CS 860: Space Bounded Computation **University of Waterloo: Winter 2026**

Lecture 8: Time-Space Tradeoffs on Branching Programs III

March 10, 2026

Instructor: Sepehr Assadi *Scribe: Daniel Jiang*

Topics of this Lecture

In the last lecture, we saw a slightly super-linear time-space tradeoff for (standard) branching programs (BPs) for decision problems, which (more or less) remains the state-of-the-art. In this lecture, we study a more restricted family of BPs—called the *random-query* model introduced by [RZ20]—that allows us to prove stronger lower bounds.

1 Random-Query Model and the Main Result

Recall from earlier lectures that in an ordinary BP, the current vertex determines which input coordinate is queried next. In the *random-query model* of [RZ20]however, at each step, the BP simply queries a random index of the input (alternatively, the BP receives a random input index and its value at each time step).

Definition 1 (Random-query branching program)**.** A **random-query branching program** is defined as in the branching-program model from Lecture 5, except that the queried index is not determined by the current vertex and instead is a uniformly at random chosen index of the input.

More precisely, let $f : [m]^n \to [o]$, and let $x = (x_1, x_2, \ldots, x_n) \in [m]^n$ be the input. The computation starts at the root. For each vertex v, there are $n \times m$ outgoing edges labelled by (i, x_i) where $i \in [n]$ and $x_i \in [m]$. At time t, the computation receives a random index $i_t \in [n]$, it then follows the edge labelled $e = (i_t, x_{i_t})$ and appending out_e to the output if it is not $\perp$. Once we reach the last layer, we halt.

Consequently, the random-query model is weaker than the standard branching-program model, since the computation can no longer choose the most useful coordinate to inspect next. Our goal is however, to understand how this loss of control affects the time-space complexity of basic computational problems.

In the original paper of [RZ20], the model was also studied under the *recurring-distribution* which means that the indices i_t will come in a repeated pattern. In this lecture, we only consider the random-query model with *independent-distribution*, where each i_t is sampled uniformly and independently. Specifically, we are interested in the following problem in this model.

Problem 1. The XOR problem is the function $XOR : \{0,1\}^n \to \{0,1\}$ defined by $XOR(x) = \bigoplus_{i=1}^{n} x_i$.

The main result of this lecture is a time-space lower bound for the XOR function in the random-query model, proved in [Din24].

Theorem 2 ([Din24])**.** *Under the random-query model, any branching program of length T and space S that computes the n-bit XOR with probability of success at least $2/3$ must satisfy*

$$S \cdot T = \Omega(n^2).$$

2 Detour: Direct-Sum Arguments

Before getting to the proof of Theorem 2, let us review a generic *hardness amplification* technique that can be quite powerful for proving various lower bounds across different computational models, the so-called *direct-sum* argument(s). The general idea is to prove a lower bound for a simple "one-bit problem" and then show the main problem implicitly requires one to solve "many" *independent* copies of this one-bit problem, thus, intuitively, should be "many times harder".

Of course, the naive statement of "if solving one instance of problem requires $\Omega(S)$ resources, then solving n independent instances should require $\Omega(nS)$ resources", does not hold in general or across all computational models. For instance, solving $\text{poly}(n)$ independent copies of USTCONN on n-vertex graphs still requires only $O(\log n)$ space (as we showed in Lecture 3 for a single graph). Even moving beyond space which is an unusual resource given it is *reusable*, we cannot even hope for a generic direct-sum for *time*. For instance, consider matrix multiplication: computing Ax for a single vector may require circuit of size $\Omega(n^2)$ (by counting arguments, for some A; we omit the proof); but computing $Ax_1, \ldots, Ax_n$ can be batched as one matrix multiplication $A[x_1 \cdots x_n]$, which can be done with circuits of $\ll n^3$ size.

Thus, direct-sum lower bounds do not come for free. Yet, we also know such results for at least some models of computation. As we will use an argument of this nature in the proof of Theorem 2, let us first see a much simpler application of these ideas in a more abstract setting to prepare for the main argument.

2.1 Decision Tree Lower Bound For OR

Let us consider the following problem.

Problem 2. The OR problem is the function $OR : \{0,1\}^n \to \{0,1\}$ defined by $OR(x) = \bigvee_{i=1}^{n} x_i$.

Recall that a decision tree is basically a branching program wherein we do not bound the width at all (thus, we are only interested in the number of queries made to the input). We can use a direct-sum (style) argument to prove a decision tree lower bound for the OR problem.

Proposition 3. *Any decision tree that computes OR with success probability at least $\geqslant \frac{2}{3}$ on the input distribution*

$$D = \begin{cases} \mathbf{0} & w.p.\ \frac{1}{2} \\ e_i & w.p.\ \frac{1}{2n} \quad \forall i \in [n] \end{cases},$$

where e_i is the unit vector in the i-th coordinate, must make $\Omega(n)$ queries in the worst-case.

Proof. Consider the (rather silly) one-bit problem $Echo : \{0,1\} \to \{0,1\}$ defined by

$$Echo(y) = y,$$

over the uniform distribution on $\{0,1\}$.

A simple observation is that any algorithm for solving $Echo$ with probability of success at least $2/3$ has *expected query complexity* at least $1/3$. To see this, suppose the algorithm queries the input bit y with probability p. Then, on the times that it does not query the input, the best it can do is to guess the answer. So the success probability of this algorithm would be $p + \frac{1-p}{2} = \frac{1+p}{2}$. If we want this value to be at least $\frac{2}{3}$, we would need $p \geq \frac{1}{3}$.

We now reduce OR to $Echo$. Let A be an algorithm that solves OR with probability at least $2/3$ under D. Construct an algorithm B that solves $Echo$ over the uniform distribution as follows:

Algorithm 1.

1. On input $y \in \{0,1\}$, generate a uniformly random index $i \in [n]$, and define $x := y \cdot e_i$.
2. Simulate the decision tree A with B on x: whenever A tries to query x_j, if $i \neq j$, B should answer 0, otherwise, B will query y and return the same answer.
3. B will output whatever A outputs.

Firstly, it is easy to verify that the distribution of inputs generated in B (taking into account the randomness of y) is the same as D. Moreover, the correct answer of A on the input of x is the value of y (as all other indices are 0), namely, $OR(x) = Echo(y)$. Thus, B also succeeds with probability at least $2/3$.

Let $q_A(x)$ be the number of queries made by A on input x and $q_B(y)$ be the same for B on y. We claim

$$\mathbb{E}\left[q_B(0)\right] = \frac{q_A(\mathbf{0})}{n}; \tag{1}$$

in other words, when the input of B is 0, then, its expected query complexity is n times smaller than that of A on the all-zero input. We note that the RHS is not a random variable as we focus on worst-case query complexity for A and thus assume all inputs have the same number of queries (by padding shorter ones with arbitrary queries).

To see this, we have,

$$\mathbb{E}\left[q_B(0)\right] = \Pr\left(A \text{ queries index } i \text{ on input } \mathbf{0}\right) = \frac{q_A(\mathbf{0})}{n},$$

because on the input $\mathbf{0}$, the distribution of index i where $y = 0$ sits, is exactly the same as all other coordinates; in other words, even given the entire vector $x = \mathbf{0}$ to A, the algorithm has no way of figuring out which index y is placed into (this is the sole reason we are focusing on the 0-inputs; this guarantee is certainly not true for the other case). Thus, since the choice of which indices to query by A is fixed by the input, the probability that index i is one of those is precisely the fraction of indices in $[n]$ queried by A.

To conclude the proof, we further argue that

$$\mathbb{E}\left[q_B(0)\right] = \mathop{\mathbb{E}}_{y}\left[q_B(y)\right],$$

where we give a uniformly at random input y to B instead of just 0. This is because before the algorithm makes a query, it will not know if $y = 0$ or $y = 1$ and thus its decision is entirely independent of the value of y; in other words, B basically queries the input bit with a fixed probability p and thus in both cases of the input, the probability it queries the input is the same.

Putting all these together, we obtained that B is an algorithm for *Echo* with expected query complexity $q_A(\mathbf{0})/n$ and success probability at least $2/3$. Thus, by our earlier observation, we need to have

$$\frac{q_A(\mathbf{0})}{n} \geqslant \frac{1}{3},$$

which implies $q_A(\mathbf{0}) \geqslant n/3$. Therefore, worst-case query complexity of A is $\Omega(n)$ as desired. □

3 Proof of Theorem 2 via Communication Complexity

We now introduce the communication problem that will serve as the main intermediate step in the proof of the BP lower bound for the XOR problem in Theorem 2.

Problem 3 (Set Disjointness)**.** The *set disjointness* problem $DISJ_{n,k}$ is a k-player communication problem. Each player P^i receives an input vector $x^{(i)} \in \{0,1\}^n$. We define

$$DISJ_{n,k}(x^{(1)}, \ldots, x^{(k)}) := \bigvee_{j=1}^{n} \bigwedge_{r=1}^{k} x_j^{(r)}.$$

That is, $DISJ_{n,k} = 1$ if and only if there exists some coordinate j such that $x_j^{(i)} = 1$ for every player $i \in [k]$ (it is rather unfortunate, but a common notation, that $DISJ(x) = 1$ if the inputs of players are not disjoint from each other, as in, there exists an element that appears in the input of all players (here, by "input" we think of each $x^{(i)}$ as being the characteristic vector of the input set in $[n]$).

The players have access to both public and private randomness.

We study $DISJ_{n,k}$ under the following distribution μ. First choose $\theta \in \{0,1\}$ uniformly at random. Then generate the columns independently as follows:

- If $\theta = 0$, choose $j \in [n]$ uniformly at random, set $x_j^{(i)} = 1$ for all i, and for each $k \neq j$, sample $x_k^{(i)}$ uniformly from $\{0,1\}$.
- If $\theta = 1$, sample every $x_k^{(i)}$ independently and uniformly from $\{0,1\}$.

In words, when $\theta = 0$ we plant one all-1 column, whereas when $\theta = 1$ all columns are fully random. Hence, under μ, the task is to guess the hidden bit θ, which is slightly (up to a factor of $n \cdot 2^{-k}$) different than solving $DISJ_{n,k}$ exactly[a]

[a] We formulated our problem this way so that the analysis in Lemma 6 will become cleaner.

There is a very large body of work on the communication complexity of the set disjointness problem. For our purpose, we need the following lower bound, also proven by [Din24], for this problem.

Theorem 4 ([Din24])**.** *For $k \geqslant 2\log n + 10$, any public-coin protocol that correctly guesses θ with probability at least $2/3$ on the distribution μ requires communicating $\Omega(n)$ bits across all players.*

We will prove the theorem later. First, we show how this communication lower bound implies our main time-space lower bound for XOR.

3.1 Random-Query Time-Space Tradeoffs for XOR from Set Disjointness

We now show how a BP for XOR in the random-query model would give a communication protocol for the distribution from Problem 3.

Lemma 5. *Let B be a branching program in the random-query model with depth $T < \frac{n^2}{4}$ and width 2^S, that computes XOR with error δ at most $\frac{1}{100}$. Set $k := 4T/n$. Then there is a public-coin protocol Π with k players and communication cost at most $O(S \cdot k)$ such that*

$$\Pr[\Pi = 1 \mid \theta = 1] \geqslant 1 - \delta - 2^{-n/20}, \qquad \Pr[\Pi = 1 \mid \theta = 0] \leqslant \frac{1}{2} + 2^{-n/20}.$$

Proof. We aim to build a public-coin communication protocol π that distinguishes between the two cases $\theta = 1$ and $\theta = 0$ in the distribution μ from Problem 3.

The idea is to use the BP B on a random string $R \in \{0,1\}^n$ and to let the players jointly simulate its random queries using their own inputs. If $\theta = 1$, this simulation should look like a genuine execution of B and thus its output should be the same as XOR of R (which the players can compute on their own). But if $\theta = 0$, one coordinate of R will never be revealed and thus the output of B will be independent of this bit, whereas the value of the bit can change the XOR, hence, making the answer not-equal-to XOR of R with probability $1/2$.

Algorithm 2. A protocol π for $DISJ_{n,k}$ from a BP B for XOR in the random-query model.

1. The players using the public randomness sample $R_1, \ldots, R_n$ at the beginning.
2. Each player P^j picks a random ordering $E_1^{(j)}, \ldots, E_m^{(j)}$ of $m := n/4$ distinct indices with $x^{(j)}_{E_r^{(j)}} = 0$.
3. If some player fails to sample m such elements, we simply consider the protocol failed.
4. Each player P^j then generates inputs $Q_1^j, \ldots, Q_m^j$ to B as shall be specified later, and run B on the input, communicating the state of B after m steps to the next player.
5. The last player outputs $\theta = 1$ if the output of B is the same as $\mathrm{XOR}(R)$ and $\theta = 0$ otherwise.

It thus remains to specify the inputs $Q_1, \ldots, Q_m$ for some fixed player in the protocol π (we drop the superscript j of the player here and in the following for simplicity). A naive approach would be to set $Q_i = (E_i, R_{E_i})$. However, this fails as we sampled E_i without replacement, but XOR expects input indices to be uniformly sampled with replacement – this means that this method of sampling does not generate inputs from the input distribution of B. However, a simple fix is to generate Q_i such that:

$$Q_1 = (E_1, R_1); \quad \text{and, for all } i > 1,$$
$$Q_i = \begin{cases} (E_j, R_{E_j}) & \text{for some random } j \in [k] \quad \text{w.p. } \frac{\ell}{n} \\ (E_{\ell+1}, R_{E_{\ell+1}}) & \text{w.p. } \frac{n-\ell}{n} \end{cases},$$

where ℓ is the number of the prefix of elements in $E_1, \ldots, E_m$ that we have used for defining $Q_1, \ldots, Q_{i-1}$ so far. Since $E_1, \ldots, E_m$ is a permutation, we see that this indeed generates a uniform sample to XOR with replacement.

We now analyze the correctness of the protocol π. First, the probability that a player fails to sample m elements, is, by Chernoff bound,

$$\Pr\left(\text{less than } m = n/4 \text{ elements in } x \in_R \{0,1\}^{n-1} \text{ are } 0\right) \leq 2\exp\left(-\frac{((n-1)/4)^2}{2(n-1)}\right) \leq 2^{-n/10},$$

where we use $(n-1)$ induces in x that do not correspond to θ are thus in both cases are chosen uniformly at random from $\{0,1\}$. Thus, by union bound over all players, we have,

$$\Pr\left(\text{there exists a player that cannot sample } m \text{ elements}\right) \leqslant k \cdot 2^{-n/10} \leqslant 2^{-n/20},$$

for sufficiently large n.

Next we analyze the success probabilities in both $\theta = 0$ case and $\theta = 1$ case.

Case $\theta = 1$. We see that the protocol perfectly simulates B, so the probability of success of Π given every player samples m bits would be

$$\Pr\left(\pi \text{ outputs } 1 \mid \theta = 1, \text{protocol did not output } \pi\right) \geqslant \Pr\left(B \text{ succeeds}\right) = 1 - \delta$$

and hence

$$\Pr\left(\pi \text{ outputs } 1 \mid \theta = 1\right) \geqslant 1 - \delta - 2^{-n/20}.$$

Case $\theta = 0$. There is an all-1 column $i \in [n]$ in this case and hence no player ever queries index i, therefore R_i never appears in the simulated execution. Thus, even though in this case the sampling protocol may not generate the same distribution as μ, the fact that B will never receive some bit in R makes the probability that it output the correct answer as $\mathrm{XOR}(R)$ no better than $\frac{1}{2}$. So, the probability of the protocol outputting 1 by union bound over this and the failure probability is

$$\Pr\left(\pi \text{ outputs } 1 \mid \theta = 0\right) \leqslant \frac{1}{2} + 2^{-n/20},$$

as desired. □

Proof of Theorem 2. Let B be a branching program in the random-query model. By standard success amplification, we may assume that B has error at most $1/100$, at the cost of increasing the depth and space by a constant factor.

Applying Lemma 5, we obtain a public-coin protocol π with communication cost $O(S \cdot k)$ such that

$$\Pr\left(\pi \text{ outputs } 1 \mid \theta = 1\right) \geqslant 1 - \delta - 2^{-n/20} \quad \text{and} \quad \Pr\left(\pi \text{ outputs } 1 \mid \theta = 0\right) \leqslant \frac{1}{2} + 2^{-n/20}.$$

Hence, the success probability of Π on the distribution μ is

$$\frac{1}{2} \cdot \Pr\left(\pi \text{ outputs } 1 \mid \theta = 1\right) + \frac{1}{2} \Pr\left(\pi \text{ outputs } 0 \mid \theta = 0\right) \geqslant \frac{1}{2}\left(1 - \delta - 2^{-n/20}\right) + \frac{1}{2} \cdot \left(\frac{1}{2} - 2^{-n/20}\right) > \frac{2}{3},$$

for sufficiently large n.

Since each player simulate $m = n/4$ time steps of the BP, we get

$$k \cdot \frac{n}{4} = T \quad \text{which implies that} \quad k = \frac{4T}{n}.$$

Since by Theorem 4 we know that the communication cost of π has to be $\Omega(n)$, we will have $O(S \cdot k) = \Omega(n)$. This, combined with the above bound, implies that

$$S \cdot T = \Omega(n^2),$$

concluding the proof. □

4 Proof of Theorem 4

The direct-sum idea from earlier will be applied in this section in a form of a communication problem. The problem $DISJ_{n,k}$ is an OR of n column-wise copies of an AND_k problem (to be defined formally below). Thus, to prove an $\Omega(n)$ communication lower bound, it is natural to first understand the "cost" of solving one copy of AND, and then argue that solving OR of all n copies would require n times as much cost.

Problem 4. The problem AND_k is defined as follows. The players receive a vector

$$y = (y_1, \ldots, y_k) \in \{0,1\}^k.$$

A bit $\theta \in \{0,1\}$ is chosen uniformly at random.

- If $\theta = 0$, then $y = \mathbf{1}$ (the all-1 column vector).
- If $\theta = 1$, then y is uniformly distributed in $\{0,1\}^k$.

The goal is to distinguish between the two cases, i.e. to output θ.

In the following, for a protocol π, let $\|\pi\|$ be the communication cost of π. We start with an easy lemma.

Lemma 6. *Let π be a protocol for $DISJ_{n,k}$ on the specified distribution of the problem. Then there exists a protocol γ for AND (on its specified distribution) such that*

$$\|\gamma\| = \|\pi\| \quad \textit{and} \quad \Pr(\gamma \textit{ outputs correctly}) = \Pr(\pi \textit{ outputs correctly}).$$

Proof. Given an input $Y = (Y_1, \ldots, Y_k) \in \{0,1\}^k$ to AND, the players construct an input $X^{(1)}, \ldots, X^{(k)}$ for π as follows.

Protocol. The players use public-randomness to sample an index $i \in [n]$. For every column $\ell \neq i$, each player P^j samples $X^{(j)}_\ell \in \{0,1\}$ independently and uniformly at random. At column i, they set $X^{(j)}_i = Y_j$ for all $j \in [k]$. Then run π on $X^{(1)}, \ldots, X^{(k)}$, and output the same answer.

Analysis. By construction $\|\gamma\| = \|\pi\|$. It remains to analyze correctness when $\theta = 1$ and $\theta = 0$.

Case $\theta = 0$. In this case, $Y = \mathbf{1}$ so the i-th column is all 1, and every other column is uniformly chosen. This is distributed exactly the same as the case $\theta = 0$ for $DISJ_{n,k}$.

Case $\theta = 1$: In this case, everything is distributed uniformly random, including the i-th column. This is distributed exactly the same as the case $\theta = 1$ for $DISJ_{n,k}$.

Thus, in both cases, the input seen by π has exactly the required distribution, implying the bound on the correctness probability of the protocol γ as well. □

It is worth comparing Lemma 6 to our proof of Proposition 3 for a decision tree lower bound of OR by reducing it to Echo. There, we designed a decision tree for Echo based on a one for OR which was *"n times more efficient"* in its resource requirement. However, Lemma 6 simply says the communication cost of the protocol for AND will be the same as the one for DISJ – this will certainly not be efficient for us as we cannot hope for a lower bound for AND stronger than $\Omega(1)$ communication. But, also recall from Proposition 3 that the measure of resource we used when going from the OR-protocol to Echo-protocol changed: we had to go from worst-case query complexity to expected query complexity (which is quite minor in that proof).

We now do the same for DISJ as well: roughly speaking, we show that the "information" revealed by AND-protocol about its input is n times less than the communication cost of DISJ-protocol (as majority

of the communication done by the DISJ-protocol should have nothing to do with the particular index we embedded the AND-instance inside). To be able to formalize this intuition, we first need to review standard tools from *information theory*.

5 Detour: A Quick Intro to Information Theory

Throughout, we use log for base two logarithm and use the convention that $0 \cdot \log(1/0) = 0$.

Entropy

Suppose you have a random variable X from a domain $\mathcal{X}$. How many bits to do you need to "encode" a sample X and send it over to someone else? Well, in the *worst-case*, definitely $\log|\mathcal{X}|$ bits is needed, just by pigeonhole principle. But what about *average-case* (over the randomness of X)? At this point, the answer depends on the distribution of X. For instance, if X has a uniform distribution, intuitively you cannot do much better than (almost) $\log|\mathcal{X}|$ bits even on average, while if almost all of the mass of X is on a single value x_0, you can do much better by encoding x_0 with a shorter message than the rest.

A rough idea is to do the following: start from the largest probability p of any element under X; we can have at most $1/p$ different $x \in \mathcal{X}$ with $\Pr(X = x) = p(x) = p$ as otherwise the sum of their probabilities will be more than 1; thus, we can encode each of these x's with $\log(1/p)$ bits and then move to the next value of p and continue like this. This way, for any element x, the *expected* length of encoding used for the element x is $p(x) \cdot \log(1/p(x))$ (ignoring all ceiling/floor issues, etc.).[1]

Based on this discussion, we define the **entropy** of a random variable X with PDF $p(x)$ for $x \in \mathcal{X}$ as:

$$\mathbb{H}(X) := \mathop{\mathbb{E}}_{x \sim X}\left[\log \frac{1}{p(x))}\right] = \sum_{x \in \mathcal{X}} p(x) \cdot \log \frac{1}{p(x)}. \tag{2}$$

We can think of $\mathbb{H}(X)$ as a "measure" of the average length of best encoding of X.[2] To test your intuition, consider determining the entropy of each the andom variables below over the domain $\mathcal{X} = \{1, \ldots, n\}$? (compare these with the encoding length of variables you "expect" to achieve by your "own" coding scheme.)

- X: uniform over $\mathcal{X}$;
- Y: deterministically equal to 1 always;
- Z: $\Pr(Z = 1) = \frac{1}{2}$ and $\Pr(Z = i) = \frac{1}{2(n-1)}$ for any other $i \neq 1 \in \mathcal{X}$.

Let us now establish several useful properties of entropy. We first need to recall *Jensen's inequality*.

Fact 7 (**Jensen's inequality**). *Let f a* concave *function and X be a random variable over $\mathcal{X}$. Then,*

$$\mathbb{E}[f(X)] \leqslant f(\mathbb{E}[X]).$$

Moreover, the equality holds iff X is deterministic or f is linear.

We now use this to establish several properties of entropy.

Property 1 (**entropy and support size**). *For a random variable X with domain $\mathcal{X}$, we have,*

$$0 \leqslant \mathbb{H}(X) \leqslant \log|\mathcal{X}|.$$

The LHS is tight iff X is deterministic and the RHS is tight iff X is uniform.

[1] This is a very handwavy argument and is only meant to help the reader put the notion of entropy in some context; that being said, this is also not too far from what happens with Huffman coding.

[2] And again, while this is not precise, this is also not too far from the truth as Huffman coding achieves average length for a random variable X which is between $\mathbb{H}(X)$ and $\mathbb{H}(X) + 1$.

Proof. The LHS is obviously true because $\mathbb{H}(X)$ is expectation of non-negative terms and the only case they are all zero is when X is deterministic. For the RHS, we apply Jensen's inequality (Fact 7) as follows (using the fact that $\log(\cdot)$ is a concave function):

$$\mathbb{H}(X) = \mathop{\mathbb{E}}_{x\sim X}\left[\log\left(1/p(x)\right)\right] \leqslant \log\left(\mathop{\mathbb{E}}_{x\sim X}\left[1/p(x)\right]\right) = \log|\mathcal{X}|.$$

Jensen's inequality will be tight here iff all $p(x)$ values are the same (making the random variable $1/p(x)$ deterministic), thus implying the second part. □

Property 2 (**sub-additivity of entropy**). *For random variables X, Y, we have,*

$$\mathbb{H}(X,Y) \leqslant \mathbb{H}(X) + \mathbb{H}(Y);$$

moreover, the equality holds iff $X \perp Y$. Here $\mathbb{H}(X,Y)$ is simply entropy of the joint *random variable (X,Y).*

Proof. We can compute $\mathbb{H}(X,Y) - (\mathbb{H}(X) + \mathbb{H}(Y))$ as follows:

$$\begin{aligned}
\mathbb{H}(X,Y) - \mathbb{H}(X) - \mathbb{H}(Y) &= \sum_{x,y} p(x,y)\log\frac{1}{p(x,y)} - \sum_{x,y} p(x,y)\log\frac{1}{p(x)} - \sum_{x,y} p(x,y)\log\frac{1}{p(y)} \\
&\qquad\qquad (p(x) = \textstyle\sum_y p(x,y) \text{ and } p(y) = \sum_{x,y} p(x,y)) \\
&= \sum_{x,y} p(x,y)\log\frac{p(x)p(y)}{p(x,y)} = \mathop{\mathbb{E}}_{(x,y)\sim(X,Y)}\left[\log\frac{p(x)p(y)}{p(x,y)}\right] \\
&\leqslant \log\left(\mathop{\mathbb{E}}_{(x,y)\sim(X,Y)}\left[\frac{p(x)p(y)}{p(x,y)}\right]\right) \qquad \text{(by Jensen's inequality)} \\
&= \log\left(\sum_{x,y} p(x)p(y)\right) = \log 1 = 0.
\end{aligned}$$

Moreover, Jensen's inequality above is tight whenever $p(x)p(y) = p(x,y)$ for all x, y, meaning $X \perp Y$. □

Conditional Entropy

Recall the motivating example for the entropy and now suppose that you have joint random variables (X,Y): Given a sample $(x,y) \sim (X,Y)$, how many bits do you need to encode x alone and send it to someone who already *knows* y? Again, we are interested in this question on *average* over choices of both X and Y. You can see that this question depends on the distribution of X, Y and the correlation between the two; for instance for X, Y with fixed marginals, say, both uniform, the answer would be very different when $X = Y$ so highly correlated vs. when $X \perp Y$.

This discussion brings us to the notion of **conditional entropy** defined as follows:

$$\mathbb{H}(X \mid Y) := \mathop{\mathbb{E}}_{y\sim Y}\left[\mathbb{H}(X \mid Y = y)\right] = \sum_{y\in\mathcal{Y}} p(y) \cdot \sum_{x\in\mathcal{X}} p(x \mid y) \cdot \log\frac{1}{p(x \mid y)}; \tag{3}$$

(notice that here $\mathbb{H}(X \mid Y = y)$ is just the entropy of random variable X' with distribution $X \mid Y = y$).

Examples. What are the conditional entropy of each of the random variables? (compare these with the encoding length you "expect" to achieve by your "own" coding scheme.)

- $\mathbb{H}(A \mid B)$: when A, B are independent and uniform over $\{0,1\}^n$;
- $\mathbb{H}(C \mid D)$: when C is uniform over $\{0,1\}^n$ and $D \in \{0,1\}$ is XOR of bits of D.
- $\mathbb{H}(E \mid F)$: when E is uniform over $\{0,1\}^n$ and F is the indicator random variable for $E = 0^n$.

Let us now present some useful properties of conditional entropy.

Property 3 (**chain rule of entropy**). *For a random variables X, Y, we have,*

$$\mathbb{H}(X, Y) = \mathbb{H}(X) + \mathbb{H}(Y \mid X).$$

Proof. We can compute $\mathbb{H}(X, Y) - (\mathbb{H}(X) + \mathbb{H}(Y \mid X))$ as follows:

$$\begin{aligned}
\mathbb{H}(X, Y) - \mathbb{H}(X) - \mathbb{H}(Y \mid X) &= \sum_{x,y} p(x, y) \log \frac{1}{p(x, y)} - \sum_{x,y} p(x, y) \log \frac{1}{p(x)} - \sum_{x} p(x) \sum_{y} p(y \mid x) \log \frac{1}{p(y \mid x)} \\
&= \sum_{x,y} p(x, y) \log \frac{1}{p(x, y)} - \sum_{x,y} p(x, y) \log \frac{1}{p(x)} - \sum_{x,y} p(x, y) \log \frac{1}{p(y \mid x)} \\
&\qquad\qquad (p(x) \cdot p(y \mid x) = p(x, y)) \\
&= \sum_{x,y} p(x, y) \log \frac{p(x) \cdot p(y \mid x)}{p(x, y)} = \sum_{x,y} p(x, y) \log \frac{p(x, y)}{p(x, y)} \\
&= \sum_{x,y} p(x, y) \log 1 = 0.
\end{aligned}$$

□

We shall note that chain rule is one of the most important properties of entropy as it gives us *additivity*.

Another property of entropy is that conditioning can only reduce its value: after all, encoding a variable X given "extra" information Y should never become harder than if we have not been given Y in the first place. Formally,

Property 4 (**conditioning cannot increase entropy**). *For a random variables X, Y, we have,*

$$\mathbb{H}(X \mid Y) \leqslant \mathbb{H}(X);$$

moreover, the equality holds iff $X \perp Y$.

Proof. By sub-additivity $\mathbb{H}(X, Y) \leqslant \mathbb{H}(X) + \mathbb{H}(Y)$ while by chain rule $\mathbb{H}(X, Y) = \mathbb{H}(Y) + \mathbb{H}(X \mid Y)$. Plugging in these two implies the result. The second part follows from the second part of Property (2). □

Let us emphasize that this property is only true when conditioning on a random variable and not an event (i.e., realization of a random variable). Can you give an example when the latter does not hold?

Mutual Information

Finally, we can talk about "information" between two random variables X, Y. In the context of the motivating examples before, we would like to quantify how much the knowledge of Y helped us in encoding X, i.e., the gap between the average encoding of X with or without the extra knowledge of Y (and vice versa).

Formally, the **mutual information** between two variables X, Y is defined as:

$$\mathbb{I}(X\,;\,Y) := \mathbb{H}(X, Y) - (\mathbb{H}(X \mid Y) + \mathbb{H}(Y \mid X)). \tag{4}$$

By applying chain rule of entropy, we can see right away that

$$\mathbb{I}(X\,;\,Y) := \mathbb{H}(X) - \mathbb{H}(X \mid Y) = \mathbb{H}(Y) - \mathbb{H}(Y \mid X). \tag{5}$$

which might sound more familiar. Finally, **conditional mutual information** is defined analogously:

$$\begin{aligned}
\mathbb{I}(X\,;\,Y \mid Z) &:= \mathbb{H}(X, Y \mid Z) - (\mathbb{H}(X \mid Z, Y) + \mathbb{H}(Y \mid Z, X)) &(6)\\
&= \mathbb{H}(X \mid Z) - \mathbb{H}(X \mid Z, Y) = \mathbb{H}(Y \mid Z) - \mathbb{H}(Y \mid Z, X). &(7)
\end{aligned}$$

Examples. What are the mutual information between the following random variables?

- $\mathbb{I}(A\,;\,B)$: when A, B are independent and uniform over $\{0,1\}^n$;
- $\mathbb{I}(C\,;\,D)$: when C is uniform over $\{0,1\}^n$ and $D \in \{0,1\}$ is XOR of bits of D.
- $\mathbb{I}(E\,;\,F)$: when E is uniform over $\{0,1\}^n$ and F is the indicator random variable for $E = 0^n$.

Some of the important properties of mutual information is as follows.

Property 5 (**mutual information is non-negative**). *For any random variables X, Y,*

$$0 \leqslant \mathbb{I}(X\,;\,Y) \leqslant \min\{\mathbb{H}(X), \mathbb{H}(Y)\}\,;$$

moreover, the LHS inequality is tight iff $X \perp Y$.

Proof. The LHS is because $\mathbb{I}(X\,;\,Y) = \mathbb{H}(X) - \mathbb{H}(X \mid Y)$ and conditioning cannot increase entropy. The RHS is by non-negativity of entropy. The second part of the result also follows from Property (4). □

Property 6 (**chain rule of mutual information**). *For any random variables X, Y, Z,*

$$\mathbb{I}(X, Y\,;\,Z) = \mathbb{I}(X\,;\,Z) + \mathbb{I}(Y\,;\,Z \mid X).$$

(note that (X, Y) is one argument of the mutual information term as a joint variable and Z the other, which are separated by ';').

Proof. By the definition of mutual information and chain rule of entropy,

$$\begin{aligned}\mathbb{I}(X, Y\,;\,Z) &= \mathbb{H}(X, Y) - \mathbb{H}(X, Y \mid Z) = \mathbb{H}(X) + \mathbb{H}(Y \mid X) - \mathbb{H}(X \mid Z) - \mathbb{H}(Y \mid Z, X)\\ &= \mathbb{H}(X) - \mathbb{H}(X \mid Z) + \mathbb{H}(Y \mid X) - \mathbb{H}(Y \mid X, Z) = \mathbb{I}(X\,;\,Z) + \mathbb{I}(Y\,;\,Z \mid X).\end{aligned}$$

□

Unlike entropy, mutual information does not behave that straightforwardly under conditioning (which is to be expected from our intuition). For instance (you should prove both statements below):

- if X, Y are independent and uniform over $\{0,1\}$ and $Z = X \oplus Y$, then,

$$\mathbb{I}(X\,;\,Y) = 0 \quad \text{but} \quad \mathbb{I}(X\,;\,Y \mid Z) = 1;$$

- on the other hand, if $X = Y = Z$ and X is uniform over $\{0,1\}$, then,

$$\mathbb{I}(X\,;\,Y) = 1 \quad \text{but} \quad \mathbb{I}(X\,;\,Y \mid Z) = 0,$$

Still, there are many cases that we can say something interesting about the effect of conditioning on a mutual information term.

Property 7 (**conditioning on an independent random variable cannot decrease information**). *For any random variables X, Y, Z, if $X \perp Z$, then,*

$$\mathbb{I}(X\,;\,Y) \leqslant \mathbb{I}(X\,;\,Y \mid Z).$$

Proof. By the definition of mutual information and the fact that conditioning cannot increase entropy,

$$\begin{aligned}\mathbb{I}(X\,;\,Y \mid Z) &= \mathbb{H}(X \mid Z) - \mathbb{H}(X \mid Y, Z)\\ &= \mathbb{H}(X) - \mathbb{H}(X \mid Y, Z) && \text{(by moreover part of Property (4) since } X \perp Z)\\ &\geqslant \mathbb{H}(X) - \mathbb{H}(X \mid Y) && \text{(as conditioning on } Z \text{ cannot increase the entropy)}\\ &= \mathbb{I}(X\,;\,Y).\end{aligned}$$

□

Distance Measures Between Distributions

We will also review some measures of "distance/difference" between distributions.

For distributions P, Q on Ω:

- **Total Variation Distance** is defined as:

$$\|P - Q\|_{TVD} = \frac{1}{2} \sum_{x \in \Omega} |P(x) - Q(x)|.$$

The total variation distance nicely captures the difference between probabilities of different events across two distributions.

Property 8. *For any two distributions P, Q on Ω,*

$$\|P - Q\|_{TVD} = \max_{E \subseteq \Omega} |P(E) - Q(E)|.$$

(We note that the absolute value is not necessary in the definition above if $P(E) - Q(E)$ is negative, we can take $P(\bar{E}) - Q(\bar{E})$ instead).

Alternatively, total variation distance can also be characterized as the best probability of success for distinguishing a sample chosen uniformly from either of P or Q. That is,

Property 9. *Suppose s is sampled from either P or Q, chosen uniformly at random. Given s, the best probability of success for determining the original distribution is*

$$\frac{1}{2} \cdot (1 + \|P - Q\|_{TVD}).$$

- **KL Divergence** is defined as

$$\mathbb{D}(P \,\|\, Q) = E_{x \sim P}\left[\log \frac{P(x)}{Q(x)}\right] = E_{x \sim P}\left[\log \frac{1}{Q(x)}\right] - \mathbb{H}(P).$$

Intuitively, it (roughly) measures the following: suppose we are using an optimal encoding for a distribution Q, but it turns out that our input is sampled from the distribution P instead; then, how much overhead we have to pay compared to encoding the input correctly according to P. A nice property of KL-divergence is how closely it is related to mutual information.

Property 10. *For any random variables X, Y,*

$$\mathbb{I}(X\,;\,Y) = \mathop{\mathbb{E}}_{y}\left[\mathbb{D}(X \mid Y = y \,\|\, X)\right].$$

- **Hellinger Distance** is defined as

$$h(P, Q) = \sqrt{1 - \sum_{x \in \Omega} \sqrt{P(x)Q(x)}}.$$

The Hellinger distance is especially convenient in communication-complexity arguments because of the "product nature" of the two terms $P(x) \cdot Q(x)$ as we shall see later.

We can relate these measures as follows.

Proposition 8 (c.f. [Lin02])**.** *For any distributions P, Q:*

$$h^2(P, Q) \leqslant \|P - Q\|_{TVD} \leqslant \sqrt{2} \cdot h(P, Q);$$

$$\|P - Q\|_{TVD} \leqslant \sqrt{\frac{1}{2} \cdot \mathbb{D}(P \,\|\, Q)};$$

$$h^2(P, Q) \leqslant \frac{1}{2} \cdot \left(\mathbb{D}(P \,\|\, \frac{P+Q}{2}) + \mathbb{D}(Q \,\|\, \frac{P+Q}{2})\right).$$

Remark. This section was only a glance into the amazing area of information theory and in no ways can do justice to this field. Interested reader is referred to the excellent textbook of Cover and Thomas [CT06] as well as numerous courses on information theory tools in TCS for further background.

6 Back to the Proof of Theorem 4

We now return to the proof of Theorem 4. Again, consider the protocols π and γ in Lemma 6. We established that γ has the same probability of success in solving its underlying AND problem as does π for solving DISJ. The rest of the proof is now to show that γ is "n times more efficient" than π, in a certain "information cost" measure we define based on mutual information, and then we prove that any protocol for AND requires $\Omega(1)$ information cost which allows us to conclude the proof.

Recall the protocol γ from Lemma 6, where Y is the random input to AND and we use I to denote the random variable for the index used to embed Y into a random column of the $DISJ_{n,k}$ instance. Finally, let M_π and M_γ denote the messages communicated by the protocols π and γ, respectively, and R_π and R_γ be their public randomness. Finally, Θ is the random variable for the answer to AND and DISJ (recall that in our reduction these quantities will be the same)

The following lemma establishes the information cost reduction we hinted at earlier.

Lemma 9. *We have,*

$$\mathbb{I}(Y\,;\,M_\gamma \mid R_\gamma, \Theta = 1) \leqslant \frac{\|\pi\|}{n}.$$

In words, when $\theta = 1$, the information revealed about Y by the message M_γ, assuming we also know the public randomness R_γ, is at most $1/n$-th of the communication cost of π.

Proof of Lemma 9. The public randomness of γ is the index I which is uniform over $[n]$ plus whatever randomness used by π and its message is the same as that of π, i.e., $M_\gamma = M_\pi$. We thus have,

$$\begin{aligned}\mathbb{I}(Y\,;\,M_\gamma \mid R_\gamma, \Theta = 1) &= \mathbb{I}(Y\,;\,M_\gamma \mid R_\pi, I, \Theta = 1)\\ &= \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}(X_i\,;\,M_\pi \mid R_\pi, I = i, \Theta = 1)\end{aligned}$$

where X_i is the content of column i of the instance X of DISJ created in the reduction. Note that when $\Theta = 1$, the distribution of X_i is the same as all other indices and in general, knowing the underlying instance X, we still cannot determine the value of i; thus, the event $I = i$ is independent of all remaining variables and we can continue by dropping conditioning on it (as it will not change the distribution of underlying variables) and have,

$$\begin{aligned}&= \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}(X_i\,;\,M_\pi \mid R_\pi, \Theta = 1)\\ &\leqslant \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}(X_i\,;\,M_\pi \mid R_\pi, X_{<i}, \Theta = 1)\end{aligned}$$

where $X_{<i}$ are all columns in DISJ with index less than i; the inequality holds by Property (7) since $X_i \perp X_{<i} \mid R_\pi, \Theta = 1$, given all columns are sampled independently from $\{0,1\}^k$ when $\Theta = 1$. We can further apply the chain rule of mutual information (Property (6)),

$$= \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}(X_1, \ldots, X_n\,;\,M_\pi \mid R_\pi, \Theta = 1)$$

$$\leqslant \frac{1}{n} \cdot \mathbb{H}(M_\pi \mid R_\pi, \Theta = 1) \qquad \text{(by Property (5))}$$
$$\leqslant \frac{1}{n} \cdot \mathbb{H}(M_\pi) \leqslant \frac{1}{n} \cdot \|\pi\|, \qquad \text{(by Property (4) and Property (1))}$$

where at the end, we used the fact that the log of the total number of different messages possible in the protocol is upper bounded by its communication cost (assuming all messages are of the same length which is without loss of generality by padding). This concludes the proof. □

We thus established the main direct-sum part of the argument: solving AND has n times less information cost compared to the communication complexity of DISJ. However, while it is trivial to prove that communication cost of any protocol for AND needs to be at least one bit, it is not clear if their information cost needs to be $\Omega(1)$ also (which is needed to conclude our proof). The next step of the proof, that gets somewhat technical and possibly less conceptual than the first step, will establish this part.

Lemma 10. *For any protocol γ for AND (on its distribution) that succeeds with probability at least 0.99,*

$$\mathbb{I}(Y\,;\, M_\gamma \mid R_\gamma, \Theta = 1) = \Omega(1).$$

To establish this lemma, we need various properties of communication protocols especially when it comes to Hellinger distance. But, instead of piling all these definitions and properties upfront, we will mention each one when we reach them. These properties were first proved in [BYJKS02].

For the rest of the proof, we drop the subscripts γ from M_γ and R_γ and simply denote them by M and R. We further write M_r for a fixed choice of public randomness r to denote the random variable for the messages of players conditioned on $R = r$ (the remaining randomness is from the private randomness of the protocol and the input).

We first have an easy step that allows us to ignore the public randomness from hereon.

Claim 11. *There exists a choice of public randomness r such that*

$$\mathbb{I}(Y\,;\, M_r \mid \Theta = 1) \leqslant 2 \cdot \mathbb{I}(Y\,;\, M \mid R, \Theta = 1), \quad \textit{and}$$
$$\Pr\left(M_r \textit{ outputs the correct answer}\right) \geqslant \frac{1}{50}.$$

Proof. By the definition of conditional mutual information,

$$\mathbb{I}(Y\,;\, M \mid R, \Theta = 1) = \mathop{\mathbb{E}}_r \left[\mathbb{I}(Y\,;\, M \mid R = r, \Theta = 1)\right] =: \mu,$$

and thus by Markov bound,

$$\Pr_r\left(\mathbb{I}(Y\,;\, M \mid R = r, \Theta = 1) > 2 \cdot \mu\right) < \frac{1}{2}.$$

Similarly,

$$\frac{1}{100} \geqslant \Pr\left(M \text{ is wrong}\right) = \mathop{\mathbb{E}}_r \left[\Pr\left(M_r \text{ is wrong} \mid R = r\right)\right] = \mathop{\mathbb{E}}_r \left[\Pr\left(M_r \text{ is wrong}\right)\right],$$

since public randomness is independent of the input. Thus, again, by Markov bound,

$$\Pr_r\left(\Pr\left(M_r \text{ is wrong}\right) > \frac{1}{50}\right) < \frac{1}{2}.$$

By union bound, there exists a choice of r that neither of the two events happen, concluding the proof. □

For the rest of the proof, we fix a choice of r as in Claim 11 and only consider the protocol π_r which is the same as π except its public randomness is fixed to be r. Thus, its message is M_r and since all other variables are independent of the public randomness, they continue to have their old distribution. From now on then, we are working with a private randomness protocols only.

For $y \in \{0,1\}^k$, denote $\bar{y} = y \oplus \mathbf{1}$, namely, by negating each bit of y. We partition all 2^k possible $y \in \{0,1\}^k$ into 2^{k-1} pairs of the form $(y, \bar{y})$. To avoid double counting, when writing a pair in (z_0, z_1), we always assume the first bit of the string z_0 is 0. I.e., any string $y = 0*$ appears in the pair $(y, \bar{y})$ whereas any string $y' = 1*$ appears in the pair $(\bar{y'}, y')$.

For our original input Y, we think of sampling Y as first sampling a pair $Z = (Z_0, Z_1)$ and then letting Y be Z_0 or Z_1 uniformly at random. For a fixed input $y \in \{0,1\}^k$, we write $M_r(y)$ to denote the random variable of the messages of π_r when run on the input y (the randomness here is now only based on the private randomness of the protocol).

Claim 12. *There exists an input $y \in \{0,1\}^k$ such that*

$$h^2(M_r(y)\,,\, M_r(\bar{y})) \leqslant \mathbb{I}(Y\,;\, M_r \mid \Theta = 1).$$

Proof. We have

$$\begin{aligned}
\mathbb{I}(Y\,;\, M_r \mid \Theta = 1) &= \mathbb{I}(Y, Z\,;\, M_r \mid \Theta = 1) && \text{(as } Z \text{ is deterministically fixed by } Y\text{)} \\
&= \mathbb{I}(Z\,;\, M_r \mid \Theta = 1) + \mathbb{I}(Y\,;\, M_r \mid Z, \Theta = 1) \\
& && \text{(by the chain rule of mutual information (Property (6)))} \\
&\geqslant \mathop{\mathbb{E}}_{z \mid \Theta=1} \left[\mathbb{I}(Y\,;\, M_r \mid Z = z, \Theta = 1)\right] \\
& && \text{(by the non-negativity of mutual information and its (conditional) definition)} \\
&= \mathop{\mathbb{E}}_{z \mid \Theta=1} \left[\mathop{\mathbb{E}}_{y \mid z, \Theta=1} \mathbb{D}\left(M_r \mid Y = y, Z = z, \Theta = 1 \,\|\, M_r \mid Z = z, \Theta = 1\right)\right] \\
& && \text{(by the connection between KL-divergence and mutual information (Property (10)))} \\
&= \mathop{\mathbb{E}}_{z \mid \Theta=1} \left[\frac{1}{2} \cdot \mathbb{D}\left(M_r(z_0) \,\|\, \frac{M_r(z_0) + M_r(z_1)}{2}\right) + \frac{1}{2} \cdot \mathbb{D}\left(M_r(z_1) \,\|\, \frac{M_r(z_0) + M_r(z_1)}{2}\right)\right]
\end{aligned}$$

as conditioned on $Z = z$, the choice of y is uniform over z_0 and z_1, and that conditioned on y, M_r simply becomes $M_r(y)$ and no longer depends on $Z = z$ and $\Theta = 1$. Continuing, we have,

$$\geqslant \mathop{\mathbb{E}}_{z \mid \Theta=1} \left[h^2(M_r(z_0), M_r(z_1))\right]. \qquad \text{(by Proposition 8)}$$

Thus, by the averaging argument, there exists a choice of $z = (y, \bar{y})$ satisfying the statement in the claim. □

The RHS of Claim 12, by Claim 11, is effectively the information cost bound we would like to bound in Lemma 10. Thus, if the RHS is $o(1)$, we will have an input y such that $h^2(M_r(y)\,,\, M_r(\bar{y})) = o(1)$, namely, the messages sent over $M_r(y)$ and $M_r(\bar{y})$ are quite similar. A priori, this does not seem contradictory, because the protocol can have the same response for all pairs $(y, \bar{y})$ as long as neither one are all-1 (all such cases correspond to the $\Theta = 1$ case). The following lemma however shows that if $h^2(M_r(y)\,,\, M_r(\bar{y})) = o(1)$, then $h^2(M_r(\mathbf{0})\,,\, M_r(\mathbf{1})) = o(1)$ also. The proof is quite similar to the rectangle property of (deterministic) communication protocols in the previous lecture, but now for randomized protocols using Hellinger distance.

Claim 13. *For any choice of $y \in \{0,1\}^k$: $h(M_r(y)\,,\, M_r(\bar{y})) = h(M_r(\mathbf{0})\,,\, M_r(\mathbf{1}))$.*

Proof. To avoid clutter in the notation, we drop the subscript r from M_r and simply denote the set of messages by M. We further write $M = M_1, M_2, \ldots, M_c$ where for $b \in [c]$, M_b denotes the b-th bit communicated in the protocol. We have

$$\begin{aligned}
&1 - h^2(M(y)\,,\, \bar{M}(\bar{y})) \\
&= \sum_m \sqrt{\Pr\left(M = m \mid Y = y\right) \cdot \Pr\left(M = m \mid Y = \bar{y}\right)} && \text{(by the definition of Hellinger distance)}
\end{aligned}$$

$$= \sum_m \sqrt{\prod_{b=1}^{c} \Pr\left(M_b = m_b \mid M_{<b} = m_{<b}, Y = y\right) \cdot \Pr\left(M_b = m_b \mid M_{<b} = m_{<b}, Y = \bar{y}\right)}$$

(by chaining the probabilities)

$$= \sum_m \sqrt{\prod_{b=1}^{c} \Pr\left(M_b = m_b \mid M_{<b} = m_{<b}, Y_{p(b)} = y_{p(b)}\right) \cdot \Pr\left(M_b = m_b \mid M_{<b} = m_{<b}, Y_{p(b)} = \bar{y}_{p(b)}\right)}$$

where $p(b) \in [k]$ denotes the player sending the b-th bit of the message; the equality holds because the message M_b only depends on prior messages received by $p(b)$ and the input of this player, not the entire players. Continuing, we have,

$$= \sum_m \sqrt{\prod_{b=1}^{c} \Pr\left(M_b = m_b \mid M_{<b} = m_{<b}, Y_{p(b)} = 0\right) \cdot \Pr\left(M_b = m_b \mid M_{<b} = m_{<b}, Y_{p(b)} = 1\right)}$$

by re-ordering the $y_{p(b)} = 0$ terms to the beginning and $y_{p(b)} = 1$ to the second term; this is the step we crucially use the "product definition" of the Hellinger distance; we further have

$$= \sum_m \sqrt{\Pr\left(M = m \mid Y = \mathbf{0}\right) \cdot \Pr\left(M = m \mid Y = \mathbf{1}\right)}$$
$$= 1 - h^2(M(\mathbf{0}), M(\mathbf{1})).$$

This concludes the proof. □

We note that as can be seen by the proof of Claim 13, we can prove a considerably more general version also that allows for exchanging any subset of the inputs of players with each other and not necessarily restricted to the case when LHS consists of messages for $y, \bar{y}$, but any types of messages. However, for our purpose, the above lemma suffices.

At this stage, we seem to be done: the answer of a correct protocol for AND cannot be the same on inputs $\mathbf{0}$ and $\mathbf{1}$ and thus the LHS should be $\Omega(1)$ which in turn makes the RHS $\Omega(1)$ and by our previous results establishes Lemma 10. Unfortunately however, this is not actually the case (yet): we are analyzing our protocol π_r over a distribution and errs with probability at most $1/50$; thus, the protocol is fully allowed to make an error on the input $\mathbf{0}$ which only happens with probability $2^{-k} \ll 1/50$. As such, we need one further step to address this part as well.

The following claim allows us to argue that distance between $M_r(\mathbf{0})$ and $M_r(\mathbf{1})$ is approximately the largest distance between any $M_r(y)$ and $M_r(\mathbf{1})$ (we will then use this to say that certainly not *all* of $y \neq \mathbf{1}$ can have similar messages as $M_r(\mathbf{1})$ as otherwise the answer cannot be correct overall with large enough probability).

Claim 14. *For any choice of $y \in \{0,1\}^k$: $h^2(M_r(y), M_r(\mathbf{1})) \leqslant 2h^2(M_r(\mathbf{0}), M_r(\mathbf{1}))$.*

Proof. Similar to the proof of Claim 13, we drop the subscript r from M_r and write the individual bits of M by $M_1, \ldots, M_c$. Fix any $y \in \{0,1\}^k$. We will actually prove the following stronger statement, which implies the original claim by non-negativity of the Hellinger distance:

$$h^2(M_r(y), M_r(\mathbf{1})) + h^2(M_r(\bar{y}), M_r(\mathbf{0})) \leqslant 2h^2(M_r(\mathbf{0}), M_r(\mathbf{1})).$$

We have

$$h^2(M_r(y), M_r(\mathbf{1})) + h^2(M_r(\bar{y}), M_r(\mathbf{0}))$$
$$= 2 - \left(\sum_m \sqrt{\Pr\left(M = m \mid Y = y\right) \Pr\left(M = m \mid Y = \mathbf{1}\right)} + \sum_m \sqrt{\Pr\left(M = m \mid Y = \bar{y}\right) \Pr\left(M = m \mid Y = \mathbf{0}\right)} \right).$$

Let $S \subseteq [k]$ be the indices in the support of y, i.e., $i \in S$ iff $y_i = 1$. As we proved in Claim 13, we have,

$$\Pr\left(M = m \mid Y = y\right) = \prod_{b=1}^{c} \Pr\left(M_b = m_b \mid M_{<b} = m_{<b}, Y_{p(b)} = y_{p(b)}\right) = q(m, \mathbf{1}_S) \cdot q(m, \mathbf{0}_{\bar{S}}),$$

for some functions q (by partitioning the terms corresponding to $y_i = 1$ for $i \in S$ and $y_i = 0$ for $i \notin S$). This is similarly true for all other inputs in our equations earlier also which allows us to write

$$\begin{aligned}
&h^2(M_r(y)\,,\, M_r(\mathbf{1})) + h^2(M_r(\bar{y})\,,\, M_r(\mathbf{0})) \\
&= 2 - \left(\sum_m \sqrt{q(m, \mathbf{1}_S) \cdot q(m, \mathbf{0}_{\bar{S}}) \cdot q(m, \mathbf{1}_S) \cdot q(m, \mathbf{1}_{\bar{S}})} + \sum_m \sqrt{q(m, \mathbf{0}_S) \cdot q(m, \mathbf{1}_{\bar{S}}) \cdot q(m, \mathbf{0}_S) \cdot q(m, \mathbf{0}_{\bar{S}})}\right) \\
&= 2 - \left(\sum_m q(m, \mathbf{1}_S) \cdot \sqrt{q(m, \mathbf{0}_{\bar{S}}) \cdot q(m, \mathbf{1}_{\bar{S}})} + \sum_m q(m, \mathbf{0}_S) \sqrt{q(m, \mathbf{1}_{\bar{S}}) \cdot q(m, \mathbf{0}_{\bar{S}})}\right) \\
&\qquad\qquad \text{(by factoring out the quadratic terms)} \\
&= 2 - \left(\sum_m (q(m, \mathbf{1}_S) + q(m, \mathbf{0}_S)) \cdot \sqrt{q(m, \mathbf{0}_{\bar{S}}) \cdot q(m, \mathbf{1}_{\bar{S}})}\right) \qquad \text{(by combining same multiplicative terms)} \\
&\leqslant 2 - \left(\sum_m \left(2\sqrt{q(m, \mathbf{1}_S) \cdot q(m, \mathbf{0}_S)}\right) \cdot \sqrt{q(m, \mathbf{0}_{\bar{S}}) \cdot q(m, \mathbf{1}_{\bar{S}})}\right) \qquad \text{(by the AM-GM inequality)} \\
&= 2 \cdot \left(1 - \sum_m \left(\sqrt{q(m, \mathbf{1}_S) \cdot q(m, \mathbf{1}_{\bar{S}}) \cdot q(m, \mathbf{0}_S) \cdot q(m, \mathbf{0}_{\bar{S}})}\right)\right) \qquad \text{(by re-ordering the terms)} \\
&= 2 \cdot \left(1 - \sum_m \sqrt{\Pr\left(M = m \mid Y = \mathbf{1}\right) \cdot \Pr\left(M = m \mid Y = \mathbf{0}\right)}\right) \qquad \text{(by the definition of } q\text{-functions)} \\
&= 2 \cdot h^2(M_r(\mathbf{0})\,,\, M_r(\mathbf{1})),
\end{aligned}$$

concluding the proof. □

Finally, we claim that for the protocol to be able to be true, its distribution of messages on $M_r(\mathbf{1})$ has to differ from at least some $M_r(y)$ which will be sufficient for us using the established claims.

Claim 15. *There exists some $y \in \{0,1\}^k$ such that $h^2(M_r(y)\,,\, M_r(\mathbf{1})) \geqslant 2/5$.*

Proof. By Claim 11, we have that

$$\begin{aligned}
\frac{49}{50} &\leqslant \Pr\left(\pi_r \text{ outputs correctly}\right) = \frac{1}{2} \cdot \Pr\left(\pi_r \text{ outputs } 1 \mid \Theta = 1\right) + \frac{1}{2} \cdot \Pr\left(\pi_r \text{ outputs } 0 \mid \Theta = 0\right) \\
&= \frac{1}{2} \cdot \mathop{\mathbb{E}}_{y \in_R \{0,1\}^k} \Pr\left(M_r(y) \text{ has output } 1\right) + \frac{1}{2} \cdot \Pr\left(M_r(\mathbf{1}) \text{ has output } 0\right).
\end{aligned}$$

This implies that

$$\mathop{\mathbb{E}}_{y \in_R \{0,1\}^k} \Pr\left(M_r(y) \text{ has output } 1\right) \geqslant \frac{48}{50} \quad \text{and} \quad \Pr\left(M_r(\mathbf{1}) \text{ has output } 1\right) \leqslant \frac{2}{50}.$$

This implies that there exists a choice of $y \in \{0,1\}^k$ such that

$$\left|\Pr\left(M_r(y) \text{ has output } 1\right) - M_r(\mathbf{1}) \text{ has output } 1\right| \geqslant \frac{46}{50}.$$

By Property (8), this implies that

$$\|M_r(y) - M_r(\mathbf{1})\|_{TVD} \geqslant \left|\Pr\left(M_r(y) \text{ has output } 1\right) - M_r(\mathbf{1}) \text{ has output } 1\right| \geqslant \frac{46}{50}.$$

This, combined with Proposition 8 which says

$$h(M_r(y)\,,\,M_r(\mathbf{1})) \geqslant \frac{1}{\sqrt{2}} \cdot \|M_r(y) - M_r(\mathbf{1})\|_{TVD},$$

implies that

$$h^2(M_r(y)\,,\,M_r(\mathbf{1})) \geqslant \frac{1}{2} \cdot \left(\frac{46}{50}\right)^2 > \frac{2}{5},$$

finalizing the proof □

Putting everything together, we can now easily prove Lemma 10.

Proof of Lemma 10. We have

$$\begin{aligned}
\mathbb{I}(Y\,;\,M \mid R, \Theta = 1) &\geqslant \frac{1}{2} \cdot \mathbb{I}(Y\,;\,M_r \mid \Theta = 1) && \text{(by Claim 11)}\\
&\geqslant \frac{1}{2} \cdot h^2(M_r(y)\,,\,M_r(\bar{y})) && \text{(by Claim 12)}\\
&= \frac{1}{2} \cdot h^2(M_r(\mathbf{0})\,,\,M_r(\mathbf{1})) && \text{(by Claim 13)}\\
&\geqslant \frac{1}{4} \cdot h^2(M_r(y)\,,\,M_r(\mathbf{1})) && \text{(by Claim 14 for the particular } y \text{ in Claim 15)}\\
&\geqslant \frac{1}{4} \cdot \frac{2}{5} = \frac{1}{10}, && \text{(by Claim 15)}
\end{aligned}$$

concluding the proof. □

Theorem 4 follows immediately from Lemma 6 and Lemma 10.

References

[BYJKS02] Z. Bar-Yossef, T.S. Jayram, R. Kumar, and D. Sivakumar. An information statistics approach to data stream and communication complexity. In *The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings.*, pages 209–218, 2002. 14

[CT06] Thomas M. Cover and Joy A. Thomas. *Elements of information theory (2. ed.)*. Wiley, 2006. 13

[Din24] Itai Dinur. Time-space lower bounds for bounded-error computation in the random-query model. In David P. Woodruff, editor, *Proceedings of the 2024 ACM-SIAM Symposium on Discrete Algorithms, SODA 2024, Alexandria, VA, USA, January 7-10, 2024*, pages 2900–2915. SIAM, 2024. 2, 4

[Lin02] Jianhua Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information theory*, 37(1):145–151, 2002. 12

[RZ20] Ran Raz and Wei Zhan. The Random-Query Model and the Memory-Bounded Coupon Collector. In Thomas Vidick, editor, *11th Innovations in Theoretical Computer Science Conference (ITCS 2020)*, volume 151 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 20:1–20:11, Dagstuhl, Germany, 2020. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. 1, 2