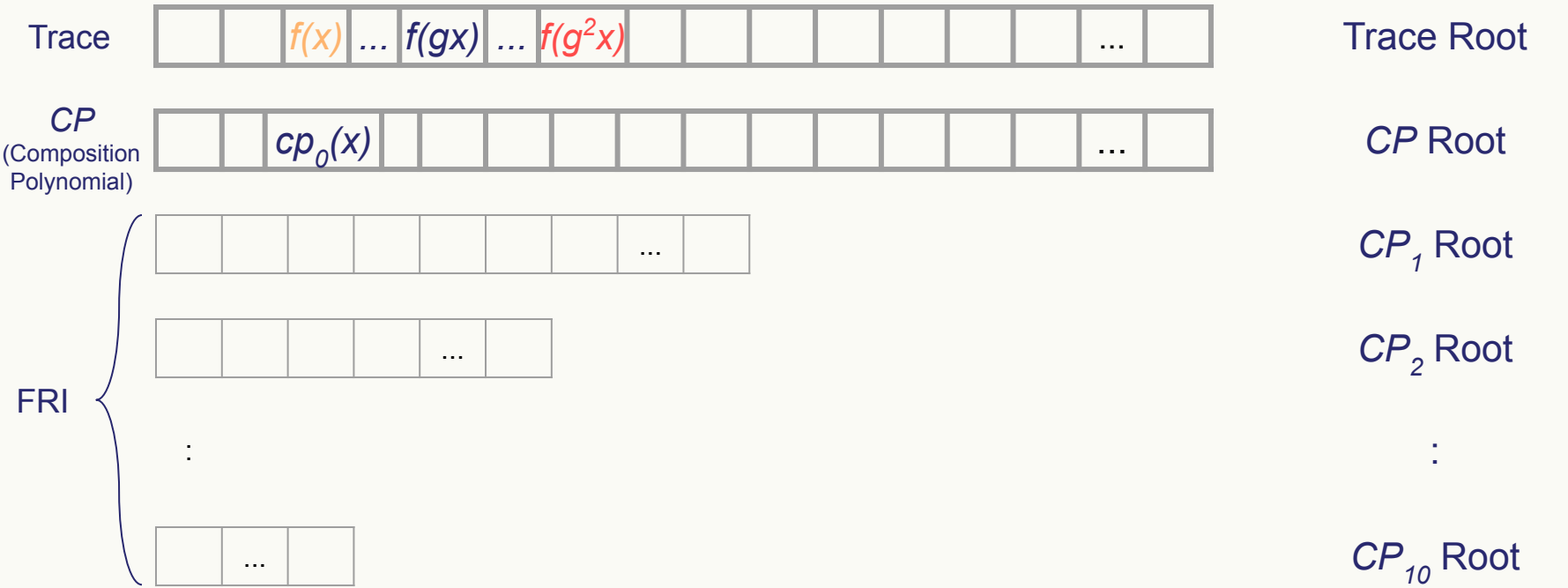
How Can the Prover Convince the Verifier?

Verifier: Sends q random elements

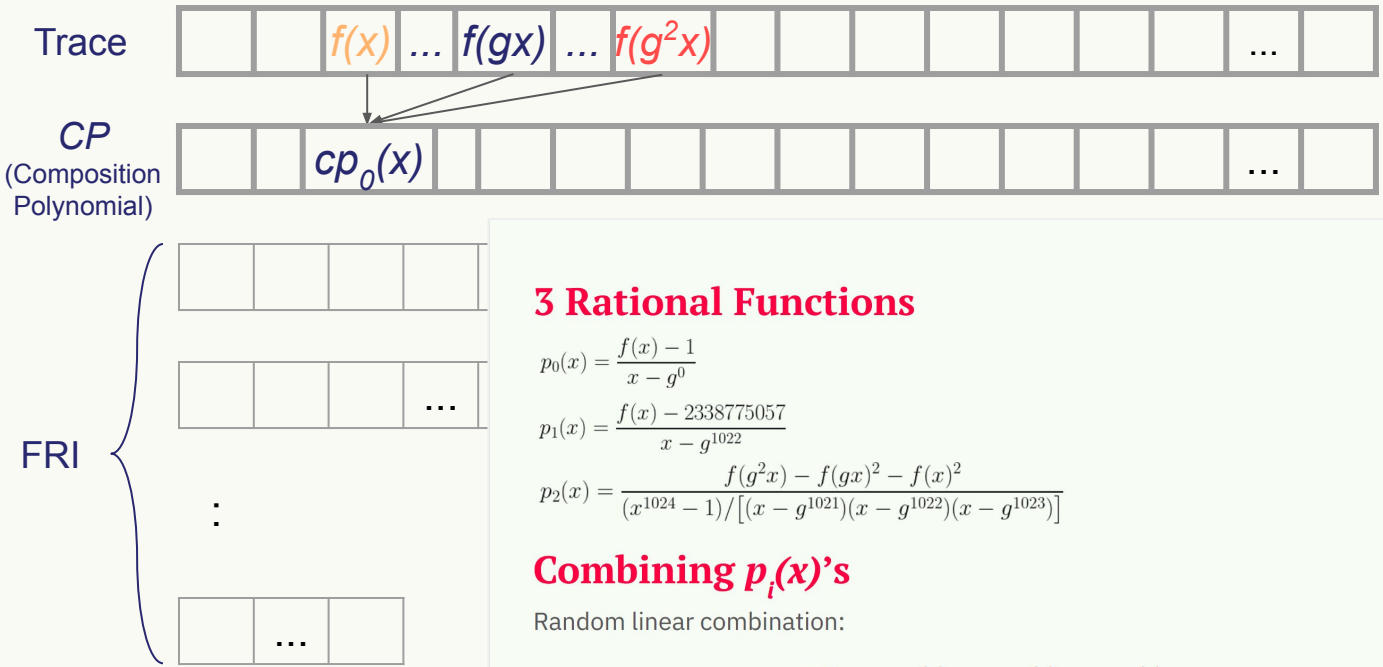
Prover: Provides a validation data for each

What is a validation data?

Trace -> *CP*


Commitment
Trace
f(x)
...
f(gx)
...
f(g^2x)
...
Trace Root
CP
(Composition
Polynomial)
cp_0(x)
...
CP Root
FRI
...
CP_1 Root
...
CP_2 Root
:
:
...
CP_10 Root

Trace -> *CP*



3 Rational Functions

$$p_0(x) = \frac{f(x) - 1}{x - g^0}$$

$$p_1(x) = \frac{f(x) - 2338775057}{x - g^{1022}}$$

$$p_2(x) = \frac{f(g^2x) - f(gx)^2 - f(x)^2}{(x^{1024} - 1)/\left[(x - g^{1021})(x - g^{1022})(x - g^{1023})\right]}$$

Combining $p_i(x)$'s

Random linear combination:

$$CP = \alpha_0 \cdot p_0(x) + \alpha_1 \cdot p_1(x) + \alpha_2 \cdot p_2(x)$$

STARKWARE STARK 101

Commitment

Trace Root

CP Root

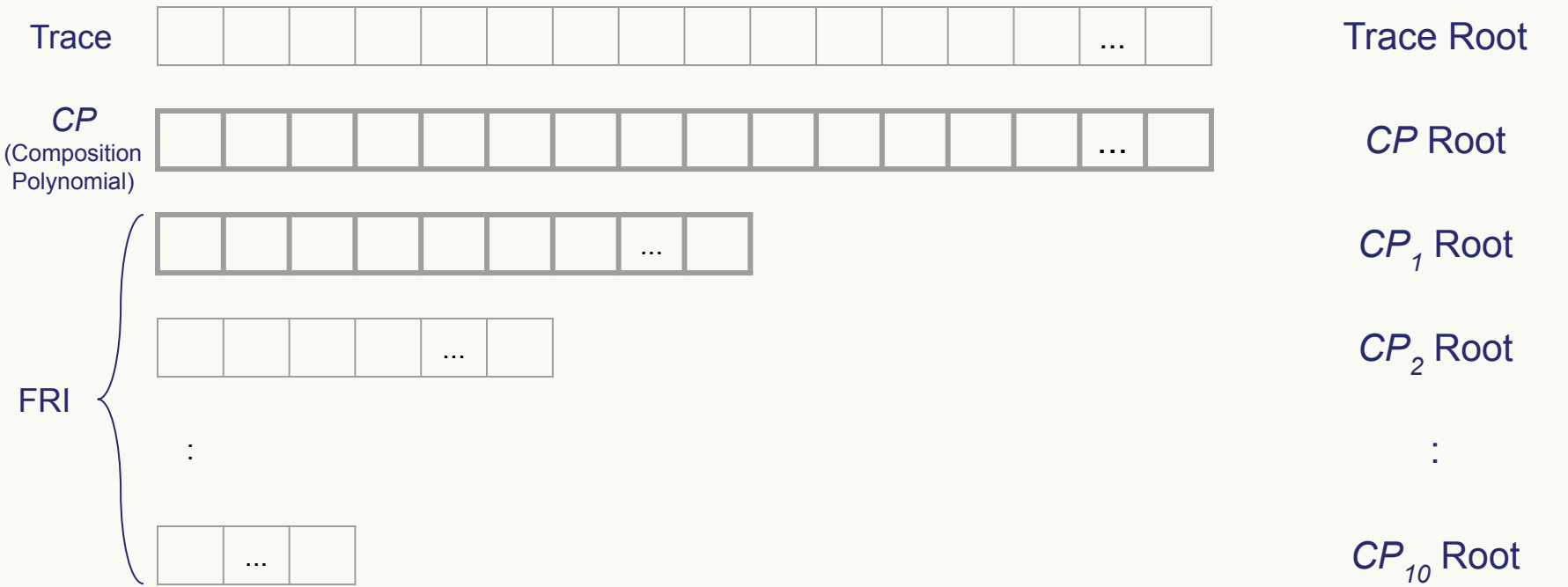
CP_1 Root

CP_2 Root

:

CP_{10} Root

FRI Step


Commitment
Trace
Trace Root
CP
(Composition
Polynomial)
CP Root
FRI
CP1 Root
CP2 Root
CP10 Root

FRI Step

$$\begin{cases} CP_i(x) = g(x^2) + xh(x^2) \\ CP_i(-x) = g(x^2) - xh(x^2) \end{cases} \longrightarrow \begin{cases} g(x^2) = \dfrac{CP_i(x) - CP_i(-x)}{2} \\ h(x^2) = \dfrac{CP_i(x) - CP_i(-x)}{2x} \end{cases}$$

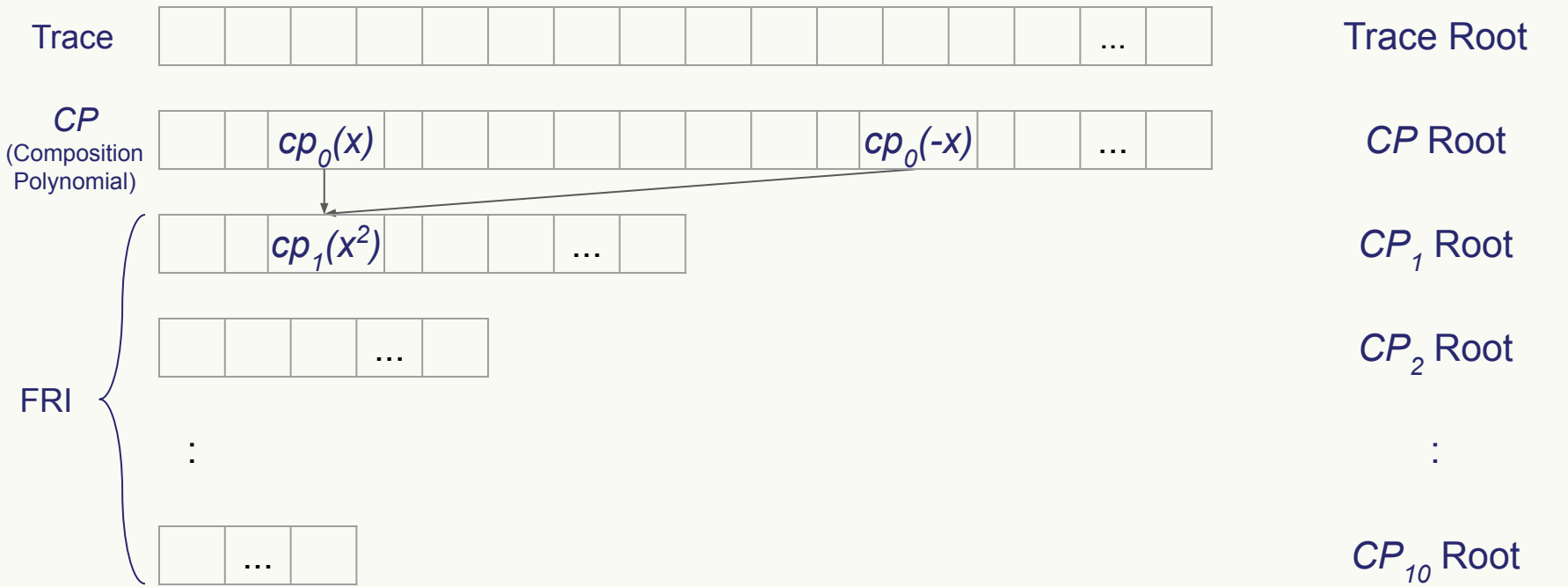
Reminder:

$$CP_{i+1}(x^2) = g(x^2) + \beta h(x^2)$$

Bottom line:

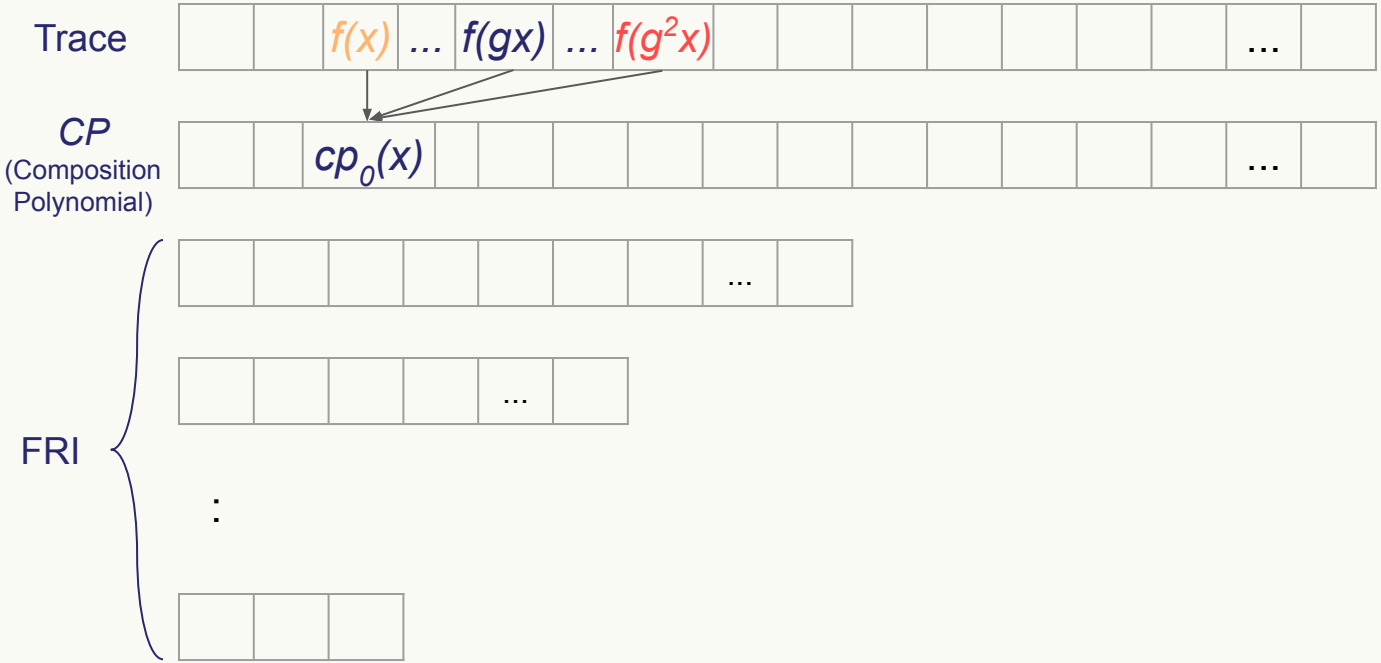
To compute $CP_{i+1}(x^2)$ *we only need* $CP_i(x)$ *and* $CP_i(-x)$

FRI Step


Commitment
Trace
Trace Root
CP
(Composition
Polynomial)
$cp_0(x)$
$cp_0(-x)$
CP Root
$cp_1(x^2)$
CP_1 Root
FRI
CP_2 Root
:
:
CP_{10} Root

Decommitment Phase (for query x)

Decommitment Phase (for query x)



Decommitment

$f(x)$ + path
$f(gx)$ + path
$f(g^2x)$ + path
$cp_0(x)$ + path

Decommitment Phase (for query x)

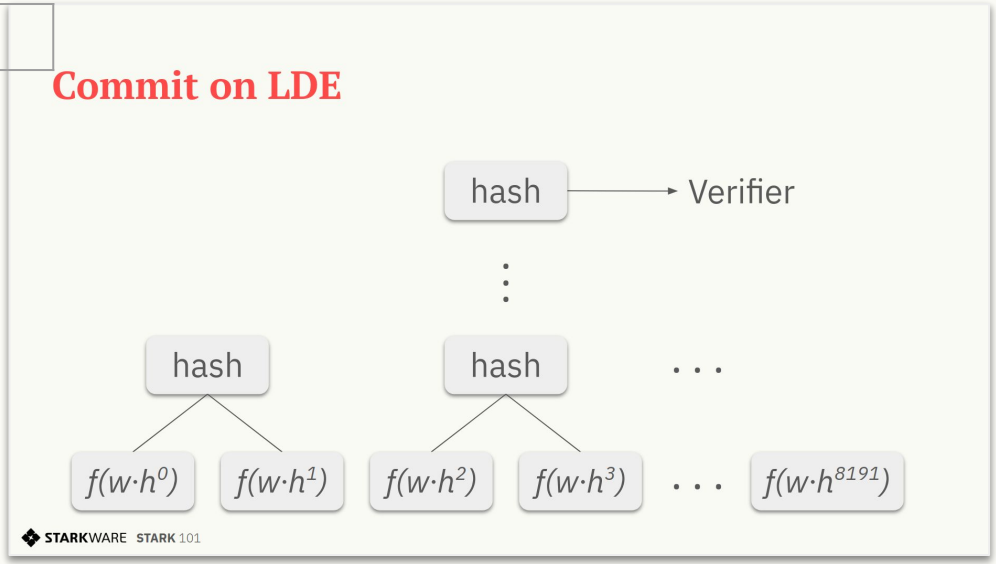
Decommitment

$f(x)$ + path
$f(gx)$ + path
$f(g^2x)$ + path
$cp_0(x)$ + path

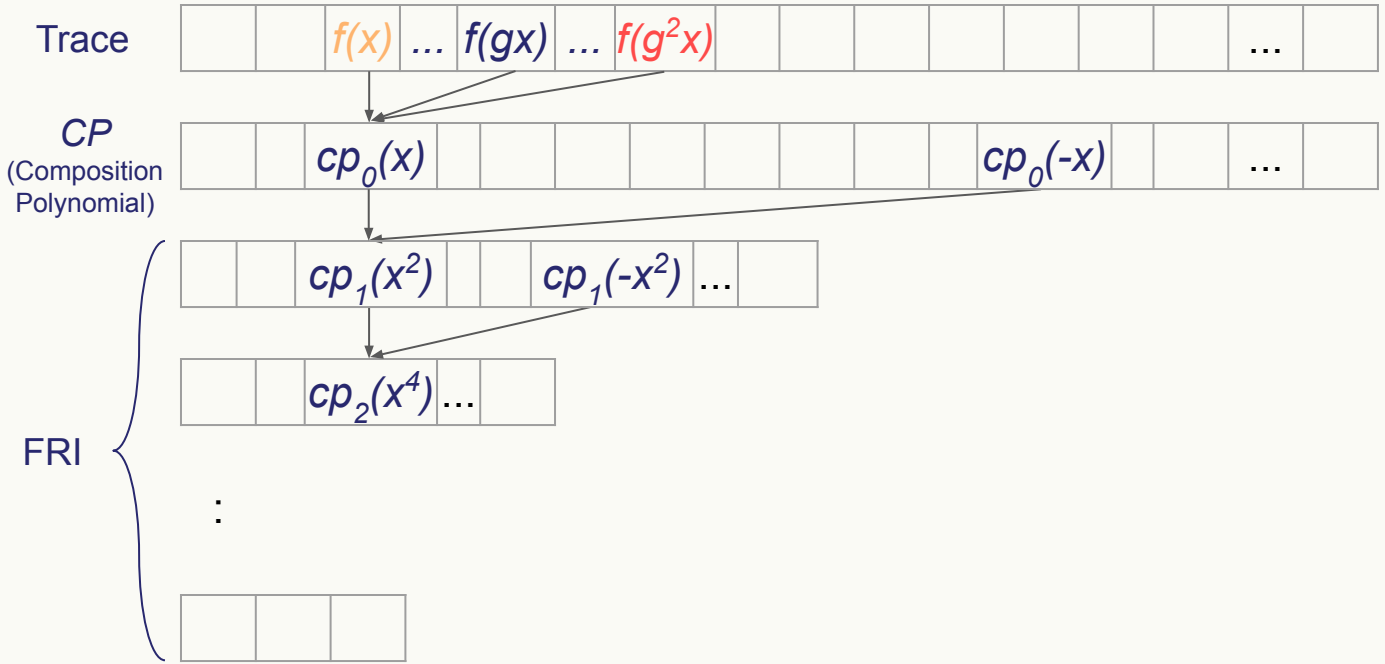
Trace

CP (Composition Polynomial)

FRI



Decommitment Phase (for query x)



Decommitment

$f(x)$ + path
$f(gx)$ + path
$f(g^2x)$ + path
$cp_0(x)$ + path
$cp_0(-x)$ + path

$cp_1(x^2)$ + path
$cp_1(-x^2)$ + path

$cp_2(x^4)$ + path
$cp_2(-x^4)$ + path

:

$cp_{10}(x^{1024})$ + path

The Entire Proof

- Commitment

- Decommitment
  - Get q random elements
  - Provide a validation data for each

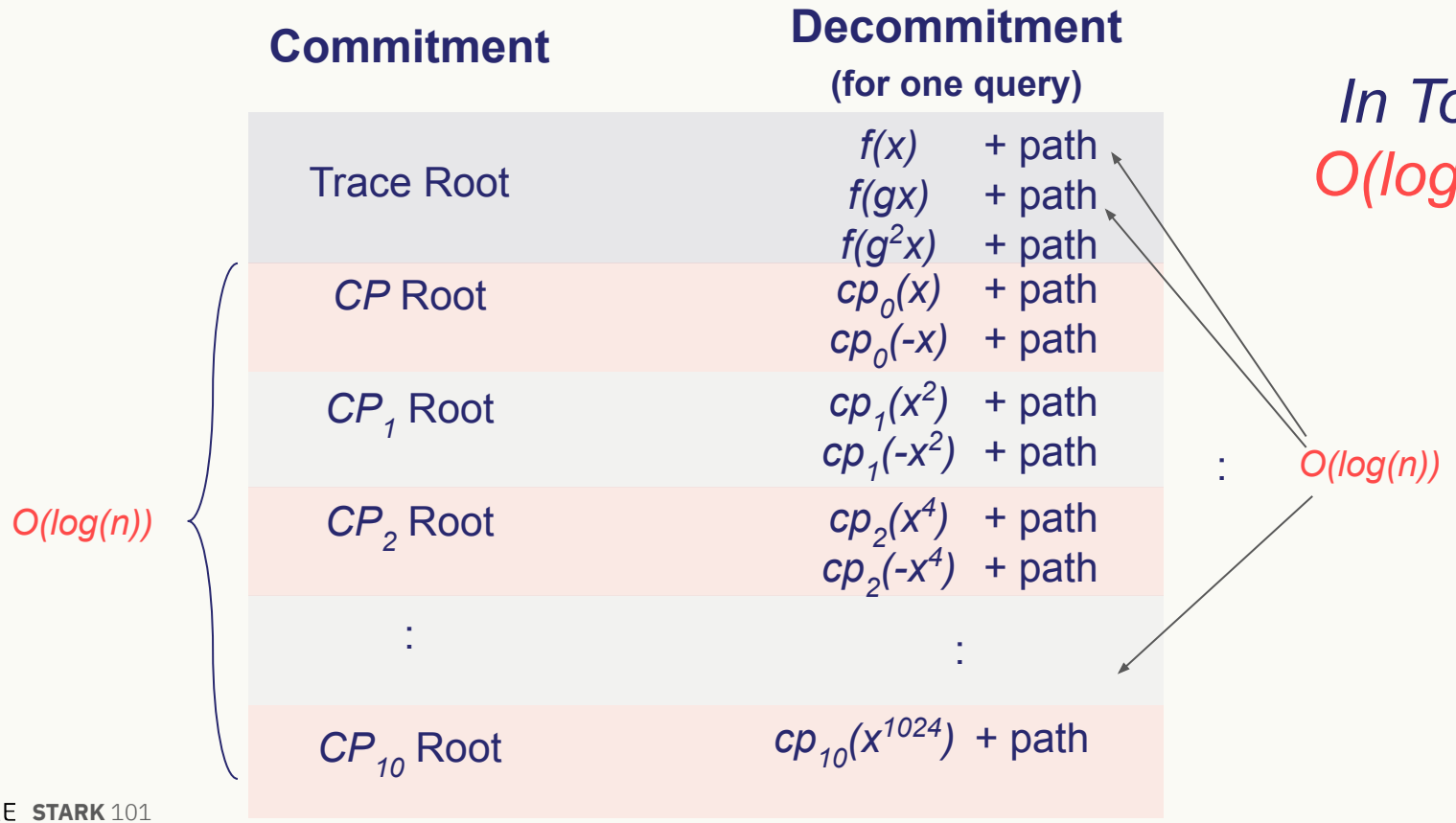
Proof Length

Commitment	Decommitment (for one query)
Trace Root	$f(x)$ + path $f(gx)$ + path $f(g^2x)$ + path
CP Root	$cp_0(x)$ + path $cp_0(-x)$ + path
CP_1 Root	$cp_1(x^2)$ + path $cp_1(-x^2)$ + path
CP_2 Root	$cp_2(x^4)$ + path $cp_2(-x^4)$ + path
:	:
CP_{10} Root	$cp_{10}(x^{1024})$ + path

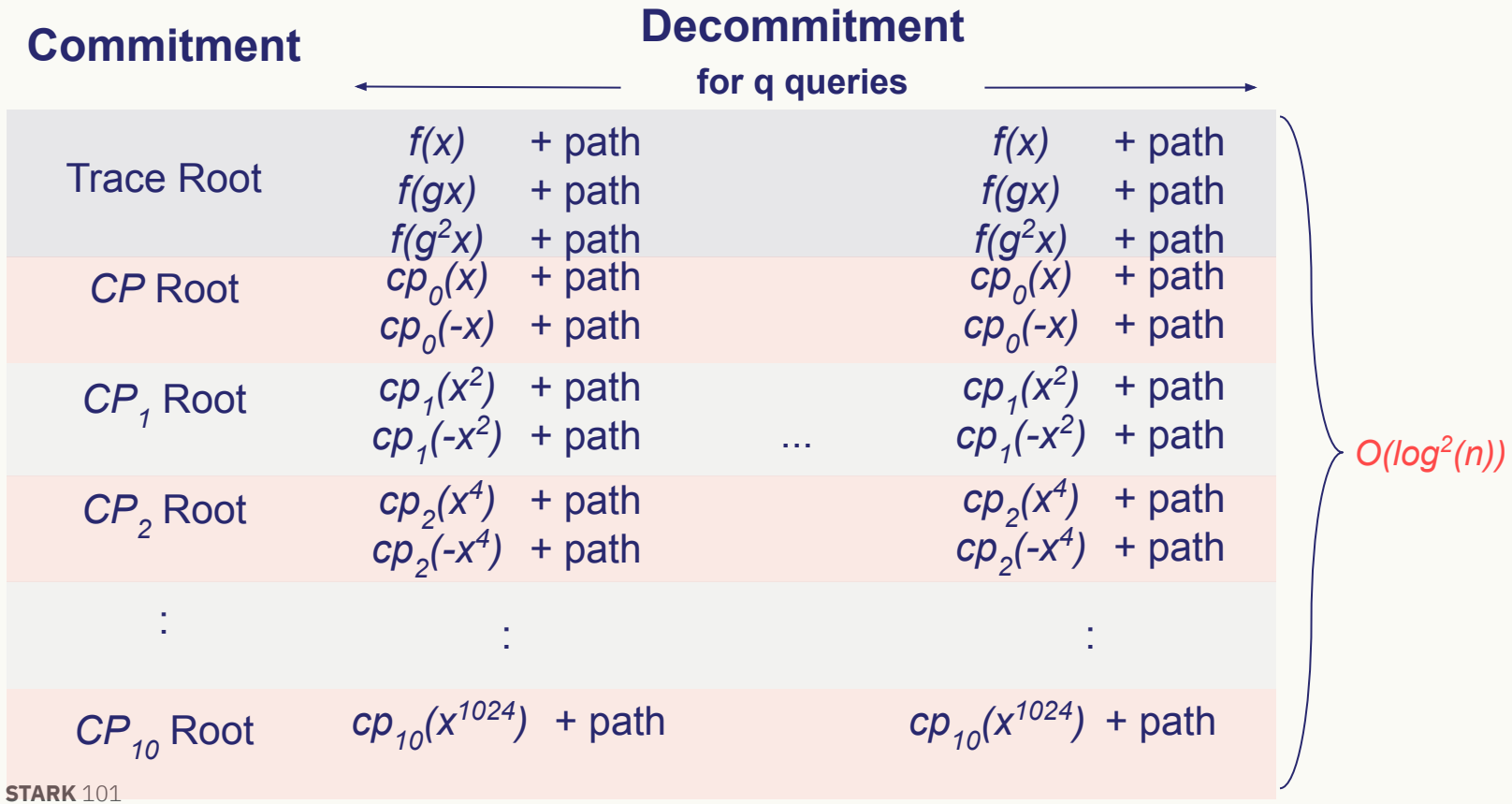
O(log(n))

n = trace length (1023)

Proof Length



Proof Length



Thank you