Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 101 (2014)

Internal compression of protocols to entropy

Balthazar Bauer[*] Shay Moran[†] Amir Yehudayoff[‡]

Abstract

We study internal compression of communication protocols to their internal entropy, which is the entropy of the transcript from the players' perspective. We first show that errorless compression to the internal entropy (and hence to the internal information) is impossible. We then provide two internal compression schemes with error. One of a protocol of Fiege et al. for finding the first difference between two strings. The second and main one is an internal compression with error $\varepsilon > 0$ of a protocol with internal entropy H^{int} and communication complexity C to a protocol with communication at most order $(H^{int}/\varepsilon)^2 \log(\log(C))$.

This immediately implies a similar compression to the internal information of public coin protocols, which exponentially improves over previously known public coin compressions in the dependence on C. It further shows that in a recent protocol of Ganor, Kol and Raz it is impossible to move the private randomness to be public without an exponential cost. To the best of our knowledge, no such example was previously known.

1 Introduction

The problem of compressing information and communication is fundamental and useful. The basic scenario, the transmission problem, was studied in Shannon's seminal work [21]. In it Alice wishes to transmit to Bob a message $u \in U$ with u that is distributed according to a known distribution μ over U. Shannon proved that the above transmission can be optimally compressed in the sense that Alice may send u to Bob using roughly $\log(1/\mu(u))$ many bits on average, and conversely if Alice sends fewer

[*] Département d'Infomatique, ENS Lyon, Lyon, France. balthazarbauer@aol.com.

[†] Departments of Mathematics and Computer Science, Technion-IIT, Israel. shaymrn@cs.technion.ac.il.

[‡] Department of Mathematics, Technion-IIT, Israel. amir.yehudayoff@gmail.com. Horev fellow – supported by the Taub foundation. Supported by ISF and BSF.

ISSN 1433-8092

than $\log(1/\mu(u))$ bits on average then information is lost. In the transmission problem, the information flow is one way, only Alice talks.

How about more complex communication protocols in which both sides are allowed to talk? The standard model for interactive communication was introduced by Yao [24]. Interactive communication, not surprisingly, allows for more efficient conversations than one way ones. For example, the following lemma (which we also use later on) demonstrates the power of interaction (and of public randomness) in handling a variant of the transmission problem in which only Bob knows the distribution μ over U.

Lemma 1.1. *Let U be a finite set, and $0 < \varepsilon < 1/2$. Assume Alice knows some $u_a \in U$ and that Bob knows a distribution μ on U which Alice does not know. Using public randomness, Alice and Bob can communicate at most $2\log(1/\mu(u_a)) + \log(1/\varepsilon) + 5$ bits, after which Bob outputs u_b so that $u_a = u_b$ with probability at least $1 - \varepsilon$.*

This lemma describes a one shot protocol (i.e. for a single instance) that enables transmission when Bob has some prior knowledge on Alice's input. A stronger version of this lemma was proved in [6] and also in [7], but since this lemma is sufficient for us and its proof is simpler than that of [6, 7] we provide its proof in Section A.3. A related result for the case when there is also an underlying distribution on Alice's input is the Slepian-Wolf theorem [22] which solves an amortized version of this problem. It is also related to the transmission problem considered by Harsha et al. [15] who studied the case that Alice knows μ and Bob wishes to sample from it.

Continuing recent works which we survey below, the main question we study is compression of interactive communication protocols. Compression of protocols, on a high level, means to simulate a given protocol π by a more efficient protocol σ whose communication complexity is roughly the information content of π. It was recently shown to be strongly related to direct sum and product questions in randomized communication complexity [6, 3, 8].

Our results include impossibility of errorless compression, and new internal compression schemes. We also provide an extensive preliminary discussion of concepts and basic facts related to compression.

1.1 A preliminary discussion

In this section we provide intuitive definitions of important concepts. See Section 2 for formal definitions.

Computation and simulation. There is a distinction between external computation and internal computation [3, 8]. A protocol externally computes a function f if an external observer can deduce the value of f from the transcript, and a protocol internally computes f if the value of f may be privately obtained by Alice and Bob but not necessarily by an external observer.

It is interesting that for deterministic protocols these two seemingly different notions coincide, so the strength of internal computation is evident only in randomized or distributional settings (the proof is given in Section A.1).

Proposition 1.2. *Let $f : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$. If π is a deterministic protocol that internally computes f then it also externally computes f.*

External and internal computations induce the corresponding types of simulations. Here we provide an intuitive meaning to the notion of simulation. In Section 2 we provide formal definitions, and discuss them in more detail. A protocol σ externally simulates the protocol π if an external observer who has access only to the public data (i.e. transcript and public randomness) of σ can deduce from it the public data of π. The protocol σ internally simulates π if each of Alice and Bob can obtain their private data of π from their private data of σ.

As an example which illustrates the difference between internal and external simulation, consider the simple case when (x, y) are jointly distributed so that $x = y$, Alice knows x, Bob knows y and π is the protocol in which Alice sends x to Bob. In this case, it is clear that the empty protocol internally simulates π but every external simulation of π must in general use many bits. This example also demonstrates that Proposition 1.2 does not hold for promise problems, when the inputs are guaranteed to come from a non rectangular set.

Information complexities. The most studied measures in the context of protocol compression are information complexities. For every communication protocol π and every distribution μ on inputs, two versions of information have been defined: The internal information [1, 3] denoted $I_\mu^{int}(\pi)$ and the external one [10] denoted $I_\mu^{ext}(\pi)$. The semantic of internal information is the amount of information the communication transcript reveals to Alice and Bob about the inputs, and the semantic of external information is the amount of information the communication transcript reveals to an external observer about the inputs. It always holds that the internal information is at most the external one, which is at most the average communication complexity $\mathsf{CC}_\mu^{avg}(\pi)$ (see e.g. [3, 16]).

The following claim shows that information provides a lower bound for errorless simulations. This generalizes the basic fact that entropy provides a lower bound for errorless transmission. This claim seems to be known but we could not find an explicit reference to it so we provide a proof in Section A.2 (the special case of deterministic external simulation was proved in [19]).

Claim 1.3. *Let π be a general protocol with input distribution μ.*

- *If σ externally simulates π without error then $\mathsf{CC}^{avg}_{\mu}(\sigma) \geq I^{ext}_{\mu}(\pi)$.*
- *If σ internally simulates π without error then $\mathsf{CC}^{avg}_{\mu}(\sigma) \geq I^{int}_{\mu}(\pi)$.*

In the other direction, [3] provided two different compression schemes for general protocols. An external compression with error that uses roughly $I^{ext}_{\mu}(\pi)\log(CC(\pi))$ bits, and an internal compression with error that uses roughly $\sqrt{I^{int}_{\mu}(\pi) \cdot CC(\pi)}$ bits. A second internal compression with error that uses at most roughly $2^{I^{int}_{\mu}(\pi)}$ bits, regardless of $CC(\pi)$, appears in [4]. Later on, [9, 20] showed that the internal compression from [3] applied to public coin protocols yields a much better compression with only order $I^{int}_{\mu}(\pi)\log(CC(\pi))$ bits. We discuss connections of these works to ours below.

Entropy complexities. We consider two additional complexity measures for compression:

The first one, which was studied in [12], is the external entropy $H^{ext}_{\mu}(\pi)$. Its semantic is how many bits are required for describing the transcript of π to an external observer. The second measure we consider is the internal entropy $H^{int}_{\mu}(\pi)$. Its semantic is the number of bits required in order to describe the transcript to Alice plus the number of bits required to describe the transcript to Bob.

Some connections between the information measures and the entropy measures are provided in the following claim.

Claim 1.4. *Let π be a protocol with input distribution μ. Then,*

$$H^{ext}_{\mu}(\pi) \geq I^{ext}_{\mu}(\pi) \quad \textit{and} \quad H^{int}_{\mu}(\pi) \geq I^{int}_{\mu}(\pi).$$

Moreover, if π does not have private randomness then

$$H^{ext}_{\mu}(\pi) = I^{ext}_{\mu}(\pi) \quad \textit{and} \quad H^{int}_{\mu}(\pi) = I^{int}_{\mu}(\pi).$$

As mentioned, in the case of one way deterministic protocols, the external entropy fully captures the compression problem. The above claim combined with Claim 1.3 implies that, more generally, for public coin protocols entropy provides a lower bound on errorless simulation. Interestingly, the authors of [12] proved that this lower bound is essentially tight. They gave an optimal external compression of general protocols (they did not state it in this language).

Theorem 1.5 ([12])**.** *A protocol π can be externally simulated*[1] *without error by a protocol σ so that $\mathsf{CC}^{avg}_\mu(\sigma) \leq O(H^{ext}_\mu(\pi))$.*

With or without error. Another important distinction is between exact simulation and simulation with error.

A meaningful example already appears in the transmission problem, when there is a distribution μ on inputs x and Alice sends a (prefix free) encoding of x to Bob. Any exact solution to this problem requires expected communication of at least $H(\mu)$. However, if μ is highly concentrated on a point but with probability ε it is uniform on the remaining elements, an empty protocol simulates μ with error while the entropy is potentially huge. So entropy and information are not in general lower bounds for simulation with error, and the lower bounds from Claim 1.3 do not hold for simulation with error.

In the other direction, we have seen that entropy (or information) provides a lower bound on errorless simualtion. We shall see below that this lower bound is not tight, that is, there are protocols with small entropy that can not be efficiently simulated without error.

1.2 Internal compression

Impossibility of errorless compression. Theorem 1.5 above provides errorless simulation to external entropy. The main compression question is, however, whether a protocol can be internally simulated with communication that is close to its internal information. We now explain why such a simulation is not available if it is required to be errorless, even for the larger internal entropy.

Theorem 1.6. *For every n and $\delta > 0$, there is a one round deterministic protocol π and input distribution μ so that $H^{int}_\mu(\pi) \leq \delta$ and $CC(\pi) \leq n$ but if σ is an errorless internal simulation of π then $\mathsf{CC}^{avg}_\mu(\sigma) \geq n - 2$.*

Our internal compression scheme and the ones from [3, 4] must therefore introduce errors. The proof of the theorem is given in Section 4.1, and it uses arguments from [16].

[1] They only considered deterministic protocols. Their result applies for general protocols since $H^{ext}_\mu(\pi) = H(T_\pi|R) \geq H(T_\pi|R, R_a, R_b)$, and their compression has expected communication order $H(T_\pi|R, R_a, R_b)$ for general protocols.

Finding the first difference. Before stating our general compression scheme, we demonstrate its ideas by an internal compression of the *finding the first difference* problem, which lies at the heart of the internal compression schemes of [3, 9, 20], and, in a nutshell, allows the players to correct their mistakes. Feige et al. [13] gave an optimal randomized protocol for this problem in terms of communication complexity (Viola [23] proved a corresponding lower bound).

Lemma 1.7 ([13])**.** *There is a public coin protocol that on inputs $x, y \in \{0,1\}^n$ externally outputs the smallest index i in which x, y differ (or outputs "equal" if $x = y$) with probability at least $1 - \varepsilon$. The communication complexity of this protocol is at most $O(\log(n/\varepsilon))$.*

The protocol of Feige et al. externally solves the problem. The following is an internal protocol for it (the protocol is presented in Section 3).

Lemma 1.8. *Let μ be a distribution on $(x,y) \in \{0,1\}^n \times \{0,1\}^n$, and let $\varepsilon > 0$. Denote by $i = i(x,y)$ the smallest index in which x, y differ (or $i =$ "equal" if $x = y$). Denote $H^{int} = H(i|x) + H(i|y)$. There is a public coin protocol and an event $\mathcal{E} \subset \{0,1\}^n \times \{0,1\}^n$ with probability $\mu(\mathcal{E}) < \varepsilon$ so that for all $(x,y) \notin \mathcal{E}$, the communication complexity of the protocol on input (x,y) is at most*

$$O\left(\log\left(\frac{1}{\mu(i|x)\cdot\mu(i|y)}\right)\log\left(\log(n)H^{int}/\varepsilon\right)\right),$$

and it internally computes i with probability at least $1 - \varepsilon$. The overall communication complexity with error ε is at most

$$O\left(\frac{H^{int}}{\varepsilon}\log\left(\log(n)H^{int}/\varepsilon\right)\right).$$

We state the lemma in this form since it hints at the core of its proof. To understand it better, it may be helpful to observe

$$H(i|x) + H(i|y) = I(i;y|x) + I(i;x|y) = \mathbb{E}_\mu \log\left(\frac{1}{\mu(i|x)\cdot\mu(i|y)}\right).$$

This protocol gives an improvement over that of [13] when the internal entropy is small. It highlights the importance of internal computation and may help to understand the more general compression below. It may also be useful in future internal compression schemes.

Main compression. We finally state our internal compression scheme (see Section 4.2

for its description). As mentioned above, such a compression must have positive error, even for one round protocols.

Theorem 1.9. *Let μ be a distribution on $\mathcal{X} \times \mathcal{Y}$ and let $\varepsilon > 0$. Let π be a protocol with inputs from μ. Then, there is a public coin protocol σ with communication complexity*

$$CC(\sigma) \leq O\left(\frac{\left(H_\mu^{int}(\pi)\right)^2}{\varepsilon^2} \cdot \log(\log(CC(\pi)))\right)$$

that internally simulates π with error ε.

As noted earlier, if π is a protocol that uses no private randomness then the internal entropy of π is equal to the internal information of π. So, for public coin protocols, Theorem 1.9 gives an internal compression to internal information with an exponential improvement over [9, 20] in terms of the dependence on $CC(\pi)$. It, therefore, also concerns the power of private randomness in saving information, which we now discuss.

Transferring private to public randomness. Every private coin protocol can be simulated by a public coin protocol with the same *communication* complexity. Conversely, Newman [18] proved that for communication complexity public randomness may be efficiently replaced by private one, when dealing with computation of relations (it however does not yield a communication efficient simulation of public coin protocols by private coin protocols). In the information complexity context the situation is opposite, every public coin protocol can be simulated by a private coin protocol with the same *information* complexity. The authors of [9, 5] showed that for information complexity private randomness may be relatively efficiently simulated by public one. If this last simulation was efficient enough, to compress general protocol it would suffice to compress public coin protocols.

Our compression shows limitations on moving private randomness to being public. A recent work of Ganor, Kol and Raz [14] shows that for every large enough $k \in \mathbb{N}$ there is a distribution μ and a private coin protocol π_0 with internal information $O(k)$ so that every protocol that internally simulates π_0 with small error must communicate at least 2^k bits. This marks the first known separation between information and communication complexities[2]. The protocol π_0 has communication complexity $O(k \cdot 2^{4^k})$ so that $\log(\log(CC(\pi_0)) = O(k)$. Together with our compression scheme, this means that there is no way to simulate π_0 using only public randomness without a cost; for example, every public coin internal simulation of π_0 with optimal communication complexity $2^{O(k)}$ must have exponentially large internal information, at least $2^{\Omega(k)}$.

[2] Part of the difficulty in proving such a separation is proving a lower bound for internal computation (rather than the more standard external computation).

Discussion of proof. Compression to internal entropy, as mentioned above, must be done in an internal way. That is, an observer of the conversation (who does not know the inputs nor the private randomness) should not be able to make much sense of it.

The only two compression schemes with this property that were previously known are from [2, 4]. The scheme from [4] is not efficient in terms of information complexity so we do not discuss it in detail here. In the scheme from [2] the players use public randomness to jointly but privately sample a possible transcript, and they communicate only to fix errors. Each error fixing costs about $\log(CC(\pi))$ communication. The main problem in analyzing their protocol is bounding the number of errors in terms of the internal information. They are able to do so but the cost is quite high[3] and the overall bound on the number of errors they show is order $\sqrt{CC(\pi) I_\mu^{int}(\pi)}$.

We take a different path which starts with the external compression of deterministic protocols of [12]. The main idea there is that a deterministic protocol induces a distribution on the leaves of the protocol tree, and that there is always a vertex u in the tree with probability mass roughly 1/2 (Lemma 2.1 below). Both players know u and they can check if the rectangle[4] it defines contains x and y with 2 bits of communication. It can easily be shown that by doing so they (roughly) learn one bit of information. This yields an optimal but external compression (an observer knows u as well).

In the internal case, there is no single node that is good for both players. Alice knows a node v_a and Bob a node v_b, which are in general arbitrary nodes in the protocol tree. The crux of our protocol is an efficient way for Alice and Bob to learn enough on v_a, v_b so that they obtain one more bit of information. We show that using Lemma 1.7 one of them, say Alice, can identify a good vertex u to focus on (roughly, u is somewhere in between v_a, v_b). Using Lemma 1.1 Alice then tries to internally transmit u to Bob. If this transmission succeeds, they indeed learn one bit of information. It turns out that even if this transmission fails, they still learn one bit of information. The transmission is indeed internal in that an external observer does not in general learn u even when Bob does. The full protocol appears in Section 4.2.

[3] On a high level this loss occurs for the following reason: If we denote by $h(p)$ the entropy of a random bit with bias $p \in [0, 1]$, then $h(\frac{1}{2} + \delta) - h(\frac{1}{2})$ is of order δ^2. The second power of δ yields the square root $CC(\pi)$ in the analysis. For public coin or deterministic protocols, the authors of [9, 20] showed how to improve the bound on the number of errors to order $I_\mu^{int}(\pi)$ where the improvement comes from that $h(\delta) - h(0)$ is of order δ.

[4] The set of inputs that reach u is a rectangle, that is, it is of the form $\mathcal{X}' \times \mathcal{Y}' \subset \mathcal{X} \times \mathcal{Y}$.

2 Preliminaries

Logarithms in this text are in base two. We provide the basic definitions needed for this text. For background and more details on information theory see the book [11] and on communication complexity see the book [17].

Information theory. The entropy of a random variable X taking values in U is defined as $H(X) = \sum_{u \in U} \Pr[X = u] \log(1/\Pr[X = u])$. The entropy of X conditioned on Y is defined as $H(X|Y) = H(X,Y) - H(Y)$. The mutual information between X, Y conditioned on Z is defined as $I(X;Y|Z) = H(X|Z) - H(X|Y,Z)$.

Protocols. A deterministic communication protocol π with inputs from $\mathcal{X} \times \mathcal{Y}$ is a rooted directed binary tree with the following structure. Edges are directed from root to leaves. Each internal node in the protocol is owned by either Alice or Bob. For every $x \in \mathcal{X}$, each internal node v owned by Alice is associated with an edge $e_{v,x}$ from v to one of the children of v. Similarly, for every $y \in \mathcal{Y}$, each internal node v owned by Bob is associated with an edge $e_{v,y}$. On input x, y, a protocol π is executed by starting at the root and following the unique path defined by x, y until reaching a leaf. We denote by $T_\pi = T_\pi(x,y)$ the leaf reached, which we also call the transcript of π with input (x,y). The length of a transcript, denoted $|T_\pi|$, is the depth of the corresponding leaf.

In a public coin protocol, Alice and Bob also have access to public randomness r that they both know. In a private coin protocol, Alice has access to a random string r_a, and Bob has access to a random string r_b. A general protocol is a protocol which uses both public and private coins. Always the four random variables $(x,y), r, r_a, r_b$ are assumed independent.

The communication complexity of a deterministic π, denoted by $\mathsf{CC}(\pi)$, is the maximum length of a transcript. For general protocols, $\mathsf{CC}(\pi)$ is defined as the maximum communication complexity over all randomness as well (i.e. over x, y, r, r_a, r_b), and $\mathsf{CC}_\mu^{avg}(\pi)$ is the expected length of a transcript over all randomness.

Computation. A deterministic protocol π externally computes a function $f : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$ if there is a map M so that $f(x,y) = M(T_\pi(x,y))$ for all x, y. A deterministic protocol π internally computes a function f if there are two maps M_a, M_b so that $M_a(x, T_\pi(x,y)) = M_b(y, T_\pi(x,y)) = f(x,y)$ for all x, y. In the randomized setting, M may depends on r, M_a may depend on r, r_a, M_b may depend on r, r_b, and the equalities should hold with probability at least $1 - \varepsilon$ over the distribution of r, r_a, r_b for all x, y. In the distributional setting, the probability is taken over x, y as well.

Simulation. Let π, σ be protocols, let μ be a distribution on the input space $\mathcal{X} \times \mathcal{Y}$ and let $\varepsilon \geq 0$. Our goal is defining when σ simulates π with error ε in the distributional setting[5]. Namely, probabilities are taken over all randomness of inputs as well as private and public coins.

The randomness in σ is s, s_a, s_b, and the randomness in π is r, r_a, r_b. We say that σ externally simulates π with error ε if there exists a function M so that, on average, for fixed (x, y), the distribution $p_\sigma^{ext} = p_\sigma^{ext}(x, y)$ of $M(T_\sigma, s)$ is ε close in L_1 distance to the distribution $p_\pi^{ext} = p_\pi^{ext}(x, y)$ of (T_π, r). That is,

$$\mathbb{E}_\mu |p_\sigma^{ext} - p_\pi^{ext}|_1 \leq \varepsilon.$$

The function M can be thought of as the dictionary translating the language of σ to the language of π.

We say that σ internally simulates π with error ε if there exist functions M_a, M_b such that, on average, for fixed (x, y), the distribution $p_\sigma^{int} = p_\sigma^{int}(x, y)$ of the pair

$$(M_a(x, T_\sigma, s, s_a), M_b(y, T_\sigma, s, s_b))$$

is ε close to the distribution $p_\pi^{int} = p_\pi^{int}(x, y)$ of the pair

$$((T_\pi, r, r_a), (T_\pi, r, r_b)).$$

That is,

$$\mathbb{E}_\mu |p_\sigma^{int} - p_\pi^{int}|_1 \leq \varepsilon.$$

In the internal case, there are two internal dictionaries, one for Alice and one for Bob.

The simulation we present in the proof of Theorem 1.9 is in fact of a stronger form. In the beginning of σ, Alice and Bob interpret the public randomness as a pair $s = (r, s')$ and their private randomness as $s_a = (r_a, s'_a)$ and $s_b = (r_b, s'_b)$. They think of r, r_a, r_b as the fixed randomness of π, and communicate in order to privately compute the fixed transcript $T_\pi = T_\pi(x, y, r, r_a, r_b)$, with error probability (over the remaining randomness s', s'_a, s'_b) of at most ε.

This stronger type of simulation is sometimes too strong to be useful, as the following example demonstrates. Consider a protocol in which x, r_a are uniform in $\{0, 1\}^n$, and Alice just sends $x + r_a \in \{0, 1\}^n$ to Bob. The transcript of this protocol is just a random noise, and its external information is indeed zero. It can, indeed, be externally simulated without error by a protocol with zero communication; interpret s as a uniform element in $\{0, 1\}^n$ and set $M(\emptyset, s) = (s, \emptyset))$. However, every strong simulation of

[5] There is also a natural variant of this definition in the randomized setting but it is not relevant for this text.

this protocol (as the one in Theorem 1.9 mentioned above) must communicate many bits. Indeed, the transcript of a strong simulation must reveal the value of $x + r_a$ to Bob, with high probability.

Information and entropy of protocols. For a distribution μ on the inputs, define

$$I_\mu^{int}(\pi) = I(T_\pi; X|Y, R, R_b) + I(T_\pi; Y|X, R, R_a)$$

and

$$I_\mu^{ext}(\pi) = I(T_\pi; X, Y|R).$$

Similarly, define

$$H_\mu^{int}(\pi) = H(T_\pi|Y, R, R_b) + H(T_\pi|X, R, R_a)$$

and

$$H_\mu^{ext}(\pi) = H(T_\pi|R).$$

Balanced nodes in trees. We use the following well known lemma (see e.g. [17]).

Lemma 2.1. *Let μ be a probability measure on the leaves of a rooted binary tree. The distribution μ may be extended to a function on all nodes in the tree by setting $\mu(v)$ to be the μ-probability that a leaf that is a successor of v is chosen. Then, there exists a node u such that either u is a leaf and $\mu(u) \geq 2/3$, or $1/3 \leq \mu(u) \leq 2/3$.*

3 Finding the first difference

Proof of Lemma 1.8. Denote by $\mathcal{E}$ the (event) set of inputs (x, y) so that

$$\mu(i|x) \cdot \mu(i|y) < 2^{-H^{int}/\varepsilon}.$$

By Markov's inequality,

$$\mu(\mathcal{E}) < \varepsilon.$$

For inputs in $\mathcal{E}$, the protocol may fail. For the rest of the proof, fix $(x, y) \notin \mathcal{E}$ and set $i = i(x, y)$.

The protocol proceeds in iterations indexed by $t \in \mathbb{N}$. For every t, Alice knows a distribution α_t on $[n] \cup \{\text{“}equal''\}$ and Bob a distribution β_t on $[n] \cup \{\text{“}equal''\}$ where we use the order $1 < 2 < \ldots < n <$ “$equal''$. They start with

$$\alpha_0(j) = \Pr_\mu[i = j|x] \;\text{ and }\; \beta_0(j) = \Pr_\mu[i = j|y]$$

for all j. Iteration t starts with Alice knowing α_t and Bob knowing β_t, and ends with an update of this distributions to $\alpha_{t+1}, \beta_{t+1}$. There are $O(H^{int}/\varepsilon)$ iterations, and the probability of failure in each iteration is $O(\delta)$ for $\delta = c\varepsilon^2/H^{int}$ for a small constant $c > 0$. The union bound implies that the overall error in the part is at most ε.

The goal of every iteration is, given α_t, β_t, to construct with probability at least $1 - O(\delta)$ distributions $\alpha_{t+1}, \beta_{t+1}$ so that (if they did not stop)

$$\alpha_{t+1}(i) \geq \alpha_t(i) \ , \quad \beta_{t+1}(i) \geq \beta_t(i)$$

and

$$\alpha_{t+1}(i) \cdot \beta_{t+1}(i) \geq \frac{3}{2} \cdot \alpha_t(i) \cdot \beta_t(i).$$

This immediately implies that the number of iterations is at most $O(H^{int}/\varepsilon)$ since we conditioned on not $\mathcal{E}$ and since α_t, β_t are always probability distributions so their maximum value is at most 1.

The protocol uses the following subroutine we call *check*(j) with error δ. It gets as input $j \in [n] \cup \{``equal''\}$ and with communication $O(\log(1/\delta))$ it externally outputs "yes" if $j = i$ and "no" if $j \neq i$. This subroutine just uses public randomness[6] to check if $x_{<j} = y_{<j}$ and $x_j \neq y_j$ for $j \in [n]$ or if $x = y$ for $j = ``equal''$.

Iteration t is performed as follows:

1. If $\alpha_t(j) > 1/3$ for some j then they check(j) with error δ. If the answer is "yes" then they stop and output j. If the answer is "no" then they update α_t, β_t to $\alpha_{t+1}, \beta_{t+1}$ by conditioning on the event $([n] \cup ``equal'') \setminus \{j\}$ and continue to the next iteration.

2. If $\beta_t(j) > 1/3$ for some j then they check(j) with error δ. If the answer is "yes" then they stop and output j. If the answer is "no" then they update α_t, β_t to $\alpha_{t+1}, \beta_{t+1}$ by conditioning on the event $([n] \cup ``equal'') \setminus \{j\}$ and continue to the next iteration.

3. Let d_a be the maximum integer so that $\alpha_t(\{1, 2, \ldots, d_a - 1\}) < 2/3$ and let d_b be the maximum integer so that $\beta_t(\{1, 2, \ldots, d_b - 1\}) < 2/3$. Alice knows d_a and Bob d_b. Using the protocol from Lemma 1.7, with communication $O(\log(\log(n)/\delta))$ they find[7] d that is between d_a, d_b with error δ.

4. They check using public randomness with error δ if $x_{<d} = y_{<d}$.

[6] For examples, using the standard randomized protocol for equality [17].

[7] Thinking of d_a, d_b as binary strings of length order $\log(n)$, to find d it suffices to find the first index in which d_a, d_b differ.

If the answer is "yes" then they update α_t, β_t to $\alpha_{t+1}, \beta_{t+1}$ by conditioning on the event $\{d, d+1, \ldots, n\} \cup \{\text{"}equal''\}$ and continue to the next iteration.

If the answer is "no" then they update α_t, β_t to $\alpha_{t+1}, \beta_{t+1}$ by conditioning on the event $\{1, 2, \ldots, d-1\}$ and continue to the next iteration.

We analyse the correctness step by step assuming that no error occurred (we have already bounded the probability of error):

1. If they output j then indeed the output is correct. Otherwise, $j \neq i$ which means that
$$\alpha_{t+1}(i) = \frac{\alpha_t(i)}{1 - \alpha_t(j)} > \frac{\alpha_t(i)}{2/3}$$
and $\beta_{t+1}(i) \geq \beta_t(i)$.

2. As in previous case.

3. If they reached here then $\alpha_t(j), \beta_t(j) \leq 1/3$ for all j. They find d that is between d_a, d_b. Assume without loss of generality that $d_a \leq d_b$. The proof in the other case is similar.

4. If $x_{<d} = y_{<d}$ then $i \geq d \geq d_a$. This implies that $\beta_{t+1}(i) \geq \beta_t(i)$. By choice,
$$\alpha_t(\{d, d+1, \ldots, n\}) = \alpha_t(d) + 1 - \alpha_t(\{1, \ldots, d\}) \leq \frac{1}{3} + \frac{1}{3} \leq \frac{2}{3},$$
which implies $\alpha_{t+1}(i) \geq 3\alpha_t(i)/2$.

   If $x_{<d} \neq y_{<d}$ then $i < d \leq d_b$. This implies that $\alpha_{t+1}(i) \geq \alpha_t(i)$. By choice,
$$\beta_t(\{1, 2, \ldots, d-1\}) \leq \frac{2}{3},$$
which implies $\beta_{t+1}(i) \geq 3\beta_t(i)/2$.

□

4 Internal compression

4.1 No errorless internal compression

Proof of Theorem 1.6. The inputs to π are $(x, y) \in \{0,1\}^n \times \{0,1\}^n$, and in it Alice just sends x to Bob bit by bit.

The input distribution μ is defined as follows. Let $\gamma > 0$ be small enough (to be determined). With probability $1-\gamma$, the input (x,y) is chosen uniformly from the set $\{(z,z)\} \subset \{0,1\}^n \times \{0,1\}^n$, and with the remaining probability γ, it is chosen uniformly from $\{0,1\}^n \times \{0,1\}^n$.

Clearly, $CC(\pi) \leq n$ and if γ is small enough then

$$H_\mu^{int}(\pi) = H(X|Y) \leq \delta.$$

Now, let σ be a errorless internal simulation of π. So, there is a map M_b so that $M_b(T_\sigma(x,y),y,s,s_b) = x$ for all x,y,s,s_b. That is, at the end of σ, Bob knows x. Let τ be the protocol obtained from σ by that Bob sends an additional bit indicating whether $x=y$ or not. So $\mathsf{CC}_\mu^{avg}(\tau) \leq \mathsf{CC}_\mu^{avg}(\sigma)+1$, and τ is a protocol that externally computes the equality function with zero error over μ. Using arguments from [16], we prove

$$I_\mu^{ext}(\tau) \geq (1-\gamma)n.$$

This finishes the proof since $\mathsf{CC}_\mu^{avg}(\tau) \geq I_\mu^{ext}(\tau)$ and $(1-\gamma)n \geq n-1$ for γ small enough.

We may assume that τ is a private coin protocol (as external information is defined as an average over the public randomness). Let t be a possible transcript of τ, and denote by $J(t)$ the set of all inputs (x,y) so that $\Pr[T_\tau(x,y)=t] > 0$. Since τ has zero error and μ has full support, $J(t) \subset \mathcal{X} \times \mathcal{Y}$ is a monochromatic rectangle. Therefore, if $(z,z) \in J(t)$ for $z \in \{0,1\}^n$ then $J(t) = \{(z,z)\}$.

Let E be indicator random variable of the event $\{(z,z)\}$. Think of $J = J(T_\tau)$ as a random variable (over the randomness in τ). Conditioned on the event $E=1$, the random variable J is basically uniform over $\{0,1\}^n$ so $H\left(J|E=1\right) = n$. It follows that

$$I\left(J;X,Y|E=1\right) = H\left(J|E=1\right) - H\left(J|X,Y,E=1\right) = n-0.$$

Finally,

$$\begin{aligned}
I_\mu^{ext}(\tau) &= I(T_\tau;X,Y) \\
&\geq I(J;X,Y) && \text{(information processing inequality)} \\
&= I(J;X,Y,E) && (X,Y \text{ determine } E) \\
&= I(J;E) + \Pr[E=1]\cdot I\left(J;X,Y|E=1\right) \\
&\qquad + \Pr[E=0]\cdot I\left(J;XY|E=0\right) && \text{(the chain rule)} \\
&\geq (1-\gamma)n.
\end{aligned}$$

□

4.2 An internal compression with error

Proof of Theorem 1.9. Let x, y be the inputs to π, let r be the public randomness, and let r_a, r_b be the private randomness. The first observation is that

$$H^{int} = H^{int}_\mu(\pi) = \mathbb{E}_{x,y,r,r_a,r_b} \log\left(\frac{1}{\mu(T_\pi|x,r,r_a)\cdot\mu(T_\pi|y,r,r_b)}\right),$$

where here $T_\pi = T_\pi(x,y,r,r_a,r_b)$. Denote by $\mathcal{E}$ the event (i.e. set of (x,y,r,r_a,r_b)) that

$$\mu(T_\pi|x,r,r_a)\cdot\mu(T_\pi|y,r,r_b) < 2^{-2H^{int}/\varepsilon}.$$

By Markov's inequality,

$$\Pr(\mathcal{E}) < \varepsilon/2.$$

When $\mathcal{E}$ occurs, the protocol σ may fail, but since it is a rare event it does not really matter. For the rest of the proof, fix $(x,y,r,r_a,r_b)\notin\mathcal{E}$ and set $T_\pi = T_\pi(x,y,r,r_a,r_b)$.

The protocol σ proceeds in iterations indexed by $t\in\mathbb{N}$. The starting point of every iteration is a distribution α_t on leaves of π that Alice knows and a distribution β_t on the leaves of π that Bob knows. These distributions reflect the current perspective of the players after the communication so far. The first distributions are

$$\alpha_0(v) = \Pr[v|x,r,r_a] \text{ and } \beta_0(v) = \Pr[v|y,r,r_b]$$

for all leaves v of the protocol tree (the probability in α_0 for example is over Bob's randomness). The goal of every iteration is to construct with probability at least $1-\delta$ distributions $\alpha_{t+1}, \beta_{t+1}$ so that

$$\alpha_{t+1}(T_\pi) \geq \alpha_t(T_\pi)\ ,\quad \beta_{t+1}(T_\pi) \geq \beta_t(T_\pi)$$

and

$$\alpha_{t+1}(T_\pi)\cdot\beta_{t+1}(T_\pi) \geq \frac{3}{2}\cdot\alpha_t(T_\pi)\cdot\beta_t(T_\pi).$$

The number of iterations is set to be at most

$$O(\log(2^{2H^{int}/\varepsilon})) = O(H^{int}/\varepsilon),$$

and the communication complexity of each iteration is at most

$$O\left(\log\left(\frac{\log(CC(\pi))}{\delta}\right)+\frac{H^{int}}{\varepsilon}+\log(1/\delta)\right).$$

Thus, setting δ smaller than $c\varepsilon^2/H^{int}$ for some small constant $c>0$, the union bound implies the overall bound on the error.

Here is how iteration t is performed:

1. Alice finds a vertex v_a promised by Lemma 2.1 with α_t, and Bob finds v_b promised by Lemma 2.1 with β_t. Denote $d_a = depth(v_a)$ and $d_b = depth(v_b)$.

2. Using the protocol from Lemma 1.7, with communication $O(\log(\log(CC(\pi))/\delta))$ they find[8] d that is between d_a, d_b with error $\delta/2$.

3. If $d_a \geq d_b$, they do the following: Let u be the ancestor of v_a at depth d and let U be the set of nodes of depth d of π . Using the protocol from Lemma 1.1 Alice sends u to Bob. They use this protocol with error parameter $\delta/2$, where Alice's input is u and Bob's input is the distribution β_t induced on U.

   If this stage takes more than $O((H^{int}/\varepsilon)+\log(1/\delta))$ bits, then they abort.

   At the end of this stage, either Bob thinks[9] he knows u as well or they have aborted.

   - If Bob thinks he knows u there are two options:

     If u is a leaf then they stop and the players internally output u.

     Otherwise, they set $\alpha_{t+1} = \alpha_t$ and β_{t+1} to be the distribution β_t conditioned on passing through u.

   - Otherwise, they aborted and they set $\beta_{t+1} = \beta_t$ and α_{t+1} to be the distribution α_t conditioned on not passing through u.

4. When $d_a < d_b$, they exchange roles.

We now analyse the performance in iteration t. For this, we assume that no error occurred. That is, that the protocols from Lemmas 1.7 and 1.1 gave the desired result (this happens with probability at least $1-\delta$). The analysis follows the outline of the protocol:

[8] Think of d_a, d_b as binary strings of length roughly $\log(CC)$.

[9] There is some small probability that Bob holds some $u' \neq u$ but he still thinks he knows u.

1. Lemma 2.1 says that there are always such nodes v_a, v_b.

2. They find d that is between d_a, d_b.

3. We distinguish between two cases:

   Bob thinks he knows u: This means that $\beta_t(u) > 0$ and so (y, r_b) is in the rectangle defined by u. Thus, $((x, r_a), (y, r_b))$ is in the rectangle defined by u, which implies that T_π is a successor of u.

   If u is a leaf then indeed $T_\pi = u$.

   Otherwise, there are two cases:

   The first is when v_b is an ancestor of u. In this case, v_b is not a leaf, $\beta_t(v_b) \geq \beta_t(u)$ and

   $$\beta_{t+1}(T_\pi) = \frac{\beta_{t+1}(T_\pi)}{\beta_t(u)} \geq \frac{\beta_{t+1}(T_\pi)}{\beta_t(v_b)} \geq \frac{\beta_{t+1}(T_\pi)}{2/3}.$$

   The second is when v_b is not an ancestor of u. In this case, $\beta_t(u) \leq 1 - \beta_t(v_b) \leq 2/3$ and

   $$\beta_{t+1}(T_\pi) = \frac{\beta_t(T_\pi)}{\beta_t(u)} \geq \frac{\beta_t(T_\pi)}{2/3}.$$

   Bob does not think he knows u: Since we assumed $\mathcal{E}$ does not occur, if u is an ancestor of T_π then

   $$\beta_t(u) \geq \beta_t(T_\pi) \geq \beta_0(T_\pi) \geq 2^{-2H^{int}/\varepsilon}.$$

   Since they aborted (we ignore possibility of error), this means that u is not an ancestor of T_π. Since u is an ancestor of v_a, $\alpha_t(u) \geq \alpha_t(v_a) \geq 1/3$. Thus, by choice,

   $$\alpha_{t+1}(T_\pi) = \frac{\alpha_t(T_\pi)}{1 - \alpha_t(u)} \geq \frac{\alpha_t(T_\pi)}{1 - \alpha_t(v_a)} \geq \frac{\alpha_t(T_\pi)}{2/3}.$$

4. When $d_a < d_b$, the proof is similar.

□

Acknowledgments

We thank Anup Rao for helpful conversations, and Makrand Sinha for comments on an earlier version of this text.

References

[1] Ziv Bar-Yossef, T. S. Jayram, Ravi Kumar, and D. Sivakumar. An information statistics approach to data stream and communication complexity. *J. Comput. Syst. Sci.*, 68(4):702–732, 2004.

[2] Boaz Barak, Mark Braverman, Xi Chen, and Anup Rao. Direct sums in randomized communication complexity. *ECCC*, 2009.

[3] Boaz Barak, Mark Braverman, Xi Chen, and Anup Rao. How to compress interactive communication. *SIAM J. Comput.*, 42(3):1327–1363, 2013.

[4] Mark Braverman. Interactive information complexity. In Howard J. Karloff and Toniann Pitassi, editors, *STOC*, pages 505–524. ACM, 2012.

[5] Mark Braverman and Ankit Garg. Public vs private coin in bounded-round information. In Javier Esparza, Pierre Fraigniaud, Thore Husfeldt, and Elias Koutsoupias, editors, *ICALP (1)*, volume 8572 of *Lecture Notes in Computer Science*, pages 502–513. Springer, 2014.

[6] Mark Braverman and Anup Rao. Information equals amortized communication. In Rafail Ostrovsky, editor, *FOCS*, pages 748–757. IEEE, 2011.

[7] Mark Braverman, Anup Rao, Omri Weinstein, and Amir Yehudayoff. Direct product via round-preserving compression. In Fedor V. Fomin, Rusins Freivalds, Marta Z. Kwiatkowska, and David Peleg, editors, *ICALP (1)*, volume 7965 of *Lecture Notes in Computer Science*, pages 232–243. Springer, 2013.

[8] Mark Braverman, Anup Rao, Omri Weinstein, and Amir Yehudayoff. Direct products in communication complexity. In *FOCS*, pages 746–755. IEEE Computer Society, 2013.

[9] Joshua Brody, Harry Buhrman, Michal Koucký, Bruno Loff, Florian Speelman, and Nikolay K. Vereshchagin. Towards a reverse newman's theorem in interactive information complexity. In *IEEE Conference on Computational Complexity*, pages 24–33. IEEE, 2013.

[10] Amit Chakrabarti, Yaoyun Shi, Anthony Wirth, , and Andrew Yao. Informational complexity and the direct sum problem for simultaneous message complexity. *Proceedings of the 42nd Annual IEEE Symposium on Foundations of Computer Science*, pages 270–278, 2001.

[11] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory.* Wiley Interscience, 2006.

[12] Martin Dietzfelbinger and Henning Wunderlich. A characterization of average case communication complexity. *Inf. Process. Lett.*, 101(6):245–249, 2007.

[13] Uriel Feige, David Peleg, Prabhakar Raghavan, and Eli Upfal. Computing with noisy information. *SIAM Journal on Computing*, (23(5)):1001–1018, 1994.

[14] Anat Ganor, Gillat Kol, and Ran Raz. Exponential separation of information and communication. *Electronic Colloquium on Computational Complexity*, 2014.

[15] Prahladh Harsha, Rahul Jain, David A. McAllester, and Jaikumar Radhakrishnan. The communication complexity of correlation. *IEEE Transactions on Information Theory*, 56(1):438–449, 2010.

[16] Gillat Kol, Shay Moran, Amir Shpilka, and Amir Yehudayoff. Direct sum fails for zero error average communication. In Moni Naor, editor, *ITCS*, pages 517–522. ACM, 2014.

[17] Eyal Kushilevitz and Noam Nisan. *Communication complexity.* Cambridge University Press, 1997.

[18] Ilan Newman. Private vs. common random bits in communication complexity. *Inf. Process. Lett.*, 39(2):67–71, 1991.

[19] Alon Orlitsky and Abbas El Gamal. Average and randomized communication complexity. *IEEE Transactions on Information Theory*, 36(1):3–16, 1990.

[20] Denis Pankratov. *Direct sum questions in classical communication complexity.* PhD thesis, University of Chicago, 2012.

[21] C.E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 623–656, 1948.

[22] David Slepian and Rahul Jack K. Wolf. Noiseless coding of correlate information sources. *IEEE Transactions on Information Theory*, (19(4)), July 1973.

[23] Emanuele Viola. The communication complexity of addition. In Sanjeev Khanna, editor, *SODA*, pages 632–651. SIAM, 2013.

[24] Andrew Chi-Chih Yao. Some complexity questions related to distributive computing (preliminary report). In Michael J. Fischer, Richard A. DeMillo, Nancy A. Lynch, Walter A. Burkhard, and Alfred V. Aho, editors, *STOC*, pages 209–213. ACM, 1979.

A Appendix

A.1 Internal deterministic computation is also external

Proof of Proposition 1.2. There are maps M_a, M_b so that for all x, y,

$$M_a(x, T_\pi(x,y)) = M_b(y, T_\pi(x,y)) = f(x,y).$$

Fix some rectangle $\rho = \{(x,y) : T_\pi(x,y) = T_\pi(x_0,y_0)\}$. For every $(x,y) \in \rho$, we know $M_a(x,\rho) = f(x,y) = f(x,y_0)$, and similarly $M_b(y_0,\rho) = f(x,y_0) = f(x_0,y_0)$. Therefore, f is constant on ρ and we can define $M(\rho) = f(x_0,y_0)$. □

A.2 Information lower bounds errorless simulation

Proof of Claim 1.3. The external case: Let σ be a protocol that externally simulated π without error. By definition of simulation, there exists a function M so that for all (x,y) so that $\mu(x,y) > 0$, it holds that $p_\sigma = p_\pi$, where p_σ is the distribution of $M(T_\sigma, s)$ and p_π is that of (T_π, r). Thus,

$$\begin{aligned}
\mathsf{CC}_\mu^{avg}(\sigma) &\geq I_\mu^{ext}(\sigma) && \text{(see e.g. [16])}\\
&= I(T_\sigma; X, Y | S) \\
&= I(T_\sigma, S; X, Y) && (S \text{ is independent of } (X,Y))\\
&\geq I(M(T_\sigma, S); X, Y) && \text{(data processing inequality)}\\
&= I(T_\pi, R; X, Y) && \text{(errorless simulation)}\\
&= I(T_\pi; X, Y | R) && (R \text{ is independent of } (X,Y))\\
&= I_\mu^{ext}(\pi).
\end{aligned}$$

The internal case: similarly to the external case,

$$\begin{aligned}
\mathsf{CC}_\mu^{avg}(\sigma) &\geq I_\mu^{ext}(\sigma)\\
&\geq I_\mu^{int}(\sigma)\\
&= I(T_\sigma, S, S_b; X|Y) + I(T_\sigma, S, S_a; Y|X)\\
&\geq I(M_b(T_\sigma, Y, S, S_b); X|Y) + I(M_a(T_\sigma, X, S, S_a); Y|X)\\
&= I(T_\pi, R, R_b; X|Y) + I(T_\pi, R, R_a; Y|X)\\
&= I_\mu^{int}(\pi).
\end{aligned}$$

□

A.3 Transmission

Proof of Lemma 1.1. They interpret the public randomness as boolean random hash functions on U. The protocol proceeds in iterations indexed by $t \in \mathbb{N}$. In iteration $t = 0$, the following is performed:

1. Alice sends $k = \lceil \log(1/\varepsilon) \rceil + 2$ hash values of u_a to Bob.
2. Bob computes the set
$$S_0 = \{u \in U : \mu(u) \in (1/2, 1]\}.$$
He compares every element of S_0 to the k hash values he received. He deletes every $s \in S_0$ that does not agree with at least one of these k hash values. Denote by S_0' the set S_0 after this deletion.

   If S_0' is empty, he sends a "0" to Alice.

   If S_0' is not empty, he sets u_b as an arbitrary element S_0', and sends "1" to Alice, and they stop.

For every $t = 1, 2, \ldots$, the following is performed (until they stop):

1. Alice sends 2 new hash values of u_a to Bob.
2. Bob computes the set
$$S_t = \{u \in U : \mu(u) \in (2^{-t-1}, 2^t]\}.$$
He compares every element of S_t to the $k + 2t$ hash values he received so far. He deletes every $s \in S_t$ that does not agree with at least one of these hash values. Denote by S_t' the set S_t after this deletion.

   If S_t' is empty, he sends a "0" to Alice.

   If S_t' is not empty, he sets u_b as an arbitrary element in S_t', and sends "1" to Alice, and they stop.

We now analyse the protocol. Let t_0 be so that $u_a \in S_{t_0}$. First, the protocol stops after at most $t_0 \leq \log(1/\mu(u_a)) + 1$ iterations, because u_a agrees with all hash values sent. Second, for every t, by the union bound,

$$\Pr[S_t' \neq \{u_a\} \cap S_t] \leq 2^{-(k+2t)} 2^{t+1} \leq 2^{-\log(1/\varepsilon)-t-1} = \frac{\varepsilon}{2^{t+1}}.$$

Thus, by the union bound, the probability that either there is some $t < t_0$ for which $S'_t \neq \emptyset$ or $S'_{t_0} \neq \{u_a\}$ is at most $\sum_{t=0}^{\infty} \varepsilon/2^{t+1} \leq \varepsilon$.

□