香港中文大學（深圳）
The Chinese University of Hong Kong, Shenzhen

SCHOOL OF DATA SCIENCE
數據科學學院

CSC 6011 - Theory of Computation

Lecture 21: IP, Sum-Check Protocol

Time: Mon & Wed, 4:00 - 5:20 PM

Location: Teaching A, 107

Instructor: Tao LIN (林涛)

TA: Yefan GAO (高叶繁)

Fall 2026

Lecture 21 – IP, Sum-Check Protocol

Last time:

- Interactive proofs and the class IP
- Graph non-isomorphism in IP
- Claim IP = PSPACE

Today:

- IP ⊆ PSPACE
- The Sum-Check Protocol

Review: Interactive Proofs

Verifier (V): probabilistic polynomial-time TM (with private random bits).

Prover (P): unlimited computational power.

Both see input w, and exchange polynomially many polynomial-size messages.

V accepts or rejects.

Defn: $A \in$ IP if some verifier V satisfies, for every input w:

- **Completeness:** If $w \in A$, some prover P makes V accept with probability $\geq 2/3$.
- **Soundness:** If $w \notin A$, every prover $\tilde{P}$ makes V accept with probability $\leq 1/3$.

Last time: NP $\subseteq$ IP, BPP $\subseteq$ IP, graph non-isomorphism $\in$ IP.

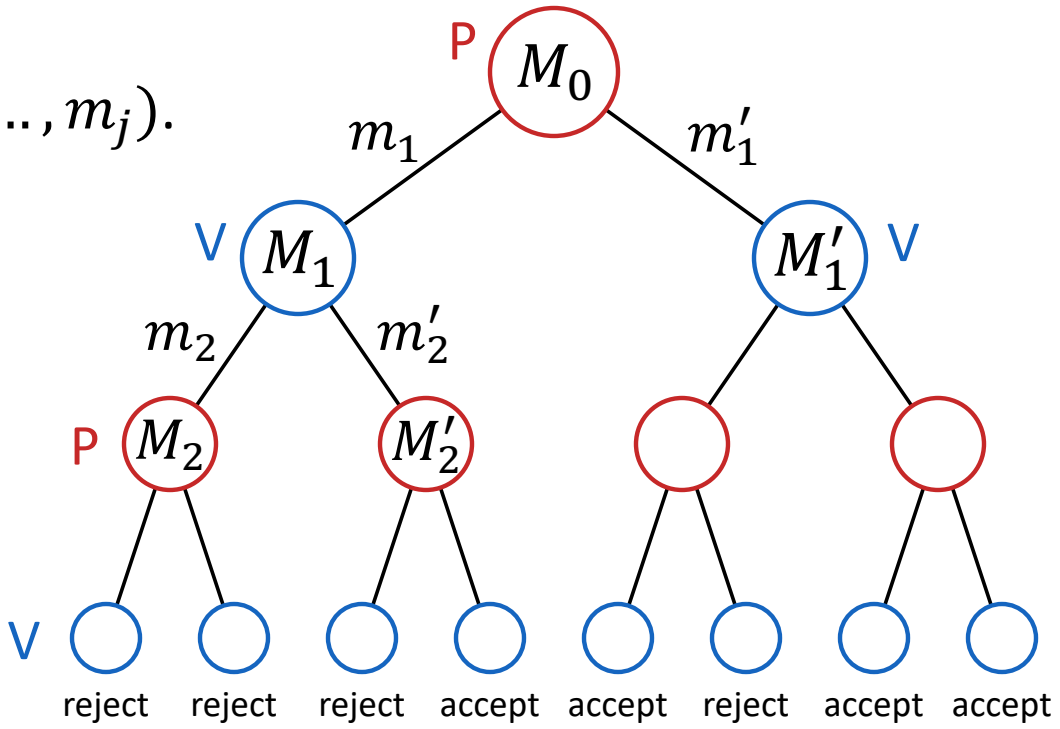
Theorem (Shamir, 1990): IP = PSPACE

Tree Representation of an Interactive Protocol

Fix an input $w \in \{0,1\}^n$ and a verifier V.

Nodes: message histories
- Nodes are associated with P or V, alternatively.
- A node on level j corresponds to a history $M_j = (m_1, \dots, m_j)$.
- The root has the empty history $M_0 = ()$

Edges: next message m_{j+1}.
- P chooses the message based on history.
- V generates the message based on history and randomness;
- V's last message is accept/reject.



Size of the tree:
- polynomial depth. # rounds of interactions is polynomial.
- Each node has at most 2^{n^c} children. 2^{n^c} possible messages of length n^c

IP ⊆ PSPACE

Let $A \in$ IP, with verifier V. Fix an input $w \in \{0,1\}^n$. At a node with message history $M_j = (m_1, \ldots, m_j)$, define the maximum probability that a prover can make V to accept, given M_j:

$$p(M_j) = \Pr[\text{V accepts } w \mid M_j] = \max_{\text{P}} \Pr[\text{V} \leftrightarrow \text{P accepts } w, \text{starting from } M_j]$$

For a leaf node: $p(M_{\text{leaf}}) = 1$ if V accepts; 0 if V rejects.

Compute $p(M_j)$ recursively:

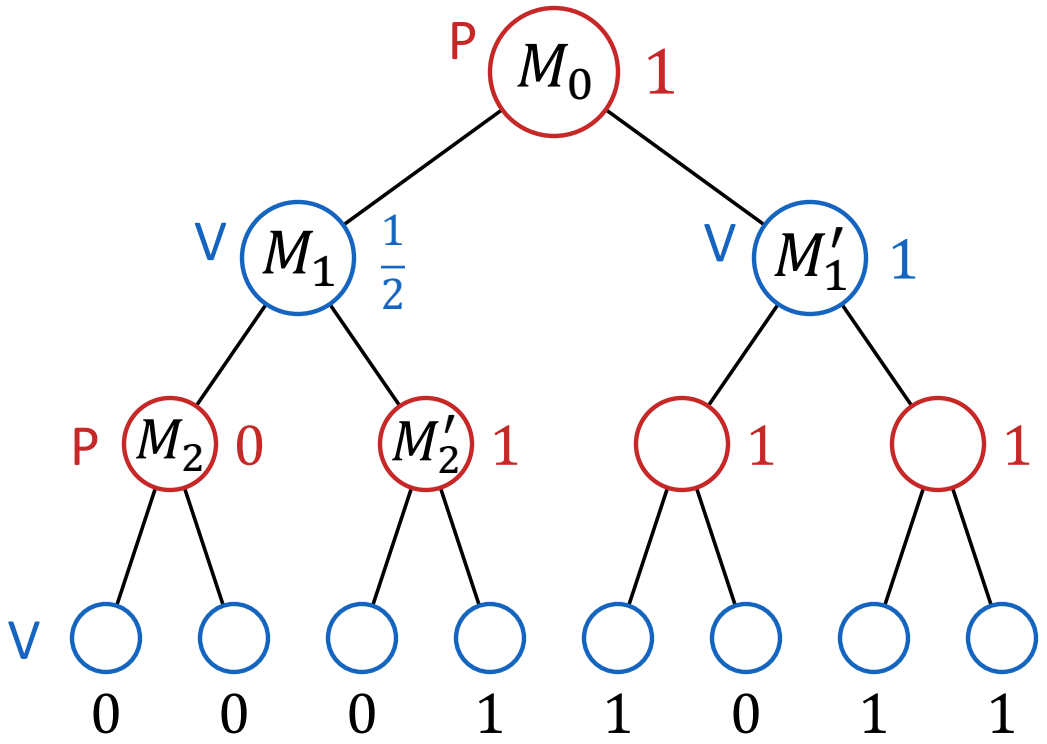
- **Prover:** $p(M_j) = \max_{m_{j+1}} p(M_{j+1})$
- **Verifier:** $p(M_j) = \sum_{m_{j+1}} \Pr[m_{j+1} \mid M_j]\, p(M_{j+1})$

The root value $p(M_0) = \max_{\text{P}} \Pr[\text{V} \leftrightarrow \text{P accepts } w]$.

By the definition of IP:

$p(M_0) \geq \frac{2}{3}$ for $w \in A$; $p(M_0) \leq \frac{1}{3}$ for $w \notin A$.

Accept w if $p(M_0) \geq \frac{1}{2}$.



Space needed:

"Depth-first" computation stores one path at a time (poly levels), using $O(n^c)$ space per level.

The Sum-Check Problem

Given an m-variable polynomial $g: \mathbb{F}_q^m \to \mathbb{F}_q$ over a finite field (arithmetic modulo q).

Known bound on the degree of each variable: $\deg_{x_i}(g) \leq d_i$ for $i = 1, \ldots, m$.

Prover provides H and wants to convince the Verifier that:

$$H = \sum_{b_1 \in \{0,1\}} \sum_{b_2 \in \{0,1\}} \cdots \sum_{b_m \in \{0,1\}} g(b_1, b_2, \ldots, b_m)$$

This sum has 2^m terms. V cannot compute it directly.

Theorem: The Sum-Check problem is in IP.

A useful tool to prove that other problems are in IP, such as:

- UNSAT: implying coNP $\subseteq$ IP
- TQBF: implying PSPACE $\subseteq$ IP (and hence IP $=$ PSPACE).

The Sum-Check Problem

Given an m-variable polynomial $g: \mathbb{F}_q^m \to \mathbb{F}_q$ over a finite field (arithmetic modulo q).

Known bound on the degree of each variable: $\deg_{x_i}(g) \leq d_i$ for $i = 1, \ldots, m$.

Prover provides H and wants to convince the Verifier that:

$$H = \sum_{b_1 \in \{0,1\}} \sum_{b_2 \in \{0,1\}} \cdots \sum_{b_m \in \{0,1\}} g(b_1, b_2, \ldots, b_m)$$

Key observation: H is equal to

$$\underbrace{\sum_{b_2 \in \{0,1\}} \cdots \sum_{b_m \in \{0,1\}} g(0, b_2, \ldots, b_m)}_{H_1(0)} + \underbrace{\sum_{b_2 \in \{0,1\}} \cdots \sum_{b_m \in \{0,1\}} g(1, b_2, \ldots, b_m)}_{H_1(1)}$$

where $H_1(X_1) = \sum_{b_2 \in \{0,1\}} \cdots \sum_{b_m \in \{0,1\}} g(X_1, b_2, \ldots, b_m)$ is a univariate polynomial of X_1, with degree $\leq d_1$, and each coefficient is the sum of exponentially many terms.

- If P can provide $H_1(X_1)$ to V and prove its correctness, then V can check $H = H_1(0) + H_1(1)$.

The Sum-Check Protocol

Start: P sends an integer S, claiming to be the sum H.

Round 1: P sends (the coefficients of) a univariate polynomial $S_1(X_1)$, claiming $S_1 = H_1$, where

$$H_1(X_1) = \sum_{b_2 \in \{0,1\}} \cdots \sum_{b_m \in \{0,1\}} g(X_1, b_2, \ldots, b_m)$$

V checks $\deg(S_1) \leq d_1$ and $S_1(0) + S_1(1) = S$.

Then, V samples a random $r_1 \in \mathbb{F}_q$ and sends it to P.

Round $i > 1$ $(i \leq m)$: P sends a univariate polynomial $S_i(X_i)$, claiming $S_i = H_i$, where

$$H_i(X_i) = \sum_{b_{i+1} \in \{0,1\}} \cdots \sum_{b_m \in \{0,1\}} g(r_1, \ldots, r_{i-1}, X_i, b_{i+1}, \ldots, b_m)$$

V checks $\deg(S_i) \leq d_i$ and $S_i(0) + S_i(1) = S_{i-1}(r_{i-1})$.

Then, V samples a random $r_i \in \mathbb{F}_q$ and sends it to P.

Final check: V checks $S_m(r_m) = g(r_1, \ldots, r_m)$.

Reject if any check fails. Otherwise accept.

Example: Honest Prover

Let $g(X_1, X_2) = X_1X_2 + X_2$. The true sum is $H = 0 + 1 + 0 + 2 = 3$.

Start: P sends $S = H = 3$.

Round 1: P sends polynomial

$$S_1(X_1) = H_1(X_1) = g(X_1, 0) + g(X_1, 1) = 0 + (X_1 + 1) = X_1 + 1.$$

V checks $S_1(0) + S_1(1) = S$: $1 + 2 = 3$, pass.

Suppose V samples $r_1 = 1$ and sends to P. $S_1(r_1) = 2$

[Then, check $S_1 = H_1$ by checking $S_1(r_1) = H_1(r_1)$.]

Round 2: P sends polynomial

$$S_2(X_2) = H_2(X_2) = g(r_1, X_2) = r_1X_2 + X_2 = 2X_2.$$

V checks $S_2(0) + S_2(1) = S_1(r_1)$: $0 + 2 = 2$, pass

Suppose V samples $r_2 = 2$. $S_2(r_2) = 4$

All checks passed: accept.

Final check: V computes and checks $g(r_1, r_2) = S_2(r_2)$: $4 = 4$, pass

Example: Cheating Prover

Question: In this example, which check fails?

Let $g(X_1, X_2) = X_1X_2 + X_2$. The true sum is $H = 0 + 1 + 0 + 2 = 3$.

Start: P sends $S = 0$, claiming to be H (a false claim).

Round 1: P sends polynomial

$$S_1(X_1) = 2X_1 - 1 \qquad \text{(but } H_1(X_1) = X_1 + 1\text{)}$$

V checks $S_1(0) + S_1(1) = S$. Pass this check? Yes: $-1 + 1 = 0$

Suppose V samples $r_1 = 1$ and sends to P. What is $S_1(r_1)$? 1

Round 2: P sends polynomial

$$S_2(X_2) = X_2 \qquad \text{(but } H_2(X_2) = 2X_2\text{)}$$

V checks $S_2(0) + S_2(1) = S_1(r_1)$. Pass this check? Yes: $0 + 1 = 1$

Suppose V samples $r_2 = 2$. What is $S_2(r_2)$? 2

Final check: V computes and checks $g(r_1, r_2) = S_2(r_2)$. Pass this check? No: $4 \neq 2$

Reject

Completeness of the Sum-Check Protocol

Start: P sends an integer S, claiming to be the sum H.

Round 1: P sends (the coefficients of) a univariate polynomial $S_1(X_1)$, claiming $S_1 = H_1$, where

$$H_1(X_1) = \sum_{b_2 \in \{0,1\}} \cdots \sum_{b_m \in \{0,1\}} g(X_1, b_2, \ldots, b_m)$$

V checks $\deg(S_1) \leq d_1$ and $S_1(0) + S_1(1) = S$.

Then, V samples a random $r_1 \in \mathbb{F}_q$ and sends it to P.

Round $i > 1$ $(i \leq m)$: P sends a univariate polynomial $S_i(X_i)$, claiming $S_i = H_i$, where

$$H_i(X_i) = \sum_{b_{i+1} \in \{0,1\}} \cdots \sum_{b_m \in \{0,1\}} g(r_1, \ldots, r_{i-1}, X_i, b_{i+1}, \ldots, b_m)$$

V checks $\deg(S_i) \leq d_i$ and $S_i(0) + S_i(1) = S_{i-1}(r_{i-1})$.

Then, V samples a random $r_i \in \mathbb{F}_q$ and sends it to P.

Final check: V checks $S_m(r_m) = g(r_1, \ldots, r_m)$.

Reject if any check fails. Otherwise accept.

Completeness:

When the prover P is honest, V accepts with probability 1.

Soundness

Claim: If P doesn't follow the protocol, then V accepts with probability at most $\frac{\sum_{i=1}^{m} d_i}{q} \leq \frac{md}{q}$.

Proof: Induction on the number of variables m:

- **Base case:** $m = 1$.
  - P sends a single message $S_1(X_1)$ claimed to equal $g(X_1)$.
  - V picks a random $r_1 \in \mathbb{F}_q$, checks that $S_1(r_1) = g(r_1)$.
  - If $S_1 \neq g$, then by Schwartz-Zippel, $\Pr_{r_1 \in \mathbb{F}_q}[\, S_1(r_1) = g(r_1) \,] \leq \frac{d_1}{q}$.

Soundness

Claim: If P doesn't follow the protocol, then V accepts with probability at most $\frac{\sum_{i=1}^{m} d_i}{q} \leq \frac{md}{q}$.

Proof: Induction on the number of variables m:

- **Inductive case:** $m > 1$.
  - P's first message $S_1(X_1)$ is claimed to equal $H_1(X_1) = \sum_{b_2 \in \{0,1\}} \cdots \sum_{b_m \in \{0,1\}} g(X_1, b_2, \ldots, b_m)$
  - Then V picks a random $r_1 \in \mathbb{F}_q$ and sends r_1 to P. They recursively check $S_1(r_1) = H_1(r_1)$.
  - If $S_1 \neq H_1$, then by Schwartz-Zippel, $\Pr_{r_1 \in \mathbb{F}_q}[\, S_1(r_1) = H_1(r_1)\,] \leq \frac{d_1}{q}$.
  - If $S_1(r_1) \neq H_1(r_1)$, then P needs to prove a false claim in the recursion.
    - The recursion applies the Sum-Check protocol to $g(r_1, X_2, \ldots, X_m)$, which has $m - 1$ variables.
    - By induction, P can convince V in the recursion with probability at most $\frac{\sum_{i=2}^{m} d_i}{q}$.
  - **Summary:** If $S_1 \neq H_1$, then the probability that V accepts is at most:

$$\Pr_{r_1 \in \mathbb{F}_q}[S_1(r_1) = H_1(r_1)] + \Pr_{r_2,\ldots,r_m \in \mathbb{F}_q}[\,\text{V accepts} \mid S_1(r_1) \neq H_1(r_1)\,]$$

$$\leq \frac{d_1}{q} + \frac{\sum_{i=2}^{m} d_i}{q} \leq \frac{\sum_{i=1}^{m} d_i}{q}$$

Costs of the protocol

Total communication: $O(md)$ field elements:

- P sends m messages, each a univariate polynomial of degree at most d.
- V sends $m-1$ messages, each being a field element.

Each field element (in $\mathbb{F}_q$) takes $O(\log q)$ bits.

V**'s runtime:** $O(md)$ field operations + time to evaluate g.

P**'s runtime:** $O(md \cdot 2^m \cdot$ time to evaluate g at one point $)$.

Mini-Quiz 21.1

In the Sum-Check protocol, how many evaluations of g does V need to do?

a) 2^m, one at every Boolean input.
b) m, one in each round.
c) Just one, $g(r_1, \ldots, r_m)$.

Quick review of today

- Reviewed IP.
- Showed IP $\subseteq$ PSPACE.
- Sum-Check Problem & Protocol.

Next time:

- Use the Sum-Check Protocol to prove coNP $\subseteq$ IP and (the main idea of) PSPACE $\subseteq$ IP

References

MIT: interactive proofs and complexity classes

Adapted from Michael Sipser, 18.404J Theory of Computation, Fall 2020, MIT.

Source: MIT OpenCourseWare, https://ocw.mit.edu

Justin Thaler: lecture notes and video

The Sum-Check Protocol, COSC 544 lecture notes, September 14, 2017.

https://people.cs.georgetown.edu/jthaler/sumcheck.pdf

Interactive Proofs (Part I), video tutorial.

https://www.youtube.com/watch?v=2XrOdfYviwA

Except for separately identified third-party material, this adapted presentation is licensed under CC BY-NC-SA 4.0.