

Communication Complexity under Randomized Protocols

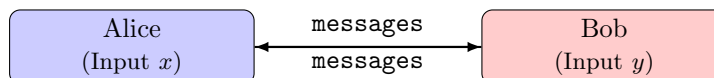
Instructor: Sruthi Sekar

Submitted by: Sagar Verma (24D0379)

Introduction

These notes cover the first part of our lecture on **randomized communication complexity**. We will discuss the setup, definitions, and a running example illustrating how randomization can lower communication costs.

1 Two-Party Communication Model



We have two players (represented by boxes, labeled “Alice” and “Bob”):

- **Alice** has a private input $x \in X$.
- **Bob** has a private input $y \in Y$.

They want to *compute* a function $f : X \times Y \rightarrow Z$. Often, we consider

$$X = \{0, 1\}^n, \quad Y = \{0, 1\}^n, \quad Z = \{0, 1\}.$$

Communication occurs via exchanged bits (represented by arrows in the diagram). The cost we focus on is the **total number of bits** exchanged in the worst case. We are not concerned with the total computation done by Alice and Bob before sending any message, and it is not included in the communication complexity.

2 Randomized Protocols

In a *randomized protocol*, Alice and Bob have access to random coins, which help reduce communication in some cases. Randomized protocol comes into below two settings:-

- **Public coins:** A shared random string accessible to both parties. For example, to check if $x = y$, they can use a shared random hash function h and compare $h(x)$ with $h(y)$. If $h(x) \neq h(y)$, they know $x \neq y$; otherwise, they assume equality with high probability.
- **Private coins:** Each party generates its own private random bits. For instance, Bob may use his private randomness to sample part of his input to decide what message to send, without revealing his full input.

Note: A “coin” refers to a source of randomness — typically a sequence of random bits — used by the protocol to make probabilistic decisions.

They can use these random bits to potentially reduce the communication cost, at the expense of allowing a small error probability ε . Formally, for every input pair (x, y) , the protocol’s output must match $f(x, y)$ with probability at least $1 - \varepsilon$.

3 Private-Coin Communication Complexity

We denote the *private-coin* communication complexity by

$$R_\varepsilon^{\text{priv}}(f) = \min_{\Pi} \{\text{bits}(\Pi) \mid \forall x, y, \Pr[\Pi(x, y) = f(x, y)] \geq 1 - \varepsilon\}.$$

Here, Π ranges over all protocols using private randomness or the worst case in random flip coin. The complexity is the minimum worst-case number of bits that must be exchanged so that the protocol computes f with error $\leq \varepsilon$.

4 Example: Equality Function

An illustrative function to see why randomness helps is the *Equality* function:

$$\text{EQ}(x, y) = \begin{cases} 1 & \text{if } x = y, \\ 0 & \text{otherwise.} \end{cases}$$

where $x, y \in \{0, 1\}^n$.

4.1 Deterministic Protocol

- Alice can simply send all n bits of her input x to Bob.
- Bob compares the received bits with his input y .
- If they match, output 1; otherwise, output 0.

This has **zero error** but costs **n** bits.

4.2 Randomized Protocol using Polynomials

Alice and Bob have inputs $a = (a_1, \dots, a_n)$ and $b = (b_1, \dots, b_n)$, respectively. They define polynomials:

$$P_A(x) = a_n x^n + \dots + a_1 x, \quad P_B(x) = b_n x^n + \dots + b_1 x$$

They want to check if $P_A(x) = P_B(x)$ without sending all coefficients.

1. Using public randomness, Alice pick a random $\alpha \in \mathbb{F}_q$.
2. Alice sends $(\alpha, P_A(\alpha))$ to Bob using $2 \log q$ bits.
3. Bob computes $P_B(\alpha)$ and accepts if it equals $P_A(\alpha)$.

The protocol uses private coins. The size of the field is chosen to be $O(n^2)$, so the total communication complexity is $O(\log n)$.

5 Alternate Protocol Using Error-Correcting Codes

An alternative way to achieve a randomized communication protocol for the Equality function (and related tasks) is via *error-correcting codes* (ECCs). Suppose Alice and Bob agree in advance on a binary ECC with the following properties:

- $\text{Enc}: \{0, 1\}^n \rightarrow \{0, 1\}^{Cn}$ is an encoding function that maps an n -bit message to a codeword of length Cn , for some constant $C > 0$.

- The code has *minimum distance* at least δn . In other words, for any distinct $x, y \in \{0, 1\}^n$, their encodings $\text{Enc}(x)$ and $\text{Enc}(y)$ differ in at least δn bit positions. (Such *asymptotically good* codes exist.)

Protocol Outline:

1. **Preprocessing:** Alice and Bob fix Enc beforehand.

2. **Alice's Side:**

- Let $x \in \{0, 1\}^n$ be Alice's input.
- Compute the codeword $c = \text{Enc}(x)$, which is in $\{0, 1\}^{Cn}$.
- For $k = O(1)$ times (chosen for the desired success probability), do:
 - Pick an index i uniformly at random from $\{1, 2, \dots, Cn\}$.
 - Send the pair (i, c_i) to Bob.
 - (This requires $\log(Cn)$ bits to specify i plus 1 bit for c_i .)

3. **Bob's Side:**

- Let $y \in \{0, 1\}^n$ be Bob's input.
- Compute $c_y = \text{Enc}(y)$.
- Upon receiving each (i, c_i) , compare c_i with the corresponding bit $c_{y,i}$.
- If they match for all (or a majority of) the sampled indices, Bob outputs $\text{EQ}(x, y) = 1$. Otherwise, he outputs 0.

Analysis:

- $\mathbf{x = y}$: If x and y are the same, then $\text{Enc}(x) = \text{Enc}(y)$. Hence each sampled bit matches with probability 1. The protocol outputs 1 with probability 1.
- $\mathbf{x \neq y}$: Since the code has minimum distance δn , $\text{Enc}(x)$ and $\text{Enc}(y)$ differ in at least δn positions. If we randomly sample an index i , the probability of hitting a differing position is at least $\frac{\delta n}{Cn} = \frac{\delta}{C}$. Thus, with probability at least δ/C , we get a mismatch that indicates $x \neq y$. Repeating the sampling k times (independently) lowers the overall error probability to about $(1 - \frac{\delta}{C})^k$. For suitable choices of constants and k , this can be driven below any $\varepsilon > 0$.

6 Error Probability Amplification

Suppose we have a randomized protocol that computes some function f with *constant* success probability $p > \frac{1}{2}$. We aim to boost that success probability to $1 - \varepsilon$ by *repetition* and *majority voting*.

6.1 Protocol Description

- Run the base protocol Π (with success probability p) r independent times on the same inputs (x, y) .
- Let the outcomes of these runs be $O_1, O_2, \dots, O_r$.
- Output the *majority* value among these r outcomes:

$$\text{Output} = \text{Majority}(O_1, O_2, \dots, O_r).$$

This procedure is often called *repetition* or *majority amplification*.

6.2 Analysis via Chernoff Bound

We want to show that this repetition can increase the success probability from p (where $p > \frac{1}{2}$) to at least $1 - \varepsilon$. Let

$$X_i = \begin{cases} 1 & \text{if the } i\text{-th run is correct (i.e., } O_i = f(x, y)), \\ 0 & \text{otherwise.} \end{cases}$$

Then each X_i is a Bernoulli(p) random variable, and the runs are independent. Let

$$S = \sum_{i=1}^r X_i$$

be the total number of successful runs. The final output is correct if $S > \frac{r}{2}$. Hence, our protocol errs only if $S \leq \frac{r}{2}$. We need to show that this event

$$\Pr\left[S \leq \frac{r}{2}\right]$$

can be made arbitrarily small by choosing r appropriately.

Since $p > \frac{1}{2}$, let $\delta = p - \frac{1}{2} > 0$. We have $\mathbb{E}[S] = pr$. Thus,

$$pr = \left(\frac{1}{2} + \delta\right)r = \frac{r}{2} + \delta r.$$

To bound $\Pr[S \leq \frac{r}{2}]$, we apply a Chernoff bound (one of several standard forms):

Chernoff Bound (One-Sided): If $X_1, \dots, X_r$ are independent Bernoulli(p) random variables with $S = \sum_{i=1}^r X_i$ and $\mathbb{E}[S] = pr$, then for any $\gamma \in (0, 1)$,

$$\Pr[S \leq (1 - \gamma)pr] \leq \exp\left(-\frac{\gamma^2}{2} pr\right).$$

In our case, we want $(1 - \gamma)pr = \frac{r}{2}$. That is:

$$(1 - \gamma)p = \frac{1}{2}.$$

Solving for γ , we get

$$\gamma = 1 - \frac{1/2}{p} = 1 - \frac{1}{2p} = \frac{2p - 1}{2p}.$$

Hence

$$\Pr\left[S \leq \frac{r}{2}\right] = \Pr\left[S \leq (1 - \gamma)pr\right] \leq \exp\left(-\frac{\gamma^2}{2} pr\right).$$

Since $\gamma = \frac{2p-1}{2p}$ is a positive constant (because $p > \frac{1}{2}$), the exponent $\frac{\gamma^2}{2} pr$ grows linearly with r . Thus we get:

$$\Pr\left[S \leq \frac{r}{2}\right] \leq \exp(-\Omega(r)).$$

By choosing r to be on the order of $\ln(\frac{1}{\varepsilon})$, we can make this probability at most ε . Specifically, pick

$$r \geq C \ln\left(\frac{1}{\varepsilon}\right)$$

for some sufficiently large constant C (depending on p and hence on δ). Then

$$\Pr\left[S \leq \frac{r}{2}\right] \leq \varepsilon.$$

This proves the *amplification*:

If a protocol is correct with probability $p > \frac{1}{2}$ in one run, then by $r = O(\log(1/\varepsilon))$ repetitions and majority vote, the new protocol is correct with probability at least $1 - \varepsilon$.

Thus, *majority voting* on repeated independent runs amplifies the success probability from a constant $p > \frac{1}{2}$ to $1 - \varepsilon$ in $\Theta(\log(1/\varepsilon))$ repetitions.

7 Public-Coin Setting (Refined Example)

In the *public-coin* model, Alice and Bob have access to a common random source. This often enables simpler or lower communication protocols compared to private-coin protocols.

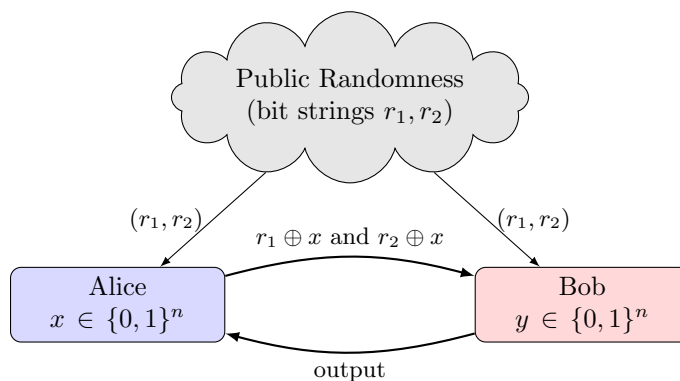
7.1 An Example Using Two Random Bits

Proposition 1. *Alice and Bob can use a simple Hamming code (or a related linear code) and employ public randomness for sampling positions. The high-level sketch is that with 2 bits of publicly chosen indices plus 1 bit from Alice about her encoded value at that index, Bob can check equality with constant success probability, which can then be boosted via amplification..*

Protocol Sketch:

1. From the common random source, Alice and Bob obtain two independent random n -bits string, say $r_1, r_2 \in \{0, 1\}^n$.
2. Bob computes the *parity* of these two bits, $r_1 \oplus x$ and $r_2 \oplus x$, and sends this **two bit** to Alice.
3. Alice uses the received bit (together with r_1 and r_2 which she also knows from the public source) to decide whether $x = y$.

The probability of error in this simple scheme is $\frac{1}{4}$ (that is, $\frac{1}{2} \times \frac{1}{2}$), since using only two random bits imposes certain limitations on detecting inequality for all $x \neq y$. However, we can repeat or amplify (see Section 6) to reduce this error further.



Analysis:

- *Communication Cost:* $3(2+1)$ bits.
- *Error Probability:* $1/4$ in a single run. Repetition plus majority voting can reduce this below any desired ε (cf. Section 6).

7.2 Why Public Coins Can Help

In general,

$$R_{\varepsilon, \text{pub}}(f) \leq R_{\varepsilon, \text{priv}}(f)$$

(Given a private coin protocol, Alice & Bob can always emulate it in the public coin world as follows: Alice can take all even positions of the common random string as her private coin & Bob can take odd ones. They can then run the private coin protocol.)

8 Newman's Theorem

A celebrated result known as *Newman's Theorem* shows that any public-coin communication protocol can be converted into a private-coin protocol with only a small overhead in both the error probability and the communication cost.

Theorem 2 (Newman). *Let $f : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}$ be any function, and let $\varepsilon, \delta > 0$. Then*

$$R_{\varepsilon+\delta}^{\text{priv}}(f) \leq R_{\varepsilon}^{\text{pub}}(f) + O(\log n + \log \frac{1}{\delta}).$$

8.1 Proof Overview

The high-level idea is that a *finite* set of public random strings will “cover” all possible inputs with small error, allowing us to simulate a public-coin protocol via private coins at an overhead of only $O(\log n + \log \frac{1}{\delta})$ bits.

Step-by-Step Construction (“Book of Random Strings”):

1. **Original Public-Coin Protocol:** Suppose the original public-coin protocol π^{pub} uses an m -bit random string $r \in \{0, 1\}^m$ and achieves error $\leq \varepsilon$ for every pair $(x, y) \in \{0, 1\}^n \times \{0, 1\}^n$.
2. **Build a Book of t Random Strings:** We choose

$$t = O\left(\frac{n}{\delta^2}\right).$$

Sample t independent random strings

$$r^{(1)}, r^{(2)}, \dots, r^{(t)} \in \{0, 1\}^m$$

from the *same* distribution used by the public protocol. Alice and Bob jointly fix this “book” of t strings before seeing their inputs.

3. **Private-Coin Simulation:**

- *Receiving Inputs:* Once in separate rooms, Alice and Bob learn $x, y \in \{0, 1\}^n$ respectively.
- *Index Selection:* Alice uses her *private coins* to choose an index $i \in [t]$ uniformly at random (no public randomness). She sends this index i to Bob using $O(\log t)$ bits.
- *Execute the Original Protocol:* Now both parties run π^{pub} as if $r^{(i)}$ were the public random string. This costs $R_{\varepsilon}^{\text{pub}}(f)$ bits of communication.

4. **Total Communication Cost:**

$$\underbrace{O(\log t)}_{\text{index}} + \underbrace{R_{\varepsilon}^{\text{pub}}(f)}_{\text{original protocol}} = R_{\varepsilon}^{\text{pub}}(f) + O(\log t) = R_{\varepsilon}^{\text{pub}}(f) + O\left(\log n + \log \frac{1}{\delta}\right).$$

We now show that the new protocol (using private randomness and the precomputed book of t strings) has error at most $\varepsilon + \delta$ on every input (x, y) .

Bad Random Strings: Call an m -bit string r *bad* for (x, y) if running the public-coin protocol with r produces an incorrect answer on input (x, y) . By assumption, for each fixed (x, y) ,

$$\Pr_r[r \text{ is bad}] \leq \varepsilon.$$

Fraction of Bad Strings in the Book: If we pick t strings $r^{(1)}, \dots, r^{(t)}$ at random, the *expected* fraction of bad strings in $\{r^{(1)}, \dots, r^{(t)}\}$ is at most ε . By applying a Chernoff bound, we see that the probability that this fraction exceeds $\varepsilon + \delta$ is at most

$$\exp(-\Theta(\delta^2 t)).$$

Since $t = O(\frac{n}{\delta^2})$, this probability can be made $< 2^{-2n}$.

Union Bound Over All (x, y) : There are at most 2^{2n} pairs (x, y) . Hence, with probability $< 2^{-2n} \times 2^{2n} = 1$, we can ensure *all* pairs (x, y) see at most $\varepsilon + \delta$ fraction of bad strings in the book. Put differently, there *exists* one fixed choice of $\{r^{(1)}, \dots, r^{(t)}\}$ such that for *every* (x, y) , the fraction of bad strings is $\leq \varepsilon + \delta$.

Conclusion: Once the set of t strings is fixed, if Alice chooses $r^{(i)}$ uniformly at random from that set, the error on any (x, y) is at most $\varepsilon + \delta$. This completes the private-coin simulation with the stated overhead in communication.

Final Remark. Theorem 2 follows: We have constructed a private-coin protocol whose error is $\varepsilon + \delta$ and whose communication is at most

$$R_\varepsilon^{\text{pub}}(f) + O(\log n + \log \frac{1}{\delta}).$$

Because $\delta > 0$ was arbitrary, we conclude

$$R_{\varepsilon+\delta}^{\text{priv}}(f) \leq R_\varepsilon^{\text{pub}}(f) + O(\log n + \log \frac{1}{\delta}),$$

which matches the statement of Newman's Theorem.

9 Distributional Communication Complexity

In the distributional setting, we do *not* consider the worst-case over all possible inputs. Instead, we assume a distribution μ over $X \times Y$. The cost measure then focuses on how many bits are needed by a *deterministic* protocol Π so that it is correct with high probability $\geq 1 - \varepsilon$ when inputs are drawn according to μ .

Definition 3 (ε -error Distributional Complexity). *For a function $f : X \times Y \rightarrow \{0, 1\}$ and a distribution μ over $X \times Y$, the ε -error distributional complexity $D_\varepsilon^\mu(f)$ is the minimum cost of a deterministic protocol Π such that*

$$\Pr_{(x,y) \sim \mu} [\Pi(x, y) = f(x, y)] \geq 1 - \varepsilon.$$

9.1 Examples

Example 1: EQ_n under Uniform Distribution. Let $f = \text{EQ}_n$, and suppose μ is the uniform distribution over $\{0, 1\}^n \times \{0, 1\}^n$.

$$\Pr_{(x,y) \sim \mu} [x = y] = \frac{1}{2^n},$$

which is extremely small for large n . Thus, a trivial protocol that *always outputs 0* will be correct except in the (rare) event $\{x = y\}$. Concretely,

$$\Pr_{(x,y) \sim \mu} [\text{output} = 0] = 1 - \frac{1}{2^n}.$$

Hence, $D_\varepsilon^\mu(\text{EQ}_n)$ can be as low as 0 bits if we allow error $\varepsilon \geq \frac{1}{2^n}$.

Example 2: DISJ under Uniform Distribution. Let $f = \text{DISJ}$, where $\text{DISJ}(x, y) = 1$ iff $x \cap y = \emptyset$. Suppose μ is again uniform on $\{0, 1\}^n \times \{0, 1\}^n$. We want the probability that two random sets (bitstrings) are disjoint. For each coordinate $i \in \{1, \dots, n\}$, the chance that both $x_i = 1$ and $y_i = 1$ is $1/4$. Therefore, the probability $\text{DISJ}(x, y) = 1$ (i.e. all coordinates do *not* simultaneously equal 1) is

$$\Pr_{(x,y) \sim \mu} [x \cap y = \emptyset] = \left(1 - \frac{1}{4}\right)^n = \left(\frac{3}{4}\right)^n.$$

For large n , $\left(\frac{3}{4}\right)^n$ is very small, so a trivial protocol that *always outputs 0* will be correct with probability

$$1 - \left(\frac{3}{4}\right)^n.$$

Hence, if $\varepsilon \geq (3/4)^n$, then 0 bits of communication suffice.

9.2 Public vs. Distributional Complexity

Theorem 4. For every distribution μ over $X \times Y$,

$$R_{\varepsilon, \text{pub}}(f) \leq D_{\varepsilon}^{\mu}(f).$$

Proof Sketch. Fix a distribution μ and suppose there is a *public-coin* protocol Π_{pub} of cost $R_{\varepsilon, \text{pub}}(f)$. We can interpret Π_{pub} as a *distribution* (over the public random string r) of deterministic protocols. Concretely, for each fixed r , $\Pi_{\text{pub}}(\cdot, \cdot; r)$ is deterministic.

Since Π_{pub} succeeds with probability at least $1 - \varepsilon$ over the choice of r and $(x, y) \sim \mu$, by the probabilistic method there must exist a particular r^* such that the deterministic protocol $\Pi(\cdot, \cdot) = \Pi_{\text{pub}}(\cdot, \cdot; r^*)$ achieves error at most ε under μ . Hence, Π is a deterministic protocol of cost $\leq R_{\varepsilon, \text{pub}}(f)$, valid for distribution μ . By definition, this implies $D_{\varepsilon}^{\mu}(f) \leq R_{\varepsilon, \text{pub}}(f)$.

Thus,

$$R_{\varepsilon, \text{pub}}(f) \leq D_{\varepsilon}^{\mu}(f).$$

□

10 Yao's Minimax Theorem (Formal Statement)

Theorem 5 (Yao's Minimax Theorem).

$$R_{\varepsilon, \text{pub}}(f) = \max_{\mu} D_{\varepsilon}^{\mu}(f),$$

where the maximum is over all distributions μ on $X \times Y$.

Interpretation (Point by Point):

1. *Public-Coin Lower Bounds via Distributional Complexity.*
2. To show $R_{\varepsilon, \text{pub}}(f) \geq C$, find a distribution μ such that *every* deterministic protocol of cost $< C$ errs on at least ε fraction (w.r.t. μ).
3. By Yao's Minimax Theorem, this implies the public-coin complexity is also $\geq C$.

10.1 Classic Example: IP_2

- $\text{IP}_2(x, y) = (x_1 \wedge y_1) \oplus \cdots \oplus (x_n \wedge y_n)$.
- Choose x, y uniformly at random from $\{0, 1\}^n$. Show any small-cost deterministic protocol has high error.
- By Minimax, we get a lower bound on $R_{\varepsilon, \text{pub}}(\text{IP}_2)$.

Why It Matters:

- Yao's Theorem connects *randomized* and *distributional* complexities.
- It simplifies lower-bound proofs: *just* exhibit a “hard distribution” μ .

10.2 Inner-Product mod 2 as a Character Function using Fourier coefficients

- **Inputs.**

$$x \in \{-1, +1\}^n \quad (\text{Alice}), \quad S \subseteq [n] \quad (\text{Bob}).$$

- **Definition.**

$$\text{IP}_2(x, S) = \chi_S(x) := \prod_{i \in S} x_i.$$

(The empty product, $S = \emptyset$, equals 1.)

- **Fourier facts (for $f : \{-1, 1\}^n \rightarrow \{-1, 1\}$).**

1. *Coefficient extraction:*

$$\mathbb{E}_x[f(x) \chi_S(x)] = \hat{f}(S).$$

2. *Parseval's identity:*

$$\sum_{S \subseteq [n]} \hat{f}(S)^2 = 1.$$

Below is a proof of lower bound for IP_2 in public coin setting that is

Theorem 6. For every $n \geq 1$,

$$R_{1/4, \text{pub}}(\text{IP}_2) \geq \frac{n}{2} - 1.$$

Plan (Yao).

- By Yao's Minimax, it is enough to find a **hard distribution** μ such that $D_{1/4}^\mu(\text{IP}_2) \geq \frac{n}{2} - 1$.
- We choose $\mu = \text{uniform}$ on $x \in \{-1, +1\}^n$, $S \subseteq [n]$, where $\text{IP}_2(x, S) = \chi_S(x) = \prod_{i \in S} x_i$.

11 Fix an Arbitrary c -bit Deterministic Protocol Π

Step 1: Partition into rectangles.

The protocol tree has at most 2^c leaves. Each leaf corresponds to a *combinatorial rectangle*

$$\mathcal{R}_\ell \times \mathcal{S}_\ell \quad (\ell = 1, \dots, 2^c),$$

and an output label $z_\ell \in \{-1, +1\}$.

3. Assume Π errs on $\leq 1/4$ mass (toward contradiction), so in terms of success it would be

$$\Pr_{(X,S) \sim \mu} [\pi(X, S) = \text{IP}_2(X, S)] \geq \frac{3}{4} \quad (2)$$

To show:

$$c \geq \frac{n}{2} - 1$$

$$(1) \quad \pi(X, S) = \sum_{i=1}^{2^c} \mathbf{1}[(X, S) \in R_i \times S_i] \cdot z_i$$

(Indicator for $(X, S) \in R_i \times S_i$)

$$= \sum_{i=1}^{2^c} \mathbf{1}[X \in R_i] \cdot \mathbf{1}[S \in S_i] \cdot z_i$$

$$(\text{denote by } \mathbf{1}_{R_i}(X), \text{ denote by } \mathbf{1}_{S_i}(S)) \quad (3)$$

$$(2) \quad \mathbb{E}_{(X,S) \sim \mu} [\pi(X, S) \cdot \text{IP}_2(X, S)] \geq \frac{3}{4}(+1) + \frac{1}{4}(-1) = \frac{1}{2}$$

(By (3)):

$$\Rightarrow \mathbb{E}_{X,S} \left[\sum_{i=1}^{2^c} \mathbf{1}_{R_i}(X) \cdot \mathbf{1}_{S_i}(S) \cdot z_i \cdot \text{IP}_2(X, S) \right] \geq \frac{1}{2}$$

$$\frac{1}{2} \leq \sum_{i=1}^{2^c} |z_i| \cdot |\mathbb{E}_{X,S} [\mathbf{1}_{R_i}(X) \cdot \mathbf{1}_{S_i}(S) \cdot \text{IP}_2(X, S)]|$$

Since $|z_i| = 1$:

$$\leq 2^c \cdot \max_i |\mathbb{E}_{X,S} [\mathbf{1}_{R_i}(X) \cdot \mathbf{1}_{S_i}(S) \cdot \text{IP}_2(X, S)]| \quad (4)$$

Claim (I): $\forall i, \quad |\mathbb{E}_{X,S} [\mathbf{1}_{R_i}(X) \cdot \mathbf{1}_{S_i}(S) \cdot \text{IP}_2(X, S)]| \leq 2^{-n/2}$

Using Claim (I) and (4):

$$\frac{1}{2} \leq 2^c \cdot 2^{-n/2} \quad \Rightarrow \quad c \geq \frac{n}{2} - 1 \quad (\text{Done})$$

11.1 Proof of Claim 1: Discrepancy of Inner Product

Claim: For all i ,

$$|\mathbb{E}_{x,S} [\mathbf{1}_{R_i}(x) \cdot \mathbf{1}_{S_i}(S) \cdot \text{IP}_2(x, S)]| \leq 2^{-n/2}$$

Technique: We use the **analysis of Boolean functions**, leveraging Fourier methods.

We begin with the expression:

$$\mathbb{E}_{x,S} [\mathbf{1}_{R_i}(x) \cdot \mathbf{1}_{S_i}(S) \cdot \text{IP}_2(x, S)] = \mathbb{E}_{x,S} [\mathbf{1}_{R_i}(x) \cdot \mathbf{1}_{S_i}(S) \cdot \chi_S(x)]$$

where $\chi_S(x) := \sum_{j \in S} x_j \bmod 2$ is the parity function (a character of the Boolean cube).

Rewrite the expectation by separating variables:

$$= \mathbb{E}_S [\mathbf{1}_{S_i}(S) \cdot \mathbb{E}_x [\mathbf{1}_{R_i}(x) \cdot \chi_S(x)]] = \mathbb{E}_S [\mathbf{1}_{S_i}(S) \cdot \widehat{\mathbf{1}_{R_i}}(S)]$$

Now we apply the Cauchy-Schwarz inequality:

$$\left| \mathbb{E}_S \left[\mathbf{1}_{S_i}(S) \cdot \widehat{\mathbf{1}_{R_i}}(S) \right] \right| \leq \sqrt{\mathbb{E}_S \left[\mathbf{1}_{S_i}(S)^2 \cdot \widehat{\mathbf{1}_{R_i}}(S)^2 \right]}$$

Since $\mathbf{1}_{S_i}(S)^2 = \mathbf{1}_{S_i}(S)$ and the uniform probability on S is $1/2^n$, we get:

$$\leq \sqrt{\frac{1}{2^n} \sum_S \widehat{\mathbf{1}_{R_i}}(S)^2}$$

Finally, by Parseval's identity:

$$\sum_S \widehat{\mathbf{1}_{R_i}}(S)^2 = \mathbb{E}_x[\mathbf{1}_{R_i}(x)^2] \leq 1$$

Putting it all together:

$$|\mathbb{E}_{x,S}[\mathbf{1}_{R_i}(x) \cdot \mathbf{1}_{S_i}(S) \cdot \text{IP}_2(x, S)]| \leq \sqrt{\frac{1}{2^n}} = 2^{-n/2}$$

11.2 Conclusion

$$D_{1/4}^{\text{unif}}(\text{IP}_2) \geq \frac{n}{2} - 1 \quad \Rightarrow \quad R_{1/4}^{\text{pub}}(\text{IP}_2) \geq \frac{n}{2} - 1$$

11.3 Remark: Uniform Distribution Does Not Help for DISJ

Uniform distribution $\mu = \text{unif}$ does not help prove:

$$R_{1/4}^{\text{pub}}(\text{DISJ}_n) = \Omega(n)$$