A medley for computational complexity: With applications of information theory, learning theory, and Ketan Mulmuley's parametric complexity technique

Loff Barreto, B.S.

Publication date
2014

Citation for published version (APA):
Loff Barreto, B. S. (2014). *A medley for computational complexity: With applications of information theory, learning theory, and Ketan Mulmuley's parametric complexity technique*. [Thesis, externally prepared, Universiteit van Amsterdam].

Chapter 4

Towards a reverse Newman's theorem in information complexity

Newman's theorem states that we can take any public-coin communication protocol and convert it into one that uses only private randomness with but a little increase in communication complexity. We consider a reversed scenario in the context of information complexity: can we take a protocol that uses private randomness and convert it into one that only uses public randomness while preserving the information revealed to each player?

We prove that the answer is yes, at least for protocols that use a bounded number of rounds. As an application, we prove new direct sum theorems through the compression of interactive communication in the bounded-round setting. To obtain this application, we prove a new one-shot variant of the Slepian-Wolf coding theorem, interesting in its own right.

Furthermore, we show that if a Reverse Newman's Theorem can be proven in full generality, then full compression of interactive communication and fully-general direct-sum theorems will result.

The results in this chapter are based on the paper:

- Joshua Brody, Harry Buhrman, Michal Koucký, Bruno Loff, Florian Speelman, and Nikolay Vereshchagin. Towards a reverse Newman's theorem in interactive information complexity. In *Proceedings of the 23rd CCC*, pages 24–33, 2013.

4.1 Introduction

Information cost was introduced by a series of papers [42, 22, 59, 23, 29] as a complexity measure for two-player communication protocols. Internal information cost measures the amount of information that each player learns about the input of the other player while executing a given protocol. In the usual setting of communication complexity we have two players, Alice and Bob, each having an input x and y, respectively. Their goal is to determine the value $f(x, y)$ for some predetermined function f. They achieve the goal by communicating to each other some amount of information about their inputs according to some *protocol.*

The usual measure considered in this setting is the *number of bits* exchanged by Alice and Bob, whereas the internal information cost measures the amount of *information* transferred between the players during the communication. Clearly, the amount of information is upper bounded by the number of bits exchanged but not vice versa. There might be a lengthy protocol (say even of exponential size) that reveals very little information about the players' inputs.

In recent years, a substantial research effort was devoted to proving the converse relationship between the information cost and the length of protocols, i.e., to proving that a protocol which reveals only I bits of information can be simulated by a different protocol which communicates only (roughly) I bits. Such results are known as *compression theorems*. [23] prove that a protocol that communicates C bits and has internal information cost I can be replaced by another protocol that communicates $O(\sqrt{I \cdot C})$ bits. For the case when the inputs of Alice and Bob are sampled from independent distributions they also obtain a protocol that communicates $O(I \cdot \log C)$ bits. These conversions do not preserve the number of rounds. In a follow-up paper, [29] consider a bounded round setting and give a technique that converts the original q-round protocol into a protocol with $O(q \cdot \log I)$ rounds that communicates $O(I + q \log \frac{q}{\varepsilon})$ bits with additional error ε.

All known compression theorems are in the randomized setting. We distinguish two types of randomness — *public* and *private*. Public random bits are seen by both communicating players, and both players can take actions based on these bits. Private random bits are seen only by one of the parties, either Alice or Bob. We use *public-coin* (*private-coin*) to denote protocols that use only public (private) randomness. If a protocol uses both public and private randomness, we call it a *mixed-coin* protocol.

Simulating a private-coin protocol using public randomness is straightforward: Alice views a part of the public random bits as her private random bits, Bob does the same using some other portion of the public bits, and they communicate according to the original private-coin protocol. This new protocol communicates the same number of bits as the original protocol and computes the same function. In the other direction, an efficient simulation of a public-coin protocol using private randomness is provided by Newman's Theorem [83]. Sending over Alice's private random bits to make them public could in general be costly as they may need e.g., polynomially many public random bits, but Newman showed that it suffices for Alice to transfer only $O(\log n + \log \frac{1}{\delta})$ random bits to be able to simulate the original public-coin protocol, up to an additional error of δ.

In the setting of information cost the situation is quite the opposite. Simulating public randomness by private randomness is straightforward: one of the players sends a part of his private random bits to the other player and then they run the original protocol using these bits as the public randomness. Since the random bits contain no information about either input, this simulation reveals no additional information about the inputs; thus the information cost of the protocol stays the same. This is despite the fact that the new protocol may communicate many more bits than the original one.

However, the conversion of a private-randomness protocol into a public-randomness protocol seems significantly harder. For instance, consider a protocol in which in the first round Alice sends to Bob her input x bit-wise XOR-ed with her private randomness. Such a message does not reveal any information to Bob about Alice's input — as from Bob's perspective he observes a random string — but were Alice to reveal her private randomness to Bob, he would learn her complete input x. This illustrates the difficulty in converting private randomness into public.

We will generally call "Reverse Newman's Theorem" (R.N.T.) a result that makes randomness public in an interactive protocol without revealing more information. This chapter is devoted to attacking the following:

> **R.N.T. Question.** *Can we take a private-coin protocol with information cost I and convert it into a public-coin protocol with the same behavior and information cost $\tilde{O}(I)$?*

Interestingly, the known compression theorems [23, 29, 60] give compressed protocols that use only public randomness, and hence as a by-product they give a conversion of private-randomness protocols into public-randomness equivalents. However, the parameters of this conversion are far from the desired ones.[1] In Section 4.4 we show that the R.N.T. question represents the core difficulty in proving full compression theorems; namely, we will prove that any public-coin protocol that reveals I bits of information can already be compressed to a protocol that uses $\tilde{O}(I)$ bits of communication, and hence a fully general R.N.T. would result in fully general compression results, together with the direct-sum results that would follow as a consequence. This was discovered independently by Denis Pankratov, who in his MSc thesis [86] extended the analysis of the [23] compression schemes to show that they achieve full compression in the case when only public randomness is used. Our compression scheme is similar but slightly different: we discovered it originally while studying the compression problem in a Kolmogorov complexity setting (as in [35]), and our proof for the Shannon setting arises from the proper "translation" of this proof; we include it for completeness and because we think it makes for a more elementary proof.

4.1.1 Main results

Our main contribution is a Reverse Newman's Theorem in the bounded-round scenario. We will show that any q-round private-coin protocol can be converted to an $O(q)$-round public-coin protocol that reveals only additional $\tilde{O}(q)$ bits of information (Theorem 4.3.1). Our techniques are new and interesting. Our main technical tool is a conversion of one-round private-randomness protocols into one-round public-randomness protocols. This conversion proceeds in two main steps. After *discretizing* the protocol so that the private randomness is sampled uniformly from some finite domain, we convert the protocol into what we call a 1-1 protocol, which is a protocol having the property that for each input and each message there is at most one choice of private random bits that

[1] We discuss the differences in more detail in Section 4.5.

will lead the players to send that message. We show that such a conversion can be done without revealing too much extra information. In the second step we take any 1-1 protocol and convert it into a public-coin protocol while leaking only a small additional amount of information about the input. This part relies on constructing special bipartite graphs that contain a large matching between the right partition and any large subset of left vertices.

Furthermore, we will prove two compression results for public-randomness protocols: a round-preserving compression scheme to be used in the bounded-round case, and a general (not round-preserving) compression scheme which can be used with a fully general R.N.T. Either of these protocols achieves much better parameters than those currently available for general protocols (that make use of private randomness as well as public). The round-preserving compression scheme is essentially a constant-round average-case one-shot version of the Slepian-Wolf coding theorem [92], and is interesting in its own right.

As a result of our R.N.T. and our round-preserving compression scheme, we will get a new compression result for general (mixed-coin) bounded-round protocols. Whereas previous results for the bounded-round scenario [29] gave compression schemes with communication complexity similar to our own result, their protocols were not round-preserving. We prove that a q-round protocol that reveals I bits of information can be compressed to an $O(q)$-round protocol that communicates $O(I+1) + q\log(\frac{qn}{\delta})$ bits, with additional error δ. As a consequence we will also improve the bounded-round direct-sum theorem of [29].

Organization of the chapter. In Section 4.3 we discuss our Reverse Newman's Theorem. In Section 4.4 we will prove our compression results. Section 4.5 will give applications to direct-sum theorems. Finally, Section 4.6 is dedicated to showing alternatives to the constructions we have presented, as well as bounds that prevent further improvement to our techniques.

4.2 Preliminaries

We use capital letters to denote random variables, calligraphic letters to denote sets, and lower-case letters to denote elements in the corresponding sets. So typically A is a random variable distributed over the set $\mathcal{A}$, and a is an element of $\mathcal{A}$. We will also use capital and lower-case letters to denote integers numbering or indexing certain sequences. We use $\Delta(A, A')$ to denote the *statistical distance* between random variables A and A':

$$\Delta(A, A') = \frac{1}{2}\sum_{a\in\mathcal{A}} |\Pr[A = a] - \Pr[A' = a]| \,.$$

4.2.1 Information theory

For a given probability random variable A distributed over the support $\mathcal{A}$, its entropy is

$$H(A) = \sum_{a\in\mathcal{A}} p_a \log\frac{1}{p_a},$$

where $p_a = \Pr[A = a]$. Given a second random variable B that has a joint distribution with A, the conditional entropy $H(A|B)$ equals

$$\mathbb{E}_{b \in B}[H(A|B = b)].$$

In this chapter, and when clear from the context, we denote a conditional distribution $A|B = b$ more succinctly by $A|b$.

4.2.1. Fact. If A has n possible outcomes then

$$H(A) \leq \log n.$$

4.2.2. Fact.

$$H(A|B) \leq H(A) \leq H(A,B), \quad H(A|B,C) \leq H(A|C) \leq H(A,B|C).$$

4.2.3. Fact.

$$H(A,B) = H(A) + H(B|A), \quad H(A,B|C) = H(A|C) + H(B|A,C).$$

We let $I(A : B)$ denote the Shannon mutual information between A and B, and $I(A : B|C)$ denote the Shannon mutual information between A and B, conditional on C:

$$I(A : B) = H(A) - H(A|B) = H(B) - H(B|A),$$
$$I(A : B|C) = H(A|C) - H(A|B,C) = H(B|C) - H(B|A,C).$$

Notice that $I(A : B|C)$ may be larger than $I(A : B)$, for instance when C is the bitwise XOR of independent A and B.

4.2.4. Fact. The following equality is called *chain rule*:

$$I(A_1, \ldots, A_k : B|C) = I(A_1 : B|C) + \sum_{i=2}^{k} I(A_i : B|C, A_1, \ldots, A_{i-1})$$

Here $A_1, \ldots, A_k$ stands for a random variable in the set of k-tuples and A_i stands for its ith projection.

4.2.5. Fact. A and B are independent conditional on C (which means that whatever outcome c of C we fix, A and B become independent conditional on the event $C = c$) if and only if $I(A : B|C) = 0$.

4.2.6. Fact. If A and B are independent conditional on D then

$$I(A : C|B, D) = I(A : BC|D) \leq I(A : C|D).$$

4.2.7. Fact. If A and C are independent conditional on the pair B, D then

$$I(A : B, C|D) = I(A : B|D).$$

4.2.8. FACT. For any two random variables A, B over the same universe $\mathcal{U}$, it holds that

$$H(A) - H(B) \leq \log(|\mathcal{U}|)\Delta\,(A, B) + 1,$$

Proof. For each u in $\mathcal{U}$, let $c_u = \min\{\Pr[A = u], \Pr[B = u]\}$, $a_u = |Pr[A = u] - c_u|$ and $b_u = |\Pr[B = u] - c_u|$. Then $\delta := \Delta\,(A, B) = \sum_u a_u = \sum_u b_u$, and $1 - \delta = \sum_u c_u$.

So let μ_c, μ_a, μ_b be distributions, with $\mu_c(u) = c_u/(1-\delta)$, $\mu_a(u) = a_u/\delta$, and $\mu_b(u) = b_u/\delta$. Then we can think of A as being generated by tossing a coin A' with bias $\Pr[A' = 1] = \delta$, and if $A' = 1$, then we sample according to μ_a, and if $A' = 0$, we sample according to μ_c. Similary we think of B as being generatedby the toss of a coin B' with the same bias, then sampling according to μ_b if $B' = 1$, and according to μ_c otherwise.

It now follows that:

$$H(A) \leq H(A, A') = H(A') + H(A|A') = H_2(\delta) + (1-\delta)H(\mu_c) + \delta H(\mu_a),$$

where H_2 is the binary entropy function. On the other hand,

$$H(B) \geq H(B|B') \geq (1-\delta)H(\mu_c).$$

This gives us the claimed bound, since $H(\mu_a) \leq \log|\mathcal{U}|$ and $H_2(\delta) \leq 1$. ■

4.2.2 Two-player protocols

We will be dealing with protocols that have both public and private randomness; this is not very common, so we will give the full definitions, which are essentially those of [23, 29]. We will be working exclusively in the distributional setting, meaning that our inputs will be drawn from some distribution, and we will be interested in the average case communication complexity, round complexity, *etc.* From here onwards, we will assume that the input is given to two players, Alice and Bob, by way of two random variables X, Y sampled from a possibly correlated distribution μ over the support $\mathcal{X} \times \mathcal{Y}$.

A *private-coin protocol* π with output set $\mathcal{Z}$ is defined as a rooted tree, called the *protocol tree*, in the following way:

1. Each non-leaf node is owned by either Alice or Bob.
2. If v is a non-leaf node belonging to Alice, then:
   (a) The children of v are owned by Bob; each child is labeled with a binary string, and the set $\mathcal{C}(v)$ of labels of v's children is prefix-free.
   (b) Associated with v is a set $\mathcal{R}_v$, and a function $M_v : \mathcal{X} \times \mathcal{R}_v \to \mathcal{C}(v)$.
3. The situation is analogous for Bob's nodes.
4. With each leaf we associate an *output value* in $\mathcal{Z}$.

On input x, y the protocol is executed as follows:

1. Set v to be the root of the protocol tree.
2. If v is a leaf, the protocol ends and outputs the value associated with v.

3. If v is owned by Alice, she picks a string $r_{A,v}$ uniformly at random from $\mathcal{R}_v$ and sends the label of $M_v(x, r_{A,v})$ to Bob, they both set $v := M_v(x, r_{A,v})$, and return to step 2.
4. If v is owned by Bob, he picks a string $r_{B,v}$ uniformly at random from $\mathcal{R}_v$ and sends the label of $M_v(x, r_{B,v})$ to Alice, they both set $v := M_v(x, r_{B,v})$, and return to step 2.

A general, or *mixed-coin*, protocol is given by a distribution over private-coin protocols. The players run such a protocol by using shared randomness to pick an index r (independently of X and Y) and then executing the corresponding private-coin protocol π_r. A protocol is called *public-coin* if every $\mathcal{R}_v$ has size 1, i.e., no private randomness is used.

We let $\pi(x, y, r, r_A, r_B)$ denote the messages exchanged during the execution of π, for given inputs x, y, and random choices r, r_A and r_B, and $\textsc{Out}_\pi(x, y, r, r_A, r_B)$ be the output of π for said execution. The random variable R is the public randomness, R_A is Alice's private randomness, and R_B is Bob's private randomness; we use Π to denote the random variable $\pi(X, Y, R, R_A, R_B)$.

4.2.9. Definition. The *worst-case communication complexity* of a protocol π, $\mathsf{CC}(\pi)$, is the maximum number of bits that can be transmitted in a run of π on any given input and choice of random strings. The *average communication complexity* of a protocol π, with respect to the input distribution μ, denoted $\mathsf{ACC}_\mu(\pi)$, is the average number of bits that are transmitted in an execution of π, for inputs drawn from μ. The *worst-case number of rounds* of π, $\mathsf{RC}(\pi)$, is the maximum depth reached in the protocol tree by a run of π on any given input. The *average number of rounds* of π, w.r.t. μ, denoted $\mathsf{ARC}_\mu(\pi)$, is the average depth reached in the protocol tree by an execution of π on input distribution μ.

4.2.10. Definition. The *(internal) information cost* of protocol π with respect to μ is:

$$\mathsf{IC}_\mu(\pi) = I(Y : \Pi, R, R_A|X) + I(X : \Pi, R, R_B|Y)$$

Here the term $I(Y : \Pi, R, R_A|X)$ stands for the amount of information Alice learns about Bob's input after the execution of the protocol (and the meaning of the second term is similar). This term can be re-written in several different ways:

$$I(Y : \Pi, R, R_A|X) = I(Y : \Pi|X, R, R_A) = I(Y : \Pi, R|X, R_A),$$
$$I(Y : \Pi, R, R_A|X) = I(Y : \Pi, R|X) = I(Y : \Pi|X, R).$$

Here the first equality holds, as Bob's input Y is independent from randomness R, R_A conditional on X, which is obvious (see Fact 4.2.6 from the preliminaries). The second equality holds, since Y is independent from randomness R conditional on X, R_A, which is also obvious.

The third equality holds, as Y is independent from R_A conditional on Π, X, R (Fact 4.2.7). This independence follows from the rectangle property of protocols: for every fixed Π, X, R the set of all pairs $((Y, R_B), R_A)$ producing the transcript Π is a rectangle and thus the pair (Y, R_B) (and hence Y) is independent from R_A conditional on Π, X, R. The fourth equality is proven similarly to the first and the second ones.

The expressions $I(Y : \Pi, R|X)$ and $I(Y : \Pi|X, R)$ for the information revealed to Alice are the most convenient ones and we will use them throughout this chapter. Similar transformations can be applied to the second term in Definition 4.2.10.

4.2.11. Definition. A protocol π is said to compute function $f : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$ with error probability ε over distribution μ if

$$\Pr_{\mu, R, R_A, R_B}[\textsc{Out}_\pi(x, y, r, r_A, r_B) = f(x, y)] \geq 1 - \varepsilon \ .$$

Many of our technical results require that the protocol uses a limited amount of randomness at each step. This motivates the following definition.

4.2.12. Definition. A protocol π is an *ℓ-discrete protocol*[2] if $|\mathcal{R}_v| = 2^\ell$ at every node of the protocol tree.

When a protocol is ℓ-discrete, we say that it uses ℓ bits of randomness for each message; when ℓ is clear from context, we omit it. While the standard communication model allows players to use an infinite amount of randomness at each step, this is almost never an issue, since one may always "round the message probabilities" to a finite precision. This intuition is captured in the following observation.

4.2.13. Observation. *Suppose π is a private-coin protocol. Then, there exists an ℓ-discrete protocol π' with $\ell = O(\log(|\mathcal{X}|) + \log(|\mathcal{Y}|) + \mathsf{CC}(\pi))$ such that (i) $\mathsf{CC}(\pi') \leq \mathsf{CC}(\pi)$, (ii) $\mathsf{RC}(\pi') \leq \mathsf{RC}(\pi)$, and (iii) for all x, y we have*

$$\Delta\left(\Pi'(x, y, R_A, R_B), \Pi(x, y, R_A, R_B)\right) \leq 2^{-\Omega(\ell)}.$$

Furthermore, for any input distribution μ, the error of π' is at most the error of π plus $2^{-\ell}$. Equally small differences hold between $\mathsf{ACC}_\mu(\pi')$, $\mathsf{ARC}_\mu(\pi')$, and their π equivalents, and $\mathsf{IC}_\mu(\pi')$ is within an additive constant of $\mathsf{IC}_\mu(\pi)$.

Proof. Let π be given by its protocol tree; for each node v, let its corresponding function be $M_v : \mathcal{X} \times \mathcal{R} \to \mathcal{C}(v)$ (if it is Alice's node) or $M_v : \mathcal{Y} \times \mathcal{R}_v \to \mathcal{C}(v)$.

We let π' be given by the same protocol tree but where the functions M_v are restricted to a finite set $\mathcal{R}'_v$ of size $\leq k = 2^{10\ell}$, with $\ell = \log|\mathcal{X}||\mathcal{Y}| + \mathsf{CC}(\pi)$. Hence by construction π' has the same worst-case communication and number of rounds as π.

Let R_v be a random variable uniformly distributed over $\mathcal{R}_v$ and R'_v be a random variable uniformly distributed over $\mathcal{R}'_v$.

[2] In a discrete protocol, we restrict only the amount of private randomness in this definition. It is perhaps natural to also restrict the public randomness, but we will not need to.

4.2.14. Claim. *For any node v of Alice's there is a choice of $\mathcal{R}'_v$ of size $\leq 2^{10\ell}$ such that*

$$|\Pr[M_v(x, R_v) = m] - \Pr[M_v(x, R'_v) = m]| \leq 2^{-4\ell}$$

for every x and m. The obvious analogue holds for Bob's nodes.

We prove that $\mathcal{R}'_v$ exists by the probabilistic method. Let $\tilde{\mathcal{R}} = \{r_1, \dots, r_k\}$ be a random variable which is a multiset obtained by picking k elements uniformly from $\mathcal{R}_v$, and define R'_v as the random variable which picks an element $r_i \in \tilde{\mathcal{R}}$ uniformly at random (counting multiplicities). Let P_m denote the random variable that is

$$P_m = \Pr[M_v(x, R'_v) = m] = \frac{\sum_{i=1}^{k} [M_v(x, r_i) = m]}{k}.$$

By linearity of expectation we find that:

$$\mathbb{E}[P_m] = \frac{\sum_{i=1}^{k} \mathbb{E}[M_v(x, r_i) = m]}{k} = \Pr[M_v(x, R_v) = m].$$

And hence by Hoeffding's inequality we conclude that:

$$\Pr[|P_m - \Pr[M_v(x, R_v) = m]| > 2^{-4\ell}] \leq 2 \exp\left(-2k2^{-8\ell}\right) \ll 2^{-\ell}.$$

Hence by a union bound there must exist a choice for $\tilde{\mathcal{R}}$ such that

$$|P_m - \Pr[M_v(x, R_v) = m]| \leq 2^{-4\ell}$$

holds for every x and m; this choice is $\mathcal{R}'_v$.

Now fix x, y; from the claim it follows that for any transcript t,

$$|\Pr[\pi(x, y) = t] - \Pr[\pi'(x, y) = t]| \leq 2^{-3\ell},$$

which in turn implies that

$$\Delta\left(\Pi(x, y, r_A, r_B), \Pi'(x, y, \mathcal{R}'^{(a)}, \mathcal{R}'^{(b)})\right) \leq 2^{-2\ell}.$$

This results in a difference of $\leq 2^{-\ell}$ in success probability, average communication complexity, and average number of rounds, for any given input distribution. The technique we used is very similar to Newman's proof of his theorem, and we could have bounded the ammount of private randomness to something exponentially smaller, while achieving similar bounds.

However, to prove that there is a small difference in information cost, we need ℓ to be as large as $\log |\mathcal{X}||\mathcal{Y}| + \mathsf{CC}(\pi)$. Begin by noting that:

$$I(\Pi : X|Y) = H(\pi(X, Y, R)|Y) - H(\pi(X, Y, R)|X, Y),$$

and then use Fact 4.2.8 to conclude that

1. $|H(\pi(X, Y, R)|Y = y) - H(\pi'(X, Y, R')|Y = y)| = O(1)$ for all y, and

2. $|H(\pi(X,Y,R)|X=x,Y=y) - H(\pi'(X,Y,R')|X=x,Y=y)| = O(1)$ for any x, y, and hence

3. $|I(\Pi : X|Y) - I(\Pi' : X|Y)| = O(1)$,

By a symmetric reasoning for Bob, we find that $|\mathsf{IC}_\mu(\pi) - \mathsf{IC}_\mu(\pi)| = O(1)$. ■

Hence, while working exclusively with discretized protocols, our theorems will also hold for non-discretized protocols, except with an additional exponentially small error term. We consider this error negligible, and hence avoid discussing it beyond this point; the reader should bear in mind, though, that when we say that we are able to simulate a discretized protocol exactly, this will imply that we can simulate any protocol with $2^{-\Omega(\ell)}$ error.

We are particularly interested in the case of one-way protocols. In a one-way protocol, Alice sends a single message to Bob, who must determine the output. A one-way protocol π is thus given by a function $M_\pi : \mathcal{X} \times \mathcal{R} \mapsto \mathcal{M}$; on input x Alice randomly generates r and sends $M_\pi(x,r)$. Note that if π is private-coin, then $\mathsf{IC}_\mu(\pi) = I(X : M(X, R_A)|Y)$, and similarly, if π is public-coin, then $\mathsf{IC}_\mu(\pi) = I(X : R, M(X,R)|Y)$.

Finally, we close this section with a further restriction on protocols, which we call 1–1. Proving an R.N.T. result for 1–1 protocols will be a useful intermediate step in the general R.N.T. proof.

4.2.15. Definition. A one-way protocol π is a 1–1 protocol if $M_\pi(x,\cdot)$ is 1–1 for all x.

4.3 Towards a Reverse Newman's Theorem

Our main result is the following:

4.3.1. Theorem (Reverse Newman's Theorem, bounded-round version). *Let π be an arbitrary, ℓ-discrete, mixed-coin, q-round protocol, and let $C = \mathsf{CC}(\pi)$, $n = \max\{\log|\mathcal{X}|, \log|\mathcal{Y}|\}$. Suppose that π's public randomness R is chosen from the uniform distribution over the set $\mathcal{R}$, and π's private randomness R_A and R_B is chosen from uniform distributions over the sets $\mathcal{R}_A$ and $\mathcal{R}_B$, respectively.*

Then there exists a public-coin, q-round protocol $\tilde{\pi}$, whose public randomness R' is drawn uniformly from $\mathcal{R} \times \mathcal{R}_A \times \mathcal{R}_B$, and that has the exact same transcript distribution, i.e., for any input pair x, y and any message transcript t,

$$\Pr[\pi(x,y,R,R_A,R_B) = t] = \Pr[\tilde{\pi}(x,y,R') = t],$$

and for any distribution μ giving the input (X,Y),

$$\mathsf{IC}_\mu(\tilde{\pi}) \leq \mathsf{IC}_\mu(\pi) + O\left(q \log\left(2n\ell\right)\right). \tag{4.1}$$

We conjecture, furthermore, that a fully general R.N.T. holds:

4.3.2. Conjecture. *Theorem 4.3.1 holds with (4.1) replaced by*

$$\mathsf{IC}_\mu(\tilde{\pi}) \le \tilde{O}(\mathsf{IC}_\mu(\pi)),$$

where $\tilde{O}(\cdot)$ suppresses terms and factors logarithmic in $\mathsf{IC}_\mu(\pi)$ and $\mathsf{CC}(\pi)$.

In Sections 4.4 and 4.5, we show that R.N.T.s imply fully general compression of interactive communication, and hence the resulting direct-sum theorems in information complexity. This results in new compression and direct-sum theorems for the bounded-round case. We believe that attacking Conjecture 4.3.2, perhaps with an improvement of our techniques, is a sound and new approach to proving these theorems.

Before proving Theorem 4.3.1 let us first remark that it suffices to show it only for protocols π without public randomness (with an absolute constant in the O-notation). To see this, fix any outcome r of the random variable R, and look at the protocol π conditioned on $R = r$. This is a protocol without public randomness, let us denote it by π_r. Using the expression

$$I(X : \Pi | Y, R) + I(Y : \Pi | X, R)$$

for information cost of π, we see that it equals the average information cost of the protocol π_r. Therefore, assuming that we are able to convert π_r into a public-coin protocol $\tilde{\pi}_r$, as in Theorem 4.3.1, we can let the protocol $\tilde{\pi}$ pick a random r and then run $\tilde{\pi}_r$. As the information cost of the resulting protocol $\tilde{\pi}$ again equals the average information cost of $\tilde{\pi}_r$, the inequality (4.1) follows from similar inequalities for π_r and $\tilde{\pi}_r$. For this reason, the theorems below will be proven for private-coin — rather than mixed-coin — protocols.

The $O(q \log(2n\ell))$-term of (4.1) suggests that we have some loss of information on each round. Indeed, Theorem 4.3.1 will be derived from its one-way version.

4.3.1 Reverse Newman's Theorem for one-way protocols

4.3.3. Theorem (R.N.T. for one-way protocols). *For any one-way private-coin ℓ-discrete protocol π there exists a one-way public-coin ℓ-discrete protocol π' such that π and π' generate the same message distributions, and for any input distribution $(X, Y) \sim \mu$, we have*

$$\mathsf{IC}_\mu(\pi') \le \mathsf{IC}_\mu(\pi) + O(\log(2n\ell)),$$

where $n = \log |\mathcal{X}|$.

Proof. We first sketch the proof. The public randomness R' used by the new protocol π' will be the very same randomness R used by π. So we seem to have very little room for changing π, but actually there is one change that we are allowed to make. Let $M_\pi : \mathcal{X} \times \mathcal{R} \mapsto \mathcal{M}$ be the function Alice uses to generate her message. It will be helpful to think of M_π as a table, with rows corresponding to possible inputs x, columns corresponding to possible choices of the private random string r, and the (x, r) entry being the message

$M_\pi(x,r)$. Noticing that r is picked uniformly, Alice might instead send message $M_\pi(x,\phi_x(r))$, where ϕ_x is some permutation of $\mathcal{R}$. In other words, she may permute each row in the table using a permutation ϕ_x for the row x. The permutation ϕ_x will "scramble" the formerly-private now-public randomness R into some new string $\tilde{r} = \phi_x(r)$ about which Bob hopefully knows nothing. This "scrambling" keeps the message distribution exactly as it was, changing only which R results in which message. We will see that this can be done in such a way that, in spite of knowing r, Bob has no hope of knowing $\tilde{r} = \phi_x(r)$, unless he already knows x to begin with.

To understand what permutation ϕ_x we need, we first note the following. Let $M' = M_{\pi'}(X,R)$ denote the message that the protocol π' we have to design sends for input X and public randomness R. Then the information cost of π' is

$$I(M',R : X|Y).$$

The information cost of the original protocol π is

$$I(M : X|Y) = I(M' : X|Y),$$

where the equality holds as the distributions of the triples (M,X,Y) and (M',X,Y) are identical (regardless of the chosen permutations ϕ_x). Thus the difference between the information costs of π' and π equals

$$I(M',R : X|Y) - I(M' : X|Y) = I(R : X|M',Y),$$

which is at most $H(R|M',Y)$. If we permute each row of the table in such a way that every message m appears in at most $d = (n \cdot \ell)^{O(1)}$ columns, then given m we can specify the column (the random-choice R) used to pick m with $O(\log n\ell)$ bits, and hence

$$H(R|M',Y) = O(\log n\ell).$$

Unfortunately, it may happen that there are no such permutations. For instance, this is the case when a row has the same message m in every column.

We will show that if all messages in a row are distinct, then we can "almost" achieve the goal: one can permute each row in such a way that with probability at least $1 - 1/n^2$ the message $M' = M_{\pi'}(X,R)$ appears in at most $d = (n \cdot \ell)^{O(1)}$ columns. Thus we first prove Theorem 4.3.3 for the special case of 1–1 protocols, i.e. for protocols where each row has distinct messages.

The proof of Theorem 4.3.3 for 1–1 protocols. We first will construct a special bipartite graph G, which we call a *matching graph.* Its left nodes will be all possible messages m and its right nodes will be all random strings r. Our strategy will be to find a way of permuting each row of our table so that for every row x and most columns r (in row x) the message $M_{\pi'}(x,r)$ in the cell (x,r) of the table is connected by an edge to r in the graph G.

4.3.4. Definition. An (m,ℓ,d,δ)-matching graph is a bipartite graph $G = (\mathcal{M} \cup \mathcal{R}, \mathcal{E})$ such that $|\mathcal{M}| = 2^m$, $|\mathcal{R}| = 2^\ell$, $\deg(u) = d$ for each $u \in \mathcal{M}$, and such that for all $\mathcal{M}' \subseteq \mathcal{M}$ with $|\mathcal{M}'| = 2^\ell$, $G_{\mathcal{M}' \cup \mathcal{R}}$ has a matching of size at least $2^\ell(1-\delta)$.

To gain some intuition about what is happening, suppose we had the following *fictional object*: an $(m, \ell, n, 0)$-matching graph — i.e., we have a degree-n graph with the property that any left-set of size $|\mathcal{R}|$ will have a perfect matching with $\mathcal{R}$ that uses only edges in the graph. Now let $\mathcal{M}_x = M_\pi(x, \mathcal{R})$ be the set of messages that π can send on input x; then in the new protocol π', $M_{\pi'}(x, r)$ is the message that is matched with r in the perfect matching between $\mathcal{M}_x$ and $\mathcal{R}$ (see Figure 4.3.1). It should be clear that π' gives each message exactly the same probability mass.

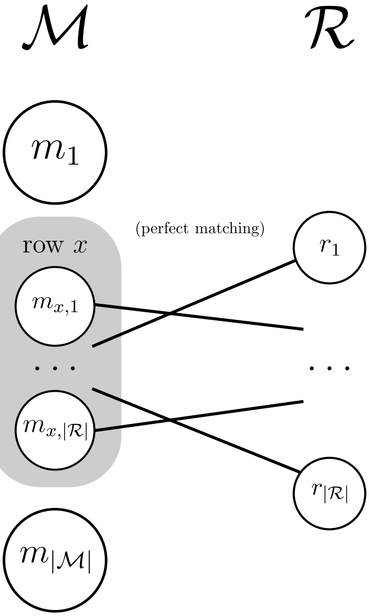

Figure 4.1: An ideal 'matching graph'.

To see that, in this new protocol π', R reveals little information about X when M' is known, notice that if we know the message $m' = M_{\pi'}(x, r)$, then in order to specify r we only need to say which edge in the graph must be followed; this is specified with $\log n$ bits because our graph has degree n. Hence $I(X : R|M) \leq H(R|M) \leq \log n$.

In truth, matching graphs with such good parameters do not exist. But we *can* have good-enough approximations, and we can show that this is enough for our purposes. These graphs are obtained through the Probabilistic Method.

4.3.5. Lemma. *For all integers $\ell \leq m$ and positive δ there is an (m, ℓ, d, δ)-matching graph with $d = O(m/\delta)$.*

In Section 4.6.1 we will show that the lemma holds also $d = O((m - \ell)/\delta^2) + \ln(1/\delta)/\delta$ (Lemma 4.6.1). That bound has better dependence on m, ℓ (especially when $m - \ell \ll m$). However, it has worse dependence on δ. In Section 4.6.2 we show a lower bound of $d = \Omega((m - \ell)/\delta)$, which almost matches our upper bounds.

Proof. Hall's theorem [48] states that if in a bipartite graph *every* left subset of cardinality $i \leq L$ has at least i neighbors then *every* left subset of cardinality $i \leq L$ has a matching in the graph.

Thus it suffices to construct a bipartite graph having this property for $L = (1-\delta)2^{\ell}$. By the union bound, a random graph[3] of degree d fails to have this property with probability at most

$$\sum_{i=1}^{L} 2^{mi} 2^{\ell i} (i/2^{\ell})^{di}.$$

Here 2^{mi} is an upper bound for the number $\binom{2^m}{i}$ of i-element left subsets $\mathcal{M}'$, $2^{\ell i}$ is an upper bound for the number of $i-1$-element right subsets $\mathcal{R}'$, and $(i/2^{\ell})^{di}$ is an upper bound for the probability that all neighbors of $\mathcal{M}'$ fall into $\mathcal{R}'$. For $L \leq (1-\delta)2^{\ell}$ this sum is upper bounded by a geometric series

$$\sum_{i=1}^{L} \left(2^m 2^{\ell} (1-\delta)^d\right)^i.$$

Thus we are done, if the base of this series $2^m 2^{\ell}(1-\delta)^d$ is less than $1/2$, say, which happens for sufficiently large $d = O(m/\delta)$. ■

Now the proof of Theorem 4.3.3 for 1–1 protocols proceeds as follows. Let $n = \log|\mathcal{X}|$ and $\ell = \log|\mathcal{R}|$. Assume without loss of generality that $\mathcal{M} = M(\mathcal{X},\mathcal{R})$; then $|\mathcal{M}| \leq 2^{n+\ell}$. Now let G be an $(n+\ell, \ell, d, \delta)$-matching graph having $\mathcal{M}$ as a subset of its left set and $\mathcal{R}$ as its right set, for $\delta = \frac{1}{n^2}$. For these parameters, we are assured by Lemma 4.3.5 that such a matching graph exists having left-degree $d = O((n+\ell)n^2)$.

We construct the new protocol π' as follows. For each $x \in \mathcal{X}$ let $\mathcal{M}_x = M(x,\mathcal{R})$ be the set of messages that might be sent on input x. Noticing that $|\mathcal{M}_x| = 2^{\ell}$, consider a partial G-matching between $\mathcal{M}_x$ and $\mathcal{R}$ pairing all but a δ-fraction of $\mathcal{M}_x$; then define a bijection $M'_x : \mathcal{R} \to \mathcal{M}_x$ by setting $M'_x(r) = m$ if (m,r) is an edge in the matching, and pairing the unmatched m and r's arbitrarily (possibly using edges not in G). Finally, set $M'(x,r) = M'_x(r)$.

Since $M'(x,r) = M'_x(r)$ for some bijection M'_x between $\mathcal{R}$ and $\mathcal{M}_x$, it is clear that M and M' generate the same transcript distribution for any input x.

Now we prove that M' does not reveal much more information than M. We have seen that the difference between the information costs of π' and π is at most $H(R|M',Y)$. Thus it suffices to show that $H(R|M',Y)$ is at most the logarithm of the left degree of the matching graph plus a constant. As $H(R|M',Y)$ is the average of $H(R|M',Y=y)$ over all choices of y, it suffices to show that

$$H(R|M',Y=y) \leq \log d + 3$$

for every y. While proving this inequality, we will drop the condition $Y=y$ to simplify notation.

[3] For each left vertex, we pick each of the d neighbours independently and uniformly from the right-set.

Let us introduce a new random variable K, which is a function of X, R, M' and takes the value 1 if (M', R) is an edge of the matching graph and is equal to 0 otherwise. Recall that for every x the pair $(M'(x,R), R)$ is an edge of the matching graph with probability at least $1 - 1/n^2$. Therefore, $K = 0$ with probability at most $1/n^2$. Call a message m *bad* if the probability that $K = 0$ conditional to $M' = m$ (that is, the fraction of rows x, among all rows containing m, such that m was not matched within the graph in the row x) is more than $1/n$. Then M' is bad with probability less than $1/n$, otherwise $K = 0$ would happen with probability greater than $1/n^2$.

The conditional entropy $H(R|M')$ is the average of

$$H(R|M' = m)$$

for m chosen according to the distribution of M'. Notice that $H(R|M' = m)$ is at most the log-cardinality of $\mathcal{X}$, because in 1–1 protocols R is a function of the pair (M', X). Thus $H(R|M' = m) \leq n$ for all m, and hence the total contribution of all bad m's in $H(R|M')$ is at most 1. Thus it suffices to show that for all good m,

$$H(R|M' = m) \leq \log d + 2.$$

To this end notice that

$$H(R|M' = m) \leq H(K|M' = m) + H(R|K, M' = m) \leq 1 + H(R|K, M' = m).$$

Thus it is enough to prove that $H(R|K, M' = m) \leq \log d + 1$ for all good m. Again, $H(R|K, M' = m)$ can be represented as the weighted sum of two terms,

$$H(R|K = 1, M' = m) \text{ and } H(R|K = 0, M' = m).$$

The former term is at most $\log d$, because when $K = 1$ and $M' = m$ we can specify R by the number of the edge (m, R) in the matching graph. The latter term is at most n, but its weight is at most $1/n$, since m is good. This completes the proof of Theorem 4.3.3 for 1-1 protocols.

The proof of Theorem 4.3.3 in general case. The general case follows naturally from the 1–1-case and the following lemma, which makes a protocol 1–1 by adding a small amount of communication.

4.3.6. Lemma (A 1–1 conversion which reveals little information). *Given a one-round ℓ-discrete private-coin protocol π, there is a one-round 1–1 ℓ-discrete private-coin protocol π' whose message is of the form*[4]

$$M_{\pi'}(x,r) = (M_\pi(x,r), J(x,r)),$$

for some function J, and such that

$$\mathsf{IC}_\mu(\pi') \leq \mathsf{IC}_\mu(\pi) + \log \ell + 1.$$

[4] On any input x and any choice of randomness r, $M_{\pi'}(x,r)$ is obtained by taking $M_\pi(x,r)$ and adding some additional communication $J(x,r)$.

Proof. We think of $M(\cdot,\cdot)$ as a table, where the inputs $x \in \mathcal{X}$ are the rows and the random choices $r \in \mathcal{R}$ are the columns, and fix some ordering $r_1 < r_2 < \dots$ of $\mathcal{R}$. The second part $J(x, r)$ of $M_{\pi'}$ will be the ordinal number of the message $M(x, r)$ inside the row x i.e.,

$$J(x,r) = |\{r' \leq r | M(x,r') = M(x,r)\}|.$$

This ensures that $M_{\pi'}$ is 1–1.

The difference between the information costs of π' and π is

$$I(M, J : X|Y) - I(M : X|Y) = I(J : X|Y, M).$$

Thus, it suffices to show that for every particular y, m we have[5]

$$I(J : X|Y = y, M = m) \leq \log \ell + 1. \tag{4.2}$$

Fix any y and m, and drop the conditions $Y = y, M = m$ to simplify the notation. By definition, $I(J : X) = H(J) - H(J|X)$. For any fixed x the random variable J has the uniform distribution over the set $\{1, 2, \dots, W_x\}$, where W_x stands for the number of occurrences of the message m in row x of the table.

Let us partition the x's into ℓ classes so that if x is in the ith class then $2^{i-1} \leq W_x < 2^i$. Let $Z = Z_{y,m}$ be the class to which X belongs. The entropy of Z is at most $\log \ell$ and hence we have

$$I(J : X) \leq I(J : X|Z) + H(Z) \leq I(J : X|Z) + \log \ell.$$

Thus it suffices to show that for every i we have

$$I(J : X|Z = i) \leq 1.$$

Notice that

$$H(J|Z = i) \leq i,$$

as for all x in the ith class we have $W_x \leq 2^i$. On the other hand,

$$H(J|X, Z = i) \geq i - 1,$$

as for every x in the ith class we have $W_x \geq 2^{i-1}$ and the distribution of J conditional to $X = x, Y = y, M = m, Z = i$ is uniform. Thus

$$I(J : X|Z = i) = H(J|Z = i) - H(J|X, Z = i) \leq i - (i - 1) = 1.$$

■

Now we are able to finish the proof of Theorem 4.3.3 in the general case. Suppose π is a given one-way private-coin ℓ-discrete protocol. Let π_2 be the 1–1 protocol guaranteed by Lemma 4.3.6, and let π_3 be the protocol constructed from π_2 in the proof of Theorem 4.3.3 for the 1–1 case. Note that π_3's message

[5] In Section 4.6.3 we will prove a corresponding lower bound, implying that this upper-bound is tight up to a constant term.

is of the form $M_{\pi_3}(X,R) = (M_\pi(X,R), J(X,R))$, since it is equidistributed with M_{π_2}. Furthermore, we have

$$\mathsf{IC}_\mu(\pi_3) \leq \mathsf{IC}_\mu(\pi) + O(\log 2n\ell).$$

Now, create a protocol π_4, which is identical to π_3, except that Alice omits $J(X,R)$. Since for each x the message $M_{\pi_4}(x,r)$ sent by π_4 equals $M(x,\phi_x(r))$ for some permutation ϕ_x of $\mathcal{R}$, it is clear that M and M' generate the same transcript distribution for any input x. And

$$\mathsf{IC}_\mu(\pi_4) \leq \mathsf{IC}_\mu(\pi_3) \leq \mathsf{IC}_\mu(\pi) + O(\log 2n\ell)\,.$$

This completes the proof of Theorem 4.3.3. ■

4.3.2 R.N.T. for many-round protocols

Let us derive Theorem 4.3.1 from Theorem 4.3.3.

Proof of Theorem 4.3.1. Let c be the constant hidden in the O-notation in Theorem 4.3.3 so that every one-round private-coin ℓ-discrete protocol π with $|\mathcal{X}|, |\mathcal{Y}| \leq 2^n$ can be converted into a one-round public-coin protocol π' generating the same distribution on transcripts with

$$\mathsf{IC}(\pi') \leq \mathsf{IC}(\pi) + c\log 2n\ell.$$

We are given a q-round private-coin protocol ρ and will simulate it by a public-coin protocol ρ' with

$$\mathsf{IC}(\rho') \leq \mathsf{IC}(\rho) + 2qc\log 2n\ell.$$

The transformation of ρ into ρ' is as one can expect: in each node v of the protocol tree ρ we use a permutation of messages that depends on the input of the player communicating in that node. More specifically, let $m_{<j}$ denote the concatenation of messages sent by ρ' up to round j. In the jth round of ρ' we apply the protocol $\rho'_{m_{<j}}$, which is obtained by the transformation of Theorem 4.3.3 from the 1-round sub-protocol $\rho_{m_{<j}}$ of ρ rooted from the node $m_{<j}$ of the protocol tree of ρ. This change does not affect the probability distribution over messages sent in each node and hence the resulting protocol ρ' generates exactly the same distribution on transcripts. The protocol ρ' uses the same randomness as ρ; however, unlike ρ it uses public and not private randomness.

We have to relate now the information cost of ρ' to that of ρ. To this end we split the information cost of ρ' into the sum of information costs of each round of ρ'. Specifically, by the Chain rule (Fact 4.2.4) the amount of information revealed by ρ' to Bob (say) equals

$$I(X : M_1, R_1, \ldots, M_q, R_q|Y) = \sum_j I(X : M_j, R_j|Y, M_{<j}, R_{<j}).$$

where R_j denotes the randomness used in the jth round of ρ' and $M_j = \rho'_{M_{<j}}(X, R_j)$ denotes the message sent in the jth round of ρ'.

From $I(R_{<j} : M_j, R_j | Y, M_{<j}) = 0$,[6] we conclude from Theorem 4.3.3 — using Facts 4.2.5 and 4.2.6 from the preliminaries — that

$$I(X : M_j, R_j | Y, M_{<j}, R_{<j}) \leq I(X : M_j, R_j | Y, M_{<j}) \leq I(X : M_j | Y, M_{<j}) + c \log 2n\ell,$$

where $I(X : M_j | Y, M_{<j})$ in the right-hand side is the information cost of the jth round of the original protocol ρ. Summing up this inequality over all $j = 1, \dots, q$ and applying the Chain rule to ρ we see that

$$I(X : M_1, R_1, \dots, M_q, R_q | Y) \leq I(X : M_1, \dots, M_q | Y) + qc \log 2n\ell.$$

The similar inequality for the amount of information revealed by ρ and ρ' to Alice is proved analogously. ■

4.4 Compression for public-coin protocols

We present in this section two results of the following general form: we will take a public-coin protocol π that reveals little information, and "compress" it into a protocol ρ that uses little *communication* to perform the same task with about the same error probability. It turns out that the results in this setting are simpler and give stronger compression than in the case where Alice and Bob have private randomness (such as in [23, 29]). We present two bounds, one that is dependent on the number of rounds of π, but which is also round-efficient, in the sense that ρ will not use many more rounds than π; and one that is independent of the number of rounds of π, but where the compression is not as good when the number of rounds of π is small. We begin with the latter.

4.4.1. Theorem. *Suppose there exists a public-coin protocol π to compute $f : \{0,1\}^n \times \{0,1\}^n \to \mathcal{Z}$ over the distribution μ with error probability δ', and let $C = \mathsf{CC}(\pi)$, $I = \mathsf{IC}_\mu(\pi)$. Then for any positive δ there is a public-coin protocol ρ computing f over μ with error $\delta' + \delta$, and with $\mathsf{ACC}_\mu(\rho) = O(I \cdot \log(2Cn/\delta))$.*

Proof. Our compression scheme is similar, but not identical, to that of [23]—the absence of private randomness allows for a more elementary proof.

It suffices to prove the theorem only for deterministic protocols—the case for public-coin protocols can be proved as follows. By fixing any outcome r of randomness R of a public-coin protocol π, we obtain a protocol π_r without public randomness and can apply Theorem 4.4.1 to π_r. The average communication length of the resulting deterministic protocol ρ_r is at most $O(I(\pi_r) \cdot \log(2Cn/\delta))$. Thus the average communication of the public-coin protocol ρ that chooses a random r and runs ρ_r will be at most $O(I \cdot \log(2Cn/\delta))$.

[6] The reader is reminded that we defined protocols so that the message in each round depends only on public randomness, the previous messages, and on a source of private randomness that is independent from the private randomness used in previous rounds. It is easy to see that such an assumption can be made.

Thus we have to show that any deterministic protocol π can be simulated with communication roughly:

$$I(Y : \Pi|X) + I(X : \Pi|Y) = H(\Pi|X) + H(\Pi|Y)$$

(the equality follows because $H(\Pi|X, Y) = 0$, since the transcript Π is a function of X and Y). As we do not relate in this theorem the round complexity of ρ to that of π, we may assume that in the protocol π every message is just a bit (and the turn to communicate does not necessarily alternate). In other words, the protocol tree has binary branching.

Given her input x, Alice knows the distribution of $\Pi|x$, and she can hence compute the conditional probability $\Pr[\pi(X, Y) = t|X = x]$ for each leaf t of the protocol tree. We will use the notation $w_a(t|x)$ for this conditional probability. Likewise Bob computes $w_b(t|y) = \Pr[\pi(X, Y) = t|Y = y]$. Now it must hold that $\pi(x, y)$ is the unique leaf such that both $w_a(t|x), w_b(t|y)$ are positive. Alice and Bob then proceed in stages to find that leaf: at a given stage they have agreed that a certain *partial transcript*, which is a node in the protocol tree of π, is a prefix of $\pi(x, y)$. Then each of them chooses a *candidate transcript*, which is a leaf extending their partial transcript (the candidate transcripts of Alice and Bob may be different). Then they find the largest common prefix (lcp) of their two candidate transcripts, i.e., find the first bit at which their candidate transcripts disagree. Now, because one of the players actually knows what that bit should be (that bit depends either on x or on y), the player who got it wrong can change her/his bit to its correct value, and this will give the new partial transcripts they agree upon. They proceed this way until they both know $\pi(x, y)$.

It will be seen that the candidate leaf can be chosen in such a way that the total probability mass under the nodes they have agreed upon halves at every correction, and this will be enough to show that Alice will only need to correct her candidate transcript $H(\Pi|X)$ times (and Bob $H(\Pi|Y)$ times) on average. Efficient protocols for finding the lcp of two strings will then give us the required bounds.

We first construct an interactive protocol that makes use of a special device, which we call *lcp box*. This is a conceptual interactive device with the following behavior: Alice takes a string u and puts it in the lcp box, Bob takes a string v and puts it in the lcp box, then a button is pressed, and Alice and Bob both learn the largest common prefix of u and v. Using an lcp box will allow us to ignore error events until the very end of the proof, avoiding an annoying technicality that offers no additional insight.

4.4.2. Lemma. *For any given probability distribution μ over input pairs and for every deterministic protocol π with information cost I (w.r.t. μ) and worst case communication C there is a deterministic protocol $\tilde{\rho}$ with zero communication computing the same function with the same error probability (w.r.t. μ) as π, and using an lcp box for C-bitstrings at most I times on average (w.r.t. μ).*

Proof. On inputs x and y, in the new protocol $\tilde{\rho}$ Alice and Bob compute weights $w_a(t|x), w_b(t|y)$ of every leaf of the protocol tree of π, as explained above. Furthermore, for every binary string s let $w_a(s|x)$ denote the sum of weights $w_a(t|x)$ over all leaves t under s. Define $w_b(s|y)$ in a similar way.

The protocol $\tilde{\rho}$ runs in stages: before each stage i Alice and Bob have agreed on a binary string $s = s_{i-1}$, which is a prefix of $\pi(x, y)$. Initially $s = s_0$ is empty.

On stage i Alice defines the candidate transcript t_a as follows: she appends 0 to $s = s_{i-1}$ if $w_a(s0|x) > w_a(s1|x)$ and she appends 1 to s otherwise. Let s' denote the resulting string. Again, she appends 0 to s' if $w_a(s'0|x) > w_a(s'1|x)$ and she appends 1 to s' otherwise. She proceeds in this way until she gets a leaf of the tree (by construction its weight is positive). Bob defines his candidate transcript t_b in a similar way. Then they put t_a and t_b in the lcp box and they learn the largest common prefix s' of t_a and t_b. By construction both $w_a(s'|x)$ and $w_b(s'|y)$ are positive and hence s' is a prefix of $\pi(x, y)$. Recall that no leaf of the protocol tree is a prefix of another leaf. Therefore either $s' = t_a = t_b$, in which case they stop the protocol, as they both know $\pi(x, y)$. Or s' is a proper prefix of both t_a and t_b. If the node s' of the protocol tree belongs to Alice, then Bob's next bit is incorrect, and otherwise Alice's next bit is incorrect. They both add the correct bit to s' and let s_i be the resulting string.

Each time Alice's bit is incorrect $w_a(s|x)$ decreases by a factor of 1/2, and similarly each time Bob's bit is incorrect $w_b(s|y)$ decreases by a factor of 1/2. At the start we have $w_a(s|x) = w_b(s|y) = 1$ and at the end we have $w_a(s|x) = w_a(\pi(x, y)|x)$ and $w_b(s|y) = w_b(\pi(x, y)|y)$. Hence they use the lcp box at most

$$\log_2 1/w_a(\pi(x, y)|x) + \log_2 1/w_b(\pi(x, y)|y)$$

times. By definition of the conditional entropy the average of $\log_2 1/w_a(\pi(X, Y)|X)$ is equal to $H(\Pi|X)$ and the average of $\log_2 1/w_b(\pi(X, Y)|Y)$ equals $H(\Pi|Y)$. Thus Alice and Bob use the lcp box at most I times on average. ■

Now we have to transform the protocol of Lemma 4.4.2 to a randomized public-coin protocol computing f that does not use an lcp box, with additional error δ. The use of an lcp box can be simulated with an error-prone implementation:

4.4.3. Lemma ([46]). *For every positive ε and every natural C there is a randomized public-coin protocol such that on input two C-bit strings x, y, it outputs the largest common prefix of x, y with probability at least $1 - \varepsilon$; its worst-case communication complexity is $O(\log(C/\varepsilon))$.*

The lemma is proven by hashing (as in the randomized protocol for equality) and binary search. From this lemma we obtain the following corollary.

4.4.4. Lemma. *Let $\tilde{\rho}$ be a protocol that computes $f : \{0,1\}^n \times \{0,1\}^n \to \mathcal{Z}$, while using an lcp box $\ell \leq 2n$ times on average for strings of length at most C. Then $\tilde{\rho}$ can be simulated with error δ by a protocol ρ that does not use an lcp box, and communicates $O(\ell \log(\frac{2Cn}{\delta}))$ bits more on average.*

Proof. The protocol ρ simulates $\tilde{\rho}$ by replacing each use of the lcp box with the protocol given by Lemma 4.4.3 with some error parameter ε (to be specified later). The simulation continues while the total communication is less than n.

Once it becomes n, we stop the simulation and Alice just sends her input to Bob.

Notice that the additional error probability introduced by the failure of the protocol of Lemma 4.4.3 is at most $\varepsilon\ell$: for each input pair (x, y) the error probability is at most $\varepsilon i(x, y)$, where $i(x, y)$ stands for the number of times we invoke lcp box for that particular pair, and the average of $\varepsilon i(x, y)$ over (x, y) equals $\varepsilon\ell$. Thus if we take $\varepsilon \leq \delta/\ell$, the error probability introduced by failures of the lcp box is it most δ.

Each call to the lcp box costs $O(\log(C/\varepsilon)$. Thus the communication of ρ is at most

$$O(\ell \log(C/\varepsilon)) + (\ell\varepsilon)(2n)$$

more on average than that of $\tilde{\rho}$. Here the first term is an upper bound for the average communication over all triples (x, y, randomness for the lcp box) such that no lcp failure occurs and the second term accounts for the average communication over all remaining triples.

Let $\varepsilon = \delta/2n$ (which is less than δ/ℓ, as we assume that $\ell \leq 2n$) so that the average communication is at most $O(\ell \log(\frac{2Cn}{\delta}) + \ell\delta) = O(\ell \log(\frac{2Cn}{\delta}))$. ■

We are now able to finish the proof of the theorem. Notice that the information cost of the initial protocol is at most $2n$. Hence we can apply Lemma 4.4.4 for $\ell = I$ to the protocol of Lemma 4.4.2. The average communication of the resulting protocol ρ is at most $O(I \cdot \log(2Cn/\delta))$. ■

The proof of Theorem 4.4.1 offers no guarantee on the number of rounds of the compressed protocol ρ. It is possible to compress a public-coin protocol on a round-by-round basis while preserving, up to a multiplicative constant, the total number of rounds used.

4.4.5. Theorem. *Suppose there exists a public-coin protocol π to compute $f : \{0,1\}^n \times \{0,1\}^n \to \mathcal{Z}$ over input distribution μ with error probability δ', and let $I = \mathsf{IC}_\mu(\pi)$ and $q = \mathsf{RC}(\pi)$. Then there exists a public-coin protocol ρ that computes f over μ with error $\delta' + \delta$, and with $\mathsf{ACC}_\mu(\rho) = O(I+1) + q\log(nq/\delta)$ and $\mathsf{ARC}_\mu(\rho) = O(q)$.*

Proof. Again it suffices to prove the theorem for deterministic protocols π. The idea of the proof is to show the result one round at a time. In round i, Alice, say, must send a certain message m_i to Bob. From Bob's point of view, this message is drawn according to the random variable $M_i = M_i(\tilde{X}, y, m_1, \dots, m_{i-1})$ where $\tilde{X}$ is Alice's input conditioned on Bob's input being y and on the messages $m_1, \dots, m_{i-1}$ that were previously exchanged. We will show that there is a sub-protocol σ_i that can simulate round i with small error by using constantly-many rounds and with

$$O(H(M_i | y, m_1, \dots, m_{i-1})) = O(I(X : M_i | y, m_1, \dots, m_{i-1}))$$

bits of communication *on average*. Then putting these sub-protocols together, and truncating the resulting protocol whenever the communication is excessive, we obtain the protocol ρ which simulates π.

The procedure to compress each round is achieved through an interactive variant of the Slepian-Wolf theorem ([92, 89, 35]). We could not apply the known theorems directly, however, since they were made to work in different settings.

In a similar fashion to the proof of Theorem 4.4.1, we will make use of a special interactive device, which we call a *transmission μ-box*, where μ is a probability distribution over input pairs (X, Y). Its behavior is as follows: one player takes a string x and puts it in the transmission box, the other player takes a string y and puts it in the box, a button is pressed, and then the second player knows x. The usage of a transmission μ-box is charged in such a way that the average cost when the input pair (X, Y) is drawn at random with respect to μ is $O(H(X|Y) + 1)$ bits of communication and $O(1)$ rounds.

4.4.6. Lemma. *Let π be any deterministic q-round protocol, and let μ be the distribution of the inputs (X, Y). Then there exists a deterministic protocol $\tilde{\rho}$ that makes use of the transmission box (each time for a different distribution) to achieve the following properties.*

1. *The average communication of $\tilde{\rho}$ is $\mathsf{ACC}_\mu(\tilde{\rho}) = O(\mathsf{IC}_\mu(\pi) + q)$;*
2. *The average number of rounds of $\tilde{\rho}$ is $\mathsf{ARC}_\mu(\tilde{\rho}) = O(q)$;*
3. *$\tilde{\rho}$ uses a transmission box q times; and*
4. *After $\tilde{\rho}$ is run on the inputs x, y, both players know $\pi(x, y)$.*

Proof. Let $\pi_{<j}(x, y)$ denote the sequence of messages sent by π in the first $j - 1$ rounds for inputs x, y. The protocol $\tilde{\rho}$ simulates π on a round-per-round basis.

Assume that in the new protocol $j - 1$ rounds were played. Let $m_{<j}$ denote the sequence of $j - 1$ messages sent earlier and let x, y stand for inputs. Assume further that in jth round of π Alice has to communicate. Her message is a function M of the sequence $m_{<j}$ and her input x. Let ν denote the probability distribution on pairs (m, y) where

$$\nu(m, y) = \Pr[M(X, m_{<j}) = m,\ Y = y | \pi_{<j}(X, Y) = m_{<j}].$$

In round j of protocol $\tilde{\rho}$, Alice puts the string $M(x, m_{<j})$ into the transmission ν-box and Bob puts there his input y and they press the button. If it is Bob's turn to communicate, then they reverse their positions.

Items 2, 3 and 4 from the statement of the Lemma follow from construction of $\tilde{\rho}$ and from the description of the transmission box. It remains to bound the average communication length of $\tilde{\rho}$. Again by assumption on the transmission box, the average communication in round j is at most $O(I_j + 1)$ where

$$I_j = H(M(X, \pi_{<j}(X, Y))|Y, \pi_{<j}(X, Y)),$$

if it is Alice's turn to communicate and

$$I_j = H(M(Y, \pi_{<j}(X, Y))|X, \pi_{<j}(X, Y)),$$

otherwise. From the chain rule (Fact 4.2.4) it follows that the sum of I_j over all j of the first type is equal to $I(\Pi : X|Y)$, while the sum of I_j over all j of the second type is equal to $I(\Pi : Y|X)$. ■

To proceed we need a protocol simulating the transmission box.

4.4.7. Lemma (Constant-round average-case one-shot Slepian–Wolf). *Let μ be the distribution of the inputs (X, Y). For every positive ε there is a public-coin communication protocol with the following properties:*

1. *For all x, y, after execution of the protocol Bob learns x with probability at least $1-\varepsilon$.*

2. *When (X, Y) are drawn according to μ, the protocol communicates an*

$$O(H(X|Y) + 1) + \log(1/\varepsilon)$$

average number of bits in $O(1)$ average number of rounds.

Contrast this to the classical Slepian–Wolf theorem, where Alice and Bob are given a stream of i.i.d. pairs $(X_1, Y_1), \ldots, (X_n, Y_n)$, and Alice gets to transmit $X_1, \ldots, X_n$ by using only one-way communication, and with an *amortized* communication of $H(X|Y)$.

Proof. Let y be Bob's given input. For a given x in the support of X, let $p(x) = \Pr[X = x|Y = y]$, and for a given subset $\mathcal{X}$ of the same support, let $p(\mathcal{X}) = \Pr[X \in \mathcal{X}|Y = y]$. Then Bob begins by arranging the x's in the support of X by decreasing order of the probability $p(x)$. He then defines the two sets

$$\mathcal{X}_1 = \{x_1, \ldots, x_{i(1)}\}, \qquad \mathcal{Z}_1 = \mathcal{X}_1,$$

where $i(1)$ is the minimal index which makes $p(\mathcal{X}_1) \geq 1/2$. Inductively, while $Z_k \neq X$, he then defines:

$$\mathcal{X}_{k+1} = \{x_{i(k)+1}, \cdots, x_{i(k+1)}\}, \qquad \mathcal{Z}_{k+1} = \mathcal{Z}_k \cup \mathcal{X}_{k+1},$$

where $i(k+1) > i(k)$ is the minimal index which makes $p(\mathcal{X}_{k+1}) \geq \frac{1-p(\mathcal{Z}_k)}{2}$. In other words, $\mathcal{X}_{k+1}$ is the smallest set which takes the remaining highest-probability x's so that they total at least half of the remaining probability mass.

Because at least one new x_i is added at every step, this inductive procedure gives Bob a finite number of sets $\mathcal{Z}_1, \ldots, \mathcal{Z}_K = X$. Then the protocol consists of applying the protocol of the following lemma, which will be proved later.

4.4.8. Lemma. *For every natural number m and every positive ε there exists a randomized public-coin protocol with the following behavior. Suppose that Bob is given a family of finite sets $\mathcal{Z}_1 \subseteq \cdots \subseteq \mathcal{Z}_K \subset \{0,1\}^m$ and Alice is given a string $z \in \mathcal{Z}_K$. Then the protocol transmits z to Bob, except with a failure probability of at most ε. For k the smallest index for which $z \in \mathcal{Z}_k$, the run of this protocol uses at most $2k+1$ rounds and $2\log|\mathcal{Z}_k| + \log\frac{1}{\varepsilon} + 4k$ bits of communication.*

Now let us bound the average number of rounds and communication complexity. First notice that $p(\mathcal{X}_k) \leq 2^{1-k}$, and hence, taking the average over Alice's inputs, we find that

$$\sum_{k=1}^{K} p(\mathcal{X}_k) 4k = O(1)$$

must upper-bound the average number of rounds, as well as the contribution of the $4k$ term to the average communication. To upper-bound the contribution of the $2 \log |\mathcal{Z}_k|$ term, we first settle that:

(i) $p(\mathcal{X}_k) \leq 2p(\mathcal{X}_{k+1}) + 2p(x_{i(k)})$, which can be seen by summing two inequalities that follow from the minimality of $i(k)$ in the definition of $\mathcal{X}_k$:

$$p(\mathcal{X}_k) - p(x_{i(k)}) \leq \frac{1 - p(\mathcal{Z}_{k-1})}{2}, \qquad \frac{1 - p(\mathcal{Z}_k)}{2} \leq p(\mathcal{X}_{k+1}),$$

after which we get

$$\frac{p(\mathcal{X}_k)}{2} - p(x_{i(k)}) \leq p(\mathcal{X}_{k+1}).$$

(ii) $|\mathcal{Z}_k| \leq \frac{1}{p(x)}$ for any $x \in \mathcal{X}_{k+1} \cup \{x_{i(k)}\}$, which follows since every $x' \in \mathcal{Z}_k$ has a higher-or-equal probability than the x's in $\mathcal{X}_{k+1} \cup \{x_{i(k)}\}$, but the sum of all the $p(x')$ still adds up to less than 1.

Now we are ready to bound the remaining term in the average communication:

$$\sum_{k=1}^{K} p(\mathcal{X}_k) \log |\mathcal{Z}_k| \leq 2 \sum_{k=1}^{K-1} p(\mathcal{X}_{k+1}) \log |\mathcal{Z}_k| + p(\mathcal{X}_K) \log |\mathcal{Z}_K| + 2 \sum_{k=1}^{K} p(x_{i(k)}) \log |\mathcal{Z}_k|$$

$$\leq 5 \sum_x p(x) \log \frac{1}{p(x)} = O(H(X|Y=y));$$

above, the first inequality follows from (i), and the second from (ii). ■

Proof of Lemma 4.4.8. The protocol is divided into stages and works as follows. On the first stage, Bob begins by sending the number $\ell_1 = \log |\mathcal{Z}_1|$ in unary to Alice, and Alice responds by picking $L_1 = \ell_1 + \log \frac{1}{\varepsilon} + 1$ random linear functions $f_1^{(1)}, \ldots, f_{L_1}^{(1)} : \mathbb{Z}_2^n \to \mathbb{Z}_2$ using public randomness, and sending Bob the hash values $f_1^{(1)}(z), \ldots, f_{L_1}^{(1)}(z)$. Bob then looks for a string $z' \in \mathcal{Z}_1$ that has the same hash values he just received; if there is such a string, then Bob says so, and the protocol is finished with Bob assuming that $z' = z$.

Otherwise, the protocol continues. At stage k, Bob computes the number $\ell_k = \log |\mathcal{Z}_k|$, and sends the number $\ell_k - \ell_{k-1}$ in unary to Alice; Alice responds by picking $L_k = \ell_k - \ell_{k-1} + 1$ random linear functions $f_1^{(k)}, \ldots, f_{L_k}^{(k)}$, whose evaluation on z she sends over to Bob. Bob then looks for a string $z' \in \mathcal{Z}_k$ that has the same hash values for all the hash functions which were picked in this and previous stages; if there is such a string, then Bob says so, and the protocol is finished with Bob assuming that $z' = z$. If the protocol has not halted in K rounds, Alice just sends her input to Bob.

An error will occur whenever a $z' \neq z$ is found that has the same fingerprint as z. The probability that this happens at stage k for a specific $z' \in \mathcal{Z}_k$ is 2^{-L}, where $L = \ell_k + k + \log \frac{1}{\varepsilon}$ is the total number of hash functions picked up to this stage. By a union bound, the probability that such a z' exists is at most $|\mathcal{Z}_k| 2^{-\ell_k} \frac{\varepsilon}{2^k} \leq \frac{\varepsilon}{2^k}$. Again by a union bound, summing over all stages k we get a total error probability of ε.

To bound the communication for $z \in \mathcal{Z}_k$, notice that sending all $\ell_1 \ldots, \ell_k$ costs Bob at most $\log |\mathcal{Z}_k| + k$ bits of total communication[7], that the total number of hash values sent by Alice is at most $\log |\mathcal{Z}_k| + 2k + \log \frac{1}{\varepsilon}$, and that Bob's reply (saying whether the protocol should continue) costs him k bits. ■

From Lemma 4.4.7 we get an analogue of Lemma 4.4.4.

4.4.9. Lemma. *Let $\tilde{\rho}$ be a protocol to compute $f : \{0,1\}^n \times \{0,1\}^n \to \mathcal{Z}$ that uses transmission boxes q times. Then, for any positive δ, $\tilde{\rho}$ can be simulated with error δ by a protocol ρ that does not use transmission boxes, and communicates $q \log(\frac{qn}{\delta}) + 1$ bits more than $\tilde{\rho}$.*

Proof. The protocol ρ simulates $\tilde{\rho}$ by replacing each use of a transmission box with the protocol given by Lemma 4.4.7 with some error parameter ε (to be specified later). The simulation continues while the total communication is less than n. Once it becomes n, we stop the simulation and Alice just sends her input to Bob.

The additional error probability introduced by the failure of the protocol of Lemma 4.4.7 is at most $q\varepsilon$. Assuming that $\varepsilon \leq \delta/q$, the error probability introduced by a transmission box failure is it most δ.

Each call of a transmission box costs $\log(1/\varepsilon)$ bits of communication more than we have charged the protocol $\tilde{\rho}$. Thus the communication of ρ is at most

$$q \log(1/\varepsilon) + (q\varepsilon)(2n)$$

longer than that of $\tilde{\rho}$. Let $\varepsilon = \delta/qn$ so that the communication of ρ be at most

$$q \log(qn/\delta) + \delta/2 \leq q \log(qn/\delta) + 1$$

more than that of $\tilde{\rho}$. ■

We are able now to finish the proof of the theorem. Applying Lemma 4.4.9 to the protocol of Lemma 4.4.6 we get the desired protocol. ■

4.5 Applications

From the combination of Theorems 4.3.1 and 4.4.5, and Observation 4.2.13, we can obtain a new compression result for general protocols.

[7] We have added 1 bit per message because, sending ℓ_i ones to Alice, Bob should append a zero to them — recall that the messages must form a prefix-free set.

4.5.1. Corollary. *Suppose there exists a mixed-coin, q-round protocol π to compute f over the input distribution μ with error probability ε, and let $C = \mathsf{CC}(\pi)$, $I = \mathsf{IC}_\mu(\pi)$, $n = \log|\mathcal{X}| + \log|\mathcal{Y}|$. Then there exists a public-coin, $O(q)$-average-round protocol ρ that computes f over μ with error $\varepsilon + \delta$, and with*

$$\mathsf{CC}(\rho) \leq O\left(I + q\log\left(\frac{qnC}{\delta}\right)\right). \tag{4.3}$$

As we will see in the following sub-section, this will result in a new direct sum theorem for bounded-round protocols. In general, given that we have already proven Theorem 4.4.1, and given that this approach shows promise in the bounded-round case, it becomes worthwhile to investigate whether we can prove Conjecture 4.3.2 with similar techniques.

4.5.1 Direct-sum theorems for the bounded-round case

The following theorem was proven in [23]:

4.5.2. Theorem. **([23], Theorem 12.)** *Suppose that there is a q-round protocol π^k that computes k copies of f with communication complexity C and error ε, over the k-fold distribution μ^k. Then there exists a q-round mixed-coin protocol π that computes a single copy of f with communication complexity C and the same error probability ε, but with information cost $\mathsf{IC}_\mu(\pi) \leq \frac{2C}{k}$ for any input distribution μ.*

As a consequence of this theorem, and of Corollary 4.5.1, we will be able to prove a direct sum theorem. The proof is a simple application of Theorem 4.5.2, and Corollary 4.5.1.

4.5.3. Theorem (Direct sum theorem for the bounded-round case). *There is some constant d such that, for any input distribution μ and any $0 < \varepsilon < \delta < 1$, if f requires, on average, at least*

$$C + q\log\left(\frac{qnC}{\delta - \varepsilon}\right)$$

bits of communication, to be computed over μ with error δ in dq (average) rounds, then $f^{\otimes k}$ requires at least kC bits of communication, in the worst case, to be computed over $\mu^{\otimes k}$ with error ε in q rounds.

4.5.2 Comparison with previous results

We may compare Corollary 4.5.1 with the results of [29]. In that paper, the nC factor is missing inside the log of equation (4.3), but the number of rounds of the compressed protocol is $O(q\log I)$ instead of $O(q)$. A similar difference appears in the resulting direct-sum theorems.

We remark that the compression of Jain et al. [60] is also achieved with a round-by-round proof. Our direct-sum theorem is incomparable with their more ambitious direct-product result. It is no surprise, then, that the communication complexity of their compression scheme is $O(\frac{qI}{\delta})$, i.e., it incurs a factor of q, whereas we pay only an additive term of $\tilde{O}(q)$. However, their direct-product result also preserves the number of rounds in the protocol, whereas in our result the number of rounds is only preserved within a constant factor.

4.6 Alternative constructions and matching lower bounds

4.6.1 A different upper bound on the degree of matching graphs

4.6.1. Lemma. *For every integer $\ell \le m$ and real $\delta > 0$ there is an (m, ℓ, d, δ)-matching graph with $d = (2 + (m - \ell)\ln 2)/\delta^2 + \ln(1/\delta)/\delta$.*

Proof. We show the existence of such a graph using a probabilistic argument. Let A and B be any sets of $M = 2^m$ left and $L = 2^\ell$ right nodes, respectively. Construct a random graph G by choosing d random neighbors independently for each $u \in A$. Different neighbors of the same node u are also chosen independently, thus they might coincide. For any $A' \subseteq A$ of size L, let $E_{A'}$ be the event that $G_{A' \cup B}$ does *not* have a matching of size $L(1-\delta)$, and let $\textsc{Bad} := \bigvee_{A'} E_{A'}$. Note that the lemma holds if $\Pr[\textsc{Bad}] < 1$.

Next, we bound $\Pr[E_{A'}]$. Let $A' = \{u_1, \ldots, u_L\}$ be any set of L left nodes. Let $\mathcal{N}(u)$ denote the neighborhood of a vertex u. Consider the following procedure for generating a matching for $G_{A' \cup B}$:

```
Find-Matching
1  Matching ← $\emptyset$
2  $V \leftarrow \emptyset$
3  for $i \leftarrow 1$ to $L$
4        if $\mathcal{N}(u_i) \not\subseteq V$
5              pick arbitrary $v_i \in \mathcal{N}(u_i) \setminus V$
6              Matching ← Matching $\cup \{(u_i, v_i)\}$
7              $V \leftarrow V \cup \{v_i\}$
8  return Matching
```

Define the indicator variables $X_1, \ldots, X_L$ as follows: $X_i = 1$ if the condition in the 4th line of Find-Matching is true and 0 otherwise. From the definition of these variables it follows that for all i and all $b = (b_1, \ldots, b_i) \in \{0, 1\}^i$ the conditional probability of $X_{i+1} = 0$ given $X_1 = b_1, \ldots, X_i = b_i$ is equal to

$$(|b|/L)^d,$$

where $|b|$ stands for Hamming weight of vector b, i.e. the number of 1s in $b = (b_1, \ldots, b_i)$. Consider also similar random variables $Y_1, \ldots, Y_L$ where the distribution of $Y_1, \ldots, Y_L$ is defined by the formula

$$\Pr[Y_{i+1} = 0 | Y_1 = b_1, \ldots, Y_i = b_i] = \begin{cases} (|b|/L)^d, & \text{if } |b| < (1 - \delta)L, \\ 1, & \text{if } |b| \ge (1 - \delta)L. \end{cases}$$

In terms of $X_1, \ldots, X_L$ the event $E_{A'}$ happens if and only if $X_1 + \cdots + X_L < (1-\delta)L$. For every string b of Hamming weight less than $(1-\delta)L$ the probabilities $\Pr[X = b]$ and $\Pr[Y = b]$ coincide. Thus it suffices to upper bound the probability $\Pr[Y_1 + \cdots + Y_L < (1-\delta)L]$. To this end consider independent random variables $Z_1, \ldots, Z_L \in \{0,1\}$, where the probability of $Z_i = 1$ is $(1-\delta)^d$.

4.6.2. Claim. $\Pr[|Y| < (1-\delta)L] \leq \Pr[|Z| < (1-\delta)L]$.

Proof. We prove this using the coupling method. We claim that there is a joint distribution of Y and Z such that the marginal distributions are as defined above, and with probability 1 it holds that $Z_i \leq Y_i$ for all i. This joint distribution is defined by the following process: we pick L independent reals $r_1, \ldots, r_L \in [0;1]$ and let

$$Z_i = \begin{cases} 0, & \text{if } r_i < (1-\delta)^d; \\ 1, & \text{otherwise.} \end{cases}$$

$$Y_i = \begin{cases} 0, & \text{if } r_i < \left(\frac{Y_1+\cdots+Y_{i-1}}{L}\right)^d \text{ and } \frac{Y_1+\cdots+Y_{i-1}}{L} < 1-\delta; \\ 1, & \text{otherwise.} \end{cases}$$

We claim that the inequality $Z_i \leq Y_i$ (holding with probability 1) implies that for every downward closed set $E \subset \{0,1\}$ it holds $\Pr[Y \in E] \leq \Pr[Z \in E]$ (we call a set E downward closed if $b \in E$ and $b' \leq b$, component-wise, implies $b' \in E$). Indeed,

$$\Pr[Y \in E] \leq \Pr[Y \in E, Z \in E] \leq \Pr[Z \in E],$$

where the first inequality holds, since E is downward closed and thus $Y \in E$ implies $Z \in E$. The set of Boolean vectors $b \in \{0,1\}^L$ of Hamming weight less than $(1-\delta)L$ is downward closed hence the statement. ■

By this lemma it suffices to upper bound the probability

$$\Pr[Z_1 + \cdots + Z_L < (1-\delta)L],$$

which can be obtained by Chernoff bound.

Let $S := \sum(1 - Z_i)$, and let $\mu := \mathbb{E}[S]$, $p = (1-\delta)^d$. Note that $\mu = pL$. Also, let $\psi := \delta/p - 1$. Using the multiplicative version of the Chernoff bound, so long as $\psi > 0$, we have

$$\begin{aligned}
\Pr[S > \delta L] &= \Pr[S > pL \cdot (\delta/p)] \\
&= \Pr[S > \mu(1+\psi)] \\
&< \left(\frac{e^{\psi}}{(1+\psi)^{(1+\psi)}}\right)^{\mu} \\
&= \exp\left(\mu\left(\frac{\delta}{p} - 1 - \frac{\delta}{p}\ln(\frac{\delta}{p})\right)\right) \\
&< \exp\left(\mu\left(\frac{\delta}{p} - \frac{\delta}{p}\ln(\frac{\delta}{p})\right)\right) \\
&= \exp\left(pL\frac{\delta}{p}\left(1 - \ln\delta + \ln p\right)\right) \\
&= \exp\left(\delta L + \delta L\ln(1/\delta) + \delta L \ln p\right) \\
&= \exp\left(\delta L\left(1 + \ln(1/\delta) + \ln p\right)\right) .
\end{aligned}$$

Thus for every set A' of L left nodes we have $\Pr[E_{A'}] < e^{\delta L(1+\ln(1/\delta)+\ln p)}$. There are $\binom{M}{L}$ subsets of A of size L. By Stirling's Formula, we have

$$\binom{M}{L} \leq \frac{(M)^L}{L!} \leq \left(\frac{Me}{L}\right)^L = \exp(L(1 + \ln M/L)) .$$

By union bound we have

$$\begin{aligned}
\Pr[BAD] &\leq \exp\left(M(1+\ln M/L)\right) \cdot \exp\left(\delta M(1+\ln(1/\delta)+\ln p)\right) \\
&= \exp\left(M + M\ln M/L + \delta M + \delta M\ln(1/\delta) + \delta M \ln p\right) \\
&< \exp\left(M + M\ln M/L + \delta M + \delta M\ln(1/\delta) - d\delta^2 M\right) \\
&< 1 ,
\end{aligned}$$

where the final inequality uses $d = (2 + \ln M/L)/\delta^2 + \ln(1/\delta)/\delta$, which also ensures that $\psi > 0$ whenever δ is sufficiently small. ■

4.6.2 A lower bound on the degree of matching graphs

4.6.3. Lemma. *An (m, ℓ, d, δ)-matching graph must have*

$$d = \Omega\left(\min\left(\frac{m-\ell}{\delta}, \delta 2^{\ell}\right)\right) .$$

Proof. We will prove that in such a bipartite graph there must exist a left-set A of size $2^m(1-4\delta)^d$ whose neighbours are contained in a right-set B of size $(1-2\delta)2^{\ell}$. If the graph is a matching graph with said parameters, it must then follow that $|A| \leq 2^{\ell}$, hence $d \geq (m-\ell)/\log(1-4\delta) = \Omega((m-\ell)/\delta)$.

We show this through the probabilistic method. Let us pick a random right-set B of size $(1-2\delta)2^{\ell}$. For a given left-node a, the probability that all its neighbours fall into B is at least

$$\binom{2^{\ell} - d}{(1-2\delta)2^{\ell} - d} \Big/ \binom{2^{\ell}}{(1-2\delta)2^{\ell}} \geq (1-2\delta)^d \left(1 - \frac{2d}{2^{\ell}}\right)^d .$$

Under the assumption that $d \leq \delta 2^{\ell}$, the left-hand side is at least $(1-4\delta)^d$.

It must then hold that for such random B, the expected number of left-nodes that map into B is $2^m(1-4\delta)$. Hence, for some choice of B, there will exist a left-set A of the same size whose neighbours are all in B. ∎

4.6.3 A lower bound for equation (4.2) of the proof of Lemma 4.3.6

4.6.4. Lemma. *There is an ℓ-discrete private-coin one-way protocol π, and a message m sent by π, such that for J defined as in Lemma 4.3.6, it holds that*

$$I(J : X|M_\pi = m) = \Omega(\log \ell).$$

Proof. Suppose Alice is given an input X uniformly distributed over $\{x_1, \ldots, x_N\}$, and private randomness uniformly distributed over $\{r_1, \ldots, r_N\}$, so that $\ell = \log N$. Let π be a one-way protocol given by

$$M_\pi(x_j, r_k) = \begin{cases} 0 & \text{if } k \leq \left\lfloor \frac{N}{j+1} \right\rfloor, \\ 1 & \text{otherwise.} \end{cases}$$

Then conditioned on $M_\pi = 0$, we will have $J(x_j, r_k) = k$. Let $M = \sum_{i=1}^{N} \lfloor \frac{N}{i+1} \rfloor$ be the size of $M_\pi^{-1}(0)$. Finally, let m denote the event $M_\pi = 0$. Then

$$\begin{aligned} I(X : J|m) &= H(X|m) - H(X|m, J) \\ &= \sum_{j=1}^{N} \frac{1}{M} \cdot \left\lfloor \frac{N}{j+1} \right\rfloor \log \frac{M}{\left\lfloor \frac{N}{j+1} \right\rfloor} - \sum_{k=1}^{N} \frac{1}{M} \cdot \left\lfloor \frac{N}{k+1} \right\rfloor \log \left\lfloor \frac{N}{k+1} \right\rfloor \\ &= \log M - \frac{2}{M} \sum_{i=1}^{N} \left\lfloor \frac{N}{i+1} \right\rfloor \log \left\lfloor \frac{N}{i+1} \right\rfloor, \end{aligned}$$

which is $\geq U$ iff:

$$2 \sum_{i=1}^{N} \left\lfloor \frac{N}{i+1} \right\rfloor \log \left\lfloor \frac{N}{i+1} \right\rfloor \leq M(\log M - U) \tag{4.4}$$

Let us denote the left-hand side with A and the right-hand side with B. Because $\frac{N}{x}$ is monotonically decreasing for $x \geq 1$, then:

$$A \leq \frac{2}{\ln 2} \int_1^{N+1} \frac{N}{x} \ln \frac{N}{x} dx.$$

The relevant primitive is $\int \frac{N}{x} \ln \frac{N}{x} dx = -\frac{1}{2} N (\ln \frac{N}{x})^2$ and hence

$$\begin{aligned} A &\leq \frac{2}{\ln 2} \left(-\frac{1}{2} N \left(\ln \frac{N}{N+1} \right)^2 + \frac{1}{2} N (\ln N)^2 \right) \\ &= \frac{2}{\ln 2} \left(N \ln N \ln(N+1) - \frac{1}{2} N (\ln(N+1))^2 \right). \end{aligned}$$

We denote this last quantity by A'. Good bounds for M are:[8]

$$N \ln N - 3N \le M = \sum_{i=1}^{N} \left\lfloor \frac{N}{i+1} \right\rfloor \le N \ln N + N$$

Let $B' := N \ln N - 3N$, so that $B \ge B'(\log B' - U)$. Then we will show that for an appropriate choice of U,

$$A' \le B'(\log B' - U)$$

and hence $A \le B$ and also $I(X : J|m) \ge U$. Equivalently,

$$A' - B' \log B' + B'U \le 0 \tag{4.5}$$

For convenience, let $\alpha = \frac{\ln(N+1)}{\ln N}$ (which goes to 1 as N goes to ∞). Then $A' = \frac{1}{\ln 2} N (\ln N)^2 (2\alpha - \alpha^2)$ and $B' \log B' = \frac{1}{\ln 2} N (\ln N)^2 + \frac{1}{\ln 2} N \ln N \ln\ln N + O(N \ln N)$. Now the proof follows from the following:

4.6.5. Claim. *$N(\ln N)^2(2\alpha - \alpha^2 - 1) \to -\frac{1}{N}$ as $N \to \infty$.*

Because under this claim, the dominant negative term in (4.5) is $\frac{1}{\ln 2} N \ln N \ln\ln N$, and thus all we need to do is set U to be $c \ln\ln N$ for some $c < \frac{1}{\ln 2}$, that this ensures (4.5) is negative. For such a choice of U, it will hold that

$$I(X : J|m) \ge U = c \ln\ln N = \Omega(\log \ell).$$

Unfortunately, l'Hopital's rule does not seem to help us, as the terms become too complicated. Instead we estimate how fast $(2\alpha - \alpha^2 - 1)$ approaches 0 as N goes to infinity. For this, let $\beta = \frac{\ln(\frac{1}{x}+1)}{\ln \frac{1}{x}}$ and let us estimate β as x approaches 0. For x close to, but different from, 0, we have:

$$\beta = 1 - \frac{1}{\ln x} \ln(x+1) = 1 - \frac{x}{\ln x} + \frac{x^2}{2 \ln x} \pm O\left(\frac{x^3}{\ln x}\right)$$

(the last equality is by the Taylor expansion of $\ln(x+1)$ around 0). We also have

$$\beta^2 = \left(1 - \frac{x}{\ln x} + \frac{x^2}{2\ln x} - O\left(\frac{x^3}{\ln x}\right)\right)^2 = \beta - \frac{x}{\ln x} + \frac{x^2}{(\ln x)^2} + \frac{x^2}{2 \ln x} \pm O\left(\frac{x^3}{(\ln x)^2}\right).$$

Hence,

$$2\beta - \beta^2 = 1 - \frac{x^2}{(\ln x)^2} \pm O\left(\frac{x^3}{(\ln x)^2}\right).$$

From this we can conclude that for $x = 1/N$, we have

$$2\alpha - \alpha^2 - 1 = -\frac{1}{N^2 (\ln N)^2} \pm O\left(\frac{1}{N^3 (\ln N)^2}\right),$$

and our claim follows. ■

[8] This is because the harmonic numbers $H_n = \sum_{i=1}^{n} 1/i$ converge to $\ln N + \gamma$ for the Euler–Mascheroni constant $\gamma \approx 0.577$.