The Sum-Check Protocol and Applications

Jonathan Katz
Google and University of Maryland

Outline

- The sum-check protocol
  - The power of IP (e.g., showing $\#P \subseteq IP$)
- The Goldwasser-Kalai-Rothblum protocol

Mathematical preliminaries

Mathematical preliminaries

Theorem

Let $\mathbb{F}$ be a field, and let $p \in \mathbb{F}[x]$ be a nonzero polynomial of degree $\leq d$. Then p has at most d roots.

Mathematical preliminaries

Theorem

Let $\mathbb{F}$ be a field, and let $p \in \mathbb{F}[x]$ be a nonzero polynomial of degree $\leq d$. Then p has at most d roots.

Theorem

Let $\mathbb{F}$ be a field, and let $p \in \mathbb{F}[x_1, \ldots, x_n]$ be a nonzero polynomial of total degree $\leq d$. Then p has at most $d \cdot |\mathbb{F}|^{n-1}$ roots.

Proof.

By induction, writing $p(x_1, \ldots, x_n) = \sum_{i=0}^{d} x_n^i \cdot p_i(x_1, \ldots, x_{n-1})$ where $p_i \in \mathbb{F}[x_1, \ldots, x_{n-1}]$ has total degree at most $d - i$. □

Mathematical preliminaries

Schwartz–Zippel lemma

Let $\mathbb{F}$ be a field, and let $p \in \mathbb{F}[x_1, \ldots, x_n]$ be a nonzero polynomial of total degree $\leq d$. Then $\Pr_{r_1,\ldots,r_n \leftarrow \mathbb{F}}[p(r_1, \ldots, r_n) = 0] \leq d/|\mathbb{F}|$.

Mathematical preliminaries

Schwartz–Zippel lemma

Let $\mathbb{F}$ be a field, and let $p \in \mathbb{F}[x_1, \ldots, x_n]$ be a nonzero polynomial of total degree $\leq d$. Then $\Pr_{r_1,\ldots,r_n \leftarrow \mathbb{F}}[p(r_1, \ldots, r_n) = 0] \leq d/|\mathbb{F}|$.

Corollary

Let $\mathbb{F}$ be a field, and let $p, p' \in \mathbb{F}[x_1, \ldots, x_n]$ be nonequal polynomials of total degree $\leq d$. Then $\Pr_{r_1,\ldots,r_n \leftarrow \mathbb{F}}[p(r_1, \ldots, r_n) = p'(r_1, \ldots, r_n)] \leq d/|\mathbb{F}|$.

Proof.

If $p \neq p'$ then $p - p'$ is a nonzero polynomial. □

The sum-check protocol

Overview

The prover and verifier have common input $p \in \mathbb{F}[x_1, \ldots, x_n]$

Overview

The prover and verifier have common input $p \in \mathbb{F}[x_1, \ldots, x_n]$

The prover wants to convince the verifier that

$$H_0 = \sum_{x_1 \in \{0,1\}} \cdots \sum_{x_n \in \{0,1\}} p(x_1, \ldots, x_n).$$

Note: the verifier could check this in time $\Omega(2^n)$, but we want a polynomial-time verifier. (For now, think of the prover as all-powerful.)

Sum-check protocol

Sum-check protocol

Common inputs: $p \in \mathbb{F}[x_1, \ldots, x_n]$, sum

$$H_0 := \sum_{x_1 \in \{0,1\}} \cdots \sum_{x_n \in \{0,1\}} p(x_1, \ldots, x_n)$$

1. For $i = 1, \ldots, n$ do:
    1. P sends $p_i(x_i) := \sum_{x_{i+1}} \cdots \sum_{x_n} p(r_1, \ldots, r_{i-1}, x_i, \ldots, x_n)$
    2. V checks the degree of p_i and that $p_i(0) + p_i(1) = H_{i-1}$
    3. V chooses $r_i \leftarrow \mathbb{F}$, sets $H_i := p_i(r_i)$, and sends r_i to P
2. V checks that $H_n = p(r_1, \ldots, r_n)$

Sum-check protocol

Sum-check protocol

Common inputs: $p \in \mathbb{F}[x_1, \ldots, x_n]$, sum

$$H_0 := \sum_{x_1 \in \{0,1\}} \cdots \sum_{x_n \in \{0,1\}} p(x_1, \ldots, x_n)$$

1. For $i = 1, \ldots, n$ do:
   1. P sends $p_i(x_i) := \sum_{x_{i+1}} \cdots \sum_{x_n} p(r_1, \ldots, r_{i-1}, x_i, \ldots, x_n)$
   2. V checks the degree of p_i and that $p_i(0) + p_i(1) = H_{i-1}$
   3. V chooses $r_i \leftarrow \mathbb{F}$, sets $H_i := p_i(r_i)$, and sends r_i to P
2. V checks that $H_n = p(r_1, \ldots, r_n)$

Completeness is clear. . .

Analysis of sum-check protocol

Theorem

Let p be an n-variate polynomial of degree d_i in each variable. Then the sum-check protocol has soundness error $\leq \sum_i d_i/|\mathbb{F}|$.

Proof.

By induction on n...

Analysis of sum-check protocol

Theorem

Let p be an n-variate polynomial of degree d_i in each variable. Then the sum-check protocol has soundness error $\leq \sum_i d_i/|\mathbb{F}|$.

Proof.

By induction on n...

Base case ($n = 1$): Say $H_0 \neq \sum_{x_1 \in \{0,1\}} p(x_1)$

Analysis of sum-check protocol

Theorem

Let p be an n-variate polynomial of degree d_i in each variable. Then the sum-check protocol has soundness error $\leq \sum_i d_i/|\mathbb{F}|$.

Proof.

By induction on n...

Base case ($n = 1$): Say $H_0 \neq \sum_{x_1 \in \{0,1\}} p(x_1)$

- If $p_1 = p$ then $p_1(0) + p_1(1) \neq H_0$ and V rejects

Analysis of sum-check protocol

Theorem

Let p be an n-variate polynomial of degree d_i in each variable. Then the sum-check protocol has soundness error $\leq \sum_i d_i/|\mathbb{F}|$.

Proof.

By induction on n...

Base case ($n = 1$): Say $H_0 \neq \sum_{x_1 \in \{0,1\}} p(x_1)$

- If $p_1 = p$ then $p_1(0) + p_1(1) \neq H_0$ and V rejects
- If $p_1 \neq p$, V accepts with probability $\Pr_{r_1}[p_1(r_1) = p(r_1)] \leq d_1/|\mathbb{F}|$

□

Analysis of sum-check protocol

Theorem

Let p be an n-variate polynomial of degree d_i in each variable. Then the sum-check protocol has soundness error $\leq \sum_i d_i/|\mathbb{F}|$.

Proof.

By induction on n...

Inductive step: Say $H_0 \neq \sum_{x_1 \in \{0,1\}} \cdots \sum_{x_n \in \{0,1\}} p(x_1, \ldots, x_n)$. Let $p_1^*(x_1) = \sum_{x_2 \in \{0,1\}} \cdots \sum_{x_n \in \{0,1\}} p(x_1, \ldots, x_n)$

Analysis of sum-check protocol

Theorem

Let p be an n-variate polynomial of degree d_i in each variable. Then the sum-check protocol has soundness error $\leq \sum_i d_i/|\mathbb{F}|$.

Proof.

By induction on n...

Inductive step: Say $H_0 \neq \sum_{x_1 \in \{0,1\}} \cdots \sum_{x_n \in \{0,1\}} p(x_1, \ldots, x_n)$. Let $p_1^*(x_1) = \sum_{x_2 \in \{0,1\}} \cdots \sum_{x_n \in \{0,1\}} p(x_1, \ldots, x_n)$

- If $p_1 = p_1^*$, then $p_1(0) + p_1(1) \neq H_0$ and V rejects

Analysis of sum-check protocol

Theorem

Let p be an n-variate polynomial of degree d_i in each variable. Then the sum-check protocol has soundness error $\leq \sum_i d_i/|\mathbb{F}|$.

Proof.

By induction on n...

Inductive step: Say $H_0 \neq \sum_{x_1 \in \{0,1\}} \cdots \sum_{x_n \in \{0,1\}} p(x_1, \ldots, x_n)$. Let $p_1^*(x_1) = \sum_{x_2 \in \{0,1\}} \cdots \sum_{x_n \in \{0,1\}} p(x_1, \ldots, x_n)$

- If $p_1 = p_1^*$, then $p_1(0) + p_1(1) \neq H_0$ and V rejects
- If $p_1 \neq p_1^*$, then $\Pr_{r_1}[p_1(r_1) \neq p_1^*(r_1)] \geq 1 - d_1/|\mathbb{F}|$

Analysis of sum-check protocol

Theorem

Let p be an n-variate polynomial of degree d_i in each variable. Then the sum-check protocol has soundness error $\leq \sum_i d_i/|\mathbb{F}|$.

Proof.

By induction on n...

Inductive step: Say $H_0 \neq \sum_{x_1 \in \{0,1\}} \cdots \sum_{x_n \in \{0,1\}} p(x_1, \ldots, x_n)$. Let $p_1^*(x_1) = \sum_{x_2 \in \{0,1\}} \cdots \sum_{x_n \in \{0,1\}} p(x_1, \ldots, x_n)$

- If $p_1 = p_1^*$, then $p_1(0) + p_1(1) \neq H_0$ and V rejects
- If $p_1 \neq p_1^*$, then $\Pr_{r_1}[p_1(r_1) \neq p_1^*(r_1)] \geq 1 - d_1/|\mathbb{F}|$
- When that is the case, $H_1 \neq \sum_{x_2 \in \{0,1\}} \cdots \sum_{x_n \in \{0,1\}} p(r_1, x_2, \ldots, x_n)$ and we can apply the induction hypothesis

□

Complexity of the sum-check protocol

Complexity of the sum-check protocol

Let T be the time to evaluate p

rounds	$O(n)$
communication	$O(\sum_i d_i)$ field elements
verifier time	$O(\sum_i d_i) + T$
prover time	$O(2^n \cdot T \cdot \sum_i d_i)$

Complexity of the sum-check protocol

Let T be the time to evaluate p

rounds	$O(n)$
communication	$O(\sum_i d_i)$ field elements
verifier time	$O(\sum_i d_i) + T$
prover time	$O(2^n \cdot T \cdot \sum_i d_i)$

Notes:

- V does not need to know anything about p (besides bounds on the degrees) until the end of the protocol
- In fact, V does not ever need to know p; it just needs the ability to evaluate p at a (random) point

#P $\subseteq$ IP

P can prove to V how many satisfying assignments a 3CNF formula ϕ has

#P ⊆ IP

P can prove to V how many satisfying assignments a 3CNF formula ϕ has

Step 1: “arithmetize” ϕ by turning it into a polynomial Φ

- $x_i \to x_i$, $\bar{x}_i \to 1 - x_i$
- $\phi_1 \wedge \phi_2 \to \Phi_1 \cdot \Phi_2$
- $\phi_1 \vee \phi_2 \to 1 - (1 - \Phi_1) \cdot (1 - \Phi_2)$

$\#P \subseteq IP$

P can prove to V how many satisfying assignments a 3CNF formula ϕ has

Step 1: "arithmetize" ϕ by turning it into a polynomial Φ

- $x_i \to x_i$, $\bar{x}_i \to 1 - x_i$
- $\phi_1 \wedge \phi_2 \to \Phi_1 \cdot \Phi_2$
- $\phi_1 \vee \phi_2 \to 1 - (1 - \Phi_1) \cdot (1 - \Phi_2)$
- Note $\phi(b_1, \ldots, b_n) = \text{false} \Rightarrow \Phi(b_1, \ldots, b_n) = 0$ and $\phi(b_1, \ldots, b_n) = \text{true} \Rightarrow \Phi(b_1, \ldots, b_n) = 1$

#P ⊆ IP

P can prove to V how many satisfying assignments a 3CNF formula ϕ has

Step 1: "arithmetize" ϕ by turning it into a polynomial Φ

- $x_i \to x_i$, $\bar{x}_i \to 1 - x_i$
- $\phi_1 \wedge \phi_2 \to \Phi_1 \cdot \Phi_2$
- $\phi_1 \vee \phi_2 \to 1 - (1 - \Phi_1) \cdot (1 - \Phi_2)$
- Note $\phi(b_1, \ldots, b_n) = \text{false} \Rightarrow \Phi(b_1, \ldots, b_n) = 0$ and $\phi(b_1, \ldots, b_n) = \text{true} \Rightarrow \Phi(b_1, \ldots, b_n) = 1$
- So, the number of satisfying assignments is exactly $\sum_{b_1,\ldots,b_n \in \{0,1\}} \Phi(b_1, \ldots, b_n)$

$\#P \subseteq IP$

A prover can prove to a verifier how many satisfying assignments a 3CNF formula ϕ has

Choose prime $q > \max\{2^n,\ 2^\lambda \cdot \sum_i d_i\}$ and view Φ as a polynomial in $\mathbb{F}_q[x_1, \ldots, x_n]$

- Although we defined Φ based on its values on $\{0,1\}^n$, nothing stops us from evaluating it on points in $\mathbb{F}_q^n$!

$\#P \subseteq IP$

A prover can prove to a verifier how many satisfying assignments a 3CNF formula ϕ has

Choose prime $q > \max\{2^n,\ 2^\lambda \cdot \sum_i d_i\}$ and view Φ as a polynomial in $\mathbb{F}_q[x_1, \ldots, x_n]$

- Although we defined Φ based on its values on $\{0,1\}^n$, nothing stops us from evaluating it on points in $\mathbb{F}_q^n$!

Step 2: run the sum-check protocol with H_0 the claimed number of satisfying assignments

IP = PSPACE

Possible to extend the previous result (using one additional idea) to show that PSPACE $\subseteq$ IP

This is tight, since it is also possible to show IP $\subseteq$ PSPACE

The GKR protocol

Motivating the GKR protocol

Say a prover wants to convince a verifier about some computation done by a machine M running in (polynomial) time T and (polynomial) space S

Motivating the GKR protocol

Say a prover wants to convince a verifier about some computation done by a machine M running in (polynomial) time T and (polynomial) space S

It is possible to reduce the computation of M to a PSPACE-complete problem, and then use PSPACE $\subseteq$ IP

- The reduction results in a formula on $n = O(S \log T)$ variables
- So the prover would require time $T^{O(S)}$!

Motivating the GKR protocol

Say a prover wants to convince a verifier about some computation done by a machine M running in (polynomial) time T and (polynomial) space S

It is possible to reduce the computation of M to a PSPACE-complete problem, and then use PSPACE $\subseteq$ IP

- The reduction results in a formula on $n = O(S \log T)$ variables
- So the prover would require time $T^{O(S)}$!

Problem: IP focuses entirely on keeping the verifier time polynomial, without regard for the prover time

- We want "doubly efficient" proofs

Mathematical preliminaries

Mathematical preliminaries

Let $f : \{0,1\}^n \rightarrow \mathbb{F}$ be a function

Mathematical preliminaries

Let $f : \{0,1\}^n \to \mathbb{F}$ be a function

$p \in \mathbb{F}[x_1, \ldots, x_n]$ is an extension of f if $p(\mathbf{b}) = f(\mathbf{b})$ for all $\mathbf{b} \in \{0,1\}^n$

Mathematical preliminaries

Let $f : \{0,1\}^n \to \mathbb{F}$ be a function

$p \in \mathbb{F}[x_1, \ldots, x_n]$ is an extension of f if $p(\mathbf{b}) = f(\mathbf{b})$ for all $\mathbf{b} \in \{0,1\}^n$

$p \in \mathbb{F}[x_1, \ldots, x_n]$ is multilinear if the degree of each variable is at most 1

Mathematical preliminaries

Lemma

If $p \in \mathbb{F}[x_1, \ldots, x_n]$ is a multilinear polynomial such that $p(\mathbf{b}) = 0$ for all $\mathbf{b} \in \{0,1\}^n$, then p is the zero polynomial.

Mathematical preliminaries

Lemma

If $p \in \mathbb{F}[x_1, \ldots, x_n]$ is a multilinear polynomial such that $p(\mathbf{b}) = 0$ for all $\mathbf{b} \in \{0,1\}^n$, then p is the zero polynomial.

Proof.

Assume not, and let $t = c \cdot \prod_{i \in S} x_i$ be a nonzero term in p with minimal total degree.

Mathematical preliminaries

Lemma

If $p \in \mathbb{F}[x_1, \ldots, x_n]$ is a multilinear polynomial such that $p(\mathbf{b}) = 0$ for all $\mathbf{b} \in \{0,1\}^n$, then p is the zero polynomial.

Proof.

Assume not, and let $t = c \cdot \prod_{i \in S} x_i$ be a nonzero term in p with minimal total degree. Then when setting all variables in S to 1, t is nonzero but all other terms are 0.

Mathematical preliminaries

Lemma

If $p \in \mathbb{F}[x_1, \ldots, x_n]$ is a multilinear polynomial such that $p(\mathbf{b}) = 0$ for all $\mathbf{b} \in \{0, 1\}^n$, then p is the zero polynomial.

Proof.

Assume not, and let $t = c \cdot \prod_{i \in S} x_i$ be a nonzero term in p with minimal total degree. Then when setting all variables in S to 1, t is nonzero but all other terms are 0. So p is nonzero in that case, a contradiction. □

Mathematical preliminaries

Theorem

Every function $f : \{0,1\}^n \to \mathbb{F}$ has a unique multilinear extension $\tilde{f}$.

Proof.

For $\mathbf{b} \in \{0,1\}^n$, define the multilinear polynomial

$$\begin{aligned} \chi_{\mathbf{b}}(x_1, \ldots, x_n) &= \prod_{i=1}^{n} (b_i x_i + (1 - b_i) \cdot (1 - \mathbf{x}_i)) \\ &= \begin{cases} 1 & \mathbf{x} = \mathbf{b} \\ 0 & \mathbf{x} \in \{0,1\}^n \setminus \mathbf{b} \end{cases} \end{aligned}$$

Mathematical preliminaries

Theorem

Every function $f : \{0,1\}^n \to \mathbb{F}$ has a unique multilinear extension $\tilde{f}$.

Proof.

For $\mathbf{b} \in \{0,1\}^n$, define the multilinear polynomial

$$\chi_{\mathbf{b}}(x_1, \ldots, x_n) = \prod_{i=1}^{n} (b_i x_i + (1 - b_i) \cdot (1 - \mathbf{x}_i)) = \begin{cases} 1 & \mathbf{x} = \mathbf{b} \\ 0 & \mathbf{x} \in \{0,1\}^n \setminus \mathbf{b} \end{cases}$$

Let $\tilde{f} = \sum_{\mathbf{b} \in \{0,1\}^n} f(\mathbf{b}) \cdot \chi_b(x_1, \ldots, x_n)$. □

Mathematical preliminaries

Theorem

Every function $f : \{0,1\}^n \to \mathbb{F}$ has a unique multilinear extension $\tilde{f}$.

Proof.

To see uniqueness, note that if g, h are both multilinear extensions of f, then $g - h$ is a multilinear polynomial that evaluates to 0 on $\{0,1\}^n$. □

Mathematical preliminaries

Given $\{f(\mathbf{b})\}_{\mathbf{b}\in\{0,1\}^n}$, how efficiently can we compute

$$\tilde{f}(\mathbf{w}) = \sum_{\mathbf{b}\in\{0,1\}^n} f(\mathbf{b}) \cdot \chi_b(w_1, \ldots, w_n)$$

(for arbitrary $\mathbf{w} \in \mathbb{F}^n$)?

Mathematical preliminaries

Given $\{f(\mathbf{b})\}_{\mathbf{b}\in\{0,1\}^n}$, how efficiently can we compute

$$\tilde{f}(\mathbf{w}) = \sum_{\mathbf{b}\in\{0,1\}^n} f(\mathbf{b}) \cdot \chi_b(w_1, \ldots, w_n)$$

(for arbitrary $\mathbf{w} \in \mathbb{F}^n$)?

Method 1: single pass over $\{f(\mathbf{b})\}_{\mathbf{b}\in\{0,1\}^n}$, time $O(n \cdot 2^n)$, space $O(n)$

- Useful when streaming $\{f(\mathbf{b})\}_{\mathbf{b}\in\{0,1\}^n}$ and want to minimize space

Mathematical preliminaries

Given $\{f(\mathbf{b})\}_{\mathbf{b}\in\{0,1\}^n}$, how efficiently can we compute

$$\tilde{f}(\mathbf{w}) = \sum_{\mathbf{b}\in\{0,1\}^n} f(\mathbf{b}) \cdot \chi_b(w_1, \ldots, w_n)$$

(for arbitrary $\mathbf{w} \in \mathbb{F}^n$)?

Method 1: single pass over $\{f(\mathbf{b})\}_{\mathbf{b}\in\{0,1\}^n}$, time $O(n \cdot 2^n)$, space $O(n)$

- Useful when streaming $\{f(\mathbf{b})\}_{\mathbf{b}\in\{0,1\}^n}$ and want to minimize space

Method 2: time and space $O(2^n)$

- Useful if $\{f(\mathbf{b})\}_{\mathbf{b}\in\{0,1\}^n}$ is stored anyway and want to minimize time
- Main idea: compute $\{\chi_{\mathbf{b}}(\mathbf{w})\}_{\mathbf{b}\in\{0,1\}^n}$ using memoization and then take the dot product with $\{f(\mathbf{b})\}_{\mathbf{b}\in\{0,1\}^n}$

Overview of the GKR protocol

Let C be a fan-in 2, layered arithmetic circuit over $\mathbb{F}$, with n inputs/outputs

- "Layered" $=$ gates at a level only connected to gates at previous level

Overview of the GKR protocol

Let C be a fan-in 2, layered arithmetic circuit over $\mathbb{F}$, with n inputs/outputs

- "Layered" $=$ gates at a level only connected to gates at previous level

P and V agree on $\mathbf{x} \in \mathbb{F}^n$; P wants to convince V that $C(\mathbf{x}) = \mathbf{y}$

Notation

Number the layers of C from 0 (output layer) to d (input layer)

- Let $S_i = 2^{s_i}$ be the number of gates at level i

Let $W_i : \{0,1\}^{s_i} \to \mathbb{F}$ be the function specifying the values on the output wires at level i (for the given input $\mathbf{x}$)

- Note V knows W_d, and the claimed value of W_0

Notation

Number the layers of C from 0 (output layer) to d (input layer)

- Let $S_i = 2^{s_i}$ be the number of gates at level i

Let $W_i : \{0,1\}^{s_i} \to \mathbb{F}$ be the function specifying the values on the output wires at level i (for the given input $\mathbf{x}$)

- Note V knows W_d, and the claimed value of W_0

Let $\text{add}_i : \{0,1\}^{s_i + 2s_{i+1}} \to \{0,1\}$ be the addition wiring predicate of layer i

- $\text{add}_i(a, b, c) = 1$ iff wire a is the sum of wires b and c

Define mult_i similarly

Notation

Key fact

$$\widetilde{W}_i(a) = \sum_{b,c\in\{0,1\}^{s_{i+1}}} \widetilde{\mathrm{add}}(a,b,c)\cdot\left(\widetilde{W}_{i+1}(b)+\widetilde{W}_{i+1}(c)\right) + \widetilde{\mathrm{mult}}(a,b,c)\cdot\left(\widetilde{W}_{i+1}(b)\cdot\widetilde{W}_{i+1}(c)\right).$$

Proof.

Both sides are multilinear in a, and agree for all $a \in \{0,1\}^{s_i}$ □

Notation

Key fact

$$\widetilde{W}_i(a) = \sum_{b,c\in\{0,1\}^{s_{i+1}}} \widetilde{\text{add}}(a,b,c)\cdot\left(\widetilde{W}_{i+1}(b)+\widetilde{W}_{i+1}(c)\right) + \widetilde{\text{mult}}(a,b,c)\cdot\left(\widetilde{W}_{i+1}(b)\cdot\widetilde{W}_{i+1}(c)\right).$$

Proof.

Both sides are multilinear in a, and agree for all $a \in \{0,1\}^{s_i}$ □

Looks like the sum-check protocol might be useful!

- Define $\tilde{p}_{i+1}(a,b,c) = \widetilde{\text{add}}_i(a,b,c)\cdot\left(\widetilde{W}_{i+1}(b)+\widetilde{W}_{i+1}(c)\right) + \widetilde{\text{mult}}_i(a,b,c)\cdot\left(\widetilde{W}_{i+1}(b)\cdot\widetilde{W}_{i+1}(c)\right)$

The GKR protocol—core idea

Common input: C and $\mathbf{x}$, which defines $W_d : \{0,1\}^{s_d} \to \mathbb{F}$

1. P sends $\mathbf{y} = C(\mathbf{x})$, which defines $W_0^* : \{0,1\}^{s_0} \to \mathbb{F}$
2. V chooses $r \leftarrow \mathbb{F}^{s_0}$, sends r to P, and sets $H_0 := \widetilde{W_0^*}(r)$
3. P, V run the sum-check protocol to show $H_0 = \sum_{b,c} \tilde{p}_1(r, b, c)$

Intuition:

- Let W_0 be the function corresponding to the correct output
- If $W_0^* \neq W_0$, then $\widetilde{W_0^*}(r) \neq \widetilde{W_0}(r)$ w.h.p.
- If $\widetilde{W_0^*}(r) \neq \widetilde{W_0}(r)$, V will reject in the sum-check protocol w.h.p.

The GKR protocol—core idea

Common input: C and $\mathbf{x}$, which defines $W_d : \{0,1\}^{s_d} \to \mathbb{F}$

1. P sends $\mathbf{y}$, which defines $W_0^* : \{0,1\}^{s_0} \to \mathbb{F}$
2. V chooses $r \leftarrow \mathbb{F}^{s_0}$, sends r to P, and sets $H_0 := \widetilde{W}_0^*(r)$
3. P, V run the sum-check protocol to show $H_0 = \sum_{b,c} \tilde{p}_1(r, b, c)$

Problem: to run the sum-check protocol, V needs to evaluate $\tilde{p}_1(r, b_1, c_1)$!

- In particular, V requires $\widetilde{W}_1(b_1)$ and $\widetilde{W}_1(c_1)$
  - We assume evaluating the rest of $\tilde{p}_1$ is easy
- If P sends W_1, then P ends up sending the entire evaluation of C...
- Instead, P sends $z_0 = \widetilde{W}_1(b_1), z_1 = \widetilde{W}_1(c_1)$ and V recursively verifies

Recursing

How to recurse?

Recursing

How to recurse?

V could check that $z_0 = \widetilde{W}_1(b_1)$ (or $z_1 = \widetilde{W}_1(c_1)$) using the sum-check protocol as before

Recursing

How to recurse?

V could check that $z_0 = \widetilde{W}_1(b_1)$ (or $z_1 = \widetilde{W}_1(c_1)$) using the sum-check protocol as before

But if V checks both in the obvious way, then V will end up running the sum-check protocol $O(2^d)$ times!

- Need a better approach ...
- We show one approach; other approaches are possible

Recursing

Recall: V needs to know $\widetilde{W}_1(b_1)$ and $\widetilde{W}_1(c_1)$

Recursing

Recall: V needs to know $\widetilde{W}_1(b_1)$ and $\widetilde{W}_1(c_1)$

Let $\ell : \mathbb{F} \to \mathbb{F}^{s_1}$ be the unique line such that $\ell(0) = b_1$ and $\ell(1) = c_1$

P sends the univariate polynomial $q = \widetilde{W}_1 \circ \ell$

Recursing

Recall: V needs to know $\widetilde{W}_1(b_1)$ and $\widetilde{W}_1(c_1)$

Let $\ell : \mathbb{F} \to \mathbb{F}^{s_1}$ be the unique line such that $\ell(0) = b_1$ and $\ell(1) = c_1$

P sends the univariate polynomial $q = \widetilde{W}_1 \circ \ell$

- V checks that q has degree $\leq s_1$
- V sets $\widetilde{W}_1(b_1) := q(0)$ and $\widetilde{W}_1(c_1) := q(1)$

Recursing

Recall: V needs to know $\widetilde{W}_1(b_1)$ and $\widetilde{W}_1(c_1)$

Let $\ell : \mathbb{F} \to \mathbb{F}^{s_1}$ be the unique line such that $\ell(0) = b_1$ and $\ell(1) = c_1$

P sends the univariate polynomial $q = \widetilde{W}_1 \circ \ell$

- V checks that q has degree $\leq s_1$
- V sets $\widetilde{W}_1(b_1) := q(0)$ and $\widetilde{W}_1(c_1) := q(1)$

V chooses $r^* \leftarrow \mathbb{F}$, sets $r_1 := \ell(r^*)$ and $H_1 := q(r^*)$, and has P prove that $H_1 = \widetilde{W}_1(r_1)$

- We have reduced checking the value of $\widetilde{W}_1$ at two points to checking its value at one point!

Overall GKR protocol

Overall, in the ith iteration

- V has a value H_i claimed to be equal to $\widetilde{W}_i(r_i)$
- P proves that $H_i = \sum_{b,c} \tilde{p}_{i+1}(r_i, b, c)$ using the sum-check protocol
  - To complete the protocol, V needs the evaluation of $\tilde{p}_{i+1}$ at a random point, which requires the evaluation of $\widetilde{W}_{i+1}$ at two random points
  - Using the previous method, we reduce this to a claim about the value of $\widetilde{W}_{i+1}$ at a single random point
- This results in a value H_{i+1} claimed to be equal to $\widetilde{W}_{i+1}(r_{i+1})$

Overall GKR protocol

Overall, in the ith iteration

- V has a value H_i claimed to be equal to $\widetilde{W}_i(r_i)$
- P proves that $H_i = \sum_{b,c} \tilde{p}_{i+1}(r_i, b, c)$ using the sum-check protocol
    - To complete the protocol, V needs the evaluation of $\tilde{p}_{i+1}$ at a random point, which requires the evaluation of $\widetilde{W}_{i+1}$ at two random points
    - Using the previous method, we reduce this to a claim about the value of $\widetilde{W}_{i+1}$ at a single random point
- This results in a value H_{i+1} claimed to be equal to $\widetilde{W}_{i+1}(r_{i+1})$

In the last iteration, V can verify the claimed value of $\widetilde{W}_d$ on its own

The GKR protocol

GKR protocol

Common input: C and $\mathbf{x}$, which defines $W_d : \{0,1\}^{s_d} \to \mathbb{F}$

P sends y, which defines $W_0 : \{0,1\}^{s_0} \to \mathbb{F}$. V chooses $r_0 \leftarrow \mathbb{F}^{s_0}$, sets $H_0 := \widetilde{W}_0(r_0)$, and sends r_0 to P. Then for $i = 0, \ldots, d-1$ do:

1. P, V run the sum-check protocol to show $H_i = \sum_{b,c} \tilde{p}_{i+1}(r_i, b, c)$
   - At the end of the protocol, V needs $\widetilde{W}_{i+1}(b_i)$, $\widetilde{W}_{i+1}(c_i)$
   - Let $\ell : \mathbb{F} \to \mathbb{F}^{s_{i+1}}$ be the line with $\ell(0) = b_i$ and $\ell(1) = c_i$
   - P sends $q_{i+1} = \widetilde{W}_{i+1} \circ \ell$ of degree at most s_{i+1}, and V uses $q_{i+1}(0), q_{i+1}(1)$ to complete the protocol
2. V chooses $r^* \leftarrow \mathbb{F}$, sets $r_{i+1} := \ell(r^*)$ and $H_{i+1} := q_{i+1}(r^*)$, and sends r_{i+1} to P

V checks that $H_d = \widetilde{W}_d(r_d)$

Analysis of the GKR protocol

There are now two sources of soundness error

- If q_{i+1} is incorrect but $q_{i+1}(r^*) = H_{i+1}$
  - Each q_{i+1} has degree at most $\log|C|$, so probability $O(|\mathbb{F}|^{-1} \cdot \log|C|)$

Analysis of the GKR protocol

There are now two sources of soundness error

- If q_{i+1} is incorrect but $q_{i+1}(r^*) = H_{i+1}$
  - Each q_{i+1} has degree at most $\log|C|$, so probability $O(|\mathbb{F}|^{-1} \cdot \log|C|)$
- The sum-check protocols
  - Each invocation of the sum-check protocol involves a polynomial in $\leq 2\log|C|$ variables, where each variable has degree $O(1)$
  - Soundness error $O(|\mathbb{F}|^{-1} \cdot \log|C|)$ per protocol

Analysis of the GKR protocol

There are now two sources of soundness error

- If q_{i+1} is incorrect but $q_{i+1}(r^*) = H_{i+1}$
  - Each q_{i+1} has degree at most $\log|C|$, so probability $O(|\mathbb{F}|^{-1} \cdot \log|C|)$
- The sum-check protocols
  - Each invocation of the sum-check protocol involves a polynomial in $\leq 2\log|C|$ variables, where each variable has degree $O(1)$
  - Soundness error $O(|\mathbb{F}|^{-1} \cdot \log|C|)$ per protocol

By a union bound, soundness error $O(d \cdot |\mathbb{F}|^{-1} \log|C|)$ overall

Complexity of the GKR protocol

Communication complexity (excluding the output)

- $\tilde{p}_{i+1}(r_i, b, c)$ is a $2s_{i+1}$-variate polynomial of degree ≤ 2 in each variable
- Each invocation of sum-check uses $O(\log |C|)$ rounds, with $O(1)$ field elements sent per round
- $O(d \log |C|)$ field elements sent overall

Complexity of the GKR protocol

Communication complexity (excluding the output)

- $\tilde{p}_{i+1}(r_i, b, c)$ is a $2s_{i+1}$-variate polynomial of degree ≤ 2 in each variable
- Each invocation of sum-check uses $O(\log |C|)$ rounds, with $O(1)$ field elements sent per round
- $O(d \log |C|)$ field elements sent overall

Round complexity $O(d \log |C|)$

Complexity of the GKR protocol

Verifier time (assuming time for $\widetilde{\mathrm{add}}_i, \widetilde{\mathrm{mult}}_i$ is dominated by other costs)

- $O(n)$ time to read input/output (and evaluate $\widetilde{W_d}$)
- $O(d \log |C|)$ additional work

Complexity of the GKR protocol

Verifier time (assuming time for $\widetilde{\text{add}}_i, \widetilde{\text{mult}}_i$ is dominated by other costs)

- $O(n)$ time to read input/output (and evaluate $\widetilde{W}_d$)
- $O(d \log |C|)$ additional work

Prover time

- Naively: in the ith iteration, P needs to evaluate p_{i+1} at $O(S_{i+1}^2)$ points; each evaluation takes time $O(S_i + S_{i+1})$
  - Total time $O(|C|^3)$
- Can do better by observing that $\widetilde{\text{add}}, \widetilde{\text{mult}}$ are sparse
  - Total time $O(|C| \log |C|)$

Recap

The sumcheck protocol is very powerful

- Can be used to show that IP is very powerful!

Recap

The sumcheck protocol is very powerful

- Can be used to show that IP is very powerful!

IP does not care about prover complexity, but in practice we (also) want the prover to be efficient (i.e., we want doubly efficient protocols)

- Also want the verifier to be as efficient as possible, not just "polynomial time"

Recap

The sumcheck protocol is very powerful

- Can be used to show that IP is very powerful!

IP does not care about prover complexity, but in practice we (also) want the prover to be efficient (i.e., we want doubly efficient protocols)

- Also want the verifier to be as efficient as possible, not just "polynomial time"

The GKR protocol takes a big step in that direction

- Note that the GKR protocol is a proof; can potentially gain more by considering arguments and using cryptography...

Thank you!