Common Tools in Succinct Proofs

Guillermo Angeris   Alex Evans

Session 2, SPLA Study Club

Outline

Quick recap

- Last session was mostly pure math
- This one, we will start doing real(-ish!) stuff

Quick recap

- Last session was mostly pure math
- This one, we will start doing real(-ish!) stuff
- Let's put some of the previous tools to the test!

High level overview

- We're going to start by discussing *models*
- Then progress onto specific tools
- Will mostly deal with 'exact' checks
- (Leave 'sparse' checks for homework/paper)

In other words...

In other words...

Let's build up the toolbox!

Outline

Why does randomness help?

- Want to certify that some given vector $x \in \mathbf{F}^n$ is *sparse*
- Say check that $\leq 10\%$ of entries are nonzero

Why does randomness help?

- Want to certify that some given vector $x \in \mathbf{F}^n$ is *sparse*
- Say check that $\leq 10\%$ of entries are nonzero
- Can start checking until we see 90% of entries are nonzero
- If the vector is large, this is not great

Randomness (continued)

- Checking every element requires $\sim n$ checks
- If $n = 2^{20}$, that's a lot of queries

Randomness (continued)

- Checking every element requires $\sim n$ checks
- If $n = 2^{20}$, that's a lot of queries
- If instead we care that $\leq 10\%$ of entries are nonzero
- Can be certain (up to 2^{-100} probability) with < 700 queries!

Randomness (continued)

- Checking every element requires $\sim n$ checks
- If $n = 2^{20}$, that's a lot of queries
- If instead we care that $\leq 10\%$ of entries are nonzero
- Can be certain (up to 2^{-100} probability) with < 700 queries!
- See homework :)

Outline

What are models?

- *Models* are how we ‘encapsulate’ interactions
- Along with cryptographic tools needed

What are models?

- *Models* are how we ‘encapsulate’ interactions
- Along with cryptographic tools needed
- We will discuss two models of interaction

Direct access model

- The first is the *direct model*

Direct access model

- The first is the *direct model*
- There exists a vector $x \in \mathbf{F}^n$
- We would like to verify some claim Q about x
- Can query entries of x (*e.g.*, can query x_i)

Coding model

- The second is the *coding model*

Coding model

- The second is the *coding model*
- We have a (known) code matrix $G \in \mathbf{F}^{m \times n}$
- There exists some message $x \in \mathbf{F}^n$
- Would like to verify some claim Q about x
- Can query individual *symbols* of the encoded message Gx

Truthfulness and commitments

- By assumption, the models require truthfulness
- In general, this is achieved via cryptographic techniques
- Ex.: the direct model can be achieved via a Merkle tree
- We elide this here! (But it is fascinating and worth studying)

Outline

Exact check model

- Exact checks work in the coding model
- Very hard in the direct access model

Exact check model

- Exact checks work in the coding model
- Very hard in the direct access model
- See homework exercises!

Set up

- For the remainder of section: fix a code matrix $G \in \mathbf{F}^{m \times n}$
- Code has distance $d > 0$
- And r will be uniformly drawn from $\{1, \ldots, m\}$
- (Think of r as drawing a uniformly random row of G)

Zero check

- Let's start with the simplest check
- Given some $x \in \mathbf{F}^n$, would like to check $x = 0$

Zero check

- Let's start with the simplest check
- Given some $x \in \mathbf{F}^n$, would like to check $x = 0$
- Randomly sample r and then check if $(Gx)_r = 0$

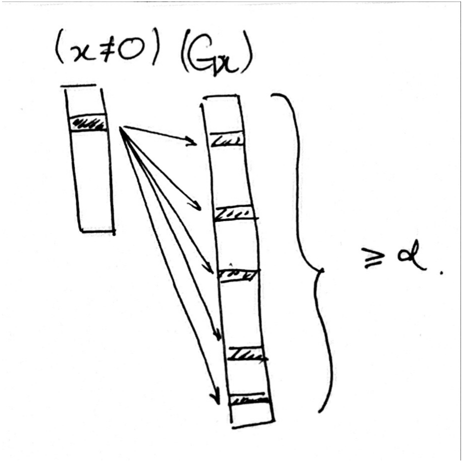
Zero check

- Let's start with the simplest check
- Given some $x \in \mathbf{F}^n$, would like to check $x = 0$
- Randomly sample r and then check if $(Gx)_r = 0$
- We can write this as

$$(Gx)_r = 0 \quad \underset{p}{\Longrightarrow} \quad x = 0$$

- Here, $p \leq 1 - d/m$

Intuition



Discussion

- Equivalent to 1D Schwartz–Zippel lemma
- But for general linear codes!
- (Schwartz–Zippel is the special case of Reed–Solomon)

Matrix zero check

- Instead of a vector x, let's check a matrix $X \in \mathbf{F}^{k \times n}$
- Ask the same question: does $X = 0$?

Matrix zero check

- Instead of a vector x, let's check a matrix $X \in \mathbf{F}^{k \times n}$
- Ask the same question: does $X = 0$?
- Idea: encode each row of X and check if *that* is zero!

Matrix zero check

- Instead of a vector x, let's check a matrix $X \in \mathbf{F}^{k \times n}$
- Ask the same question: does $X = 0$?
- Idea: encode each row of X and check if *that* is zero!
- Let $\tilde{x}_i$ denote ith row, then

$$(G\tilde{x}_1)_r = 0, \ \ldots, \ (G\tilde{x}_n)_r = 0 \quad \underset{p}{\Longrightarrow} \quad X = 0$$

Matrix zero check

- Instead of a vector x, let's check a matrix $X \in \mathbf{F}^{k \times n}$
- Ask the same question: does $X = 0$?
- Idea: encode each row of X and check if *that* is zero!
- Let $\tilde{x}_i$ denote ith row, then

$$(G\tilde{x}_1)_r = 0, \ \ldots, \ (G\tilde{x}_n)_r = 0 \quad \underset{p}{\Longrightarrow} \quad X = 0$$

- What should p be here? (See homework 1, exercise 5!)

Matrix zero check (equiv.)

- Another (identical) way of writing this:
  1. Pick random row r from G
  2. Use row to take linear combination of cols $x_1, \ldots, x_n$ of X
  3. Check if this linear combination is zero

Matrix zero check (equiv.)

- Another (identical) way of writing this:
  1. Pick random row r from G
  2. Use row to take linear combination of cols $x_1, \ldots, x_n$ of X
  3. Check if this linear combination is zero
- We can write this as:

$$G_{r1}x_1 + \cdots + G_{rn}x_n = 0 \quad \underset{p}{\Longrightarrow} \quad X = 0$$

Reduced matrix zero check

- Let's get funky!

Reduced matrix zero check

- Let's get funky!
- The vector zero check goes

$$\text{Vector} \stackrel{?}{=} 0 \quad \rightarrow \quad \text{Scalar} \stackrel{?}{=} 0$$

Reduced matrix zero check

- Let's get funky!
- The vector zero check goes

$$\text{Vector} \stackrel{?}{=} 0 \quad \rightarrow \quad \text{Scalar} \stackrel{?}{=} 0$$

- The matrix zero check goes

$$\text{Matrix} \stackrel{?}{=} 0 \quad \rightarrow \quad \text{Vector} \stackrel{?}{=} 0$$

Reduced matrix zero check

- Let's get funky!
- The vector zero check goes

$$\text{Vector} \stackrel{?}{=} 0 \quad \to \quad \text{Scalar} \stackrel{?}{=} 0$$

- The matrix zero check goes

$$\text{Matrix} \stackrel{?}{=} 0 \quad \to \quad \text{Vector} \stackrel{?}{=} 0$$

- Can we 'put them together'?

Reduced matrix zero check

- Let's get funky!
- The vector zero check goes

$$\text{Vector} \stackrel{?}{=} 0 \quad \to \quad \text{Scalar} \stackrel{?}{=} 0$$

- The matrix zero check goes

$$\text{Matrix} \stackrel{?}{=} 0 \quad \to \quad \text{Vector} \stackrel{?}{=} 0$$

- Can we 'put them together'? (Yes! See homework)

Aside: reduced matrix zero check

- This check is a Schwartz–Zippel generalization
- Indeed, we get the same bounds for Reed–Solomon codes
- (But can get better ones! See §3.1.3 of the paper)
- Shows Schwartz–Zippel is not tight (by a very small factor...)

Vector subspace check

- The ‘final’ boss

Vector subspace check

- The 'final' boss
- Let's start with some matrix $X \in \mathbf{F}^{k \times n}$
- Want to check if all columns of X are in a subspace $V \subseteq \mathbf{F}^k$
- Can we do this cheaply?

Vector subspace check

- Let's do something similar to matrix zero check:
  1. Pick random row r
  2. Use row for linear combination of columns of X
  3. Check if linear combination lies in V

Vector subspace check

- Let's do something similar to matrix zero check:
  1. Pick random row r
  2. Use row for linear combination of columns of X
  3. Check if linear combination lies in V
- We can write this as

$$G_{r1}x_1 + \cdots + G_{rn}x_n \in V \quad \underset{p}{\Longrightarrow} \quad x_i \in V, \quad i = 1, \ldots, n$$

Proof via probabilistic implications

- Let C be parity check matrix for V
- Then

$$G_{r1}x_1 + \cdots + G_{rn}x_n \in V \;\; \text{implies} \;\; C(G_{r1}x_1 + \cdots + G_{rn}x_n) = 0$$

Proof via probabilistic implications

- Let C be parity check matrix for V
- Then

$$G_{r1}x_1 + \cdots + G_{rn}x_n \in V \quad \text{implies} \quad C(G_{r1}x_1 + \cdots + G_{rn}x_n) = 0$$

- But, by linearity (see homework 1, problem 1)

$$0 = C(G_{r1}x_1 + \cdots + G_{rn}x_n) = G_{r1}(Cx_1) + \cdots + G_{rn}(Cx_n)$$

- Note the last quantity!

Proof via probabilistic implications (cont.)

- From before,

$$G_{r1}(Cx_1) + \cdots + G_{rn}(Cx_n) = 0$$

- But, this is just the matrix zero check!

Proof via probabilistic implications (cont.)

- From before,

$$G_{r1}(Cx_1) + \cdots + G_{rn}(Cx_n) = 0$$

- But, this is just the matrix zero check! So,

$$G_{r1}(Cx_1) + \cdots + G_{rn}(Cx_n) = 0 \quad \underset{p}{\Longrightarrow} \quad Cx_1 = \cdots = Cx_n = 0$$

Proof via probabilistic implications (cont.)

- From before,

$$G_{r1}(Cx_1) + \cdots + G_{rn}(Cx_n) = 0$$

- But, this is just the matrix zero check! So,

$$G_{r1}(Cx_1) + \cdots + G_{rn}(Cx_n) = 0 \quad \underset{p}{\Longrightarrow} \quad Cx_1 = \cdots = Cx_n = 0$$

- By definition of parity check matrix then

$$Cx_1 = \cdots = Cx_n = 0 \quad \text{implies} \quad x_1, \ldots, x_n \in V$$

Putting it all together

- We only used one implication with error p (matrix zero check)
- Rest were all ‘standard’ implications (*i.e.*, zero error)
- This means that we can write:

$$G_{r1}x_1 + \cdots + G_{rn}x_n \in V \quad \underset{p}{\Longrightarrow} \quad x_i \in V, \quad i = 1, \ldots, n$$

- Where p is the same as the matrix zero check from before

Outline

Analogues of exact checks

- There is a sparse analogue of each exact check
- We will not explore proofs here (as they are more complicated)
- But check the paper (and the homework!) for some of these

Sparse checks

- We will use two types of sparse checks for next lecture
- One will be in homework (standard sparsity check)
- The other will have a proof outline next lecture

Sparse checks

- We will use two types of sparse checks for next lecture
- One will be in homework (standard sparsity check)
- The other will have a proof outline next lecture
- It's good to be comfortable with notation before that :)

Outline

Summary

- Rewrote a bunch of tools used in many papers!
- Some maybe felt familiar
- But did all of them in very short order

Next lecture

- We will see how to apply these tools to explain FRI
- (As a quasi-'capstone' project)
- But many more applications exist :)