



Direct Sums in Randomized Communication Complexity

Boaz Barak* Mark Braverman Xi Chen Anup Rao

May 6, 2009

Abstract

We prove a direct sum theorem for randomized communication complexity. Ignoring logarithmic factors, our results show that:

- Computing n copies of a function requires \sqrt{n} times the communication.
- For average case complexity, given any distribution μ on inputs, computing n copies of the function on n independent inputs sampled according to μ requires \sqrt{n} times the communication for computing one copy.
- If μ is a product distribution, computing n copies on n independent inputs sampled according to μ requires n times the communication.

We also study the complexity of computing the parity of n evaluations of f , and obtain results analogous to those above.

Our results are obtained by designing new compression schemes that can compress the communication in interactive processes that do not reveal too much information about their inputs. This generalizes the notion of traditional compression, which can be viewed as compressing protocols that involve only one way communication.

*Department of Computer Science, Princeton University, boaz@cs.princeton.edu. Supported by NSF grants CNS-0627526, CCF-0426582 and CCF-0832797, US-Israel BSF grant 2004288 and Packard and Sloan fellowships.

1 Introduction

Does computing n copies of a function require n times the computational effort? In this paper, we give the first non-trivial answer to this question for the model of randomized communication complexity.

Communication complexity measures the complexity of a function $f(x, y)$ in terms of the number of bits that two parties need to communicate with each other to determine the value of the function, if each party knows one of the inputs. The study of communication complexity has had many applications. We refer the reader to the book [KN97] for an introduction.

Known results prior to our work applied to more restricted models of communication. Feder et al. [FKNN95] showed that in the model of *deterministic* communication complexity, where the participants are not allowed to use random bits, computing n copies requires at least $n\sqrt{c}$ bits of communication, where c is the deterministic communication complexity of one copy of the function. Earlier results were obtained for the randomized model under strong restrictions, for example restricting the number of rounds of communication to be a constant [CSWY01, JRS03, JRS05, HJMR07].

It turns out that the direct sum question is related to the problem of compressing the communication in an interactive protocol. It is well known how to compress a message from a distribution X so that the average length of the message is the Shannon entropy $H(X)$. In this paper we address the analogous question for the case of interactive communication protocols. A single message is simply the case of an interactive protocol with one way communication.

1.1 Our Results

1.2 The Direct Sum Question

We give new bounds on the randomized communication complexity of the direct sum of any function.

Given a function $f : \mathcal{X} \times \mathcal{Y} \rightarrow \mathcal{Z}$, we define the function $f^n : \mathcal{X}^n \times \mathcal{Y}^n \rightarrow \mathcal{Z}^n$ to be the concatenation of the evaluations:

$$f^n(x_1, \dots, x_n, y_1, \dots, y_n) \stackrel{def}{=} (f(x_1, y_1), f(x_2, y_2), \dots, f(x_n, y_n)).$$

Denote by $R_\rho(f)$ the communication complexity of the best randomized public coin protocol for computing f that errs with probability at most ρ . In this paper we show:

Theorem 1.1 (Direct Sum for Randomized Communication Complexity). *For every $\alpha > 0$,*

$$R_\rho(f^n) \cdot \log(R_\rho(f^n)/\alpha) \geq \Omega(R_{\rho+\alpha}(f)\alpha\sqrt{n})$$

[Theorem 1.1](#) is obtained using Yao's min-max principle from an analogous theorem for *distributional* communication complexity. For a distribution μ on the inputs $\mathcal{X} \times \mathcal{Y}$, we write $D_\rho^\mu(f)$ to denote the communication complexity of the best protocol (randomized or deterministic) that computes f with probability of error at most ρ when the inputs are sampled according to μ . We write μ^n to denote the distribution on n inputs, where each is sampled according to μ independently. Our direct sum for result for distributional complexity is stated in the following theorem:

Theorem 1.2 (Direct Sum for Distributional Communication Complexity).

$$D_\rho^{\mu^n}(f^n) \cdot \log(D_\rho^{\mu^n}(f^n)/\alpha) \geq \Omega(D_{\rho+\alpha}^\mu(f)\alpha\sqrt{n})$$

The communication complexity bound of [Theorem 1.2](#) only grows as the square root of the number of repetitions. However, in the case that the distribution on inputs is a product distribution, we obtain a direct sum theorem that is optimal up to logarithmic factor:

Theorem 1.3 (Direct Sum for Product Distributions). *If μ is a product distribution, then*

$$D_{\rho}^{\mu^n}(f^n) \cdot \text{polylog}(D_{\rho}^{\mu^n}(f^n)/\rho) \geq \Omega\left(D_{4\rho}^{\mu}(f)\rho n\right)$$

1.3 XOR Lemma for communication complexity

When n is very large in terms of the other quantities, the above theorems can be superseded by trivial arguments, since f^n must require at least n bits of communication just to describe the output. Our next set of theorems show that almost the same bounds apply to the complexity of the XOR (or more generally sum modulo K) of n copies of f , where the trivial arguments do not hold. Assume that the output of the function f is in the ring \mathbb{Z}_K for some integer K , and define

$$f^{+n}(x_1, \dots, x_n, y_1, \dots, y_n) \stackrel{\text{def}}{=} \sum_{i=1}^n f(x_i, y_i).$$

We have the following results for the complexity of f^{+n} :

Theorem 1.4 (XOR Lemma for Randomized Communication Complexity).

$$R_{\rho}(f^{+n}) \cdot \log(R_{\rho}(f^{+n})/\alpha) \geq \Omega\left(\left(R_{\rho+\alpha}(f) - 2\log K\right)\alpha\sqrt{n}\right)$$

Theorem 1.5 (XOR Lemma for Distributional Communication Complexity).

$$D_{\rho}^{\mu^n}(f^{+n}) \cdot \log(D_{\rho}^{\mu^n}(f^{+n})/\alpha) \geq \Omega\left(\left(D_{\rho+\alpha}^{\mu}(f) - 2\log K\right)\alpha\sqrt{n}\right)$$

Theorem 1.6 (XOR Lemma for Product Distributions). *If μ is a product distribution, then*

$$D_{\rho}^{\mu^n}(f^{+n}) \cdot \text{polylog}(D_{\rho}^{\mu^n}(f^{+n})/\rho) \geq \Omega\left(\left(D_{4\rho}^{\mu}(f) - 2\log K\right)\rho n\right)$$

Remark 1.7. If $f : \mathbb{Z}_K \times \mathbb{Z}_K \rightarrow \mathbb{Z}_K$ is itself the sum function, then the communication complexity of f^{+n} does not grow at all, since there is a trivial protocol to compute $\sum_i(x_i + y_i) = \sum_i x_i + \sum_j y_j$ using $2\log K$ bits. This suggests that some kind of additive loss (like the $\log K$ term above) is necessary in the above theorems.

1.4 Compressing Communication Protocols

An important step towards proving our results is something that may be of independent interest. We give new way to compress the communication in a communication protocol. Essential to our results is an information theory based measure of the complexity of a protocol that we call the *information content* of a protocol. Given a protocol π and random variables that denote inputs X, Y , let $\pi(X, Y)$ denote the random variable of the concatenation of the public randomness in the protocol with the messages that are transmitted when it is run. One natural way to measure the information revealed by a protocol, used by several earlier works [[CSWY01](#), [BYJKS04](#), [SS02](#), [Ab193](#)], is to measure the mutual information of the messages and public randomness in the protocol with the inputs, i.e. $I(XY; \pi(X, Y))$. The measure we use is slightly different, though it turns out to be the same as the one above when the inputs X, Y are independent of each other:¹

¹The measure we use here was also considered by Bar-Yossef et al. [[BYJKS04](#)] but was not explicitly defined.

Definition 1.8. Given a distribution μ on inputs X, Y , and protocol π , we call the quantity

$$IC_\mu(\pi) \stackrel{def}{=} I(X; \pi(X, Y)|Y) + I(Y; \pi(X, Y)|X)$$

the *information content* of π .

The first term in the above sum intuitively measures the information about X contained in the messages that the second player *doesn't already know*. Similarly, the second term measures the information about Y that is revealed to the player holding X that she didn't already know. Note that the information content of a protocol can be very different from the mutual information of the messages and the inputs. For example, if μ is a distribution where $X = Y$ always, then the information content is always 0, though the mutual information between the messages and the inputs can be very large. However, it is easy to check that $IC_\mu(\pi) = I(XY; \pi(X, Y))$, in the case that μ is a product distribution. It is also easy to check that if π is deterministic, then $IC_\mu(\pi) = H(\pi(X, Y)|Y) + H(\pi(X, Y)|X)$, which is the same as $H(\pi(X, Y))$ if X, Y are independent.

We shall then prove:

Theorem 1.9. *For every distribution μ , every protocol π , and every $\epsilon > 0$, there exists functions π_x, π_y , and a protocol τ such that $|\pi_x(X, \tau(X, Y)) - \pi(X, Y)| < \epsilon$, $\Pr[\pi_x(X, \tau(X, Y)) \neq \pi_y(Y, \tau(X, Y))] < \epsilon$ and $CC(\tau) \leq \sqrt{CC(\pi) \cdot IC_\mu(\pi)} \frac{\log(CC(\pi)/\epsilon)}{\epsilon}$.*

The condition $|\pi_x(X, \tau(X, Y)) - \pi(X, Y)| < \epsilon$ ensures that the transcript of τ specifies a unique leaf in the protocol tree for π in such a way that this leaf is ϵ -close to the leaf sampled by π . The condition that $\Pr[\pi_x(X, \tau(X, Y)) \neq \pi_y(Y, \tau(X, Y))] < \epsilon$ guarantees that with high probability both players achieve a consensus on what the sampled leaf was. Thus, the triple τ, π_x, π_y specify a new protocol that can be viewed as a compression of π .

Thus, our results can be viewed as some kind of generalization of the traditional notion of compression, which applies to the more restricted case of deterministic one way protocols.

1.5 Techniques

The high level approach that we use was introduced in the work of Chakrabarti, Shi, Wirth and Yao [CSWY01]. We shall focus on the approach to proving [Theorem 1.2](#) (direct sum for distributional complexity), since this is where most of the challenges arise. We prove [Theorem 1.2](#) via a reduction — we first show that if there is a protocol that computes f^n (resp. f^{+n}) under the distribution μ^n with small communication complexity, then we can use it to obtain a protocol that computes a single copy with small information content (though its communication complexity remains as large as in the original protocol). The following theorem was implicit in the work of Bar-Yossef et al. [BYJKS04], though for completeness we shall write the proof in this paper.

Theorem 1.10. *For every μ, f, ρ there exists a protocol τ computing f on inputs drawn from μ with probability of error at most ρ and communication at most $D_\rho^{\mu^n}(f^n)$ such that $IC_\mu(\tau) \leq \frac{2D_\rho^{\mu^n}(f^n)}{n}$.*

The key idea involved in proving the above theorem is a way to split dependencies between the inputs that arose in the study of lowerbounds for the communication complexity of disjointness and in the study of parallel repetition [KS92, Raz92, Raz98]. We give the proof in [Section 4](#).

Once we have a theorem of the above type, we show how to take any protocol whose information content is small, and compress it. There are several challenges that need to be overcome in doing

this. An interesting case to consider is a protocol where the players alternate sending each other messages, and transmitted message is just a bit with information content $\epsilon \ll 1$. In this case, we cannot afford to even transmit one bit to simulate each of the messages, since that would incur an overhead of $1/\epsilon$, which would be too large for our application. This barrier was one of the big stumbling blocks for earlier works, which is why their results applied only when the number of rounds in the protocols were forced to be small.

We give two protocols to solve this problem. Our more efficient solution, which gives a protocol with communication complexity within polylogarithmic factors of the information content only applies when the input distribution μ is a product distribution. The idea here is to have the players use shared randomness to guess entire blocks of messages that they might have exchanged. Given a particular sequence of messages, the players can then communicate to estimate the correct probability with which this sequence was supposed to occur. The players can then either accept the sequence or resample a new sequence in order to get a final sample that behaves in a way that is close to the distribution of the original protocol. There are several technical challenges involved in getting this to work. The fact that the inputs of the players are independent is important for the players to decide how many messages the players should try to sample at once. When the players' inputs are dependent, they cannot estimate how many messages they should sample before the information content becomes too high, and we are unable to make this approach work.

In the above simulation, the sampled messages will not come from a distribution that is the same as the original one, or even close to it. Instead, what we are able to guarantee is that no particular outcome is more than a constant factor (say 4 times) more likely than in the original protocol. Thus, if the original protocol had a probability of error of at most ρ , our simulation must have a probability of error of at most 4ρ .

Our general solution, that applies even to non-product distributions, is quite different. In this simulation, we have each of the players sample each of their potential transmissions ahead of time. In other words, for every prefix v of messages, each player samples the next bit of the interaction according to the best guess that they have for how this bit is distributed. The players do this using shared randomness, in a way that guarantees that if their guesses are close, then the probability that they sample the same bit is high. Once they have each sampled the possible interactions, we show how the players can communicate a few bits with each other to resolve the inconsistencies in their samples in such a way that the final outcome is actually statistically close to the distribution of the original protocol. Unfortunately, in this case, the cost of the simulation is not close to the information content of the original protocol, but can only be bounded in terms of the geometric mean between the information content and the communication complexity of the original protocol.

We get around this issue by having the players each guess the *entire transcript* of the protocol in a correlated way, using public randomness. Since each player does not have enough information to correctly sample a transcript, the players will necessarily make many mistakes in their guesses. The players will then communicate with each other to fix the inconsistencies in their transcripts, until they have converged to a transcript that is consistent with both of their inputs.

2 Preliminaries

Notation. We reserve capital letters for random variables and distributions, calligraphic letters for sets, and small letters for elements of sets. Throughout this paper, we often use the notation $|b$ to denote conditioning on the event $B = b$. Thus $A|b$ is shorthand for $A|B = b$. Given a sequence

of symbols $A = A_1, A_2, \dots, A_k$, we use $A_{\leq j}$ denote the prefix of length j .

We use the standard notion of *statistical/ total variation* distance between two distributions.

Definition 2.1. Let D and F be two random variables taking values in a set \mathcal{S} . Their *statistical distance* is

$$|D - F| \stackrel{\text{def}}{=} \max_{\mathcal{T} \subseteq \mathcal{S}} (|\Pr[D \in \mathcal{T}] - \Pr[F \in \mathcal{T}]|) = \frac{1}{2} \sum_{s \in \mathcal{S}} |\Pr[D = s] - \Pr[F = s]|$$

If $|D - F| \leq \epsilon$ we shall say that D is ϵ -close to F . We shall also use the notation $D \stackrel{\epsilon}{\approx} F$ to mean D is ϵ -close to F .

2.1 Information Theory

Definition 2.2 (Entropy). The *entropy* of a random variable X is $H(X) \stackrel{\text{def}}{=} -\sum_x \Pr[X = x] \log(1/\Pr[X = x])$. The *conditional entropy* $H(X|Y)$ is defined to be $\mathbb{E}_{y \in \mathcal{R}_Y} [H(X|Y = y)]$.

Fact 2.3. $H(AB) = H(A) + H(B|A)$.

Definition 2.4 (Mutual Information). The *mutual information* between two random variables A, B , denoted $I(A; B)$ is defined to be the quantity $H(A) - H(A|B) = H(B) - H(B|A)$. The *conditional mutual information* $I(A; B|C)$ is $H(A|C) - H(A|BC)$.

In analogy with the fact that $H(AB) = H(A) + H(B|A)$,

Proposition 2.5. Let C_1, C_2, D, B be random variables. Then

$$I(C_1 C_2; B|D) = I(C_1; B|D) + I(C_2; B|C_1 D).$$

The previous proposition immediately implies the following:

Proposition 2.6 (Super-Additivity of Mutual Information). Let C_1, C_2, D, B be random variables such that for every fixing of D , C_1, C_2 are independent. Then

$$I(C_1; B|D) + I(C_2; B|D) \leq I(C_1 C_2; B|D).$$

We also use the notion of *divergence*, which is a different way to measure the distance between two distributions:

Definition 2.7 (Divergence). The informational divergence between two distributions is $\mathbb{D}(A||B) \stackrel{\text{def}}{=} \sum_x A(x) \log(A(x)/B(x))$.

For example, if B is the uniform distribution on $\{0, 1\}^n$ then $\mathbb{D}(A||B) = n - H(A)$.

Proposition 2.8. $\mathbb{D}(A||B) \geq |A - B|^2$.

Proposition 2.9. Let A, B, C be random variables in the same probability space. For every a in the support of A and c in the support of C , let B_a denote $B|A = a$ and B_{ac} denote $B|A = a, C = c$. Then $I(A; B|C) = \mathbb{E}_{a, c \in \mathcal{R}_{A, C}} [\mathbb{D}(B_{ac}||B_c)]$

The above facts imply the following easy proposition:

Proposition 2.10. *With notation as in Proposition 2.9, for any random variables A, B , $\mathbb{E}_{a \in_{\mathbb{R}} A} [|B_a - B|] \leq \sqrt{I(A; B)}$.*

Proof.

$$\begin{aligned} \mathbb{E}_{a \in_{\mathbb{R}} A} [|B_a - B|] &\leq \mathbb{E}_{a \in_{\mathbb{R}} A} \left[\sqrt{\mathbb{D}(B_a || B)} \right] \\ &\leq \sqrt{\mathbb{E}_{a \in_{\mathbb{R}} A} [\mathbb{D}(B_a || B)]} && \text{by convexity} \\ &= \sqrt{I(A; B)} && \text{by Proposition 2.9} \end{aligned}$$

□

2.2 Communication Complexity

Let \mathcal{X}, \mathcal{Y} denote the set of possible inputs to the two players, who we name P_x, P_y . In this paper², we view a *private coins protocol* for computing a function $f : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{Z}_K$ as a binary tree with the following structure:

- Each node is *owned* by one of P_x or by P_y
- For every $x \in \mathcal{X}$, each internal node v owned by P_x is associated with a distribution $O_{v,x}$ supported on the children of v . Similarly, for every $y \in \mathcal{Y}$, each internal node v owned by P_y is associated with a distribution $O_{v,y}$ supported on the children of v .
- The leaves of the protocol are labelled by output values.

On input x, y , the protocol π is executed as in Figure 1.

Generic Communication Protocol
<ol style="list-style-type: none"> 1. Set v to be the root of the protocol tree. 2. If v is a leaf, the computation ends with output the value in the label of v. Otherwise, the player owning v samples a child of v according to the distribution associated with her input for v and sends a bit to the other player to indicate which child was sampled. 3. Set v to be the newly sampled node and return to the previous step.

Figure 1: A communication protocol.

A public coin protocol is a distribution on private coins protocols, run by first using shared randomness to sample an index r and then running the corresponding private coin protocol π_r . Every private coin protocol is thus a public coin protocol. The protocol is called deterministic if all distributions labeling the nodes have support size 1.

²The definitions we present here are equivalent to the classical definitions and are more convenient for our proofs.

Definition 2.11. The *communication complexity* of a public coin protocol π , denoted $\text{CC}(\pi)$, is the maximum depth of the protocol trees in the support of π .

Given a protocol π , $\pi(x, y)$ denotes the concatenation of the public randomness with all the messages that are sent during the execution of π . We call this the *transcript* of the protocol. We shall use the notation $\pi(x, y)_j$ to refer to the j 'th transmitted bit in the protocol. We write $\pi(x, y)_{\leq j}$ to denote the concatenation of the public randomness in the protocol with the first j message bits that were transmitted in the protocol. Given a transcript, or a prefix of the transcript, v , we write $\text{CC}(v)$ to denote the number of message bits in v (i.e. the length of the communication).

We often assume that every leaf in the protocol is at the same depth. We can do this since if some leaf is at depth less than the maximum, we can modify the protocol by adding dummy nodes which are always picked with probability 1, until all leaves are at the same depth. This does not change the communication complexity.

Definition 2.12 (Communication Complexity notation). For a function $f : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{Z}_K$, a distribution μ supported on $\mathcal{X} \times \mathcal{Y}$, and a parameter $\rho > 0$, $D_\rho^\mu(f)$ denotes the communication complexity of the cheapest deterministic protocol for computing f on inputs sampled according to μ with error ρ . $R_\rho(f)$ denotes the cost of the best randomized public coin protocol for computing f with error at most ρ on *every* input.

We shall use the following simple fact, first observed by Yao:

Fact 2.13 (Yao's Min-Max). $R_\rho(f) = \max_\mu D_\rho^\mu(f)$.

Remark 2.14 (Information content of private vs. public coins protocols.). Another way to view the difference between public coins and private coins protocols is that the public randomness is considered part of the protocol's transcript. But even if the randomness is short compared to the overall communication complexity, making it public can have a dramatic effect on the information content of the protocol. (As an example, consider a protocol where one party sends a message of $x \oplus r$ where x is its input and r is random. If the randomness r is private then this message has zero information content. If the randomness is public then the message completely reveals the input. (This protocol may seem trivial since its communication complexity is larger than the input length, but in fact we will be dealing with exactly such protocols, as our goal will be to "compress" communication of protocols that have very large communication complexity, but very small information content.)

2.3 Finding differences in inputs

We use the following lemma of Feige et al. [FPRU94]:

Lemma 2.15 ([FPRU94]). *There is a randomized public coin protocol τ with communication complexity $O(\log(k/\epsilon))$ such that on input two k bit strings x, y , it outputs the first index $i \in [k]$ such that $x_i \neq y_i$ with probability at least $1 - \epsilon$, if such an i exists.*

For completeness, we include the proof (based on hashing) in [Appendix C](#).

3 Proof of main theorem

In this section, we prove [Theorem 1.2](#), showing a direct sum for distributional communication complexity even in the case where the input distribution is not necessarily a product distribution. By Yao's minimax principle, for every function f , $R_\rho(f) = \max_\mu D_\rho^\mu$. Thus [Theorem 1.2](#) implies [Theorem 1.1](#) and [Theorem 1.5](#) implies [Theorem 1.4](#). So we shall focus on proving [Theorem 1.2](#) and its XOR Lemma analog [Theorem 1.5](#).

By [Theorem 1.9](#), the main step to establish [Theorem 1.2](#) is to give an efficient simulation of a protocol with small information content by a protocol with small communication complexity. We shall thus prove

Theorem 3.2 (Restated). *For every distribution μ , every protocol π , and every $\epsilon > 0$, there exists functions π_x, π_y , and a protocol τ such that $|\pi_x(X, \tau(X, Y)) - \pi(X, Y)| < \epsilon$, $\Pr[\pi_x(X, \tau(X, Y)) \neq \pi_y(Y, \tau(X, Y))] < \epsilon$ and $\text{CC}(\tau) \leq \sqrt{\text{CC}(\pi) \cdot \text{IC}_\mu(\pi)} \frac{\log(\text{CC}(\pi)/\epsilon)}{\epsilon}$.*

Proof of direct sum theorem from [Theorem 3.2](#). Before proving [Theorem 3.2](#), let's see how we can use it to get our main result ([Theorem 1.2](#)). Let π be any protocol computing f^n on inputs drawn from μ^n with probability of error less than ρ . Then by [Theorem 1.9](#), there exists a protocol τ_1 computing f on inputs drawn from μ with error at most ρ with $\text{CC}(\tau_1) \leq \text{CC}(\pi)$ and $\text{IC}_\mu(\tau_1) \leq 2\text{CC}(\pi)/n$. Next, applying [Theorem 3.2](#) to the protocol τ_1 gives that there must exist a protocol τ_2 computing f on inputs drawn from μ with error at most $\rho + \alpha$ and

$$\begin{aligned} \text{CC}(\tau_2) &\leq O\left(\sqrt{\text{CC}(\tau_1)\text{IC}_\mu(\tau_1)} \log(\text{CC}(\tau_1)/\alpha)/\alpha\right) \\ &= O\left(\sqrt{\text{CC}(\pi)\text{CC}(\pi)/n} \log(\text{CC}(\pi)/\alpha)/\alpha\right) \\ &= O\left(\frac{\text{CC}(\pi) \log(\text{CC}(\pi)/\alpha)/\alpha}{\sqrt{n}}\right) \end{aligned}$$

This proves [Theorem 1.2](#). □

Proof of the XOR Lemma. The proof for [Theorem 1.5](#) (XOR Lemma for distributional complexity) is very similar. First, we show an XOR-analog of [Theorem 1.9](#):

Theorem 3.1. *For every distribution μ , there exists a protocol τ computing f with probability of error ρ over the distribution μ with $\text{CC}(\tau) \leq D_\rho^{\mu^n}(f^{+n}) + 2\log K$ such that if τ' is the protocol that is the same as τ but stops running after $D_\rho^{\mu^n}(f^{+n})$ message bits have been sent, then $\text{IC}_\mu(\tau') \leq \frac{2D_\rho^{\mu^n}(f^{+n})}{n}$.*

Now let π be any protocol computing f^{+n} on inputs drawn from μ^n with probability of error less than ρ . Then by [Theorem 3.1](#), there exists a protocol τ_1 computing f on inputs drawn from μ with error at most ρ with $\text{CC}(\tau_1) \leq \text{CC}(\pi) + 2\log K$ and such that if τ'_1 denotes the first $\text{CC}(\pi)$ bits of the message part of the transcript, $\text{IC}_\mu(\tau'_1) \leq 2\text{CC}(\pi)/n$. Next, applying [Theorem 3.2](#) to the protocol τ'_1 gives that there must exist a protocol τ'_2 simulating τ'_1 on inputs drawn from μ with

error at most $\rho + \alpha$ and

$$\begin{aligned} \text{CC}(\tau'_2) &\leq O\left(\sqrt{\text{CC}(\tau'_1)\text{IC}_\mu(\tau'_1)\log(\text{CC}(\tau'_1)/\alpha)/\alpha}\right) \\ &= O\left(\sqrt{\text{CC}(\pi)\text{CC}(\pi)/n\log(\text{CC}(\pi)/\alpha)/\alpha}\right) \\ &= O\left(\frac{\text{CC}(\pi)\log(\text{CC}(\pi)/\alpha)/\alpha}{\sqrt{n}}\right) \end{aligned}$$

Finally we get a protocol for computing f by first running τ'_2 and then running the last $2\log K$ bits of π . Thus we must have that $O\left(\frac{\text{CC}(\pi)\log(\text{CC}(\pi)/\alpha)/\alpha}{\sqrt{n}}\right) + 2\log K \leq D_{\rho+\alpha}^\mu(f)$, as in the theorem. \square

4 Reduction to Small Information Content

We now prove Theorems 1.9 and 3.1, showing that the existence protocol with communication complexity C for f^n (or f^{+n}) implies a protocol for f with information content roughly C/n .

Theorem 1.9 (Restated). *For every μ, f, ρ there exists a protocol τ computing f on inputs drawn from μ with probability of error at most ρ and communication at most $D_\rho^{\mu^n}(f^n)$ such that $\text{IC}_\mu(\tau) \leq \frac{2D_\rho^{\mu^n}(f^n)}{n}$.*

Theorem 3.1 (Restated). *For every distribution μ , there exists a protocol τ computing f with probability of error ρ over the distribution μ with $\text{CC}(\tau) \leq D_\rho^{\mu^n}(f^{+n}) + 2\log K$ such that if τ' is the protocol that is the same as τ but stops running after $D_\rho^{\mu^n}(f^{+n})$ message bits have been sent, then $\text{IC}_\mu(\tau') \leq \frac{2D_\rho^{\mu^n}(f^{+n})}{n}$.*

Proof. Fix, $\mu, f, n, \rho, \epsilon$ as in the statement of the theorems. We shall prove Theorem 1.9 first. Theorem 3.1 will easily follow by the nature of our proof. To prove Theorem 1.9, we show how to use the best protocol for computing f^n to get a protocol with small information content computing f . Let π be a deterministic protocol with communication complexity $D_\rho^{\mu^n}(f^n)$ computing f^n with probability of error at most ρ .

Let $(X_1, Y_1), \dots, (X_n, Y_n)$ denote random variables distributed according to μ^n . Let $\pi(X^n, Y^n)$ denote the random variable of the transcript (which is just the concatenation of all messages, since this is a deterministic protocol) that is obtained by running the protocol π on inputs $(X_1, Y_1), \dots, (X_n, Y_n)$. We define random variables $W = W_1, \dots, W_n$ where each W_i takes values in the disjoint union $\mathcal{X} \uplus \mathcal{Y}$ so that each $W_i = X_i$ with probability $1/2$ and $W_i = Y_i$ with probability $1/2$. Let W^{-i} denote $W_1, \dots, W_{i-1}, W_{i+1}, \dots, W_n$.

Our new protocol τ shall operate as in Figure 2.

The probability that the protocol τ makes an error on inputs sampled from μ is at most the probability that the protocol π makes an error on inputs sampled from μ^n . It is also immediate that $\text{CC}(\tau) = \text{CC}(\pi)$. All that remains is to bound the information content $\text{IC}_\mu(\tau)$. We do this by relating it to the communication complexity of π . Given any fixing of i, w^{-i} , we use $\tau_{i, w^{-i}}(X, Y)$ to denote the messages sent during the private randomness phase of the protocol. As usual, $\tau(X, Y) = I, W^{-I}, \tau_{I, W^{-I}}(X, Y)$ denotes the concatenation of the public randomness used in τ with the messages sent during its execution.

Protocol τ
<p>Public Randomness Phase :</p> <ol style="list-style-type: none"> 1. The players sample $i, w^{-i} \in_{\mathbb{R}} I, W^{-I}$ using public randomness. <p>Private Randomness Phase :</p> <ol style="list-style-type: none"> 1. P_x sets $x_i = x$, P_y sets $y_i = y$. 2. For every $j \neq i$, P_x samples X_j conditioned on the value of w^{-i}. 3. For every $j \neq i$, P_y samples Y_j conditioned on the value of w^{-i}. 4. The players simulate π on the inputs $x_1, \dots, x_n, y_1, \dots, y_n$ and output the i'th output of π.

Figure 2: A protocol simulating π

Then we have that:

$$\begin{aligned}
& \sum_{i=1}^n I(X_i Y_i; \pi(X^n, Y^n) | W) \\
& \leq I(X_1 \cdots X_n Y_1 \cdots Y_n; \pi(X^n, Y^n) | W) && \text{by Proposition 2.6} \\
& \leq \text{CC}(\pi) && \text{since } H(\pi(X^n, Y^n) | W) \leq \text{CC}(\pi) \\
& = D_{\rho}^{\mu^n}(f^n) && (1)
\end{aligned}$$

For any index i , we have that

$$\begin{aligned}
I(X_i Y_i; \pi(X^n, Y^n) | W) &= I(X_i Y_i; \pi(X^n, Y^n) | W_i W^{-i}) \\
&= \mathbb{E}_{w_i \in_{\mathbb{R}} W_i} [I(X_i Y_i; \pi(X^n, Y^n) | w_i W^{-i})] \\
&= \frac{\mathbb{E}_{x_i \in_{\mathbb{R}} X_i} [I(X_i Y_i; \pi(X^n, Y^n) | x_i W^{-i})] + \mathbb{E}_{y_i \in_{\mathbb{R}} Y_i} [I(X_i Y_i; \pi(X^n, Y^n) | y_i W^{-i})]}{2} \\
&= \frac{I(Y_i; \pi(X^n, Y^n) | X_i W^{-i}) + I(X_i; \pi(X^n, Y^n) | Y_i W^{-i})}{2} && (2)
\end{aligned}$$

Now note that in our protocol, for every i , we have that $X, Y, \tau_{i, W^{-i}}(X, Y), W^{-i}$ have exactly the same distribution as $X_i, Y_i, \pi(X^n, Y^n), W^{-i}$. Thus

$$\begin{aligned}
nI_{\text{C}\mu}(\tau) &= n(I(X; \tau(X, Y) | Y) + I(Y; \tau(X, Y) | X)) \\
&= \sum_{i=1}^n I(Y; \tau_{i, W^{-i}}(X, Y) | X W^{-i}) + I(X; \tau_{i, W^{-i}}(X, Y) | Y W^{-i}) \\
&= \sum_{i=1}^n I(Y_i; \pi(X^n, Y^n) | X_i W^{-i}) + I(X_i; \pi(X^n, Y^n) | Y_i W^{-i}) && (3)
\end{aligned}$$

Equation 1, Equation 2 and Equation 3 imply that

$$D_{\rho}^{\mu^n}(f^n) \geq \frac{nIC_{\mu}(\tau)}{2}$$

Remark 4.1. The analysis above can be easily improved to get the bound $IC_{\mu}(\tau) \leq CC(\tau)/n$ by taking advantage of the fact that each bit of the transcript gives information about at most one of the players' inputs, but for simplicity we do not prove this here.

This completes the proof for [Theorem 1.9](#). The proof for [Theorem 3.1](#) is very similar. As above, we let π denote the best protocol for computing f^{+n} on inputs sampled according to μ^n . Analogous to τ as above, we define the simulation γ as in [Figure 3](#).

Protocol γ
<p>Public Randomness Phase :</p> <ol style="list-style-type: none"> 1. The players sample $i, w^{-i} \in_{\mathbb{R}} I, W^{-I}$ using public randomness. <p>Private Randomness Phase :</p> <ol style="list-style-type: none"> 1. P_x sets $x_i = x$, P_y sets $y_i = y$. 2. For every $j \neq i$, P_x samples X_j conditioned on the value of w^{-i}. 3. For every $j \neq i$, P_y samples Y_j conditioned on the value of w^{-i}. 4. The players simulate π on the inputs $x_1, \dots, x_n, y_1, \dots, y_n$ to compute $z \in \mathbb{Z}_K$. 5. P_x computes $\sum_{j \neq i, w_j = y_j} f(x_j, w_j)$ and sends this sum to P_y 6. P_y outputs the value of the function as $z - \sum_{j \neq i, w_j = y_j} f(x_j, w_j) - \sum_{j \neq i, w_j = x_j} f(w_j, y_j)$.

Figure 3: A protocol simulating π

As before, the probability that the protocol γ makes an error on inputs sampled from μ is at most the probability that the protocol π makes an error on inputs sampled from μ^n , since there is an error in $\gamma_{i, w^{-i}}$ if and only if there is an error in the computation of z . It is also immediate that $CC(\gamma) = CC(\pi) + 2 \log K$.

Let $\gamma'(X, Y)$ denote the concatenation of the public randomness and the messages of γ upto the computation of z . Then, exactly as in the previous case, we have the bound:

$$IC_{\mu}(\gamma') \leq 2CC(\gamma)/n$$

This completes the proof. □

5 Small information complexity \rightarrow small communication complexity

We now prove our main technical theorem, [Theorem 3.2](#):

Theorem 3.2 (Restated). *For every distribution μ , every protocol π , and every $\epsilon > 0$, there exists functions π_x, π_y , and a protocol τ such that $|\pi_x(X, \tau(X, Y)) - \pi(X, Y)| < \epsilon$, $\Pr[\pi_x(X, \tau(X, Y)) \neq \pi_y(Y, \tau(X, Y))] < \epsilon$ and $\text{CC}(\tau) \leq \sqrt{\text{CC}(\pi) \cdot \text{IC}_\mu(\pi) \frac{\log(\text{CC}(\pi)/\epsilon)}{\epsilon}}$.*

Proof. In order to prove the theorem, we consider the protocol tree \mathcal{T} for π_r , for every fixing of the public randomness r . If R is the random variable for the public randomness used in π , we have that

Claim 5.1. $\text{IC}_\mu(\pi) = \mathbb{E}_R [\text{IC}_\mu(\pi_R)]$

Proof.

$$\begin{aligned}
 \text{IC}_\mu(\pi) &= I(\pi(X, Y), X|Y) + I(\pi(X, Y), Y|X) \\
 &= I(R\pi_R(X, Y), X|Y) + I(R\pi_R(X, Y), Y|X) \\
 &= I(R, X|Y) + I(R, Y|X) + I(\pi_R(X, Y), X|YR) + I(\pi_R(X, Y), Y|XR) \\
 &= I(\pi_R(X, Y), X|YR) + I(\pi_R(X, Y), Y|XR) \\
 &= \mathbb{E}_R [\text{IC}_\mu(\pi_R)]
 \end{aligned}$$

□

It will be convenient to describe protocol π_r in a non-standard, yet equivalent way in [Figure 4](#).

Protocol π_r
<p>Sampling Phase :</p> <ol style="list-style-type: none"> 1. For every non-leaf node w in the tree, the player who owns w samples a child according to the distribution given by her input and the public randomness r. This leaves each player with a subtree of the original protocol tree, where each node has out-degree 1 or 0 depending on whether or not it is owned by the player. <p>Path Finding Phase :</p> <ol style="list-style-type: none"> 1. Set v to be the root of the tree. 2. If v is a leaf, the computation ends with the value of the node. Else, the player to whom v belongs communicates one bit to the other player to indicate which of the children was sampled. 3. Set v to the sampled child and return to the previous step.

Figure 4: π restated

For some error parameters β, γ , we define a randomized protocol $\tau_{\beta, \gamma}$ that will simulate π and use the same protocol tree. The idea behind the simulation is to avoid communicating by guessing what the other player's samples look like. The players shall make many mistakes in doing this, but they shall then use [Lemma 2.14](#) to correct the mistakes and end up with the correct transcript. Our simulation is described in [Figure 5](#).

Protocol $\tau_{\beta, \gamma}$
<p>Public Sampling Phase :</p> <ol style="list-style-type: none"> 1. Sample r according to the distribution of the public randomness in π. <p>Correlated Sampling Phase :</p> <ol style="list-style-type: none"> 1. For <i>every</i> non-leaf node w in the tree, let κ_w be a uniformly random element of $[0, 1]$ sampled using public randomness. 2. On input x, y, player P_x (resp. P_y) defines the tree \mathcal{T}_x (resp. \mathcal{T}_y) in the following way: for each node w, P_x (resp. P_y) includes the edge to the left child if $\Pr[\pi_r(X, Y) \text{ reaches the left child} \pi_r(X, Y) \text{ reaches } w \text{ and } X = x] > \kappa_w$ (resp. if $\Pr[\pi_r(X, Y) \text{ reaches the left child} \pi_r(X, Y) \text{ reaches } w \text{ and } Y = y] > \kappa_w$). Otherwise, the right child is picked. <p>Path Finding Phase :</p> <ol style="list-style-type: none"> 1. Each of the players computes the unique path in their trees that leads from the root to a leaf. The players then use Lemma 2.14, communicating $O(\log(n/\beta))$ bits to find the first node at which their respective paths differ, if such a node exists. The player that does not own this node corrects this edge and recomputes his path. They repeatedly correct their paths in this way $\sqrt{\text{CC}(\pi) \cdot \text{IC}_\mu(\pi)}/\gamma$ times.

Figure 5: The simulation of π

Define $\pi_x(x, \tau_{\beta, \gamma}(x, y))$ (resp. $\pi_y(y, \tau_{\beta, \gamma}(x, y))$) to be leaf of the final path computed by P_x (resp. P_y) in the protocol $\tau_{\beta, \gamma}$ (see [Figure 5](#)). The definition of the protocol $\tau_{\beta, \gamma}$ implies immediately the following upper bound on its communication complexity

$$\text{CC}(\tau_{\beta, \gamma}) = O(\sqrt{\text{CC}(\pi) \cdot \text{IC}_\mu(\pi)} \log(\text{CC}(\pi)/\beta)/\gamma) . \quad (4)$$

Let $V = V_0, \dots, V_{\text{CC}(\pi)}$ denote the “right path” in the protocol tree of $\tau_{\beta, \gamma}$. That is, every i , V_{i+1} is the child of V_i that is sampled by the owner of V_i . Observe that this path has the right distribution, since every child with exactly the right conditional probability by the corresponding owner. That is, we have the following claim:

Claim 5.2. *For every x, y, r , the distribution of $V|xyr$ as defined above is the same as the distribution of the sampled transcript in the protocol π .*

This implies in particular, that

$$I(X; V|rY) + I(X; V|rY) = \text{IC}_\mu(\pi_r) .$$

Given two fixed trees $\mathcal{T}_x, \mathcal{T}_y$ as in the above protocol, we say there is a *mistake* at level i if the out-edges of V_{i-1} are inconsistent in the trees. We shall first show that the expected number of mistakes that the players make is small.

Lemma 5.3. $\mathbb{E}[\# \text{ of mistakes in simulating } \pi_r | r] \leq \sqrt{\text{CC}(\pi) \cdot \text{IC}_\mu(\pi_r)}$.

Proof. For $i = 1, \dots, \text{CC}(\pi)$, we denote by C_{ir} the indicator random variable for whether or not a mistake occurs at level i in the protocol tree for π_r , so that the number of mistakes is $\sum_{i=1}^{\text{CC}(\pi)} C_{ir}$.

We shall bound $\mathbb{E}[C_i]$ for each i . A mistake occurs at a vertex w at depth i exactly when $\Pr[V_{i+1} = 0 | x \wedge V_{\leq i} = w] \leq \kappa_w < \Pr[V_{i+1} | y \wedge V_{\leq i} = w]$ or $\Pr[V_{i+1} = 0 | y \wedge V_{\leq i} = w] \leq \kappa_w < \Pr[V_{i+1} | x \wedge V_{\leq i} = w]$. Thus a mistake occurs at $v_{\leq i}$ with probability at most $|(V_i | xv_{<i}r) - (V_i | yv_{<i}r)|$.

If v_{i-1} is owned by P_x , then $V_i | xv_{<i}r$ has the same distribution as $V_i | xyv_{<i}r$. In this case [Proposition 2.10](#) gives that

$$\begin{aligned}
& \mathbb{E}_{xyv_{<i} \in_{\mathbb{R}} XYV_{<i}} [|(V_i | xv_{<i}r) - (V_i | yv_{<i}r)|] \\
&= \mathbb{E}_{xyv_{<i} \in_{\mathbb{R}} XYV_{<i}} [|(V_i | xyv_{<i}r) - (V_i | yv_{<i}r)|] \\
&\leq \mathbb{E}_{v_{<i} \in_{\mathbb{R}} V_{<i}} \left[\sqrt{I(X; V_i | Yv_{<i}r)} \right] && \text{by Proposition 2.10} \\
&\leq \sqrt{\mathbb{E}_{v_{<i} \in_{\mathbb{R}} V_{<i}} [I(X; V_i | Yv_{<i}r)]} && \text{by convexity} \\
&\leq \sqrt{\mathbb{E}_{v_{<i} \in_{\mathbb{R}} V_{<i}} [I(X; V_i | Yv_{<i}r) + I(Y; V_i | Xv_{<i}r)]} \\
&= \sqrt{I(X; V_i | YV_{<i}r) + I(Y; V_i | XV_{<i}r)}
\end{aligned}$$

Similarly, we see that the same inequality holds if v_{i-1} is owned by P_y .

Thus we get that

$$\begin{aligned}
\mathbb{E}[C_{ir}] &\leq \sqrt{I(X; V_i | YV_{i-1}r) + I(Y; V_i | XV_{i-1}r)} \\
&= \sqrt{I(X; V_i | YV_{<i}r) + I(Y; V_i | XV_{<i}r)}
\end{aligned}$$

where the last equality follows from the fact that $V_{<i-1}$ is determined by V_{i-1} . Finally we apply the Cauchy Schwartz inequality to conclude that

$$\begin{aligned}
\mathbb{E} \left[\sum_{i=1}^{\text{CC}(\pi)} C_{ir} \right] &= \sum_{i=1}^{\text{CC}(\pi)} \mathbb{E}[C_{ir}] \\
&\leq \sqrt{\text{CC}(\pi) \sum_{i=1}^{\text{CC}(\pi)} \mathbb{E}[C_{ir}]^2} \\
&\leq \sqrt{\text{CC}(\pi) \sum_{i=1}^{\text{CC}(\pi)} [I(X; V_i | YV_{<i}r) + I(Y; V_i | XV_{<i}r)]} \\
&= \sqrt{\text{CC}(\pi) (I(X; V^{\text{CC}(\pi)} | Yr) + I(Y; V^{\text{CC}(\pi)} | Xr))} \\
&= \sqrt{\text{CC}(\pi) \cdot \text{IC}_\mu(\pi_r)}
\end{aligned}$$

□

We then get that overall the expected number of mistakes is small:

Lemma 5.4. $\mathbb{E}[\# \text{ of mistakes in simulating } \pi] \leq \sqrt{\text{CC}(\pi) \cdot \text{IC}_\mu(\pi)}$.

Proof.

$$\begin{aligned} \mathbb{E}[\# \text{ of mistakes in simulating } \pi] &= \mathbb{E}_R[\# \text{ of mistakes in simulating } \pi_R] \\ &\leq \mathbb{E}_R \left[\sqrt{\text{CC}(\pi) \cdot \text{IC}_\mu(\pi_R)} \right] \\ &\leq \sqrt{\mathbb{E}_R[\text{CC}(\pi) \cdot \text{IC}_\mu(\pi_R)]} \\ &= \sqrt{\text{CC}(\pi) \cdot \text{IC}_\mu(\pi)} \end{aligned}$$

□

Lemma 5.5. *The distribution of the leaf sampled by $\tau_{\beta, \gamma}$ is $\gamma + \beta \frac{\sqrt{\text{CC}(\pi) \cdot \text{IC}_\mu(\pi)}}{\gamma}$ -close to the distribution of the leaf sampled by π .*

Proof. We show that in fact the probability that both players do not finish the protocol with the leaf $V_{\text{CC}(\pi)}$ is bounded by $\gamma + \beta \frac{\sqrt{\text{CC}(\pi) \cdot \text{IC}_\mu(\pi)}}{\gamma}$. This follows from a simple union bound — the leaf $V_{\text{CC}(\pi)}$ can be missed in two ways: either the number of mistakes on the correct path is larger than $\sqrt{\text{CC}(\pi) \cdot \text{IC}_\mu(\pi)}/\gamma$ (probability at most γ by Lemma 5.6 and Markov's inequality) or our protocol fails to detect all mistakes (for each mistake this happens with probability β). □

We set $\beta = \gamma^2/\text{CC}(\pi)$. Then, since $\text{CC}(\pi) \geq \text{IC}_\mu(\pi)$, we get that the protocol errs with probability at most $\rho + 2\gamma$. On the other hand, by (5.2), the communication complexity of the protocol is at most $O(\sqrt{\text{CC}(\pi) \cdot \text{IC}_\mu(\pi)} \log(\text{CC}(\pi)/\beta)/\gamma) = O(\sqrt{\text{CC}(\pi) \cdot \text{IC}_\mu(\pi)} \log(\text{CC}(\pi)/\gamma)/\gamma)$. Setting $\epsilon = 2\gamma$ proves the theorem. □

6 Proofs for the Product Case

In this section we argue how to get a linear bound in the case that μ is a product distribution. We shall prove Theorem 1.3.

Throughout this section we assume that the distribution on X, Y is a product distribution.

Again, we prove this via a reduction. We start with a protocol π with $\text{CC}(\pi) = D_\rho^{\mu^n}(f^n)$ such that π computes f^n with probability of error at most ρ on inputs sampled according to μ^n . Our first step shall be to get a protocol that computes f^n but whose messages are *smoothed out* in the sense that every bit in the protocol is relatively close to being unbiased. We define a protocol that is a simulation of π in the following way: for a parameter β that we shall fix later, every time a player wants to send a bit in π , she instead sends $1000 \frac{\log(\text{CC}(\pi)/\gamma)}{\beta^2}$ bits which are each independently chosen to be the correct value with probability $1/2 + \beta$. The receiving player takes the majority of the bits sent to reconstruct the intended transmission. By the Chernoff bound, we have that the probability that any transmission is received incorrectly is at most $\gamma/\text{CC}(\pi)$. By the union bound, this means that for every input, the distribution of the simulated transcript is γ -close to the correct distribution. We have thus argued the following claim:

Claim 6.1. For every f, μ, ρ, γ , there exists a protocol π computing f^n with probability of error $\rho + \gamma$ on the distribution μ such that

$$\text{CC}(\pi) = O\left(\frac{D_\rho^{\mu^n}(f^n) \log(D_\rho^{\mu^n}(f^n)/\gamma)}{\beta^2}\right)$$

and for every x, y, v, i we have that

$$\Pr[\pi(x, y)_{i+1} = 1 | v^i] \in [1/2 - \beta, 1/2 + \beta].$$

In analogy with [Theorem 1.9](#) and [Theorem 3.1](#), [Claim 6.1](#) leads to the following results:

Theorem 6.2 (Reduction to Small Information Content). For every $\gamma, \beta, \mu, f, \rho$ there exists a protocol τ computing f on inputs drawn from μ with probability of error at most $\rho + \gamma$, such that

$$\text{CC}(\tau) \leq O\left(\frac{D_\rho^{\mu^n}(f^n) \log(D_\rho^{\mu^n}(f^n)/\gamma)}{\beta^2}\right)$$

and

$$\text{IC}_\mu(\tau) \leq O\left(\frac{\text{CC}(\tau) \log \text{CC}(\tau)}{n}\right).$$

Further, for every x, y, j, t , we have the j 'th bit of the messages satisfies $\Pr[\tau(x, y)_j = 1 | \tau(x, y)_{\leq j-1} = t] \in [1/2 - \beta, 1/2 + \beta]$.

For f^{+n} we have the following theorem:

Theorem 6.3 (Reduction to Small Information Content). For every distribution μ , there exists a protocol τ computing f with probability of error $\rho + \gamma$ over the distribution μ with

$$\text{CC}(\tau) \leq O\left(\frac{D_\rho^{\mu^n}(f^{+n}) \log(D_\rho^{\mu^n}(f^{+n})/\gamma)}{\beta^2}\right) + 2 \log K$$

such that if τ' is the protocol that simulates all but the last $2 \log K$ bits of τ , then

$$\text{IC}_\mu(\tau') \leq O\left(\frac{\text{CC}(\tau') \log \text{CC}(\tau')}{n}\right)$$

Further, for every x, y, j, t , we have the j 'th bit of the messages satisfies $\Pr[\tau'(x, y)_j = 1 | \tau(x, y)_{\leq j-1} = t] \in [1/2 - \beta, 1/2 + \beta]$.

To complete the proof, we need to show how to simulate the protocols τ in the above theorems with small communication complexity. We shall do this by proving the following theorem:

Theorem 6.4. There exists a constant k such that for every $\epsilon > 0$, if π is a protocol such that for every x, y, v, i we have that

$$\Pr[\pi(x, y)_{i+1} = 1 | v_{\leq i}] \in \left[\frac{1}{2} - \frac{1}{k \log(\text{CC}(\pi)/\epsilon)}, \frac{1}{2} + \frac{1}{k \log(\text{CC}(\pi)/\epsilon)}\right]$$

Then for every product distribution μ on inputs X, Y there exists a protocol τ and a function p such that for every x, y and every transcript l of π ,

$$\frac{\Pr[p(\tau(x, y)) = l]}{\Pr[\pi(x, y) = l]} \leq \exp(O(\epsilon))$$

and the expected communication complexity of τ under the distribution μ is at most $\exp(O(\epsilon)) \text{IC}_\mu(\pi)$.

We use the above theorems to prove our final theorem about product distributions:

Proof of Theorem 1.3. Let π be a protocol satisfying the conclusions of Theorem 6.2 with β set to $\frac{1}{k' \log(D_\rho^{\mu^n}(f^n)/\epsilon)}$ for some constant k' . k' can be chosen to be large enough so that $\beta = \frac{1}{k' \log(D_\rho^{\mu^n}(f^n)/\epsilon)} \leq \frac{1}{k \log(\text{CC}(\pi)/\epsilon)}$ as in Theorem 6.4.

By Theorem 6.4, we get a protocol computing f on the distribution μ with error $(\rho+\gamma) \exp(O(\epsilon))$ and expected communication $\exp(O(\epsilon)) D_\rho^{\mu^n}(f^n) \text{polylog}(D_\rho^{\mu^n}(f^n)/\gamma)/n$. Markov's inequality implies that by interrupting the protocol if it runs for too long, we can get a protocol that errs with probability $(\rho+\gamma) \exp(O(\epsilon))+\lambda$ and has communication complexity $\exp(O(\epsilon)) D_\rho^{\mu^n}(f^n) \text{polylog}(D_\rho^{\mu^n}(f^n)/\gamma)/\lambda n$. Set $\lambda = \gamma = \rho$ and let $\exp(O(\epsilon))$ be a small enough constant to prove the theorem. \square

The proof for Theorem 1.6 is almost exactly the same, so we omit it.

6.1 Proof of Theorem 6.4

It only remains to prove Theorem 6.4. Set $\beta = 1/k \log(\text{CC}(\pi)/\epsilon)$. We need the following definition:

Definition 6.5 (Conditional Divergence). Given a protocol π , a prefix v of the transcript and $j \in [\text{CC}(v)]$, we define the j 'th step divergence cost as

$$\mathbb{D}_{x,j}^\pi(v) \stackrel{def}{=} \mathbb{D}((\pi(x, Y)_j | v_{\leq j-1}) || (\pi(X, Y)_j | v_{\leq j-1}))$$

$$\mathbb{D}_{y,j}^\pi(v) \stackrel{def}{=} \mathbb{D}((\pi(X, y)_j | v_{\leq j-1}) || (\pi(X, Y)_j | v_{\leq j-1}))$$

We define the divergence cost for the whole prefix as the sum of the step divergence costs

$$\mathbb{D}_x^\pi(v) \stackrel{def}{=} \sum_{j=1}^{\text{CC}(v)} \mathbb{D}_{x,j}^\pi(v), \quad \mathbb{D}_y^\pi(v) \stackrel{def}{=} \sum_{j=1}^{\text{CC}(v)} \mathbb{D}_{y,j}^\pi(v)$$

It is easy to check that

$$\mathbb{E}_{X, Y, \pi(X, Y)} [\mathbb{D}_X^\pi(\pi(X, Y)) + \mathbb{D}_Y^\pi(\pi(X, Y))] = \text{IC}_\mu(\pi)$$

Thus the conditional divergence is in some sense a measure of the amount of information revealed by the relevant prefix of the transcript. Observe that $\mathbb{D}_x^\pi(v)$ is a function only of x and v . Further, we have that if the node corresponding to $v_{\leq j-1}$ is owned by x , then $\mathbb{D}_{y,j}^\pi(v) = 0$, since conditioned on $v_{\leq j-1}$, Y is independent of V_j .

We use the fact that the bits in our protocol are close to uniform to show that the step divergence is at most $O(\beta)$ for each step:

Proposition 6.6. For every j , $\mathbb{D}_{x,j}^\pi(v)$ and $\mathbb{D}_{y,j}^\pi(v)$ are bounded by $O(\beta)$.

Proof. This follows from the fact that all probabilities for each step lie in $[1/2 - \beta, 1/2 + \beta]$. The worst the divergence between two distributions that lie in this range can be is clearly $\log\left(\frac{1/2+\beta}{1/2-\beta}\right) = \log(1 + O(\beta)) = O(\beta)$. \square

Next, for every prefix v of the transcript, and inputs x, y , we define a subset of the prefixes of potential transcripts that start with v , \mathcal{B}_{vxy} in the following way: we include w in \mathcal{B}_{vxy} if and only if for every w' that is a strict prefix of w ,

$$\max \left\{ \sum_{j=\text{CC}(v)+1}^{\text{CC}(w')} \mathbb{D}_{x,j}^{\pi}(w'), \sum_{j=\text{CC}(v)+1}^{\|w'\|} \mathbb{D}_{y,j}^{\pi}(w') \right\} < \beta,$$

and we have that w itself is either a leaf or satisfies

$$\max \left\{ \sum_{j=\text{CC}(v)+1}^{\text{CC}(w)} \mathbb{D}_{x,j}^{\pi}(w), \sum_{j=\text{CC}(v)+1}^{\|w\|} \mathbb{D}_{y,j}^{\pi}(w) \right\} \geq \beta.$$

The set \mathcal{B}_{vxy} has the property that every path from v to a leaf of the protocol tree must intersect exactly one element of \mathcal{B}_{vxy} , i.e. if we cut all paths at the point where they intersect \mathcal{B}_{vxy} , we get a protocol tree that is a subtree of the original tree. We define the distribution B_{vxy} on the set \mathcal{B}_{vxy} as the distribution on \mathcal{B}_{vxy} induced by the protocol π . Namely we sample from B_{vxy} by sampling from $\pi(x, y)|v$ and then taking the unique vertex of \mathcal{B}_{vxy} that the sampled path intersects. Similarly, we define the distributions B_{vx}, B_{vy}, B_v on \mathcal{B}_{vxy} to be the distributions obtained by first sampling a path according to $\pi(x, Y)|v, \pi(X, y)|v, \pi(X, Y)|v$ and then taking the unique vertices in \mathcal{B}_{vxy} that these paths intersect. For every transcript w , the players can compute the element of \mathcal{B}_{vxy} that intersects the path w by communicating $2 \log \text{CC}(\pi)$ bits.

Given, these definitions, we are now ready to describe our simulation protocol. The protocol proceeds in rounds. In each round the players shall use rejection sampling to sample some consecutive part of the transcript.

6.1.1 A single round

The first protocol, shown in [Figure 6](#) assumes that we have already sampled the prefix v . We define the protocol for some constant t that we shall set later.

Note that $B_v(w) = \prod_{i=\text{CC}(v)+1}^{\text{CC}(w)} \Pr[\pi(X, Y)_{\leq i+1} = w_{\leq i+1} | \pi(X, Y)_{\leq i} = w_{\leq i}]$. We write $B_v^x(w)$ to denote the part of this product that corresponds to nodes sampled by P_x , and $B_v^y(w)$ to denote the part that corresponds to nodes sampled by P_y . Thus $B_v = B_v^x B_v^y$. We use B_{vx}^x, B_{vx}^y etc to denote the analogous functions. Then note that since X, Y are independent, we have that $B_{vx}^y = B_v^y$. Thus we get

$$\left(\frac{B_{vx}}{B_v} \right) \left(\frac{B_{vy}}{B_v} \right) = \left(\frac{B_v^y B_{vxy}^x}{B_v^x B_v^y} \right) \left(\frac{B_{vxy}^y B_v^x}{B_v^x B_v^y} \right) = \frac{B_{vxy}^x B_{vxy}^y}{B_v^x B_v^y} = \frac{B_{vxy}}{B_v} \quad (5)$$

This suggests that our protocol should pick a transcript distributed according to B_{vxy} . We shall argue that the subsequent prefix of the transcript sampled by the protocol in [Figure 6](#) cannot be sampled with much higher probability than what it is sampled with in the real distribution. Let B'_{vxy} denote the distribution of the accepted prefix of $\tau_{v,t}$.

Claim 6.7 (No sample gets undue attention). *For every prefix w ,*

$$B'_{vxy}(w)/B_{vxy}(w) \leq 2 \exp \left(-\Omega \left(\frac{\log t - O(\beta)}{\beta} \right) \right)$$

Protocol $\tau_{v,t}$
<p>1. Both players use public randomness to sample a path according to $\pi(X, Y) v$ and communicate $2 \log \text{CC}(\pi)$ bits to sample an element w of \mathcal{B}_{vxy} according to the distribution B_v.</p> <p>2. P_x samples a bit a_1 which is 1 with probability</p> $\min \left\{ \frac{B_{vx}(w)}{tB_v(w)}, 1 \right\}.$ <p>3. P_y samples a bit a_2 which is 1 with probability</p> $\min \left\{ \frac{B_{vy}(w)}{tB_v(w)}, 1 \right\}.$ <p>4. If both a_1 and a_2 were 1, they accept w. Else they repeat the protocol.</p>

Figure 6: The protocol to sample a subsequent part of the transcript

We shall also show that the expected communication complexity of this protocol is not too high:

Claim 6.8 (Small number of rounds). *The expected communication complexity of τ_v is at most*

$$\frac{O(t^2)}{1 - \exp\left(-\Omega\left(\frac{\log t - O(\beta)}{\beta}\right)\right)}$$

[Claim 6.7](#) and [Claim 6.8](#) will follow from the following claim:

Claim 6.9.

$$\Pr_{w \in_R \mathcal{B}_{vxy}} \left[\frac{B_{vx}(w)}{B_v(w)} \geq t \right] \leq \exp\left(-\Omega\left(\frac{\log t - O(\beta)}{\beta}\right)\right), \quad \Pr_{w \in_R \mathcal{B}_{vxy}} \left[\frac{B_{vy}(w)}{B_v(w)} \geq t \right] \leq \exp\left(-\Omega\left(\frac{\log t - O(\beta)}{\beta}\right)\right)$$

Let us first argue that [Claim 6.7](#) follows from [Claim 6.9](#).

Proof of Claim 6.7. Set a to be the function that maps any $w \in \mathcal{B}_{vxy}$ to $\min\left\{(1/t)\frac{B_{vx}(w)}{B_v(w)}, 1\right\} \cdot \min\left\{(1/t)\frac{B_{vy}(w)}{B_v(w)}, 1\right\}$. Set $a' = (1/t)\frac{B_{vx}(w)}{B_v(w)}(1/t)\frac{B_{vy}(w)}{B_v(w)}$. Then clearly $a'(w) \geq a(w)$ for every w . Applying [Equation 4](#), we get

$$a' = (1/t^2) \left(\frac{B_{vx}}{B_v}\right) \left(\frac{B_{vy}}{B_v}\right) = (1/t^2) \frac{B_{vxy}}{B_v}$$

Thus $B_{vxy} = \beta a' \cdot B_v$ for some constant β . By [Proposition B.3](#), applied to a', a and the distributions $D = B_{vxy}, D' = B'_{vxy}$, we have that for every w ,

$$\frac{B'_{vxy}(w)}{B_{vxy}(w)} \leq \frac{1}{1 - \Pr_{w \in_R \mathcal{B}_{vxy}}[a'(w) \geq a(w)]}$$

On the other hand, by the union bound and [Claim 6.9](#),

$$\Pr_{w \in_{\mathbb{R}} \mathcal{B}_{vxy}} [a'(w) \geq a(w)] \leq \Pr_{w \in_{\mathbb{R}} \mathcal{B}_{vxy}} \left[\frac{B_{vx}(w)}{B_v(w)} \geq t \vee \frac{B_{vy}(w)}{B_v(w)} \geq t \right] \leq 2 \exp \left(-\Omega \left(\frac{\log t - O(\beta)}{\beta} \right) \right)$$

Since $1/(1-z) \leq 1 + O(z)$ for $z \in (0, 1/10)$, we get [Claim 6.7](#). \square

Now we show [Claim 6.8](#) assuming [Claim 6.9](#).

Proof of [Claim 6.8](#). We shall use [Proposition B.4](#). We need to estimate the probability that the first round of $\tau_{v,t}$ accepts its sample. This probability is exactly

$$\sum_{w \in \mathcal{B}_{vxy}} B_v(w) \min \left\{ (1/t) \frac{B_{vx}(w)}{B_v(w)}, 1 \right\} \cdot \min \left\{ (1/t) \frac{B_{vy}(w)}{B_v(w)}, 1 \right\}$$

Let $A \subset \mathcal{B}_{vxy}$ denote the set $\{w : \frac{B_{vx}(w)}{B_v(w)} \leq t \wedge \frac{B_{vy}(w)}{B_v(w)} \leq t\}$. Then we see that the above sum can be lower bounded:

$$\begin{aligned} & \sum_{w \in \mathcal{B}_{vxy}} B_v(w) \min \left\{ (1/t) \frac{B_{vx}(w)}{B_v(w)}, 1 \right\} \cdot \min \left\{ (1/t) \frac{B_{vy}(w)}{B_v(w)}, 1 \right\} \\ & \geq (1/t^2) \sum_{w \in A} B_v(w) \left(\frac{B_{vx}(w)}{B_v(w)} \right) \left(\frac{B_{vy}(w)}{B_v(w)} \right) = (1/t^2) \sum_{w \in A} B_{vxy}, \end{aligned}$$

where the last equality follows from [Equation 4](#).

Finally, we see that [Claim 6.9](#) implies that $\sum_{w \in A} B_{vxy} \geq 1 - \exp \left(-\Omega \left(\frac{\log t - O(\beta)}{\beta} \right) \right)$. [Proposition B.4](#) then gives the bound we need. \square

Next we prove [Claim 6.9](#). To do this we shall need to use a simple generalization of Azuma's inequality, which we prove in [Appendix A](#).

Proof of [Claim 6.9](#). Let W be a random variable distributed according to B_{vxy} . Set $Z_{\text{CC}(v)+1}, \dots, Z_{\text{CC}(\pi)}$ to be real valued random variables such that if $i \leq \text{CC}(W)$,

$$Z_i = \log \left(\frac{\Pr[\pi(x, Y)_i = W_i | vW_{\leq i-1}]}{\Pr[\pi(X, Y)_i = W_{\leq i} | vW_{\leq i-1}]} \right).$$

If $i > \text{CC}(W)$, set $Z_i = 0$. Observe that $\mathbb{E}[Z_i | w_{\leq i-1}] = \mathbb{D}_{x,i}^\pi(w)$. We also have that

$$\begin{aligned} \sum_{i=\text{CC}(v)+1}^{\text{CC}(\pi)} Z_i &= \log \left(\frac{\Pr[\pi(x, Y)^{\text{CC}(w)} = w | v]}{\Pr[\pi(X, Y)^{\text{CC}(w)} = w | v]} \right) \\ &= \log \left(\frac{B_{vx}(w)}{B_v w} \right) \end{aligned} \tag{6}$$

Next set $T_i = Z_i - \mathbb{E}[Z_i | Z_{i-1}, \dots, Z_1]$. Note that $\mathbb{E}[T_i | T_{i-1}, \dots, T_1] = 0$ (in fact the stronger condition that $\mathbb{E}[T_i | Z_{i-1}, \dots, Z_1] = 0$ holds). For every $w \in \mathcal{B}_{vxy}$, we have that

$$\begin{aligned} \sup(T_i | w_{\leq i-1}) &\leq \max \left\{ \log \left(\frac{\Pr[\pi(x, Y)_i = 0 | w_{\leq i-1}]}{\Pr[\pi(X, Y)_i = 0 | w_{\leq i-1}]} \right), \log \left(\frac{\Pr[\pi(x, Y)_i = 1 | w_{\leq i-1}]}{\Pr[\pi(X, Y)_i = 1 | w_{\leq i-1}]} \right) \right\} \\ \inf(T_i | w_{\leq i-1}) &\geq \min \left\{ \log \left(\frac{\Pr[\pi(x, Y)_i = 0 | w_{\leq i-1}]}{\Pr[\pi(X, Y)_i = 0 | w_{\leq i-1}]} \right) - \mathbb{D}_{x,i}^\pi(w), \log \left(\frac{\Pr[\pi(x, Y)_i = 1 | w_{\leq i-1}]}{\Pr[\pi(X, Y)_i = 1 | w_{\leq i-1}]} \right) - \mathbb{D}_{x,i}^\pi(w) \right\} \end{aligned}$$

By [Proposition 2.8](#) and using the fact that $\pi(x, Y) = 1 \in [1/2 - \beta, 1/2 + \beta]$ we can bound

$$\begin{aligned} \sup(T_i | w_{\leq i-1}) &\leq \log \left(\frac{1/2 - \beta + \sqrt{\mathbb{D}_{x,i}^\pi(w)}}{1/2 - \beta} \right) \\ &= \log \left(1 + O \left(\sqrt{\mathbb{D}_{x,i}^\pi(w)} \right) \right) \\ &= O \left(\sqrt{\mathbb{D}_{x,i}^\pi(w)} \right) \end{aligned} \tag{7}$$

$$\begin{aligned} \inf(T_i | w_{\leq i-1}) &\geq \log \left(\frac{1/2 - \beta}{1/2 - \beta + \sqrt{\mathbb{D}_{x,i}^\pi(w)}} \right) - \mathbb{D}_{x,i}^\pi(w) \\ &= \log \left(1 - O \left(\sqrt{\mathbb{D}_{x,i}^\pi(w)} \right) \right) \\ &= -O \left(\sqrt{\mathbb{D}_{x,i}^\pi(w)} \right), \end{aligned} \tag{8}$$

as long as $\beta < 1/10$.

[Equation 6](#) and [Equation 7](#) imply that for $w \in \mathcal{B}_{vxy}$,

$$\sum_{i=\text{CC}(v)+1}^{\text{CC}(\pi)} (\sup(T_i) - \inf(T_i) | w_{\leq i-1})^2 \leq \sum_{i=\text{CC}(v)+1}^{\text{CC}(\pi)} O(\mathbb{D}_{x,i}^\pi(w)) = O(\beta) \tag{9}$$

For every w ,

$$\begin{aligned} \left(\sum_{i=\text{CC}(v)+1}^{\text{CC}(\pi)} T_i \right) | w &= \left(\sum_{i=\text{CC}(v)+1}^{\text{CC}(\pi)} Z_i \right) | w - \sum_{i=\text{CC}(v)+1}^{\text{CC}(\pi)} \mathbb{E}[Z_i | w_{\leq i-1}] \\ &= \sum_{i=\text{CC}(v)+1}^{\text{CC}(w)} \log \left(\frac{\Pr[\pi(x, Y)_i = w_i | v\pi(x, Y)_{\leq i-1} = w_{\leq i-1}]}{\Pr[\pi(X, Y)_i = w_i | \pi(X, Y)_{\leq i-1} = vw_{\leq i-1}]} \right) - \sum_{i=\text{CC}(v)+1}^{\text{CC}(w)} \mathbb{D}_{x,i}^\pi(w) \\ &\geq \log \left(\frac{B_{vx}(w)}{B_v w} \right) - O(\beta) \end{aligned} \tag{10}$$

where the last inequality follows from the definition of \mathcal{B}_{vxy} , [Proposition 6.6](#) and [Equation 5](#).

Thus we can use [Theorem A.1](#) to bound

$$\begin{aligned}
& \Pr_{w \in_{\mathbb{R}} B_{vxy}} \left[\frac{B_{vx}(w)}{B_v(w)} \geq t \right] \\
& \leq \Pr_{w \in_{\mathbb{R}} B_{vxy}} \left[\log \left(\frac{B_{vx}(w)}{B_v(w)} \right) \geq \log t \right] \\
& \leq \Pr \left[\sum_{i=\text{CC}(v)+1}^{\text{CC}(\pi)} T_i \geq \log t - O(\beta) \right] && \text{by Equation 9} \\
& \leq \exp \left(-\Omega \left(\frac{\log t - O(\beta)}{\sum_{i=\text{CC}(v)+1}^{\text{CC}(\pi)} (\sup(T_i) - \inf(T_i) | w_{\leq i-1})^2} \right) \right) \\
& \leq \exp \left(-\Omega \left(\frac{\log t - O(\beta)}{\beta} \right) \right) && \text{by Equation 8}
\end{aligned}$$

□

6.1.2 The whole protocol

Our final protocol for computing f is shown in [Figure 7](#).

Protocol τ_t
<ol style="list-style-type: none"> 1. The players publicly sample the public randomness $v \in_{\mathbb{R}} R$ for π. 2. The players repeatedly run $\tau_{v,t}$ to get a new prefix v. They stop only when they reach a leaf of the protocol tree for π.

Figure 7: The protocol to sample a subsequent part of the transcript

We first argue that our simulation returns the correct answer with decent probability. We shall actually argue that the probability for any returned transcript does not increase by too much. To ease notations, let us set

$$\alpha \stackrel{\text{def}}{=} \exp \left(-\Omega \left(\frac{\log t - O(\beta)}{\beta} \right) \right)$$

Set t to be a large enough constant so that $\alpha = \exp(-\Omega(1/\beta)) = \exp(-\Omega(\log(\text{CC}(\pi)^k/\epsilon^k))$. Set k to be large enough so that $\alpha \leq \epsilon/\log \text{CC}(\pi)$.

Let L denote the random variable of the sampled transcript returned by τ_t . Then by [Claim 6.7](#), we get that for every leaf l ,

$$\frac{\Pr[L = l|xy]}{\Pr[\pi(x, y) = l]} \leq (1 + \alpha)^{\text{CC}(\pi)} = \exp(O(\epsilon)) \tag{11}$$

Next we bound the expected communication of the protocol. First observe that if the protocol accepts a leaf l , then the protocol must have involved $O(\mathbb{D}_x^\pi(l) + \mathbb{D}_y^\pi(l))$ rounds. The expected number of bits communicated in each of these rounds is independent of l by [Proposition B.1](#), and is $\frac{t^2}{1-\alpha}$ by [Claim 6.8](#). Thus the expected communication complexity of the protocol can be bounded

$$\begin{aligned}
& \mathbb{E}_{x,y,l \in_{\mathbb{R}} X,Y,L} \left[O \left((\mathbb{D}_x^\pi(l) + \mathbb{D}_y^\pi(l)) \frac{t^2}{1-\alpha} \right) \right] \\
&= \frac{O(t^2)}{1-\alpha} \mathbb{E}_{x,y \in_{\mathbb{R}} X,Y} \left[\sum_l \Pr[L = l | x, y] (\mathbb{D}_x^\pi(l) + \mathbb{D}_y^\pi(l)) \right] \\
&\leq O(1) \mathbb{E}_{x,y \in_{\mathbb{R}} X,Y} \left[\sum_l \exp(O(\epsilon)) \Pr[\pi(x, y) = l] (\mathbb{D}_x^\pi(l) + \mathbb{D}_y^\pi(l)) \right] \quad \text{by Equation 10} \\
&= \exp(O(\epsilon)) \mathbb{E}_{X,Y} [\mathbb{D}_X^\pi(\pi(X, Y)) + \mathbb{D}_Y^\pi(\pi(X, Y))] \\
&= \exp(O(\epsilon)) \text{IC}_\mu(\pi)
\end{aligned}$$

This completes the proof of [Theorem 6.4](#).

7 Acknowledgements

We thank Noga Alon, Emanuel Milman, Alex Samorodnitsky, Avi Wigderson and Amir Yehudayoff for useful discussions.

A A simple generalization of Azuma's inequality

We shall need the following theorem, whose proof appears in [\[JHM⁺98\]](#). For completeness, we reproduce the part of the proof we need here:

Theorem A.1 (Azuma). *Let T_1, \dots, T_k be real valued random variables such that for every i , we have $\mathbb{E}[T_i | T_{i-1}, \dots, T_1] \leq 0$. Set $A_i = (\sup(T_i) - \inf(T_i) | T_{i-1}, \dots, T_1)^2$. Then if $\sum_{i=1}^k A_i \leq c$, for every $\alpha > 0$,*

$$\Pr \left[\sum_{i=1}^k T_i \geq \alpha \right] \leq \exp(2\alpha^2/c).$$

To prove the theorem, we need the following lemma appearing as Lemma 2.6 in [\[JHM⁺98\]](#):

Lemma A.2. *Let X be a real valued random variable with $\mathbb{E}[X] = 0$ and $X \in [a, b]$ almost surely. Then $\mathbb{E}[\exp(X)] \leq \exp\left(\frac{(b-a)^2}{8}\right)$.*

Proof of Theorem A.1. First, we assume without loss of generality that $\mathbb{E}[T_i | T_{i-1}, \dots, T_1] \leq 0$. We can do this by changing each random variable T_i to $T_i - \mathbb{E}[T_i | T_{i-1}, \dots, T_1]$. This does not change any of the conditions above, and only increases $\Pr[\sum_i T_i \geq \alpha]$.

By Markov's inequality, for every positive λ we have

$$\Pr \left[\sum_{i=1}^k T_i \geq \alpha \right] = \Pr \left[\exp \left(\lambda \sum_{i=1}^k T_i \right) > \exp(\lambda \alpha) \right] \leq \mathbb{E} \left[\exp \left(\lambda \sum_{i=1}^k T_i \right) \right] \exp(-\lambda \alpha)$$

Next we show by induction on k that $\mathbb{E} \left[\exp \left(\lambda \sum_{i=1}^k T_i \right) \right] \leq \sup \left(\prod_{i=1}^k \mathbb{E} [\exp(\lambda T_i) | T_{i-1}, \dots, T_1] \right)$. The case $k = 1$ is trivial. For general k we compute

$$\begin{aligned} \mathbb{E} \left[\exp \left(\lambda \sum_{i=1}^k T_i \right) \right] &= \mathbb{E} \left[\exp(\lambda T_1) \mathbb{E} \left[\exp \left(\lambda \sum_{i=2}^k T_i \right) | T_1 \right] \right] \\ &\leq \mathbb{E} [\exp(\lambda T_1)] \sup \left(\prod_{i=2}^k \mathbb{E} [\exp(\lambda T_i) | T_{i-1}, \dots, T_1] \right) && \text{by induction} \\ &= \sup \left(\mathbb{E} [\exp(\lambda T_1)] \prod_{i=2}^k \mathbb{E} [\exp(\lambda T_i) | T_{i-1}, \dots, T_1] \right) \\ &= \sup \left(\prod_{i=1}^k \mathbb{E} [\exp(\lambda T_i) | T_{i-1}, \dots, T_1] \right) \end{aligned}$$

Thus we can bound

$$\begin{aligned} \Pr \left[\sum_{i=1}^k T_i \geq \alpha \right] &\leq \exp(-\lambda \alpha) \sup \left(\prod_{i=1}^k \mathbb{E} [\exp(\lambda T_i) | T_{i-1}, \dots, T_1] \right) \\ &\leq \exp(-\lambda \alpha) \sup \left(\prod_{i=1}^k \exp \left(\frac{\lambda^2 A_i^2}{8} \right) \right) && \text{by Lemma A.2} \\ &= \exp(-\lambda \alpha) \sup \left(\exp \left(\frac{\sum_{i=1}^k \lambda^2 A_i^2}{8} \right) \right) \\ &= \exp(-\lambda \alpha) \exp \left(\sup \left(\frac{\sum_{i=1}^k \lambda^2 A_i^2}{8} \right) \right) \\ &\leq \exp(-\lambda \alpha + \lambda^2 c/8) \end{aligned}$$

Setting $\lambda = 4\alpha/c$, we get that

$$\Pr \left[\sum_{i=1}^k T_i \geq \alpha \right] \leq \exp(-2\alpha^2/c)$$

□

B Analyzing Rejection Sampling

In this section we give some basic facts about *rejection sampling*. For a distribution C supported on some finite set \mathcal{C} and a function $a : \mathcal{C} \rightarrow [0, 1]$, [Figure 8](#) describes a generic rejection sampling algorithm.

Algorithm Rejection Sampling.
<ol style="list-style-type: none"> 1. Sample an element $z \in_{\mathbb{R}} C$. 2. Accept it with probability $a(z)$, else go to the first step.

Figure 8: Generic Rejection Sampling

We prove some simple properties of this kind of sampling. Let D' denote the random variable of the sampled element. Let R denote the random variable that counts the number of rounds before the algorithm accepts the sample. Then we see that D' is independent of R , since for any integers c, c' , $D'|R = c$ has the same distribution as $D'|R = c'$.

Proposition B.1. *D' is independent of R .*

We then see that $D'(w) = \Pr[(R = 1) \wedge w \text{ is accepted}] / \Pr[R = 1] = C(w)a(w) / \Pr[R = 1]$. We have shown the following claim:

Claim B.2. *For some constant α , $D' = \alpha a \cdot C$.*

Set $b = a' - a$. Then $D = \beta a' \cdot C = \beta C \cdot (a + b)$. Thus, by [Claim B.2](#), there exists a distribution D'' such that D' is a convex combination $D = \beta' D'' + (1 - \beta') D'$. In particular, this implies that $\frac{D'(w)}{D(w)} \leq \frac{1}{1 - \beta'}$. We bound $\beta' \leq \Pr[D' \in \text{Supp}(D'')] = \Pr_{w \in_{\mathbb{R}} D}[a'(w) \geq a(w)]$. This gives us the following two bounds:

Proposition B.3. *Let $D = \beta a' \cdot C$ be a distribution such that $a'(w) \geq a(w)$ for every w . Then for every w , $\frac{D'(w)}{D(w)} \leq \frac{1}{1 - \Pr_{w \in_{\mathbb{R}} D}[a'(w) \geq a(w)]}$.*

Proposition B.4. *The expected number of rounds that the above protocol runs for is $1 / \Pr[R = 1]$.*

Proof. From the construction, we see that $\mathbb{E}[R] = \Pr[R = 1] + (1 - \Pr[R = 1])(1 + \mathbb{E}[R])$. Rewriting this, we get $\mathbb{E}[R] = 1 / \Pr[R = 1]$. □

C Finding The First Difference in Inputs

Proof Sketch for [Lemma 2.14](#). Without loss of generality, we assume that $k = 2^t$ for an integer t (if not, we can always pad the input strings with 0's until the lengths are of this form before running the protocol). For a parameter C , we define a labeled tree of depth $C \log(k/\epsilon) = C(t + \log(1/\epsilon))$ as follows. The root of the tree is labeled by the interval $[1, 2^t]$. For i ranging from 0 to $t - 1$, every node at depth i labeled by $[a, b]$ has two children, corresponding to splitting the interval $[a, b]$ into equal parts. Thus the left one is labeled by the interval $[a, b - 2^{t-i+1}]$ and the right one is labeled by $[a + 2^{t-i+1}, b]$. Thus at depth t there are 2^t nodes, each labeled by $[a, a]$ for distinct a 's from $[2^t]$. Every node at depth $\geq t$ has exactly one child, labeled the same as the parent.

In the protocol, the players shall try to narrow down where the first difference in their inputs is by taking a walk on the tree. At each step, the players first check that the interval they are on is correct, and then try to narrow down their search. For any integer $a \in [n]$, let x_a denote

the prefix of x of length a . To check whether a given interval $[a, b]$ contains the index that they seek, the players will use public randomness to pick random functions $h_1 : \{0, 1\}^a \rightarrow [18]$ and $h_2 : \{0, 1\}^b \rightarrow [18]$ and compare $h_1(x_a)$ with $h_1(y_a)$ and $h_2(x_b)$ with $h_2(y_b)$. The probability of getting an incorrect answer is thus at most $1/9$.

For a parameter C , the protocol works as follows:

1. The players set v to be the root of the tree.
2. The players run the tests described above to check whether the index with the first difference lies in the interval corresponding to v and in those corresponding to v 's children. If the tests are consistent, and indicate that the interval for v does not contain the index, the players set v to be the parent of the old v (or leave it unchanged if v is the root). If the tests are consistent and indicate that the interval of one of the children contains the index, the players set v to be that child. If the tests are inconsistent, the players leave v unchanged.
3. Step 2 is repeated $C(t + \log(1/\epsilon))$ times.
4. If the final vertex is labeled by an interval of the form $[a, a]$, output a . Else conclude that the input strings are equal.

To analyze the protocol, fix x and y . Note that if $x = y$, then the protocol never fails. So let us assume that $x \neq y$ and assume that a is the first index at which x, y differ. Then let w denote the vertex in the tree of largest depth that is labeled by $[a, a]$. Next we direct the edges of the tree so that at every vertex, the only outgoing edge points to the neighbor that is closer to w in terms of shortest path distance. Then observe that at every step of our protocol, v is changed to a neighbor that is closer to w with probability at least $2/3$. Further, our protocol succeeds as long as the number of correct steps on the tree exceeds the number of incorrect steps by t . This happens as long as the number of correct steps is at least $C/2(t + \log(1/\epsilon)) + t/2$. Since the expected number of correct steps is $2C/3(t + \log(1/\epsilon))$, we get that the bad event happens only when we deviate from the expected number by $C/6(t + \log(1/\epsilon)) - t/2 > (C/6 - 1/2)(t + \log(1/\epsilon))$. By the Chernoff bound, the probability that this happens is at most $\exp(\Omega((C/6 - 1/2)^2(t + \log(1/\epsilon))))$. Setting C to be a large enough constant makes this error at most ϵ .

□

References

- [Ab193] Farid Ab1ayev. Lower bounds for one-way probabilistic communication complexity. In Andrzej Lingas, Rolf Karlsson, and Svante Carlsson, editors, *Proceedings of the 20th International Colloquium on Automata, Languages, and Programming*, volume 700 of *LNCS*, pages 241–252. Springer-Verlag, 1993.
- [BYJKS04] Ziv Bar-Yossef, T. S. Jayram, Ravi Kumar, and D. Sivakumar. An information statistics approach to data stream and communication complexity. *Journal of Computer and System Sciences*, 68(4):702–732, 2004.
- [CSWY01] Amit Chakrabarti, Yaoyun Shi, Anthony Wirth, and Andrew Yao. Informational complexity and the direct sum problem for simultaneous message complexity. In Bob

Werner, editor, *Proceedings of the 42nd Annual IEEE Symposium on Foundations of Computer Science*, pages 270–278, Los Alamitos, CA, October 14–17 2001. IEEE Computer Society.

- [FKNN95] Tomàs Feder, Eyal Kushilevitz, Moni Naor, and Noam Nisan. Amortized communication complexity. *SIAM Journal on Computing*, 24(4):736–750, 1995.
- [FPRU94] Uriel Feige, David Peleg, Prabhakar Raghavan, and Eli Upfal. Computing with noisy information. *SIAM Journal on Computing*, 23(5):1001–1018, 1994.
- [HJMR07] Prahladh Harsha, Rahul Jain, David A. McAllester, and Jaikumar Radhakrishnan. The communication complexity of correlation. In *IEEE Conference on Computational Complexity*, pages 10–23. IEEE Computer Society, 2007.
- [JRS03] Rahul Jain, Jaikumar Radhakrishnan, and Pranab Sen. A direct sum theorem in communication complexity via message compression. In Jos C. M. Baeten, Jan Karel Lenstra, Joachim Parrow, and Gerhard J. Woeginger, editors, *ICALP*, volume 2719 of *Lecture Notes in Computer Science*, pages 300–315. Springer, 2003.
- [JRS05] Rahul Jain, Jaikumar Radhakrishnan, and Pranab Sen. Prior entanglement, message compression and privacy in quantum communication. In *IEEE Conference on Computational Complexity*, pages 285–296. IEEE Computer Society, 2005.
- [JHM⁺98] Mark Jerrum, Michel Habib, Colin McDiarmid, Jorge L. Ramirez-Alfonsin, and Bruce Reed. *Probabilistic Methods for Algorithmic Discrete Mathematics*, volume 16 of *Algorithms and Combinatorics*. Springer-Verlag, 1998.
- [KS92] Bala Kalyanasundaram and Georg Schnitger. The probabilistic communication complexity of set intersection. *SIAM Journal on Discrete Mathematics*, 5(4):545–557, November 1992.
- [KN97] Eyal Kushilevitz and Noam Nisan. *Communication complexity*. Cambridge University Press, Cambridge, 1997.
- [Raz98] Ran Raz. A parallel repetition theorem. *SIAM Journal on Computing*, 27(3):763–803, June 1998.
- [Raz92] Razborov. On the distributed complexity of disjointness. *TCS: Theoretical Computer Science*, 106, 1992.
- [SS02] Michael Saks and Xiaodong Sun. Space lower bounds for distance approximation in the data stream model. In ACM, editor, *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*, pages 360–369. ACM Press, 2002.