Université Paris Diderot (Paris 7)
Sorbonne Paris Cité

Institut de Recherche en Informatique Fondamentale (IRIF)

Thèse

présentée pour l'obtention du diplôme de

Docteur de l'Université Paris Diderot,
spécialité Informatique

à l'École doctorale de Sciences Mathématiques de Paris Centre
(ED 386)

Information Theory for Multi-Party Peer-to-Peer Communication Protocols

Théorie de l'Information pour Protocoles de Communication Peer-to-Peer

Par :
Florent Urrutia

Directeur de thèse : Iordanis Kerenidis

Soutenue publiquement le 25 mai 2018 *devant le jury constitué de :*

Amit Chakrabarti,	Professeur	Dartmouth College	*Rapporteur*
Christoph Dürr,	Directeur de recherche	LIP6	*Examinateur*
Yuval Ishai,	Professeur	Technion	*Examinateur*
Iordanis Kerenidis,	Directeur de recherche	IRIF	*Directeur*
Sophie Laplante,	Professeur des universités	IRIF	*Présidente du jury*
Toniann Pitassi,	Professeur	University of Toronto	*Rapporteuse*
Adi Rosén,	Directeur de recherche	IRIF	*Examinateur*
Aslan Tchamkerten,	Professeur associé	Télécom ParisTech	*Examinateur*

Abstract

I started my PhD studies in fall 2014, in the LIAFA at University Paris VII, which then became the IRIF. During these years, Iordanis Kerenidis and Adi Rosén supervised my main project, centred around the notion of multi-party communication protocols. I also worked under the supervision of François le Gall on improving the complexity of algorithms for the multiplication of rectangular matrices. This manuscript presents the results of my research with Iordanis Kerenidis and Adi Rosén. The results of the work on matrix multiplication have been published in [GU18].

This thesis is concerned with the study of multi-party communication protocols in the asynchronous message-passing peer-to-peer model. We introduce two new information measures, the *Public Information Complexity* (PIC) and the *Multi-party Information Complexity* (MIC), study their properties and how they are related to other fundamental quantities in distributed computing such as communication complexity and randomness complexity. We then use these two measures to study the parity function and the disjointness function. A detailed description of the content of this thesis is given at the end of the introduction.

Keywords: communication protocol, multi-party communication, peer-to-peer model, Public Information Complexity, PIC, Multi-party Information Complexity, MIC, communication complexity, information theory, randomness complexity, privacy.

Résumé

J'ai commencé mon doctorat à l'automne 2014 au LIAFA à l'Université Paris VII, qui est ensuite devenu l'IRIF. Durant ces années, Iordanis Kerenidis et Adi Rosén ont encadré mon projet principal, ayant pour sujet la notion de protocoles de communication peer-to-peer. J'ai également travaillé avec François le Gall pour améliorer la complexité des algorithmes de multiplication de matrices rectangulaires. Ce manuscrit présente le résultat de mes recherches avec Iordanis Kerenidis et Adi Rosén. Les résultats de mon travail sur la multiplication matricielle ont été publiés dans [GU18].

Cette thèse a pour sujet les protocoles de communication peer-to-peer asynchrones. Nous introduisons deux mesures basées sur la théorie de l'information, la *Public Information Complexity* (PIC) et la *Multi-party Information Complexity* (MIC), étudions leurs propriétés et leur relation avec d'autres mesures fondamentales en calcul distribué, telles que la communication complexity et la randomness complexity. Nous utilisons ensuite ces deux mesures pour étudier la fonction parité et la fonction disjointness. Une description détaillée du contenu de cette thèse est donnée à la fin de l'introduction.

Mots-clefs : protocole de communication, modèle peer-to-peer, Public Information Complexity, PIC, Multi-party Information Complexity, MIC, communication complexity, théorie de l'information, randomness complexity, privacy.

Acknowledgements

I first have to thank all the people who helped me to start my PhD: the administration staff from ENS and LIAFA, in particular Isabelle Delais, for her help in dealing with all the formalities necessary to the start of a PhD; Iordanis Kerenidis, Pierre Fraigniaud and Claire Mathieu who vouched for me and assisted me in the quest for a funding; and my family for the financial and logistic support. I also have to thank the many friends who hosted me during my long apartment hunt.

Most of all, I have to thank Iordanis Kerenidis and Adi Rosén, who worked with me during all this time. Thank you for having the knowledge I don't have, for having a sharp mind able to support my vague intuitions, for having been patient and persevering when we were only making slow progress, for being too optimistic about what we can do, for having enough clairvoyance to see where our random walk was leading us, and sometimes for even making it converging, all while giving me enough freedom and having enough trust in me to allow me to explore the areas I was interested in and to be stubborn enough to keep working, perhaps unreasonably, on problems I could sometimes not solve. Thank you for the interesting and stimulating research talks, for having introduced me to other researchers, and for giving me some opportunities to see a bit more of the world.

Thank you, Amit Chakrabarti and Toniann Pitassi, for having taken the time to review my manuscript and for the suggestions you provided. I also thank Christoph Dürr, Yuval Ishai, Sophie Laplante and Aslan Tchamkerten for having accepted to sit in my jury and to attend the defence.

I also want to thank François le Gall who welcomed me in Kyoto, introduced me to a different research field and provided excellent working conditions, and Rafail Ostrovsky who welcomed me for a short stay in Los Angeles.

Thanks to Xinyi for having done her best to prevent me from working too much and for having turned my apartment into a liveable place. Our holidays together were an important part of my PhD.

Thanks to all the students who shared my office and my floor for the friendly working atmosphere.

My last thanks go to my two mathematics teachers, Guillaume Roussel, who taught me how to think well, quickly and elegantly, and gave me the interest in mathematics which led me to this day, and Denis Favennec, who gave me the hard training which made me strong enough to solve (most of) the problems I faced until completing this PhD.

Contents

Introduction

Historical Background

Communication complexity, first introduced by Yao [Yao82], has become a major topic of research in Theoretical Computer Science, both for its own sake, and as a tool which has yielded important results (mostly lower bounds) in various theoretical computer science fields such as circuit complexity, streaming algorithms, or data structures (e.g. [KN97, MNSW95, GG10, SHK$^+$10, FHW12]). Communication complexity is a measure for the amount of communication needed in order to solve a problem whose input is distributed among several players. Informally, it answers the question "How many bits must the players transmit to solve a distributed problem?". The two-party case, where two players cooperate in order to compute a function of their respective inputs, has been widely studied and has produced a large number of interesting and important results, upper and lower bounds; yet major questions in this area are still open today (e.g. the log-rank conjecture, cf. [Lov14]).

The multi-party case, where $k \geq 3$ players cooperate in order to compute a function of their inputs, is much less understood. A number of sub-models have been considered in the literature for the multi-party communication setting: the *number-in-hand* model (NIH), where each player has a private input, is maybe the most natural one, while in the *number-on-the-forehead* model (NOF), each player i knows all inputs x_j, $j \neq i$, i.e. the "inputs" of all players except its own. For the communication pattern, a number of variants exist as well. In the *blackboard* model, the players communicate by broadcasting messages (or writing them on a "blackboard"). In the *coordinator* model, there is an additional entity, the coordinator, and all players communicate back and forth only with the coordinator. The most natural setting is, however, the *message-passing* model, also known as *peer-to-peer* model, where each pair of players is given a private channel to communicate (cf. [KN97] for more details on the different variants). Most of the work realized on multi-party communication complexity focuses on the number-on-the-forehead model and/or the blackboard model, to which some of the techniques developed in the study of two-party protocols have been generalized. This is the case, for example, of the partition (into rectangles) bound, and of the discrepancy method. In contrast, only few lower bound techniques are available for the number-in-hand model, and most of the methods developed in the two-party case appear to be unsuitable, or at least unsatisfactory, for that model.

Lower bounds obtained in the coordinator model can be transferred to the peer-to-peer model at the cost of a $\log(k)$ factor, where k is the number of players, since any peer-to-peer protocol can be simulated in the coordinator model by having the players attach to any message the identity of the destination of that message. The loss of this factor is unavoidable when the communication protocols rely on a flexible communication pattern. Such configurations arise naturally for mobile communicating devices. A practical example would be communicating cars exchanging information with the nearby cars in order to avoid collisions. Constructions based on the *pointer jumping* problem are also likely to be harder in the coordinator model, as solving the problem usually requires exchanging information in a specific order determined by the inputs of the players. On the other hand, other functions, for example the parity function, have the same communication complexity in the peer-to-peer and in the coordinator models. Thus, it is important to develop lower bound techniques which apply directly in the peer-to-peer model.

A powerful tool recently introduced for the study of two-party communication protocols is the measure of *Information complexity* (or *cost*). This measure, first defined in [BCKO93, CSWY01], extends the notion of information theory, originally introduced by Shannon [Sha48], to interactive settings. Information complexity is a measure of how much *information*, about each other's input, the players must learn during the course of the protocol in order to succeed in computing the function correctly. Since the information complexity can easily be shown to provide a lower bound on the communication complexity, this measure has proven to be a strong and useful tool for obtaining lower bounds on the two-party communication complexity in a sequence of papers (e.g. [BYJKS02, BBCR10, BR11, Bra12]).

An interesting property of information complexity is that it satisfies a *direct sum*. The direct sum question, one of the most fundamental questions in complexity theory, asks whether solving n independent copies of the same problem must cost (in a given measure) n times the cost of solving a single instance. In the case of communication complexity, this question has been studied in e.g. [FKNN95, CSWY01, Sha03, JRS03, HJMR10, BBCR10, Kla10, Jai15] and in many cases it remains open whether a direct sum property holds.

Another important question is the relation between the information complexity of a function and its communication complexity. We would like to know if it is possible to compute a function by sending a number of bits which is not (too much) more than the information the protocol actually has to reveal. Put differently, is it always possible to *compress* the communication cost of a protocol to its information cost? For the two-party case it is known that perfect compression is not possible [GKR15a, GKR15b]. Still, several interesting compression results are known. The equality between information cost and amortized communication cost is shown in [BR11, Bra12], and other compression techniques are given in [BBCR10, BMY15, BBK$^+$13, Pan15]. It remains open if one can compress interactive communication up to some small loss (for example logarithmic in the size of the input).

Unfortunately, information complexity cannot be extended in a straightforward manner to the multi-party setting. The celebrated results on information-theoretic private computation [BOGW88, CCD88] state that if the number of players is at least 3, then *any function* can be computed by a randomized protocol such that *no information* about the inputs is revealed to the players (other than what is implied by the value of the function and their own input). This implies that the information complexity of all functions is too low to provide a meaningful lower bound on the communication complexity in the natural multi-party peer-to-peer setting. Therefore, information complexity and its variants have rarely been used to obtain lower bounds on multi-party communication complexity, and only in settings which do not allow for private protocols (and most notably not in the natural peer-to-peer setting). One example of such work is [HRVZ15] which introduces a notion of external information cost in the coordinator model of [DF89] to study maximum matching in a distributed setting. Among the interesting works on multi-party communication which are not based on information complexity are [CRR14, CR15] which study the influence of the topology of the network and [PVZ12, WZ14] which introduce the techniques of *symmetrization* and *composition*, further developed along with other reduction techniques in [WZ11, WZ13]. Another example is the notion of strong fooling sets, introduced in [CK16] to study deterministic communication complexity of discreet protocols, also defined in [CK16].

In certain circumstances, we would like that the players, while being able to compute the value of the function, retain as much privacy as possible about their input. Informally, we would like that the players learn nothing about the others' input but the value of the function. The question of when such a protocol is possible, and how to design it, has been posed in the field of cryptography [Yao82]. In cryptography, the notion of security is computational: we assume that the players have a limited computation power, and we want to design a protocol which ensures that the players cannot get more information than they should be able to. Constructions based on trapdoor one-way functions [GMW87, CDvdG88] answer this question. A stronger notion of security is information-theoretic security. Instead of relying on cryptographic assumptions, we now aim for unconditionally secure protocols. As discussed above, it is well known that in the multi-party ($k \geq 3$) number-in-hand peer-to-peer setting, unlike in the two-party case, any function can be privately computed [BOGW88, CCD88]. Private computation is attained through the use of private randomness. The amount of randomness needed in order to compute privately a given function has been many times referred to as the *randomness complexity* of that function. The interest of randomness complexity lies in the fact that true randomness is considered as a costly resource, and in the fact that the randomness complexity has been shown to be related to other complexity measures, such as the circuit size of the function or its sensitivity. For example, it has been shown in [KOR96] that a boolean function f has a linear size circuit if and only if f has constant randomness complexity. A few works [BDSPV99, KM97, GR05] prove lower bounds on the randomness complexity of the parity function. The parity and other modulo sum functions are, to the best of our knowledge, the only functions for which randomness complexity lower bounds are available.

Content of this thesis

Contributions

In this thesis, we will focus on the number-in-hand, peer-to-peer model. This setting has been studied, in the context of communication complexity, less than the other settings, probably due to the difficulty of tracking the distributed communication patterns that occur during a run of a protocol in this model. It is, however, not only the most natural one, and the one that occurs the most in real systems, but also the setting widely studied in the distributed algorithms and distributed computation communities.

Our main goal is to introduce novel information-theoretical measures for the study of number-in-hand, peer-to-peer multi-party protocols, coupled with a natural model that, among other things, allows private protocols. We attempt to fill the gap in the study of peer-to-peer communication complexity, and, further, create a bridge between the research fields of communication complexity and distributed computation.

We propose a model that, on one hand, is a very natural peer-to-peer model, and very close to the model used in the distributed computation community, and, at the same time, does have properties that allow one to analyze protocols in terms of their information complexity. While at first sight the elaboration of such a model does not seem to be a difficult task, many fundamental and technical issues render this task non-trivial. For example, one would like to define a notion of "transcript" that would guarantee both a relation between the length of the transcript and the communication complexity, and at the same time will contain *all* the information that the players get and use while running the protocol. The difficulty in elaborating such a model may be the reason for which hardly any work studied communication complexity in a multi-party peer-to-peer setting. We propose the model and prove a number of fundamental properties that allow one to analyze protocols in that model. Our model allows for private protocols. We also show that our model is at least sometimes stronger than the models that have been previously used in most of the papers dealing with multi-party communication complexity, and that if one seeks to accurately understand the natural peer-to-peer model, suppressing polylog-factor inaccuracies, one has to study directly the peer-to-peer model.

We first define the new measure of *Public Information Complexity* (PIC), as a tool for the study of multi-party communication protocols, and of quantities such as their communication complexity, or the amount of randomness they require in the context of information-theoretic private computation. Intuitively, our measure captures a combination of the amount of information about the inputs that the players leak to other players, and the amount of randomness that the protocol uses. By proving lower bounds on PIC for a given multi-party function f, we are able to give lower bounds on the communication complexity of f and on the amount of randomness needed to privately compute f. The crucial point is that the public information complexity of functions, in our multi-party model, is not always zero, unlike their information complexity.

We go on to show a number of interesting properties and applications of our new notion:

- The public information complexity is a lower bound on the communication complexity and an upper bound on the information complexity. In fact, it can be strictly larger than the information complexity.
- The difference between the public information complexity and the information complexity provides a lower bound on the amount of randomness used in a protocol. We show that in the two-party setting, the use of private coins is required in order to achieve the optimal information cost.
- We compress communication protocols to their PIC (up to logarithmic factors), by extending to the multi-party setting the work of Brody et al. [BBK+13] and Pankratov [Pan15].
- We show that one can approach the central question of direct sum in communication complexity by trying to prove a direct sum result for PIC. Indeed, we show that a direct sum property for PIC implies a certain direct sum property for communication complexity.
- We explicitly calculate the zero-error public information complexity of the k-party, n-bit parity function (Par_k^n), where a player outputs the bit-wise parity of the inputs. We show that the PIC of this function is $n(k-1)$. This result is tight and it also establishes that the amount of randomness needed for a private protocol that computes this function is $\Omega(n)$.

We then introduce an information-theoretic measure that we call *Multi-party Information Complexity* (MIC). MIC is a natural extension of the two-party information cost, and can be interpreted as summing over all players i the sum of two terms: what player i learns on the other players' inputs, and what player i leaks about its input.

- We show that MIC can be used as a lower bound on the communication complexity.

- We prove a direct sum property for product distributions which allows us to obtain tight bounds on the MIC of the parity function Par_k^n.

- We also study the disjointness function by introducing the specific measure of SMIC (*Switched Multi-party Information Cost*) and by following the approach of [BEO$^+$13]. We prove a specific direct sum property and obtain a bound of $\Omega(kn)$ on the multi-party information complexity of the k-player n-bit disjointness function in our peer-to-peer model, which leads to a tight bound of $\Omega(kn)$ on its communication complexity for $n \geq \alpha k$, where α is a constant. From a quantitative point of view, our result for the disjointness function improves by a $\log k$ factor the lower bound that could be deduced for our model from results on the disjointness function in the coordinator model [BEO$^+$13].

- We further prove, by relating SMIC to PIC, that any private protocol for the disjointness function must use at least $\Omega(n)$ private coins. The importance of this result lies in that it is the first such lower bound that grows with the size of the input while the output remains a single bit, by contrast to the sum function from [BDSPV99] or the bitwise parity function that we also study in this thesis.

Organization

The thesis is organized as follows.

In Chapter 1, we introduce the notion of communication protocols and our communication model. We review the traditional notions of communication complexity (CC) and information complexity (IC), and the relation between them. We describe how information complexity was used in the context of two-party communication protocols, and explain why a new approach is needed when one is interested in studying multi-party protocols.

In Chapter 2, we introduce the new notion of *public information cost* and study its properties. In particular, we show that the public information cost can be a tool for the study of communication complexity.

We then give an introduction on private computation and randomness complexity. We show why the possibility of private computation turns the information complexity into a meaningless quantity in the multi-party setting. We explain why on the contrary the public information cost is a pertinent notion, and study the connection between public information cost and randomness complexity.

We present several lower bound techniques for the public information cost, and study the public information cost of the functions **And** and **Parity**. This allows us to compute the randomness complexity of the parity function.

Last, we give some background on the notion of direct sum, review the problem of compressing communication protocols, and explain the links between the two. We then explain why the public information cost is related to them.

In Chapter 3, we introduce the new notion of *multi-party information cost*, show that it satisfies a direct sum property for product distributions and that it can be used as a lower bound on the communication complexity. We then prove a lower bound on the MIC and the CC of the parity function.

We introduce the measure of SMIC and use it to prove a bound on the MIC and the CC of the function **Disjointness**. We then prove that SMIC is also a lower bound on PIC, which leads to a bound on the randomness complexity of the disjointness function.

The model of communication introduced in Chapter 1 and its analysis, as well as the content of Chapter 2, come from work with Iordanis Kerenidis and Adi Rosén and was published in [KRU16].

The content of Chapter 3 comes from work with Adi Rosén [RU17]. The idea of using SMIC in the peer-to-peer model originates from a discussion with Rotem Oshman.

Chapter 1

Multi-party Protocols, Communication and Information

1.1 General notations

We start by defining a number of notations. We denote by k the number of players. We often use n to denote the size (in bits) of the input to each player. Calligraphic letters will be used to denote sets. Upper case letters will be used to denote random variables, and given two random variables X and Y, we will denote by XY the joint random variable (X, Y). Given a string (of bits) s, $|s|$ denotes the length of s. Using parentheses we denote an ordered set (family) of items, e.g. (Y_i). Given a family (Y_i), Y_{-i} denotes the sub-family which is the family (Y_i) *without* the element Y_i. The letter X will usually denote the input to the players, and we thus use the shortened notation X for (X_i), *i.e.* the input to all players. The letter π will be used to denote a protocol. log is the binary logarithm. Given a function $f : A \to B$, $f^{\otimes n}$ denotes the function $(f, \dots, f) : A^n \to B^n$. Given a distribution μ on a set U, $\mu^{\otimes n}$ denotes the distribution on the set U^n defined by $\mu^{\otimes n}(u_1, \dots, u_n) = \prod_{i=1}^{n} \mu(u_i)$. Denote by $\overline{1}^t$ the vector of length t, each entry consisting of the bit 1. Denote by $\overline{e}^t_{a_1,\dots,a_d}$ the vector obtained from $\overline{1}^t$ by changing the bit 1 into the bit 0 at indexes $a_1, \dots, a_d$. To simplify notations, we sometimes omit the superscript t when $t = k$, and write $\overline{e}_{a_1,\dots,a_d}$ or $\overline{1}$. For $x \in \{0, 1\}^k$ and $b \in \{0, 1\}$, let $x_{[i \leftarrow b]}$ represent the input obtained from x by replacing the i^{th} bit of x by b. Last, δ_{ij} is the Kronecker delta, having value 1 if $i = j$ and value 0 otherwise.

1.2 Information theory

We give a reminder on basic information theory, as introduced by Shannon in [Sha48]. A good reference about information theory is the classic book of Cover and Thomas [CT06].

All the distributions considered here are defined over discrete domains.

Definition 1.2.1. *The entropy of a random variable X is*

$$H(X) = \sum_x \Pr[X = x] \log \left(\frac{1}{\Pr[X = x]} \right).$$

The conditional entropy $H(X \mid Y)$ is defined as $\mathbb{E}_y[H(X \mid Y = y)]$.

Proposition 1.2.2. *For any random variables X and Y, $H(X \mid Y) \leq H(X)$.*

The entropy of a random variable is always non negative. It is a measure of the information that it contains, or equivalently, its uncertainty. This is the measure that we will use throughout all the thesis. The classic theorem of Shannon gives an additional light on the meaning of the entropy: when sending n independent copies of a random variable X, the limit when n goes to infinity of the required number of bits per copy is $H(X)$.

As the random variables we will work with are strings of bits, it is pertinent to see the entropy as the "number of unknown bits".

Theorem 1.2.3 (Shannon)**.** *For any prefix-free finite set $\mathcal{X} \subseteq \{0,1\}^*$ and any random variable X with support supp$(X) \subseteq \mathcal{X}$, it holds*

$$H(X) \leq \mathbb{E}[|X|].$$

Definition 1.2.4. *The mutual information between two random variables X,Y is*

$$I(X;Y) = H(X) - H(X \mid Y).$$

The mutual information of X and Y conditioned on Z is

$$I(X;Y \mid Z) = H(X \mid Z) - H(X \mid YZ).$$

The mutual information measures the change in the entropy of X when one learns the value of Y. The mutual information satisfies the following important properties.

Proposition 1.2.5. *The mutual information is non negative: for any random variables X,Y,Z,*

$$I(X;Y \mid Z) \geq 0.$$

Proposition 1.2.6. *The mutual information is symmetric: for any random variables X,Y,Z,*

$$I(X;Y \mid Z) = I(Y;X \mid Z).$$

Proposition 1.2.7. *For any random variables X, Y and Z, $I(X;Y \mid Z) = 0$ if and only if X and Y are independent conditioned on every possible value of Z.*

We will use extensively the following propositions, known under the name of *chain rules*. It allows one to decompose the entropy of a couple of random variables.

Proposition 1.2.8 (Chain rule for the entropy)**.** *For any random variables A, B, C,*

$$H(AB \mid C) = H(B \mid C) + H(A \mid BC).$$

Proposition 1.2.9 (Chain rule for the mutual information)**.** *For any random variables A, B, C, D,*

$$I(AB;C \mid D) = I(A;C \mid D) + I(B;C \mid DA).$$

Proposition 1.2.10 (Generalized chain rule for the mutual information)**.** *For any random variables $A_1 \dots A_p$, $B_1 \dots B_q$, C,*

$$I(A_1 \dots A_p; B_1 \dots B_q \mid C) = \sum_{i=1}^{p} \sum_{j=1}^{q} I(A_i; B_j \mid (A_r)_{r<i} (B_s)_{s<j} C).$$

Proof.

$$\begin{aligned} I(A_1 \dots A_p; B_1 \dots B_q \mid C) &= \sum_{i=1}^{p} I(A_i; B_1 \dots B_q \mid (A_r)_{r<i} C) \\ &\quad \text{(by Proposition 1.2.9)} \\ &= \sum_{i=1}^{p} \sum_{j=1}^{q} I(A_i; B_j \mid (A_r)_{r<i} (B_s)_{s<j} C) \text{ (idem).} \end{aligned}$$

⌟

The *data processing inequality* expresses the fact that information can only be lost when applying a function to a random variable.

Proposition 1.2.11. *For any random variables X, Y, and any function f,*

$$H(f(X) \mid Y) \leq H(X \mid Y).$$

Proposition 1.2.12 (Data processing inequality for mutual information)**.** *For any random variables X, Y, Z, and any function f*

$$I(X; f(Y) \mid Z) \leq I(X; Y \mid Z).$$

We will occasionally make use of the two following lemmas, which allow to add or remove a random variable from the conditioning.

Lemma 1.2.13 ([Bra12])**.** *For any random variables A, B, C, D such that $I(B; D \mid AC) = 0$,*

$$I(A; B \mid C) \geq I(A; B \mid CD).$$

Lemma 1.2.14 ([Bra12])**.** *For any random variables A, B, C, D such that $I(B; D \mid C) = 0$,*

$$I(A; B \mid C) \leq I(A; B \mid CD).$$

We will also use the following technical lemma.

Lemma 1.2.15. *Let A, B, C, D, $\phi = \varphi(C, B)$ be random variables.*

$$I(A; B \mid CD) = 0 \Longrightarrow I(A; \phi \mid CD) = 0.$$

Proof.

$$\begin{aligned} I(A; \phi \mid CD) &= I(A; \varphi(C, B) \mid CD) \\ &= \mathop{\mathbb{E}}_{c}[I(A; \varphi(c, B) \mid C = c, D)] \\ &\leq \mathop{\mathbb{E}}_{c}[I(A; B \mid C = c, D)] \text{ (by data processing inequality 1.2.12)} \\ &\leq I(A; B \mid CD). \end{aligned}$$

⌟

We present an unusual variant of mutual information, that we name *information leak.*

Definition 1.2.16. *The information leak in a random variable X when a random variable Y takes value y is*

$$\tilde{I}(X; Y = y) = H(X) - H(X \mid Y = y).$$

The information leak in a random variable X when a random variable Y takes value y conditioned on the fact that a random variable Z has value z is

$$\tilde{I}(X; Y = y \mid Z = z) = H(X \mid Z = z) - H(X \mid Y = y, Z = z).$$

Note that unlike the mutual information I, the information leak is not a symmetric quantity. The information leak can be used instead of the mutual information to get more control over the structure of the variable Y. The mutual information is actually the average information leak, as shown by the following proposition.

Proposition 1.2.17.

$$\mathop{\mathbb{E}}_{y}[\tilde{I}(X;Y=y)] = I(X,Y)$$

and

$$\mathop{\mathbb{E}}_{y,z}[\tilde{I}(X;Y=y \mid Z=z)] = I(X,Y \mid Z).$$

Proof.

$$\begin{aligned}\mathop{\mathbb{E}}_{y}[\tilde{I}(X;Y=y)] &= \mathop{\mathbb{E}}_{y}[H(X) - H(X \mid Y=y)] \\ &= H(X) - H(X \mid Y) \\ &= I(X,Y)\end{aligned}$$

and

$$\begin{aligned}\mathop{\mathbb{E}}_{y,z}[\tilde{I}(X;Y=y \mid Z=z)] &= \mathop{\mathbb{E}}_{y,z}[H(X \mid Z=z) - H(X \mid Y=y, Z=z)] \\ &= H(X \mid Z) - H(X \mid YZ) \\ &= I(X,Y \mid Z).\end{aligned}$$

⌟

When there is no ambiguity, we may write $\tilde{I}(X;y)$ for $\tilde{I}(X;Y=y)$, $\tilde{I}(X;y \mid z)$ for $\tilde{I}(X;Y=y \mid Z=z)$ so as to work with lighter notations.

Proposition 1.2.18 (Chain rule for the information leak)**.** *For any random variables A, B, C, D,*

$$\tilde{I}(A;bc \mid d) = \tilde{I}(A;b \mid d) + \tilde{I}(A;c \mid bd).$$

Proof.

$$\begin{aligned}\tilde{I}(A;bc \mid d) &= H(A \mid d) - H(A \mid bcd) \\ &= H(A \mid d) - H(A \mid bd) + H(A \mid bd) - H(A \mid bcd) \\ &= \tilde{I}(A;b \mid d) + \tilde{I}(A;c \mid bd).\end{aligned}$$

⌟

We will also use a convenient statistical tool called Hellinger distance.

Definition 1.2.19. *Let P and Q be two distributions over a domain Ω. The Hellinger distance between P and Q is*

$$h(P,Q) = \frac{1}{\sqrt{2}}\sqrt{\sum_{\omega\in\Omega} \mid \sqrt{P(\omega)} - \sqrt{Q(\omega)} \mid^2}.$$

It can be checked that it satisfies the triangular inequality. When writing a Hellinger distance, we will mostly use the following identity.

Proposition 1.2.20. *Let P and Q be two distributions over a domain Ω.*

$$h(P,Q)^2 = 1 - \sum_{\omega\in\Omega}\sqrt{P(\omega)Q(\omega)}.$$

Hellinger distance can be related to mutual information by the following relation.

Lemma 1.2.21 ([BYJKS02])**.** *Let η_0, η_1 be two distributions. Suppose that Y is generated as follows: we first select S uniformly in $\{0,1\}$, and then sample $Y$from η_S. Then $I(S,Y) \geq h(\eta_0,\eta_1)^2$.*

Another useful measure is the statistical distance.

Definition 1.2.22. *Let P and Q be two distributions over a domain Ω. The statistical distance between P and Q is*

$$\Delta(P,Q) = \max_{\Omega'\subseteq\Omega} \mid P(\Omega') - Q(\Omega') \mid .$$

Hellinger distance and statistical distance are related by the following relation.

Lemma 1.2.23. *Let P and Q be two distributions. $h(P,Q) \geq \frac{1}{\sqrt{2}}\Delta(P,Q)$.*

1.3 Two-party protocols

Before introducing a framework for multi-party communication, we first start by giving a light introduction to two-party communication protocols.

In the two-party setting, two players, Alice with input $x \in \mathcal{X}$ and Bob with input $y \in \mathcal{Y}$, wish to compute a function of their joint inputs $f(x,y)$. Each player is also given a private random tape, and a public random tape is also available. In order to become able to compute the function f, the players are allowed to exchange messages, according to a fixed protocol. This means that the content of the messages sent by Alice is determined by functions of her input, her private random tape and the public random tape, as well as the messages she has received so far, and similarly for Bob.

1.3.1 Information complexity

Information complexity measures the amount of information that must be transmitted so that the players can compute a given function of their joint inputs. One of its main uses is to provide a lower bound on the communication complexity of a function. In the two-party setting, this measure led to interesting results on various functions such as AND and **Disjointness**.

In the two-party case, two interesting information measures coexist. *External* information complexity (cf. [CSWY01, BYJKS02, Bra12]) represents how much information an external observer would get by observing the transcript of the protocol.

Definition 1.3.1. *The external information complexity of a protocol π on distribution μ is*

$$\mathsf{IC}_{\mu}^{ext}(\pi) = I(XY; \Pi).$$

Internal information complexity represents how much information the players learn about each other's input during the protocol.

Definition 1.3.2. *The internal information complexity of a protocol π on distribution μ is*

$$\mathsf{IC}_{\mu}(\pi) = I(X; \Pi \mid Y) + I(Y; \Pi \mid X).$$

The two following propositions describe the relation between internal information cost and external information cost in the two-party case.

Proposition 1.3.3 ([Bra12])**.** *For any protocol π and any input distribution μ,*

$$\mathsf{IC}_{\mu}(\pi) \leq \mathsf{IC}_{\mu}^{ext}(\pi).$$

Proposition 1.3.4 ([Bra12])**.** *For any protocol π and any input product distribution μ,*

$$\mathsf{IC}_{\mu}(\pi) \geq \mathsf{IC}_{\mu}^{ext}(\pi).$$

1.3.2 Communication complexity

Communication complexity, introduced in [Yao79], measures how many bits of communication are needed in order for a set of players to compute with error ϵ a given function of their inputs.

Definition 1.3.5. *The communication complexity is defined as the worst case, over the possible inputs and the possible randomness, of the number of bits sent by all players during the protocol. We denote the communication complexity of a protocol π by $\mathsf{CC}(\pi)$.*

Definition 1.3.6. *The communication complexity of a function f is*

$$\mathsf{CC}(f) = \inf_{\pi \text{ computing } f} \mathsf{CC}(\pi).$$

Most lower bound techniques for the communication complexity in the two-party setting rely on the fundamental notion of combinatorial rectangle. A communication protocol can be seen as the realization of a partition of the player's inputs set into rectangles. This is because every message sent during the protocol cuts the set of inputs into two subset: the inputs compatible with this message, and the others. As the protocol must allow the players to compute the function, the partition realized by the protocol is said monochromatic. This means that the function to be computed by the players is constant on the subsets of the partition. Bounds on the size of monochromatic partitions thus lead to bounds on the communication complexity. Various methods, such as the fooling set method, the rank method, and the discrepancy method in the randomized case, aim at bounding the size of possible monochromatic partitions

More interesting to us is the link between information and communication. The following proposition relates the communication complexity and the information complexity.

Proposition 1.3.7 ([BR11])**.** *For any protocol π and input distribution μ, $\mathsf{CC}(\pi) \geq \mathsf{IC}_\mu(\pi)$. Thus, for any function f, $\mathsf{CC}(f) \geq \mathsf{IC}(f)$.*

1.4 Multi-party communication

Communication protocols are a theoretical model for distributed systems. Such systems are ubiquitous, and communications protocols naturally arise in the study of computer systems, networks, computer architecture, etc. A communication model is a formal construction which represents the informal idea of a discussion among a group of people, called the *players*. Several points must be specified.

- We first need to choose how to model the players, which usually means choosing a computation model. In most of the work realized in the distributed computation community, the players are Turing machines. We are usually not interested in the computation time, the focus being on the interaction between the players.

- It is also important to decide what resources are available to the players. A communication model is called *synchronous* if the players have access to a common clock, otherwise it is called *asynchronous*. We can or not assume the existence of a string of random bits available to the players, as well as the existence of a private string of random bits for each player.

- We finally need to describe how the players communicate. In the *blackboard* model, there is a common board that all the players can see and on which they can all write. In the *peer-to-peer* model, any player is able to send messages directly to any other player, through a private channel. In the *coordinator* model introduced in [DF89], a specific player is able to send and receive messages from any player, whereas the other players can only communicate with that player.

In a given communication model, a communication protocol is a formal description of the behaviour of the players. Communication protocols can be studied for diverse reasons. In the field of distributed computation, one is interested in the case where the players wish to compute a function, the input of which is distributed among the players. In the *number-on-the-forehead* model, for every player i there is a variable X_i which is available to all players but player i. In the *number-in-hand* model, every player i knows the variable X_i.

We then need to define what it means for the protocol to succeed in computing the function. We could for example require that all players are at the end able to compute the function, or that only one designated player has to be able to compute the value of the function. In some cases, we may want to only consider protocols which eventually stop. We may also allow the players to make mistakes.

1.4.1 Model of communication

We define here a natural communication model which is a slight restriction of the general asynchronous peer-to-peer model. Its restriction is that for a given player at a given time, the set of players from which that player waits for a message can be determined by that player's own local view. This allows us to define information-theoretical tools that pertain to the transcripts of the protocols, and at the same time to use these tools as lower bounds for communication complexity. This restriction however does not exclude the existence of private protocols, as other special cases of the general asynchronous model do. We observe that without such a restriction the information revealed by the execution of a protocol might be higher than the number of bits transmitted and that, on the other hand, practically all multi-party protocols in the literature are implicitly defined in our model.

In the next subsection, we will compare our model to the general one and to other restricted ones and explain the usefulness and logic of our specific model.

We now describe formally the communication model we will work with. It is an *asynchronous multi-party number-in-hand peer-to-peer* model. To make the discussion simpler we assume a global time which is *unknown* to the players.

Each player i has an input X_i and has access to a source of private randomness R_i. We will use the notation R for (R_i), i.e. the private randomness of all players. A source of public randomness R^p is also available to all players. We will call a protocol with no private randomness a *public-coins protocol*, and a protocol with no public randomness a *private-coins* protocol. Each player has unbounded local computation power.

We will consider a family of k players and a family of k functions $f = (f_i)_{i \in [\![1,k]\!]}$, with $\forall\ i \in [\![1,k]\!], f_i : \prod_{l=1}^{k} \mathcal{X}_l \to \mathcal{Y}_i$. Each player i is given some input $x = (x_i) \in \prod_{i=1}^{k} \mathcal{X}_i$ and has to compute $f_i(x)$. Thus, $\mathcal{X}_l$ denotes the set of possible inputs of player l, and $\mathcal{Y}_i$ denotes the set of possible outputs of player i. We will usually denote the set of possible inputs as $\mathcal{X}$. Thus $\mathcal{X} \subseteq \mathcal{X}_1 \times \cdots \times \mathcal{X}_k$.

Every pair of players is connected by a bidirectional communication link that allows them to send messages in both directions. There is no bound on the delivery time of a message, but every message is delivered in finite time, and the communication link maintains FIFO order in each of the two directions. Each player has a special write-only output tape.

Given a specific time we define the *view* of player i, denoted D_i, as the input X_i of that player, its private randomness R_i, the public randomness R^p and the messages read so far by player i. The protocol of each player i runs in *local* rounds. In each round, player i sends messages to some subset of the other players. The identity of these players, as well as the content of these messages, depend on the current view of player i. The player also decides whether to write a (nonempty) string on its output tape. Then, the player waits for messages from a certain subset of the other players, this subset being also determined by the current view of the player. Then the (local) round of player i terminates. Note that the fact that the receiving of the incoming messages comes as the last step of the local round comes only to emphasize that the sending of the messages and the writing on the output tape are a function of only the messages received in previous local rounds. To make it possible for the player to identify the arrival of the *complete* message that it waits for, we require that each message sent by a player in the protocol be self-delimiting.

Denote by D_i^l the view of player i at the end of local round l, $l \geq 0$, where the beginning of the protocol is considered round 0. Formally, a protocol π is defined by a sequence of functions for each player i, parametrized by the *local* round l, $l \geq 1$:

- $S_i^{l,s} : D_i^{l-1} \to 2^{\{1,\ldots,k\}\setminus\{i\}}$, defining the set of players to which player i *sends* the messages.

- $m_{i,j}^l : D_i^{l-1} \to \{0,1\}^*$, for all $j \in S_i^{l,s}(D_i^{l-1})$, defining the content of the messages player i sends. Each such message has to be self-delimiting.

- $O_i^l : D_i^{l-1} \to \{0,1\}^*$, defining what the player i writes on its output tape.

- $S_i^{l,r} : D_i^{l-1} \to 2^{\{1,\ldots,k\}\setminus\{i\}}$, defining the set of players from which player i waits to *receive* a message.

For simplicity we also assume that a protocol must eventually stop. That is, for all possible inputs and all possible assignments for the random sources, there is eventually no message in transit. Note that a player may not know locally that the protocol ended. Note also that the model does not impose "coherence" between the players. That is, the model does not preclude the possibility that a certain player waits indefinitely for a message that is never sent to it. Similarly, it may happen that a player sends a message which is never read.

A meaningful variant of this model (the **return** variant) consists in imposing that the players should *return* their output, in the sense that outputting and halting the local program are simultaneous events. In this case, in each round every player has to decide whether he should stop and output or go on with the communication. Formally, for every player i and every round l, there is a function $O_i^l : D_i^{l-1} \to \{0,1\}^+ \cup \perp$, defining whether or not the local program of player i stops and returns its output. If the value is $\perp$ then no output occurs. If the value is $y \in \{0,1\}^+$, then the local program stops and the player returns the value y.

We will also use at some point a special case of our model, where the sets $S_i^{l,s}$ and $S_i^{l,r}$ are a function of i and l only, and not of the entire current view of the player. This is a natural special case for protocols which we call **oblivious protocols**, where the communication pattern is fixed and is not a function of the input or the randomness. The messages themselves remain a function of the view of the players.

Note that all these models also allow for private protocols.

We define the transcript of the protocol of player i, denoted Π_i, as the concatenation of the messages read by player i from the links of the sets $S_i^{1,r}, S_i^{2,r}, \ldots$, ordered by local round number, and within each round by the index of the player. We denote by $\overleftrightarrow{\Pi_i}$ the concatenation of Π_i together with a similar concatenation $\overrightarrow{\Pi_i}$ of the messages sent by player i to the sets $S_i^{1,s}, S_i^{2,s} \ldots$ We denote by $\Pi_{i \to j}$ the concatenation of the messages sent by player i to player j during the course of the protocol. The transcript of the (whole) protocol, denoted Π, is obtained by concatenating all the Π_i ordered by player index.

In Chapter 3, we will use a different representation of the transcripts of the players, in order to stock more information. We call this representation the *channel representation.* In order not to make the notations heavy, we will denote it in the same way as the representation described in the previous paragraph. There will be, however, no risk of confusion, as the channel representation will not be used at the same time as the other representation. The channel representation is defined as follows.

We first define $k(k-1)$ basic transcripts $\Pi_{i,j}^r$, denoting the transcript of the messages read by player i from its link from player j, and another $k(k-1)$ basic transcripts $\Pi_{i,j}^s$, denoting the transcript of the messages sent by player i on its link to player j. We then define the transcript of player i, Π_i, as the $2(k-1)$-tuple of the $2(k-1)$ basic transcripts $\Pi_{i,j}^r, \Pi_{i,j}^s$, $j \in [\![1,k]\!] \setminus \{i\}$. The transcript of the whole protocol Π is defined as the k-tuple of the k player transcripts Π_i, $i \in [\![1,k]\!]$. We denote by $\Pi_i(x,r)$ the transcript of player i when protocol π is run on input x and on randomness (public and private of all players) r, and similarly by $\overleftarrow{\Pi_i}(x,r)$ the partial transcript of player i composed only of the incoming messages.

Observe that while $\Pi_{i,j}^r$ is always a prefix of $\Pi_{j,i}^s$, the definition of a protocol does not imply that they are equal. Further observe that each bit sent in π appears in Π at most twice.

When working with oblivious protocols, we will sometimes need to refer to individual messages. We define a natural enumeration of the messages of an oblivious protocol. We first define a sequence of lots of messages. In each lot there is at most one message of any of the $k(k-1)$ directional links. The order of the messages is defined by the order of the lots, and inside each lot the messages are ordered according to the lexicographic order of the identities of the sender and receiver of each message. The messages are assigned to lots as follows. The messages sent at the start of the protocol form a first lot; once these messages are delivered, new messages are sent by the players, which form the second lot, and so on. Formally, the messages assigned to lot $s \geq 1$ are defined inductively after lots $s' < s$ have been defined. To define the messages of lot $s > 1$, proceed as follows for each player i: run the protocol π, and whenever player i is waiting for a message, extract it from the already defined lots (lots $s' < s$), if that message is assigned to one of them. Continue until a needed message is not available (i.e. the protocol "gets stuck"), or after player i sends a message not yet assigned to a lot $s' < s$. In the latter case, assign to lot s all the messages sent by player i in the same local round (i.e. for any player i and local round r, all messages sent by player i in local round r are in the same lot). This enumeration of the messages respects the intuitive "temporal causality" of the protocol. Denote by $(\overrightarrow{T_i^l})_{l\geq 0}$ the family of messages sent by player i in the protocol π, ordered according to the enumeration that we just defined. Similarly, denote by $(\overleftarrow{T_i^l})_{l\geq 0}$ the family of messages received by player i. Denote by $j(i,l)$ the player receiving $\overrightarrow{T_i^l}$. For any l_0, let $\overrightarrow{T_i^{<l_0}}$ be the random variable representing the so-far history, i.e. all messages to and from player i until the time of message $\overrightarrow{T_i^{l_0}}$. In a similar way, define $\overleftarrow{T_i^{<l_0}}$ to be the random variable representing the history of the messages to and from player i until the time of message $\overleftarrow{T_i^{l_0}}$. We also define a function $l'(i,l)$ such that every message $\overrightarrow{T_i^l}$ sent by player i is received by player $j(i,l)$ as $\overleftarrow{T_j^{l'}}$ where l' is a short cut for $l'(i,l)$ and j for $j(i,l)$.

We now define what it means for a protocol to compute a family of functions. We will give most of the definitions for the case where all functions f_i are the same function, that we denote by f. The definitions in the case of family of functions are similar.

Definition 1.4.1. *For $\epsilon \geq 0$, a protocol π ϵ-computes a function $f : \mathcal{X} \to \mathcal{Y}$ if for all $(x_1, \ldots, x_k) \in \mathcal{X}$:*

1. *For all possible assignments for the random sources R_i, $1 \leq i \leq k$, and R^p, every player eventually writes on its output tape a non-empty string.*

2. *With probability (over all random sources) at least $1 - \epsilon$ the following*

event occurs: each player i writes on its output tape the value $f(x)$, i.e. the correct value of the function.

The allowed error ϵ, implicit in many of the contexts, will be written explicitly as a superscript when necessary.

We also consider the notion of *external computation.*

Definition 1.4.2. *For $\epsilon \geq 0$, a protocol π is an ϵ-error protocol externally computing $f : \mathcal{X} \to \mathcal{Y}$ if there exists a deterministic function θ taking as input the possible transcripts of π and satisfying*

$$\forall\ x \in \mathcal{X}, \Pr[\theta(\Pi(x)) = f(x)] \geq 1 - \epsilon$$

where we use the channel representation for Π.

Note that a protocol may compute a function without *externally* computing it. This is because the function θ in Definition 1.4.2 must take as input only the transcript of the protocol, and none of the players' input.

1.4.2 Information complexity

The following lemma formalizes the fact that a protocol computing a function f must distinguish inputs having different images by f, and that this can be seen in the distribution of the transcript of the protocol. A stronger version can be found in [BYJKS02].

Lemma 1.4.3. *Let f be a k-party function, and let π be an ϵ-error protocol externally computing f. If x and y are two inputs such that $f(x) \neq f(y)$, then $h(\Pi(x), \Pi(y)) \geq \dfrac{1-2\epsilon}{\sqrt{2}}$ where we use the channel representation for* Π.

Proof. By Lemma 1.2.23, we only need to show that $\Delta(\Pi(x), \Pi(y)) \geq 1-2\epsilon$. By definition, there exists a function θ taking as input the possible transcripts of π and satisfying $\forall\ z \in \mathcal{X}, \Pr[\theta(\Pi(z)) = f(z)] \geq 1 - \epsilon$. Let $\Omega' = \theta^{-1}(f(x))$.

$$\Delta(\Pi(x), \Pi(y)) \geq \mid \Pr[\Pi(x) \in \Omega'] - \Pr[\Pi(y) \in \Omega'] \mid .$$

We have $\Pr[\Pi(x) \in \Omega'] = \Pr[\theta(\Pi(x) = f(x)] \geq 1 - \epsilon$ and

$$\begin{aligned}
\Pr[\Pi(y) \in \Omega'] &= \Pr[\Pi(y) \in \theta^{-1}(f(x))] \\
&\leq \Pr[\Pi(y) \notin \theta^{-1}(f(y))] \\
&\quad \text{(as since } f(x) \neq f(y),\ \theta^{-1}(f(x)) \cap \theta^{-1}(f(y)) = \varnothing) \\
&\leq 1 - \Pr[\Pi(y) \in \theta^{-1}(f(y))] \\
&\leq 1 - (1 - \epsilon) = \epsilon.
\end{aligned}$$

Thus

$$\begin{aligned}\Delta(\Pi(x),\Pi(y)) &\geq \Pr[\Pi(x)\in\Omega'] - \Pr[\Pi(y)\in\Omega'] \\ &\geq (1-\epsilon)-\epsilon = 1-2\epsilon.\end{aligned}$$

⌟

Last, the following lemma formalizes the fact that if a player is able to compute a function with small error at the end of a protocol, then from the point of view of this player the value of the function at the end of the protocol has little entropy.

Lemma 1.4.4. *Let π be a k-party communication protocol, let $i \in [\![1,k]\!]$ and $\epsilon \in \left[0,\frac{1}{2}\right]$. If player i ϵ-computes a binary function f in π, then $H(f(X) \mid X_i R_i R^p \Pi_i) \leq h(\epsilon)$, where h is the binary entropy function. The result also holds if Π_i is the channel representation.*

Proof. Let θ be the function which takes as parameter (x_i, r_i, r^p, π_i) and returns the output of player i at the end of the protocol π, given his input x_i, his local randomness r_i, the public randomness r^p and the transcript π_i. Define the random variable $P = \theta(X_i, R^p, R_i, \Pi_i)$ and the random variable $M = 1 - \delta_{f(X),P}$ i.e. the indicator variable of the event $F(X) \neq P$. Observe that

$$\begin{aligned}\Pr(M=1) &= \mathbb{E}[M] \\ &= \sum_x \Pr(X=x)\,\mathbb{E}[M \mid X=x] \\ &= \sum_x \Pr(X=x)\Pr(M=1 \mid X=x) \\ &\leq \sum_x \Pr(X=x)\times\epsilon \quad \text{(as player } i \text{ } \epsilon\text{-computes f)} \\ &\leq \epsilon.\end{aligned}$$

Thus we have:

$$\begin{aligned}H(f(X) \mid X_i R_i R^p P i_i) &\leq H(f(X) \mid P) \quad \text{(data processing inequality)} \\ &\leq H(M \mid P) \quad \text{(as given } P \text{ there is a bijection} \\ &\quad \text{between } f(X) \text{ and } M) \\ &\leq H(M) \\ &\leq h(\Pr(M=1)) \quad \text{(as M is binary)} \\ &\leq h(\epsilon) \quad \text{(as } h \text{ is increasing on } [0,1/2]).\end{aligned}$$

⌟

We now focus on designing an analogue to the information cost for the multi-party setting. The notion of internal information cost for two-party protocols (cf. [Bra12]) can be easily generalized to any number of players:

Definition 1.4.5. *The internal information cost of a k-party protocol π with respect to input distribution μ is the sum of the information revealed to each player about the inputs of the other players:*

$$\mathsf{IC}_\mu(\pi) = \sum_{i=1}^{k} I(X_{-i}; \Pi_i \mid X_i R_i R^p).$$

The definition we give above, when restricted to two players is the same as in [Bra12], even though they look slightly different. This is because we explicit the role of the randomness, which will later allow us to bound the amount of randomness needed for private protocols in the multi-party setting.

The *internal information complexity* of a function f with respect to input distribution μ, as well as the *internal information complexity* of a function f, can be defined for the multi-party case based on the information cost of a protocol, just as in the 2-party case.

Definition 1.4.6. *The internal information complexity of a function f, with respect to input distribution μ is the infimum of the internal information cost over all protocols computing f on input distribution μ:*

$$\mathsf{IC}_\mu(f) = \inf_{\pi \text{ computing } f} \mathsf{IC}_\mu(\pi).$$

Definition 1.4.7. *The internal information complexity of a function f is the infimum, over all protocols π computing f, of the information cost of π when run on the worst input distribution for that protocol:*

$$\mathsf{IC}(f) = \inf_{\pi \text{ computing } f} \sup_{\mu} \mathsf{IC}_\mu(\pi).$$

The information revealed to a given player by a protocol can be written in several ways. The two following propositions illustrate why the Definition 1.4.5 of information complexity is coherent with the traditional definition of information complexity in the two-party case.

Proposition 1.4.8. *For any protocol π, for any player i:*

$$I(X_{-i}; \overleftrightarrow{\Pi_i} \mid X_i R_i R^p) = I(X_{-i}; \Pi_i \mid X_i R_i R^p).$$

Proof. For any protocol π, for any player i:

$$\begin{aligned}
I(X_{-i}; \overleftrightarrow{\Pi}_i \mid X_iR_iR^p) &= I(X_{-i}; \overrightarrow{\Pi}_i\Pi_i \mid X_iR_iR^p) \\
&= I(X_{-i}; \Pi_i \mid X_iR_iR^p) + I(X_{-i}; \overrightarrow{\Pi}_i \mid X_iR_iR^p\Pi_i) \\
&\quad \text{(using the chain rule, Proposition 1.2.9)} \\
&= I(X_{-i}; \Pi_i \mid X_iR_iR^p) \\
&\quad \text{(as } \overrightarrow{\Pi}_i \text{ can be deduced from } X_iR_iR^p\Pi_i\text{).}
\end{aligned}$$

⌟

Proposition 1.4.9. *In the return variant of our model (cf. Subsection 1.4.1), for any protocol π in which the sets $S_i^{l,s}$ and $S_i^{l,r}$ do not depend on the private randomness of the players, for any player i:*

$$I(X_{-i}; \overleftrightarrow{\Pi}_i \mid X_iR^p) = I(X_{-i}; \overleftrightarrow{\Pi}_i \mid X_iR_iR^p).$$

Proof. We denote $\overleftrightarrow{\Pi}_i^{<j} = B_0 \dots B_{j-1}$. We have:

$$\begin{aligned}
I(X_{-i}; \overleftrightarrow{\Pi}_i \mid X_iR_iR^p) &= I(X_{-i}; B_0 \dots B_q \mid X_iR_iR^p) \\
&= \sum_j I(X_{-i}; B_j \mid X_iR_iR^p\overleftrightarrow{\Pi}_i^{<j}) \\
&\quad \text{(using the chain rule, Proposition 1.2.9).}
\end{aligned}$$

We show that $I(X_{-i}; R_i \mid X_iR^p\overleftrightarrow{\Pi}_i^{<j}) = 0$. By Proposition 1.2.7, it is equivalent to show that conditioned on each possible value of $(X_i, R^p, \overleftrightarrow{\Pi}_i^{<j})$, the random variables X_{-i} and R_i are independent. To do this, we fix a value $(x_i, r, \overleftrightarrow{\pi}_i^{<j})$ of $(X_i, R^p, \overleftrightarrow{\Pi}_i^{<j})$, take a value x_{-i} of X_{-i} such that the event $X_{-i} = x_{-i}$ is compatible with the event $(X_i, R^p, \overleftrightarrow{\Pi}_i^{<j}) = (x_i, r, \overleftrightarrow{\pi}_i^{<j})$, and prove that for any possible value r_i of R_i, the quantity $\Pr[r_i \mid x_i\ r\ \overleftrightarrow{\pi}_i^{<j}\ x_{-i}]$ does not depend on x_{-i}.
We have

$$\Pr[r_i \mid x_i\ r\ \overleftrightarrow{\pi}_i^{<j}\ x_{-i}] = \sum_{r_{-i}} \Pr[r_i \mid x_i\ r\ \overleftrightarrow{\pi}_i^{<j}\ x_{-i}r_{-i}] \Pr[r_{-i} \mid x_i\ r\ \overleftrightarrow{\pi}_i^{<j}\ x_{-i}].$$

We define the set function

$$\rho : (x_i, r\ \overleftrightarrow{\pi}_i^{<j}, x_{-i}, r_{-i}) \mapsto \{r_i \mid \overleftrightarrow{\Pi}_i^{<j}(x_i, x_{-i}, r, r_i, r_{-i}) = \overleftrightarrow{\pi}_i^{<j}\},$$

which represents the set of possible values of r_i, such that inputs x_i, x_{-i}, public randomness r and private randomness r_i, r_{-i} will lead to a transcript

of player i beginning by $\overleftrightarrow{\pi}_i^{<j}$.
Note that $\Pr[r_{-i} \mid x_i \; r \; \overleftrightarrow{\pi}_i^{<j} \; x_{-i}] \neq 0 \iff \rho(x_i, r \; \overleftrightarrow{\pi}_i^{<j}, x_{-i}, r_{-i}) \neq \emptyset$.
Also note that $r_i \notin \rho(x_i, r, \overleftrightarrow{\pi}_i^{<j}, x_{-i}, r_{-i}) \iff \Pr[r_i \mid x_i \; r \; \overleftrightarrow{\pi}_i^{<j} \; x_{-i} r_{-i}] = 0$ and that $\forall \; r_i \in \rho(x_i, r, \overleftrightarrow{\pi}_i^{<j}, x_{-i}, r_{-i})$

$$\Pr[r_i \mid x_i \; r \; \overleftrightarrow{\pi}_i^{<j} \; x_{-i} r_{-i}] = \frac{1}{|\rho(x_i, r, \overleftrightarrow{\pi}_i^{<j}, x_{-i}, r_{-i})|}.$$

We prove that the function $\rho(x_i, r, \overleftrightarrow{\pi}_i^{<j}, \cdot, \cdot)$ is constant on the set $\{(x'_{-i}, r'_{-i})) \mid \Pr[r'_{-i} \mid x_i \; r \; \overleftrightarrow{\pi}_i^{<j} \; x'_{-i}] \neq 0\}$. Let r_{-i} such that $\Pr[r_{-i} \mid x_i \, r \; \overleftrightarrow{\pi}_i^{<j} \; x_{-i}] \neq 0$ and (x'_{-i}, r'_{-i}) such that $\Pr[r'_{-i} \mid x_i \, r \; \overleftrightarrow{\pi}_i^{<j} \; x'_{-i}] \neq 0$. Let $r_i \in \rho(x_i, r \overleftrightarrow{\pi}_i^{<j}, x_{-i}, r_{-i})$ and $r'_i \in \rho(x_i, r, \overleftrightarrow{\pi}_i^{<j}, x'_{-i}, r'_{-i})$. We get that $\overleftrightarrow{\Pi}_i^{<j}(x_i, r, x'_{-i}, r_i, r'_{-i}) = \overleftrightarrow{\pi}_i^{<j}$: this comes from the fact that $\overleftrightarrow{\Pi}_i^{<j}(x_i, r, x_{-i}, r_i, r_{-i}) = \overleftrightarrow{\pi}_i^{<j} = \overleftrightarrow{\Pi}_i^{<j}(x_i, r, x'_{-i}, r'_i, r'_{-i})$ and from the fact that the actions of the players are based on their current view (this claim, whose formal proof is complicated, will be proved rigorously in Chapter 3, cf. Lemma 3.3.2). Thus $r_i \in \rho(x_i, r, \overleftrightarrow{\pi}_i^{<j}, x'_{-i}, r'_{-i})$.
For the same reason, $r'_i \in \rho(x_i, r, \overleftrightarrow{\pi}_i^{<j}, x_{-i}, r_{-i})$.

We have shown that $\rho(x_i, r, \overleftrightarrow{\pi}_i^{<j}, x_{-i}, r_{-i}) = \rho(x_i, r, \overleftrightarrow{\pi}_i^{<j}, x'_{-i}, r'_{-i})$.

Let $\alpha(x_i, r, \overleftrightarrow{\pi}^{<j})$ be the constant value of the function $|\rho(x_i, r, \overleftrightarrow{\pi}_i^{<j}, \cdot, \cdot)|$.

$$\begin{aligned}
\Pr[r_i \mid x_i \; r \; \overleftrightarrow{\pi}_i^{<j} \; x_{-i}] &= \sum_{r_{-i}} \Pr[r_i \mid x_i \; r \; \overleftrightarrow{\pi}_i^{<j} \; x_{-i} \; r_{-i}] \Pr[r_{-i} \mid x_i \; r \; \overleftrightarrow{\pi}_i^{<j} \; x_{-i}] \\
&= \sum_{r_{-i}} \frac{1}{|\rho(x_i, r, \overleftrightarrow{\pi}_i^{<j}, x_{-i}, r_{-i})|} \Pr[r_{-i} \mid x_i \; r \; \overleftrightarrow{\pi}_i^{<j} \; x_{-i}] \\
&= \sum_{r_{-i}} \frac{1}{\alpha(x_i, r, \overleftrightarrow{\pi}^{<j})} \Pr[r_{-i} \mid x_i \; r \; \overleftrightarrow{\pi}_i^{<j} \; x_{-i}] \\
&= \frac{1}{\alpha(x_i, r, \overleftrightarrow{\pi}^{<j})}
\end{aligned}$$

which is independent of x_{-i}. Thus $\Pr[r_i \mid x_i \; r \; \overleftrightarrow{\pi}_i^{<j} \; x_{-i}] = \Pr[r_i \mid x_i \; r \; \overleftrightarrow{\pi}_i^{<j}]$. We have shown that the random variables X_{-i} and R_i are independent conditioned on $(X_i, R^p, \overleftrightarrow{\Pi}_i^{<j})$, and thus $I(X_{-i}; R_i \mid X_i R^p \overleftrightarrow{\Pi}_i^{<j}) = 0$. By Lemma 1.2.14, this implies

$$I(X_{-i}; B_j \mid X_i R^p R_i \overleftrightarrow{\Pi}_i^{<j}) \geq I(X_{-i}; B_j \mid X_i R^p \overleftrightarrow{\Pi}_i^{<j}).$$

A similar reasoning, along with Lemma 1.2.13, leads to

$$I(X_{-i}; B_j \mid X_i R^p R_i \overleftrightarrow{\Pi}_i^{<j}) \leq I(X_{-i}; B_j \mid X_i R^p \overleftrightarrow{\Pi}_i^{<j}).$$

We have:

$$\begin{aligned} I(X_{-i}; \overleftrightarrow{\Pi}_i \mid X_i R^p R_i) &= \sum_j I(X_{-i}; B_j \mid X_i R_i R^p \overleftrightarrow{\Pi}_i^{<j}) \\ &= \sum_j I(X_{-i}; B_j \mid X_i R^p \overleftrightarrow{\Pi}_i^{<j}) \\ &= I(X_{-i}; \overleftrightarrow{\Pi}_i \mid X_i R^p) \\ &\quad \text{(using the chain rule, Proposition 1.2.9).} \end{aligned}$$

⌟

As we will see in details in Subsection 2.4.3, IC is not an interesting quantity in multi-party computation, as there always exists a protocol with low information cost. For this reason, we cannot use IC to study other interesting notions such as the *communication complexity*.

1.4.3 Communication complexity

Definitions of communication complexity in the multi-party case are identical to the two-party case (cf. Subsection 1.3.2). With our usual notations, $\mathsf{CC}(\pi)$ is equal to the maximal length of the transcript of π over all possible inputs, private randomness and public randomness.

We can also consider the *distributional* error.

Definition 1.4.10. *Given an input distribution μ, a protocol is said to compute a function with distributional error ϵ if the probability over the input and the randomness of the protocol that the protocol fails is at most ϵ. The distributional communication complexity $\mathsf{D}^{\epsilon}_{\mu}(f)$ of a function f is the communication complexity of the best protocol computing f with distributional error ϵ.*

Communication complexity and distributional communication complexity are related by the following lemma.

Lemma 1.4.11 (Yao's minimax lemma)**.** *For any function f,*

$$\mathsf{CC}^{\epsilon}(f) = \sup_{\mu} \mathsf{D}^{\epsilon}_{\mu}(f).$$

We will also occasionally need the two following quantities.

Definition 1.4.12. *The average-case communication complexity of a protocol π with respect to the input distribution μ, denoted $\mathsf{AvCC}_{\mu}(\pi)$, is the expected number of bits that are transmitted in an execution of π for inputs distributed according to μ and uniform randomness.*

Definition 1.4.13. *The ϵ-error amortized communication complexity of a function f is defined as $\mathsf{AmCC}^{\epsilon}_{\mu}(f) = \lim_{n\to\infty} \frac{\mathsf{CC}(f^{\otimes n})}{n}$, where $f^{\otimes n}$ denotes the task of computing f with error ϵ for each coordinate.*

In many practical settings, as communication can be considered as a costly resource, communication complexity is a natural cost measure. When looking at a specific function, one is often interested in designing an optimal protocol in terms of communication, that is to say a protocol where the number of bits exchanged is minimal. We want to be able to compute the communication complexity of a given function. While one can prove upper bounds on the communication complexity of a function by exhibiting a protocol computing this function, proving lower bounds can be more challenging and often requires the use of specific techniques.

The results from the field of communication complexity have many applications. They range from applied fields like network protocols and VLSI chips design to theoretical results in circuit complexity. More details on the applications of communication complexity can be found in [KN97] and the references therein. Most of the work realized on multi-party communication complexity focuses on the number-on-the-forehead model. Some of the techniques presented in Subsection 1.3.2, which were developed in the study of two-party protocols, have actually been generalized to the number-on-the-forehead model. In contrast, few lower bound techniques are available in the number-in-hand model.

In the coordinator model, a technique called *symmetrization* was introduced in [PVZ12], and it was shown how to use it to study functions such as the bit-wise parity and AND functions. Reductions were also used in [WZ11, WZ14, WZ13] to prove lower bounds on communication.

Information theory has been used to prove lower bounds on communication of the disjointness function in the broadcast model [BYJKS02, CKS03, Gro09, BO15]. Information complexity was then used in [BEO$^+$13] to prove a lower bound for the disjointness function in the coordinator model. This result was extended in [CM15] to the function *Tribes*.

We will see in the next two chapters that information theory can also be used to obtain lower bounds on the communication complexity in the peer-to-peer model.

1.4.4 Comparison with other models

We provide here a comparison between our model of communication and the other similar models which have been used in the literature.

The model of communication that we defined is a restriction of the general asynchronous model in the sense that the players have in our model a local round structure, while in the general asynchronous model the players can react upon reception of a message, regardless of its origin. The local round structure that we impose allows us to define measures similar to the information complexity that we will show to have desirable properties and to be of use. Notice that the general asynchronous model is problematic in this respect since one bit of communication can bring up to $\log(k)$ bits of information, as not only the content of the message but also the identity of the sender may reveal information. Thus, in the general asynchronous model, information is not a lower bound on communication.

The following protocol is an example of the above mentioned issue in the general asynchronous model. Given 4 players A, B, C and D, it allows A to transmit its input bit x to B, in such a way that the content of the messages transiting through every communication link is independent of x.

A: If $x = 0$ send 0 to C; after receiving 0 from C, send 0 to D.
If $x = 1$ send 0 to D; after receiving 0 from D, send 0 to C

B: After receiving 0 from a player, send 0 back to that player.

C,D: After receiving 0 from A send 0 to B. After receiving 0 from B send 0 to A.

One can see that B learns the value of x from the order of the messages it gets.

In our case, the sets $S_i^{l,s}$ and $S_i^{l,r}$ are determined by the current view of the player, (Π_i) contains only the content of the messages, and thus the desirable relation between the communication and the information is maintained. The local round structure prevents protocols from using the classic network coding routines. On the other hand, this restriction is natural, does not seem to be very restrictive (practically all protocols in the literature adhere to our model), and does not exclude the existence of private protocols, as the private protocols described in [BOGW88, CCD88] (and further work) are defined in the synchronous setting, and thus can be adapted to our communication model (the sets $S_i^{l,s}$ and $S_i^{l,r}$ always consist of all the players and hence are even independent of the current views).

There has been a long series of works about multi-party communication protocols in different models, for example [CKS03, Gro09, Jay09, PVZ12, CRR14, CR15]. Several papers consider a restricted class of protocols working in the *coordinator model*: an additional player with no input can communicate privately with each player, and the players can only communicate with the coordinator.

We first note that the coordinator model does not yield exact bounds for the multi-party communication complexity in the peer-to-peer model (neither in our model nor in the general one). Namely, a protocol in the peer-to-peer model can be transformed into a protocol in the coordinator model with an $O(\log k)$ multiplicative factor in the communication complexity, by sending any message to the coordinator with a $O(\log k)$-bit label indicating its destination (cf. also [PVZ12, EOPV13]). This factor is sometimes necessary, e.g. for the q-`index` function, where player i, $0 \leq i \leq k-1$, holds an input bit x_i, player k holds q indices $0 \leq j_\ell \leq k-1$, $1 \leq \ell \leq q$, and player k should learn the vector $(x_{j_1}, \ldots, x_{j_q})$: in the coordinator model the communication complexity of this function is $\Theta(\min\{k, q \log k\})$, while in both peer-to-peer models there is a protocol for this function that sends only (at most) $\min\{k, 2q\}$ bits, where player k just queries the appropriate other players. Another example is the `permutation` functional defined as follows: Given a permutation $\sigma : [\![1,k]\!] \rightarrow [\![1,k]\!]$, each player i has for input a bit b_i and $\sigma^{-1}(\sigma(i)-1)$ and $\sigma^{-1}(\sigma(i)+1)$ (i.e. each player has for input the indexes of the players before and after itself in the permutation).[1] For player i the function f_i is defined as $f_i = b_{\sigma^{-1}(\sigma(i)+1)}$ (i.e. the value of the input bit of the next player in the permutation σ). The natural protocol is valid in our model, and the communication complexity of this function in our model is k (each player sends its input bit to the right player). On the other hand, in the coordinator model $\Omega(k \log k)$ bits of communication are necessary. But this multiplicative factor between the complexities in the two models is not always necessary: the communication complexity of the parity function is $\Theta(k)$ both in the peer-to-peer models and in the coordinator model.

Moreover, when studying private protocols in the peer-to-peer model, the coordinator model does not offer any insight. In the (asynchronous) coordinator model, described in [DF89] and used for instance in [BEO$^+$13], if there is no privacy requirement with respect to the coordinator, it is trivial to build a private protocol by having all the players send their input to the coordinator, and the coordinator return the results to the players. If there is a privacy requirement with respect to the coordinator, then if there is a random source shared by all the players (but not the coordinator), privacy is always possible using the protocol of [FKN94]. If no such source exists, privacy is impossible in general. This follows from the results of Braverman et al. [BEO$^+$13] who showed a lower bound on the total internal information complexity of all parties (including the coordinator) for the disjointness function in that model. By contrast, our model allows for private protocols.

[1] All additions are modulo k. This is a promise problem.

Chapter 2

Public Information Cost

2.1 Definition and properties

We now introduce a new information-theoretic quantity which can be used instead of IC in the multi-party setting. It will also allow us to study the *randomness complexity* of distributed problems (Section 2.4).

Definition 2.1.1. *For any k-player protocol π and any input distribution μ, we define the public information cost of π:*

$$\mathsf{PIC}_\mu(\pi) = \sum_{i=1}^{k} I(X_{-i}; \Pi_i R_{-i} \mid X_i R_i R^p).$$

The difference between PIC and IC is the presence of the other parties private coins, R_{-i}, in the formula. If π is a protocol using only public randomness, then for any input distribution μ, $\mathsf{PIC}_\mu(\pi) = \mathsf{IC}_\mu(\pi)$, hence the name public information cost. The role of private coins in communication protocols has been studied for example in [BG14, BBK$^+$13, Koz15].

The public information cost measures both the information about the inputs learned by the players and the information that is hidden by the use of private coins. It can be decomposed, using the chain rule, into two terms, making explicit the contribution of the internal information cost and of the private randomness of the players.

Proposition 2.1.2. *For any k-player protocol π and any input distribution μ,*

$$\mathsf{PIC}_\mu(\pi) = \mathsf{IC}_\mu(\pi) + \sum_{i=1}^{k} I(R_{-i}; X_{-i} | X_i \Pi_i R_i R^p).$$

The meaning of the second term is the following. At the end of the protocol, player i knows its input X_i, its private coins R_i, the public coins R^p and its transcript Π_i. Suppose that the private randomness R_{-i} of the other players is now revealed to player i. This brings to it some new information $I(R_{-i}; X_{-i}|X_i\Pi_i R_i R^p)$ about the inputs X_{-i} of the other players.

We also define the public information complexity of a function.

Definition 2.1.3. *For any function f and any input distribution μ, we define the quantity*

$$\mathsf{PIC}_\mu(f) = \inf_{\pi \text{ computing } f} \mathsf{PIC}_\mu(\pi).$$

Definition 2.1.4. *For any f, we define the quantity*

$$\mathsf{PIC}(f) = \inf_{\pi \text{ computing } f} \sup_\mu \mathsf{PIC}_\mu(\pi).$$

In fact, as we show below, the public information cost of a function is equal to its internal information cost in a setting where only public randomness is allowed.

Theorem 2.1.5. *For any function f and input distribution μ,*

$$\mathsf{PIC}_\mu(f) = \inf_{\pi \text{ computing } f, \text{ using only public coins}} \mathsf{IC}_\mu(\pi)$$

and

$$\mathsf{PIC}(f) = \inf_{\pi \text{ computing } f, \text{ using only public coins}} \sup_\mu \mathsf{IC}_\mu(\pi).$$

Proof. It suffices to show than one can turn any protocol π into a public-coins protocol π' such that for all input distributions μ, $\mathsf{PIC}_\mu(\pi') = \mathsf{PIC}_\mu(\pi)$. Fix a protocol π and an input distribution μ. Let R_i denote the private randomness of player i in π, and $R = (R_i)$. Define π', where the players act as they do in π, but use public randomness instead of their private randomness whenever they need to use it. For this, split the public random tape into two sub-tapes $R = (R_i)$, to be used instead of the private randomness of each player, and R^p, to be used as public randomness. We have

$$\begin{aligned}\mathsf{PIC}_\mu(\pi') &= \sum_{i=1}^{k} I(X_{-i};\Pi_i \mid X_i R R^p)\\ &= \sum_{i=1}^{k}[I(X_{-i};\Pi_i R_{-i} \mid X_i R_i R^p) - I(X_{-i};R_{-i} \mid X_i R_i R^p)]\\ &\quad\text{(chain rule, Proposition 1.2.9)}\\ &= \sum_{i=1}^{k} I(X_{-i};\Pi_i R_{-i} \mid X_i R_i R^p)\\ &= \mathsf{PIC}_\mu(\pi).\end{aligned}$$

⌟

The following property of the public information cost will be useful for zero-error protocols.

Theorem 2.1.6. *For any function f, for any input distribution μ,*

$$\mathsf{PIC}^0_\mu(f) = \mathsf{IC}^{\mathsf{det}}_\mu(f)$$

where $\mathsf{IC}^{\mathsf{det}}_\mu(f) = \inf_{\pi \text{ deterministic protocol computing } f} \mathsf{IC}_\mu(\pi)$.

Proof. Let $\delta > 0$. Let π be a zero-error protocol for f such that $\mathsf{PIC}_\mu(\pi) \le \mathsf{PIC}^0_\mu(f) + \frac{\delta}{2}$. By Theorem 2.1.5, one can assume that π has no private randomness.

$$\begin{aligned}\mathsf{PIC}_\mu(\pi) &= \sum_{i=1}^{k} I(X_{-i};\Pi_i \mid X_i R^p)\\ &= \sum_{i=1}^{k} \mathop{\mathbb{E}}_r [I(X_{-i};\Pi_i \mid X_i, R^p = r)]\\ &= \mathop{\mathbb{E}}_r \left[\sum_{i=1}^{k} I(X_{-i};\Pi_i \mid X_i, R^p = r)\right].\end{aligned}$$

Letting $t(r) = \sum_{i=1}^{k} I(X_{-i};\Pi_i \mid X_i, R^p = r)$, it holds $\mathsf{PIC}_\mu(\pi) = \mathop{\mathbb{E}}_r [t(r)]$. Let r_0 be a value of the public random tape such that $t(r_0) \le \mathsf{PIC}_\mu(\pi) + \frac{\delta}{2}$ and define π^0 as the protocol behaving like π on the random tape r_0. Note that π^0 is a deterministic zero-error protocol computing f.

$$\begin{aligned}
\mathsf{IC}_\mu(\pi^0) &= \sum_{i=1}^{k} I(X_{-i}; \Pi_i^0 \mid X_i)\\
&= \sum_{i=1}^{k} I(X_{-i}; \Pi_i \mid X_i R = r_0)\\
&= t(r_0)\\
&\leq \mathsf{PIC}_\mu(\pi) + \frac{\delta}{2}\\
&\leq \mathsf{PIC}_\mu(f) + \delta.
\end{aligned}$$

δ being arbitrary, this concludes the proof.

⌟

We also define an external public information cost.

Definition 2.1.7. *For any k-player protocol π and any input distribution μ, we define the external public information cost of π:*

$$\mathsf{PIC}_\mu^{\mathsf{ext}}(\pi) = \sum_{i=1}^{k} I(X_i; \overleftrightarrow{\Pi_i} R_i R^p).$$

The following theorem makes the link between public information cost and external public information cost. Intuitively, it means that PIC at least takes into account the information that each player leaks about his input to someone who has access to all the messages that involve this player. It can be seen as a multi-party equivalent of proposition 1.3.4.

Theorem 2.1.8. *For any oblivious protocol π, for any input product distribution μ,*

$$\mathsf{PIC}_\mu(\pi) \geq \mathsf{PIC}_\mu^{\mathsf{ext}}(\pi).$$

Proof. Let π be an oblivious k-party communication protocol, and let μ be an input product distribution. We first assume that π is deterministic. We thus have $\mathsf{PIC}_\mu^{\mathsf{ext}}(\pi) = \sum\limits_{i=1}^{k} I(X_i; \overleftrightarrow{\Pi_i})$. Note that using the chain rule (Proposition 1.2.9), we have for all $i \in [\![1, k]\!]$,

$$\begin{aligned}
I(X_i; \overleftrightarrow{\Pi_i}) &= \sum_{l} I(X_i; \overrightarrow{T_i^{l}} \mid \overrightarrow{T_i^{<l}}) + \sum_{l} I(X_i; \overleftarrow{T_i^{l}} \mid \overleftarrow{T_i^{<l}})\\
&= \sum_{l} I(X_i; \overrightarrow{T_i^{l}} \mid \overrightarrow{T_i^{<l}}),
\end{aligned}$$

where we used the fact that every term of the second sum is 0, as for any l, conditioned on $\overleftarrow{T_i^{<l}}$, X_i is independent of the variable $\overleftarrow{T_i^l}$ (Proposition 1.2.7).

Starting from the definition of PIC and using the chain rule, we can decompose it as a sum over all messages received in the protocol:

$$\mathsf{PIC}_\mu(\pi) = \sum_i \sum_l I(X_{-i}; \overleftarrow{T_i^l} \mid \overleftarrow{T_i^{<l}} X_i).$$

We rearrange the sum by considering the messages from the point of view of the sender rather than the receiver. Remember that j is a short notation for $j(i,l)$, and l' is a short notation for $l'(i,l)$.

$$\mathsf{PIC}_\mu(\pi) = \sum_i \sum_l I(X_{-j}; \overrightarrow{T_i^l} \mid \overleftarrow{T_j^{<l'}} X_j).$$

We will show that for any message $\overrightarrow{T_i^l}$,

$$I(X_{-j}; \overrightarrow{T_i^l} \mid \overleftarrow{T_j^{<l'}} X_j) \geq I(X_i; \overrightarrow{T_i^l} \mid \overrightarrow{T_i^{<l}}).$$

As $\overrightarrow{T_i^l}$ is determined by X_i and $\overrightarrow{T_i^{<l}}$, $H(\overrightarrow{T_i^l} \mid X_i \overrightarrow{T_i^{<l}}) = 0$,
and we have $I(X_i; \overrightarrow{T_i^l} \mid \overrightarrow{T_i^{<l}}) = H(\overrightarrow{T_i^l} \mid \overrightarrow{T_i^{<l}})$, and similarly
$I(X_{-j}; \overrightarrow{T_i^l} \mid \overleftarrow{T_j^{<l'}} X_j) = H(\overrightarrow{T_i^l} \mid \overleftarrow{T_j^{<l'}} X_j)$. Thus

$$\begin{aligned} I(X_i; \overrightarrow{T_i^l} \mid \overrightarrow{T_i^{<l}}) \leq I(X_{-j}; \overrightarrow{T_i^l} \mid \overleftarrow{T_j^{<l'}} X_j) &\iff H(\overrightarrow{T_i^l} \mid \overrightarrow{T_i^{<l}}) \leq H(\overrightarrow{T_i^l} \mid \overleftarrow{T_j^{<l'}} X_j) \\ &\iff I(\overrightarrow{T_i^l}; \overrightarrow{T_i^{<l}}) \geq I(\overrightarrow{T_i^l}; \overleftarrow{T_j^{<l'}} X_j). \end{aligned}$$

We show that $I(\overrightarrow{T_i^l}; \overrightarrow{T_i^{<l}}) = I(\overrightarrow{T_i^l}; \overrightarrow{T_i^{<l}} \overleftarrow{T_j^{<l'}} X_j)$, which implies that the last inequality is true. For this, using the chain rule we just need to show that $I(\overrightarrow{T_i^l}; \overleftarrow{T_j^{<l'}} X_j \mid \overrightarrow{T_i^{<l}}) = 0$. Notice that given the value of $\overrightarrow{T_i^{<l}}$, $\overrightarrow{T_i^l}$ is determined by X_i and thus by the data processing inequality (Proposition 1.2.12) $I(X_i; \overleftarrow{T_j^{<l'}} X_j \mid \overrightarrow{T_i^{<l}}) \geq I(\overrightarrow{T_i^l}; \overleftarrow{T_j^{<l'}} X_j \mid \overrightarrow{T_i^{<l}})$, and so we just have to show that $I(X_i; \overleftarrow{T_j^{<l'}} X_j \mid \overrightarrow{T_i^{<l}}) = 0$, which we now do.

Note that $(\overleftarrow{T_j^{<l'}}, X_j)$ is a function of $(X_{-i}, \overrightarrow{T_i^{<l}})$. The data processing inequality implies that $I(X_i; \overleftarrow{T_j^{<l'}} X_j \mid \overrightarrow{T_i^{<l}}) \leq I(X_i; X_{-i} \overrightarrow{T_i^{<l}} \mid \overrightarrow{T_i^{<l}})$ and thus $I(X_i; \overleftarrow{T_j^{<l'}} X_j \mid \overrightarrow{T_i^{<l}}) \leq I(X_i; X_{-i} \mid \overrightarrow{T_i^{<l}})$. We show that $I(X_i; X_{-i} \mid \overrightarrow{T_i^{<l}}) = 0$.

$$\begin{aligned} I(X_i; X_{-i} \mid \overrightarrow{T_i^{<l}}) &= H(X_i \mid \overrightarrow{T_i^{<l}}) - H(X_i \mid X_{-i} \overrightarrow{T_i^{<l}}) \\ &= -H(X_i) + H(X_i \mid \overrightarrow{T_i^{<l}}) + H(X_i \mid X_{-i}) - H(X_i \mid X_{-i} \overrightarrow{T_i^{<l}}) \\ &\quad \text{(as } X_i, X_{-i} \text{ are independent)} \\ &= -I(X_i; \overrightarrow{T_i^{<l}}) + I(X_i; \overrightarrow{T_i^{<l}} \mid X_{-i}) \end{aligned}$$

Thus we just need to prove that $I(X_i; T_i^{\overrightarrow{<l}} \mid X_{-i}) \leq I(X_i; T_i^{\overrightarrow{<l}})$. From now on the proof is similar to the proof that the internal IC of a protocol is lower than its external IC (cf. [Bra12]).

Let us write $T_i^{\overrightarrow{<l}} = M_i^1 \dots M_i^t$, and let $M_i^{<p} = M_i^1 \dots M_i^{p-1}$. Using the chain rule, $I(X_i; T_i^{\overrightarrow{<l}} \mid X_{-i}) \leq I(X_i; T_i^{\overrightarrow{<l}})$ if and only if

$$\sum_{p=1}^{t} I(X_i; M_i^p \mid X_{-i} M_i^{<p}) \leq \sum_{p=1}^{t} I(X_i; M_i^p \mid M_i^{<p}).$$

We prove the inequality term-wise, for any p. If M_i^p is received by player i, as M_i^p is a function of $(X_{-i}, M_i^{<p})$, $I(X_i; M_i^p \mid X_{-i} M_i^{<p}) = 0$ and the inequality holds. Similarly, if M_i^p is sent by player i, $I(X_{-i}; M_i^p \mid X_i M_i^{<p}) = 0$ and applying Lemma 1.2.13, we get that

$$I(X_i; M_i^p \mid X_{-i} M_i^{<p}) \leq I(X_i; M_i^p \mid M_i^{<p}).$$

We have shown so far that for any message $T_i^{\overrightarrow{l}}$,

$$I(X_{-j}; T_i^{\overrightarrow{l}} \mid T_j^{\overleftarrow{<l'}} X_j) \geq I(X_i; T_i^{\overrightarrow{l}} \mid T_i^{\overrightarrow{<l}}).$$

Summing over i and l, we get

$$\mathsf{PIC}_\mu(\pi) \geq \sum_{i=1}^{k} \sum_{l} I(X_i; T_i^{\overrightarrow{l}} \mid T_i^{\overrightarrow{<l}})$$

and thus

$$\mathsf{PIC}_\mu(\pi) \geq \sum_{i} I(X_i; \overleftrightarrow{\Pi_i}) = \mathsf{PIC}_\mu^{\mathsf{ext}}(\pi).$$

Let us now consider the case where π is a randomized protocol. Denote by π^r the deterministic protocol built from π by fixing all the randomness

(R, R^p) of the protocol to the value r.

$$\begin{aligned}
\mathsf{PIC}_\mu(\pi) &= \sum_{i=1}^{k} I(X_{-i}; \Pi_i R_{-i} \mid X_i R_i R^p) \\
&= \sum_{i=1}^{k} H(X_{-i}; \mid X_i R_i R^p) - H(X_{-i} \mid X_i R_i R^p \Pi_i R_{-i}) \\
&= \sum_{i=1}^{k} H(X_{-i} \mid X_i) - H(X_{-i} \mid X_i R_i R^p \Pi_i R_{-i}) \\
&= \sum_{i=1}^{k} \mathop{\mathbb{E}}_{r}[H(X_{-i} \mid X_i) - H(X_{-i} \mid X_i \Pi_i, (R_{-i} R_i R^p) = r)] \\
&= \mathop{\mathbb{E}}_{r}\left[\sum_{i=1}^{k} (H(X_{-i} \mid X_i) - H(X_{-i} \mid X_i \Pi_i, (R_{-i} R_i R^p) = r))\right] \\
&= \mathop{\mathbb{E}}_{r}\left[\sum_{i=1}^{k} I(X_{-i}; \Pi_i^r \mid X_i)\right] \\
&= \mathop{\mathbb{E}}_{r}[\mathsf{PIC}_\mu(\pi^r)]
\end{aligned}$$

and

$$\begin{aligned}
\mathsf{PIC}_\mu^{\mathsf{ext}}(\pi) &= \sum_{i=1}^{k} I(X_i; \overleftrightarrow{\Pi}_i R_i R^p) \\
&\leq \sum_{i=1}^{k} I(X_i; \overleftrightarrow{\Pi}_i R_{-i} R_i R^p) \\
&\leq \sum_{i=1}^{k} \Big(H(X_i) - H(X_i \mid \overleftrightarrow{\Pi}_i R_{-i} R_i R^p)\Big) \\
&\leq \sum_{i=1}^{k} \Big(H(X_i) - \mathop{\mathbb{E}}_{r}[H(X_i \mid \overleftrightarrow{\Pi}_i, (R_{-i} R_i R^p) = r)]\Big) \\
&\leq \mathop{\mathbb{E}}_{r}\left[\sum_{i=1}^{k} \Big(H(X_i) - H(X_i \mid \overleftrightarrow{\Pi}_i, (R_{-i} R_i R^p) = r)\Big)\right] \\
&\leq \mathop{\mathbb{E}}_{r}\left[\sum_{i=1}^{k} \Big(I(X_i; \overleftrightarrow{\Pi_i^r}))\Big)\right] \\
&\leq \mathop{\mathbb{E}}_{r}\left[\mathsf{PIC}_\mu^{\mathsf{ext}}(\pi^r)\right].
\end{aligned}$$

π^r being a deterministic protocol, we have for any r, $\mathsf{PIC}_\mu(\pi^r) \geq \mathsf{PIC}^{\mathsf{ext}}_\mu(\pi^r)$, and taking the expectation over r,

$$\mathsf{PIC}_\mu(\pi) \geq \mathsf{PIC}^{\mathsf{ext}}_\mu(\pi).$$

⌟

2.2 The two-party case

We prove that the zero-error public information cost of the two-party function AND is $\log(3) \simeq 1.58$, while, as shown in [BGPW13], $\mathsf{IC}^0(\mathsf{AND}) \simeq 1.49$. This shows that in the zero-error case, even in the two party setting IC and PIC can be different. This implies that the protocol that achieves the optimal information cost for AND must use private coins. We remark also that in [BGPW13] it is shown that the external information cost of AND, that we do not consider here, is $\log(3)$. It is not clear whether there exists a general relation between zero-error public information cost and external information cost.

Proposition 2.2.1. *For two players,* $\mathsf{PIC}^0(\mathsf{AND}) = \log(3) \simeq 1.58$.

Proof. We first prove that there exists a protocol π^* for AND such that $\sup_\mu \mathsf{PIC}_\mu(\pi^*) = \inf_\pi \sup_\mu \mathsf{PIC}_\mu(\pi)$, where the infimum is over all protocols π computing AND. To this end we will prove that for any protocol π for AND it holds that $\sup_\mu \mathsf{PIC}_\mu(\pi^*) \leq \sup_\mu \mathsf{PIC}_\mu(\pi)$, where π^* is a protocol for AND that we define below.

Let π be a zero-error protocol for AND. By Proposition 2.1.6, we can assume that π is deterministic. Suppose for example that the first player sending a nonconstant bit is player 1. The protocol π being deterministic, player 1 is either sending his bit x or sending $1 - x$. Player 2 is then able to compute the value of AND, and as player 1 must be able to compute AND at the end of the protocol, the optimal protocol consists in player 2 sending back the value of AND to player 1. We thus define the protocol π^* defined as follows: player 1 sends its input bit x to player 2, who can now compute $\mathsf{AND}(X, Y)$ and sends to player 1 this value.

We compute the value of $\mathsf{PIC}_\mu(\pi^*)$, for μ defined as follows: $X \sim \mathbf{Ber}(\alpha, 1-\alpha)$ and $Y \sim \mathbf{Ber}(\beta, 1-\beta)$.

$$\begin{aligned}\mathsf{PIC}_\mu(\pi^*) &= I(X; \Pi^* | Y) + I(Y; \Pi^* | X) \\ &= H(X \mid Y) + \left[\alpha I(Y; \Pi^* | X = 0) + (1-\alpha) I(Y; \Pi^* | X = 1)\right].\end{aligned}$$

When $X = 0$, player 1 does not learn anything about Y, while when $X = 1$, player 1 learns the value of Y, as $\mathsf{AND}(X, Y) = Y$. Thus

$$\mathsf{PIC}_\mu(\pi^*) = H(X \mid Y) + (1 - \alpha)H(Y \mid X = 1).$$

Define X', having the same probability distribution as X, and Y', having the same probability distribution as "Y conditioned on $X = 1$", with X' and Y' independent. Note that $H(X' \mid Y') = H(X') = H(X) \geq H(X \mid Y)$ and that $H(Y' \mid X' = 1) = H(Y') = H(Y \mid X = 1)$. Thus, in order to maximize $\mathsf{PIC}_\mu(\pi^*)$, we can assume that X and Y are independent. Then,

$$\mathsf{PIC}_\mu(\pi^*) = H(X) + (1 - \alpha)H(Y).$$

For any α, $\mathsf{PIC}_\mu(\pi*)$ is maximal for $\beta = \frac{1}{2}$ and we then have $\mathsf{PIC}_\mu(\pi^*) = H(X) + (1 - \alpha)$. Thus we study the function

$f : [0, 1] \to \mathbb{R}$

$\alpha \mapsto -\alpha \log(\alpha) + (\alpha - 1) \log(1 - \alpha) + 1 - \alpha.$

f is continuous on $[0, 1]$ and differentiable on $]0, 1[$. For $\alpha \in]0, 1[$, we have: $f'(\alpha) = -\log(\alpha) - 1 + \log(1 - \alpha) + 1 - 1 = \log(\frac{1}{\alpha} - 1) - 1.$

f' is continuous and decreasing on $]0, 1[$, and admits the unique root $\frac{1}{3}$. $\mathsf{PIC}_\mu(\pi^*)$ is thus maximized for $\alpha = \frac{1}{3}$, its value being $f(\alpha) = \log(3)$.

⌟

2.3 Public information cost and communication complexity

The public information cost is a lower bound for the communication complexity.

Theorem 2.3.1. *For any protocol π and input distribution μ,*

$$\mathsf{CC}(\pi) \geq \frac{1}{2}\mathsf{PIC}_\mu(\pi) - k.$$

Thus, for any function f,

$$\mathsf{CC}(f) \geq \frac{1}{2}\mathsf{PIC}(f) - k.$$

Proof.

$$\begin{aligned}
\mathsf{PIC}_\mu(\pi) &= \sum_{i=1}^{k} I(X_{-i}; R_{-i} \mid X_i R_i R^p) + I(X_{-i}; \Pi_i \mid X_i R R^p) \\
&\quad \text{(using the chain rule, Proposition 1.2.9)} \\
&= \sum_{i=1}^{k} I(X_{-i}; \Pi_i \mid X_i R R^p) \text{ (by Proposition 1.2.7)} \\
&= \sum_{i=1}^{k} H(\Pi_i \mid X_i R R^p) \text{ (since } H(\Pi_i \mid X R R^p) = 0\text{)} \\
&\leq \sum_{i=1}^{k} H(\Pi_i) \text{ (by Proposition 1.2.2).}
\end{aligned}$$

For any i, one can encode Π_i into a variable Π'_i such that the set of possible values of Π'_i is prefix-free: replace every bit b in Π_i by bb and add 01 at the end. We have $H(\Pi'_i) = H(\Pi_i)$, and $\mathbb{E}[|\Pi'_i|] = 2\,\mathbb{E}[|\Pi_i|] + 2$. Using Theorem 1.2.3, for each i, $H(\Pi'_i)$ is upper bounded by the expected size of Π'_i. As the expected size of Π is equal to the sum over i of the expected size of Π_i, we get

$$\mathsf{CC}(\pi) \geq \mathbb{E}[|\Pi|] \geq \frac{1}{2}\mathsf{PIC}_\mu(\pi) - k.$$

⌟

This means that we are able to use the public information cost to prove lower bounds on the communication complexity of functions in the multiparty setting, just as one could use the information cost to prove lower bounds on the communication complexity of functions in the two-party setting.

In the oblivious setting, we can get a better bound.

Theorem 2.3.2. *In the oblivious setting, for any protocol π and input distribution μ,*

$$\mathsf{CC}(\pi) \geq \mathsf{PIC}_\mu(\pi).$$

Thus, for any function f,

$$\mathsf{CC}(f) \geq \mathsf{PIC}(f).$$

Proof. As in the proof of Theorem 2.3.1, we have $\mathsf{PIC}_\mu(\pi) \leq \sum_{i=1}^{k} H(\Pi_i)$. The protocol π being oblivious, for any $i \in [\![1, k]\!]$, $H(\Pi_i)$ is of fixed size, and thus prefix-free. By Theorem 1.2.3,

$$\mathsf{PIC}_\mu(\pi) \le \mathop{\mathbb{E}}_{\mu}[|\Pi|] \le \mathsf{CC}(\pi).$$

$\lrcorner$

2.4 Randomness complexity

2.4.1 Private computation

In certain circumstances, we would like that the players, while being able to compute the value of the function, retain as much privacy as possible about their input. Informally, we would like that the players learn nothing about the others' input, but the value of the function.

The question of when such a protocol is possible, and how to design it, has been posed in the field of cryptography [Yao82]. In cryptography, the notion of security is computational: we assume that the players have a limited computation power, and we want to design a protocol which ensures that the players cannot get more information than they should be able to. Constructions based on trapdoor one-way functions [GMW87, CDvdG88] answer this question.

Here we are interested in a different notion of security. Instead of relying on cryptographic assumptions, we aim for unconditionally secure protocols, i.e. information-theoretic secure. A protocol π is said to *privately* compute a given function if at the end of the execution of the protocol, the players have learned nothing but the value of that function. We note that the literature devoted to private computation usually focuses on *synchronous* protocols, which are a special case of *oblivious* protocols as defined in Subsection 1.4.1. Moreover, most of the literature on private computation deals with zero-error protocols. Therefore, in the rest of this section, we will restrict ourselves to the case of oblivious zero-error protocols. The definitions for the case of epsilon-error privacy are similar. Formally:

Definition 2.4.1. *A k-player oblivious protocol π computing a family of functions (f_i) is private if for every player $i \in [\![1,k]\!]$, for any pair of inputs $x = (x_1, \dots, x_k)$ and $x' = (x'_1, \dots, x'_k)$ such that $f_i(x) = f_i(x')$ and $x_i = x'_i$, for all possible private random tapes r_i of player i, and all possible public random tapes r^p, it holds that for any transcript T*

$$\Pr[\Pi_i = T \mid R_i = r_i \ ; \ X = x \ ; \ R^p = r^p] = \Pr[\Pi_i = T \mid R_i = r_i \ ; \ X = x' \ ; \ R^p = r^p]$$

where the probability is over the randomness R_{-i}.

It is well known that in the multi-party number-in-hand peer-to-peer setting (for $k \geq 3$), unlike in the two-party case, any function can be privately computed.

Theorem 2.4.2 ([BOGW88, CCD88])**.** *Any family of functions of more than two variables can be computed by a private protocol.*

We detail shortly the scheme of [BOGW88]. Their construction can be divided in three steps.

1. Preparing one's own input for the computation. This is similar to Shamir's secret sharing scheme [Sha79]. The idea is to encode each input into a polynomial, and to share values of this polynomial among the players. The value can then be extracted by realizing an interpolation, which a single player cannot do by himself.

2. Computing the function. The function is computed by simulating a circuit computing it. The gates of the circuit are translated to operations on the polynomials.

3. Recovering the value of the function. For this, the players share values in order to be able to interpolate the polynomial encoding the value of the function.

The works presented in [BOGW88, CCD88] actually deal with a stronger notion of privacy, that we only define in an informal way here.

Definition 2.4.3. *A protocol is said to be t-private if any set of at most t players is not able to get more information after the protocol than they had jointly from their inputs and from the value of the function.*

Theorem 2.4.4 ([BOGW88],[CCD88])**.** *Any function of $n > 2$ variables can be computed by a t-private protocol if $t \leq \dfrac{n}{3}$*

Note that privacy as we defined it earlier is equivalent to 1-privacy.

2.4.2 Randomness complexity

The private protocols presented in the previous subsection make use of the private randomness of the players. One may wonder what amount of randomness is needed in order to compute privately a given function. This leads to the notion of randomness complexity. Several definitions have been used, the ones we adopt here are the following.

Definition 2.4.5. *A communication protocol is said to be d-random if on any run the total number of private coins used by the players is at most d.*

Definition 2.4.6. *The randomness complexity $\mathcal{R}(f)$ of a function f is the minimal integer d such that there exists a d-random private protocol computing f.*

We will mainly use a finer notion:

Definition 2.4.7. *The randomness complexity of a protocol π on input distribution μ is defined as $\mathcal{R}_\mu(\pi) = H(\Pi \mid XR^p)$.*

Once the input and the public coins are fixed, the entropy of the transcript of a protocol comes solely from the private randomness. Thus for any input distribution μ, $\mathcal{R}_\mu(\pi)$ provides a lower bound on the entropy of the private randomness used by the players in the protocol π. Note that by the results presented in [KY76], this is equivalent to the average number of coin tosses needed by the protocol.

Definition 2.4.8. *The randomness complexity of a function f on input distribution μ is defined as*

$$\mathcal{R}_\mu(f) = \min_{\pi \text{ private protocol computing } f} \mathcal{R}_\mu(\pi).$$

The following lemma is immediate.

Lemma 2.4.9. *Let d be an integer. If there exists an input distribution μ such that $\mathcal{R}_\mu(f) \leq d$, then $\mathcal{R}(f) \leq d$.*

This means that in order to lower-bound the randomness complexity of a function f, we only have to find an input distribution μ such that the randomness complexity of the function f on μ is high.

The literature on private protocols only deals with the case of *synchronous* protocols, where the players communicate according to a global round structure. At every round, each player sends a message to every other player. In addition, each player has an output tape. In order to ensure that no player is ever engaged in an infinite computation process, it is required that on any input and randomness assignment, every player eventually stops sending messages. That is, for a synchronous protocol π let $t_i(x, r)$ be the smallest integer such that if π is run on (x, r) then player i does not send any message and does not write on its output tape after round $t_i(x, r)$. If no such integer exists then $t_i(x, r) = \infty$. The requirement is that for every player i, input x, and randomness assignment r, $t_i(x, r) < \infty$.

The following lemma implies that any synchronous protocol is actually an oblivious protocol. Therefore, when working on randomness complexity, we will usually focus on the case of oblivious protocols.

Lemma 2.4.10. *Let π be a synchronous protocol. If for any i, x, and r, $t_i(x,r) < \infty$, then there exists an integer t_f such that for any i, x, and r, $t_i(x,r) < t_f$.*

Proof. Let π be a synchronous protocol. The protocol π is fully described by its *state graph*. In this oriented graph, the vertices encode the state of the players, i.e. the inputs of the players, the messages received in previous rounds and the random bits read so far, and there is an edge from vertex U to vertex V if the state of vertex V can be reached from the state of vertex U in one communication round. Note that in any round, each player can only pick the messages he will send from a finite set, even though his private random tape is infinite. This is because in any state, the number of random bits that a given player can read is bounded (otherwise, there would exist a random string on which the player would never stop reading). This ensures that in the state graph of π, every vertex has finite degree. If $\nexists\, t_f \mid \forall\, i, x, r, t_i(x,r) < t_f$, the state graph of π is infinite. Then, by König's lemma (cf. [Kle02]), there must be an infinite path in the state graph of π, which corresponds to the existence of a triple (i,x,r) such that $t_i(x,r) = \infty$. ⌟

The question of how much privacy one can retain thanks to the use of private randomness also makes sense for two-party protocols, even though in this case the existence of private protocols is not guaranteed. This question is well understood in the case of bounded round protocols [BG14].

The interest of randomness complexity lies in the fact that true randomness is usually considered as a costly resource. Besides, randomness complexity is also related to other notions in theoretical computer science. A good example is [KOR96], where it was shown that a boolean function f has a linear size circuit if and only if f has constant randomness complexity.

2.4.3 Private computation and information complexity

In the previous subsection we gave the historical definitions of private protocols. Here we express the notion of privacy in terms of entropy and information.

Proposition 2.4.11. *A protocol π is private if and only if for all input distributions μ, $\sum_{i=1}^{k} I(X_{-i};\Pi_i \mid X_i R_i R^p f_i(X)) = 0$.*

Proof. By Proposition 1.2.7, Definition 2.4.1 is equivalent to the following:

$$\forall\, i, I(X_{-i};\Pi_i \mid X_i R_i R^p f_i(X)) = 0\ .$$

Since I is non-negative, this is equivalent to

$$\sum_{i=1}^{k} I(X_{-i};\Pi_i \mid X_i R_i R^p f_i(X)) = 0\ .$$

⌟

Theorem 2.4.2 and Proposition 2.4.11 lead to the following lemma.

Lemma 2.4.12. *For any family of functions $f = (f_i)_{i\in[\![1,k]\!]}$ of more than two variables and any input distribution μ, $\mathsf{IC}_\mu(f) \le \sum_{i=1}^{k} H(f_i(X))$, where X is distributed according to μ.*

Proof. Let π be a k-player private protocol computing f. Fix a distribution μ on the inputs.

$$\begin{aligned}\mathsf{IC}_\mu(\pi) &= \sum_{i=1}^{k} I(X_{-i};\Pi_i \mid X_i R_i R^p)\\ &\le \sum_{i=1}^{k} I(X_{-i};\Pi_i f_i(X) \mid X_i R_i R^p)\\ &\le \sum_{i=1}^{k} [I(X_{-i}; f_i(X) \mid X_i R_i R^p) + I(X_{-i};\Pi_i \mid X_i R_i R^p f_i(X))]\\ &\quad \text{(chain rule, Proposition 1.2.9)}\\ &\le \sum_{i=1}^{k} I(X_{-i}; f_i(X) \mid X_i R_i R^p) \quad \text{(Proposition 2.4.11)}\\ &\le \sum_{i=1}^{k} H(f_i(X)).\end{aligned}$$

Thus, $\mathsf{IC}_\mu(f) \le \mathsf{IC}_\mu(\pi) \le \sum_{i=1}^{k} H(f_i(X))$.

⌟

This lemma shows that IC is not a pertinent notion in the multi-party setting. In particular, it cannot be used to obtain any meaningful lower bound on the communication complexity, since its value is always upper-bounded by the entropy of the functions.

2.4.4 Private computation and public information cost

The public information cost, on the other hand, is not affected by the existence of private protocols. As we have seen in Section 2.1, PIC is minimized by public-coins protocol:

Theorem (2.1.5). *For any function f and input distribution μ,*

$$\mathsf{PIC}_\mu(f) = \inf_{\pi \textit{ computing } f, \textit{ using only public coins}} \mathsf{IC}_\mu(\pi)$$

and

$$\mathsf{PIC}(f) = \inf_{\pi \textit{ computing } f, \textit{ using only public coins}} \sup_\mu \mathsf{IC}_\mu(\pi).$$

This means that the public information cost of private protocols may still be high, even if they have a low information cost.

We will see now that the difference between the public information cost of a protocol and its information cost can provide a lower bound on the randomness complexity of a function.

Theorem 2.4.13. *Let $f = (f_i)$ be a family of functions of k variables. For any oblivious protocol π computing f and any input distribution μ, it holds:*

$$\mathcal{R}_\mu(\pi) \geq \frac{\mathsf{PIC}_\mu(\pi) - \mathsf{IC}_\mu(\pi)}{k-1}.$$

Thus running an oblivious protocol for f with information cost I_μ on μ requires a protocol with randomness complexity $\mathcal{R}_\mu(\pi) \geq \dfrac{\mathsf{PIC}_\mu(f) - I_\mu}{k-1}$.

Proof. Fix a protocol π computing f and a distribution μ on inputs.

We first prove that $\forall\ i, I(R_{-i}; R_i \mid X\overleftrightarrow{\Pi}_i R^p) = 0$, which we will need during the proof of the theorem.

$$\begin{aligned}
I(R_i; R_{-i} \mid X\overleftrightarrow{\Pi}_i R^p) &= H(R_i \mid X\overleftrightarrow{\Pi}_i R^p) - H(R_i \mid R_{-i}X\overleftrightarrow{\Pi}_i R^p)\\
&= H(R_i \mid X\overleftrightarrow{\Pi}_i R^p) - H(R_i \mid XR^p) + \\
&\quad H(R_i \mid XR^pR_{-i}) - H(R_i \mid R_{-i}X\overleftrightarrow{\Pi}_i R^p)\\
&\text{(as } R_i,\ R_{-i},\ R^p \text{ and } X \text{ are independent)}\\
&= -I(R_i; \overleftrightarrow{\Pi}_i \mid XR^p) + I(R_i; \overleftrightarrow{\Pi}_i \mid XR^pR_{-i}).
\end{aligned}$$

Thus we just need to prove that $I(R_i; \overleftrightarrow{\Pi}_i \mid XR^pR_{-i}) \leq I(R_i; \overleftrightarrow{\Pi}_i \mid XR^p)$. From now on the proof is similar to the proof that the internal IC of a protocol is lower than its external IC (cf. [Bra12]).

Let us write $\overleftrightarrow{\Pi}_i = \Pi_i^1 \dots \Pi_i^t$ where the Π_i^p represent the messages ordered by local round of player i, and let $\Pi_i^{<p} = \Pi_i^1 \dots \Pi_i^{p-1}$. Using the chain rule (Proposition 1.2.9),

$$I(R_i; \overleftrightarrow{\Pi}_i \mid XR^pR_{-i}) = \sum_{p=1}^{t} I(R_i; \Pi_i^p \mid XR^pR_{-i}\Pi_i^{<p})$$

and

$$I(R_i; \overleftrightarrow{\Pi}_i \mid XR^p) = \sum_{p=1}^{t} I(R_i; \Pi_i^p \mid XR^p\Pi^{<p}).$$

We prove the inequality

$$\sum_{p=1}^{t} I(R_i; \Pi_i^p \mid XR^pR_{-i}\Pi_i^{<p}) \leq \sum_{p=1}^{t} I(R_i; \Pi_i^p \mid XR^p\Pi_i^{<p})$$

term-wise, for any p. If Π_i^p is received by player i, as Π_i^p is a function of $(X, R^p, R_{-i}, \Pi_i^{<p})$, $I(R_i; \Pi_i^p \mid XR^pR_{-i}\Pi_i^{<p}) = 0$ and the inequality holds. Similarly, if Π_i^p is sent by player i, $I(R_{-i}; \Pi_i^p \mid XR^pR_i\Pi_i^{<p}) = 0$ and applying Lemma 1.2.13, we get that $I(R_i; \Pi_i^p \mid XR^pR_{-i}\Pi_i^{<p}) \leq I(R_i; \Pi_i^p \mid XR^p\Pi_i^{<p})$.

We now go back to the main proof.

$$\begin{aligned}
\mathsf{PIC}_\mu(\pi) &= \sum_{i=1}^{k} I(X_{-i}; \Pi_i R_{-i} \mid X_i R_i R^p) \\
&\leq \sum_{i=1}^{k} I(X_{-i}; \overleftrightarrow{\Pi}_i R_{-i} \mid X_i R_i R^p) \\
&\leq \sum_{i=1}^{k} I(X_{-i}; \overleftrightarrow{\Pi}_i \mid X_i R_i R^p) + \sum_{i=1}^{k} I(R_{-i}; X_{-i} \mid X_i R_i R^p \overleftrightarrow{\Pi}_i) \\
&\text{(using the chain rule, Proposition 1.2.9)} \\
&\leq \mathsf{IC}_\mu(\pi) + \sum_{i=1}^{k} I(R_{-i}; X_{-i} \mid X_i R_i R^p \overleftrightarrow{\Pi}_i) \quad \text{(Proposition 1.4.8)} \\
&\leq \mathsf{IC}_\mu(\pi) + \sum_{i=1}^{k} I(R_{-i}; X R_i \overleftrightarrow{\Pi}_i \mid R^p) \\
&\text{(using the chain rule and the fact that I is non-negative).}
\end{aligned}$$

This formulation is similar to Proposition 2.1.2. We could have got an equality if we had proved a statement similar to the one of Proposition 1.4.8 at the second line, but we do not need it here.

For any i,

$$\begin{aligned}
I(R_{-i}; XR_i\overleftrightarrow{\Pi}_i \mid R^p) &\leq I(R_{-i}; \overleftrightarrow{\Pi}_i X \mid R^p) + I(R_{-i}; R_i \mid X\overleftrightarrow{\Pi}_i R^p) \text{ (chain rule)}\\
&\leq I(R_{-i}; \overleftrightarrow{\Pi}_i X \mid R^p)\\
&\leq I(R_{-i}; \Pi X \mid R^p)\\
&\leq \sum_{j\neq i} I(R_j; \Pi X \mid R_{<j,\neq i} R^p) \text{ (chain rule)}\\
&\leq \sum_{j\neq i} I(R_j; \Pi X \mid R^p)\\
&\quad \text{(by Lemma 1.2.13 with } I(R_j; R_{<j,\neq i} \mid X\Pi R^p) = 0).
\end{aligned}$$

Thus

$$\mathsf{PIC}_\mu(\pi) \leq \mathsf{IC}_\mu(\pi) + (k-1)\sum_{i=1}^{k} I(R_i; \Pi X \mid R^p)$$

$$\begin{aligned}
\mathsf{PIC}_\mu(\pi) &\leq \mathsf{IC}_\mu(\pi) + (k-1)\sum_{i=1}^{k} I(R_i; \Pi X \mid R_{<i} R^p) \text{ (by Lemma 1.2.14)}\\
&\leq \mathsf{IC}_\mu(\pi) + (k-1) I(R; \Pi X \mid R^p) \text{ (chain rule)}\\
&\leq \mathsf{IC}_\mu(\pi) + (k-1)(I(R; X \mid R^p) + I(R; \Pi \mid XR^p)) \text{ (idem)}\\
&\leq \mathsf{IC}_\mu(\pi) + (k-1) I(R; \Pi \mid XR^p)\\
&\leq \mathsf{IC}_\mu(\pi) + (k-1) H(\Pi \mid XR^p)\\
&\leq \mathsf{IC}_\mu(\pi) + (k-1)\mathcal{R}_\mu(\pi).
\end{aligned}$$

⌟

Combining Theorem 2.4.13 and Lemma 2.4.12, we can bound the randomness required to run a private protocol.

Theorem 2.4.14. *For any $f = (f_i)$ family of functions of k variables, for any input distribution μ,*

$$\mathcal{R}_\mu(f) \geq \frac{\mathsf{PIC}_\mu(f) - \sum_{i=1}^{k} H(f_i(X))}{k-1}.$$

Using Theorem 2.4.13, we can also give a lower bound on the randomness one needs for protocols that are allowed to leak some limited amount of information about the inputs of the players.

The following theorem apply to general protocols, but gives a bound slightly weaker than Theorem 2.4.13.

Theorem 2.4.15. *Let $f = (f_i)$ be a family of functions of k variables. Let π be a protocol for f. For any input distribution μ, it holds:*

$$H_\mu(\Pi \mid XR^p) \geq \frac{\mathsf{PIC}_\mu(\pi) - \mathsf{IC}_\mu(\pi)}{k}.$$

Thus, running a protocol for f with information cost I_μ requires entropy

$$H_\mu(\Pi \mid XR^p) \geq \frac{\mathsf{PIC}_\mu(f) - I_\mu}{k}.$$

Proof. Fix a protocol π computing f and a distribution μ on inputs.

Define Q_i as

$$Q_i = I(X_{-i}; R_{-i} \mid X_i R_i \Pi_i R^p).$$

By Proposition 2.1.2 we have,

$$\mathsf{PIC}(\pi) = \mathsf{IC}(\pi) + \sum_{i=1}^{k} Q_i.$$

Now,

$$\begin{aligned} Q_i &= I(X_{-i}; R_{-i} \mid X_i R_i \Pi_i R^p) \\ &\leq I(X_{-i}\Pi_i; R_{-i} \mid X_i R_i R^p) \\ &\text{(using the chain rule, Proposition 1.2.9)} \\ &\leq I(X_{-i}; R_{-i} \mid X_i R_i R^p) + I(\Pi_i; R_{-i} \mid X_i R_i X_{-i} R^p) \quad \text{(chain rule)} \\ &\leq I(\Pi_i; R_{-i} \mid X R_i R^p) \\ &\leq H(\Pi_i \mid X R_i R^p) \\ &\leq H(\Pi \mid XR^p). \end{aligned}$$

Thus,

$$\mathsf{PIC}(\pi) \leq \mathsf{IC}(\pi) + k \cdot H(\Pi \mid XR^p).$$

⌟

2.4.5 The randomness complexity of the parity function

The parity function is the canonical problem for zero-error multi-party computation. The k-party parity problem with n-bit inputs Par_k^n is defined as follows. Each player i receives n bits $(x_i^p)_{p\in[\![1,n]\!]}$ and player 1 has to output the bitwise sum modulo 2 of the inputs:

$$\mathsf{Par}_k^n(x) = \left(\bigoplus_{i=1}^{k} x_i^1, \bigoplus_{i=1}^{k} x_i^2, \ldots, \bigoplus_{i=1}^{k} x_i^n \right).$$

There is a simple private protocol which computes Par_k^n while using n bits of private randomness: player 1 uses a private random n-bit string r and sends to player 2 the string $x_1 \oplus r$; then, player 2 computes the bit-wise parity of its input with the message and sends $x_2 \oplus x_1 \oplus r$ to player 3; the players continue this process until player 1 receives back the message $x_k \oplus \ldots \oplus x_1 \oplus r$. Player 1 then takes the bit-wise parity of this message with the private string r to compute the value of the parity function. It is easy to see that this protocol has information cost equal to n, since player 1 only learns the value of the function and all other players learn nothing. We thus see that information cost cannot provide here a lower bound that scales with k.

A good part of the work realized in private multi-party computation focuses on the parity function. In [KR98], the authors studied the relation between the number of random bits allowed and the number of rounds necessary to compute the parity function. The communication complexity of t-privately computing the parity function has been studied in [CK93]. A series of paper tried to characterize the randomness complexity of the parity function. In [BDSPV99], the authors proved lower bounds on the randomness complexity of t-private protocols for function of sensitivity n, among which Par_k^n. They got a tight bound in the case of total privacy, i.e. for $t = n - 2$, but unfortunately, their bounds are only interesting for values of t close to n. For other values of t, this was complemented by [KM97], but for the sole function Par_k^1. The work of [GR05] provided improvements for Par_k^1 for values of t between 2 and $\log(n)$, and is tight for constant t.

We will prove the following lower bound on the randomness complexity of the parity function Par_k^n in Subsection 2.5.2.

Theorem 2.4.16. *There exists an input distribution μ such that*

$$\mathcal{R}_\mu(\mathsf{Par}_k^n) \geq \frac{k-2}{k-1} n.$$

Our result appears even more interesting when you consider the fact that given n bits of private randomness, there exist several protocols privately computing Par_k^n and having good properties. For example, it was shown in [KOR98] that when the players have to compute n instances of Par sequentially (i.e. they receive one instance, compute the parity, receive the next instance, and so on), it is possible to design a protocol which, by using the random bits in the computation of several instances, compute the n instances of parity in only $\mathcal{O}(n)$ rounds. This is to compare to the naive $\Theta(nk)$ protocol using one random bit per instance.

Theorem 2.4.16 shows that one cannot amortize the cost of computing several instances of the parity function, if the cost measure is the randomness complexity. However, it should be noted that in the context of private multi-party computation, amortization is sometimes possible for other cost measures. In particular, it was shown in [FY92] that the communication complexity of t-privately computing several instances of the parity function can be amortized if t is big enough.

2.5 Lower bounds techniques for the public information cost

In this section, we present techniques which can be used to prove lower bounds on the public information cost. We will focus on oblivious protocols. Note that the private protocol we described in Subsection 2.4.5 is oblivious.

2.5.1 Linearity

There could be several meaningful ways to define the notion of *linear functions* in the context of communication protocols. It is reasonable to conjecture that for linear functions such as the parity function, the function *sum modulo d*, or the function *sum over a field K*, linear protocols, which are protocols in which messages are linear functions of the input, are optimal in term of information, at least in the zero-error setting. As it is not clear how general this conjecture should be, we will focus here on the parity function, but the definitions and proofs in this section can be adapted to a more general setting. We define formally the notions of *linear functions* and *linear protocols.*

Definition 2.5.1. *A function f of k variables is linear if*

$$\exists\ (s_i) \in \{0,1\}^k \mid f(x) = \bigoplus_{i=1}^{k} s_i x_i.$$

Definition 2.5.2. *A k-player communication protocol π, where the players have inputs $(x_i)_{i\in[\![1,k]\!]}$, is linear if it is oblivious and if any message sent in the protocol is a xor-combination of inputs. Equivalently, π is linear if it is oblivious and if any message sent by player i is a xor-combination of x_i and messages previously received by player i.*

It is very natural to believe that linear protocols are optimal for the parity function. If we could prove this conjecture, we could restrict our study to linear protocols, which by nature lead to elegant proofs. Theorem 2.5.7 can be seen as a proof of this conjecture in the oblivious zero-error setting.

We first prove the following general lemma.

Lemma 2.5.3. *Let $X_1, \dots, X_k$ be uniformly distributed random variables in $\{0,1\}$. Let $\mathcal{S} = \{S_1 = \alpha_1, \dots, S_t = \alpha_t\}$ be a set of xor-equations on (X_i), i.e. for each $l \in [\![1,t]\!]$, $\alpha_l \in \{0,1\}$ and S_l is of the form $\bigoplus_{i=1}^{k} m_i^l X_i$ where $m_i^l \in \{0,1\}$. Suppose the system $\mathcal{S}$ admits one solution in $E = \mathbb{F}_2^k$.*

Then for all xor-combination random variable $B = \bigoplus_{i=1}^{k} b_i X_i$, we have :

$$H(B \mid \mathcal{S}) \in \{0,1\}.$$

Proof. Define $M = (m_j^i)_{i,j} \in \mathcal{M}_{t,k}(\mathbb{F}_2)$ the matrix of the system $\mathcal{S}$, and $\alpha = (\alpha_i) \in \mathcal{M}_{t,1}(\mathbb{F}_2)$. Let $B = \bigoplus_{j=1}^{k} b_j X_j$ be a xor-combination random variable, where $(b_j) \in \mathcal{M}_{1,k}(\mathbb{F}_2)$. Define $M' \in \mathcal{M}_{t+1,k}(\mathbb{F}_2)$ the matrix obtained from M when adding a new line $(b_j)_{j\in[\![1,k]\!]}$. Define $\alpha_0 \in \mathcal{M}_{t+1,1}$ the vector obtained by adding 0 at the end of α and $\alpha_1 \in \mathcal{M}_{t+1,1}$ the vector obtained by adding 1 at the end of α. Define $\mathcal{S}_0$ the system $M'X = \alpha_0$ and $\mathcal{S}_1$ the system $M'X = \alpha_1$.

Each solution of the system $\mathcal{S}$ is either a solution of $\mathcal{S}_0$ or $\mathcal{S}_1$.

- Suppose only one of the two systems $\mathcal{S}_0$ and $\mathcal{S}_1$ admits a solution. Then $H(B \mid \mathcal{S}) = 0$.
- Suppose both of the two systems $\mathcal{S}_0$ and $\mathcal{S}_1$ admit a solution. Then the number of solutions of each one of the affine systems $\mathcal{S}_0$ and $\mathcal{S}_1$ is equal to the number of solution of the linear system $M'X = 0$, and thus $H(B \mid \mathcal{S}) = 1$.

$\lrcorner$

The following lemma illustrates the fact that we can eliminate repetitions in linear protocols.

Lemma 2.5.4. *If there exists a public-coins linear protocol π computing a linear function f, there exists a public-coins linear protocol π' computing f such that for any input distribution μ, $\mathsf{PIC}_\mu(\pi') \leq \mathsf{PIC}_\mu(\pi)$ and*

$$\forall\, i, \forall\, j \neq i,\ I\left((\Pi'_{l\to i})_{l\notin\{i,j\}}; \Pi'_{j\to i} \mid X_i R^p f(X)\right) = 0.$$

Proof. Define π' in the following way: the players act as in π, but each player i, before sending a bit B to a player j, checks that $H(B \mid X_j f(X) \Pi_j^{<B} R^p) \neq 0$, where $\Pi_j^{<B}$ denotes the bit received by player j before B (we take as a convention that within one round, bits are received in the order induced by the index of the player who sends it). If it is equal to 0, then player i omits to send the bit B. Note that the protocol being oblivious, player i can check the value of $H(B \mid X_j f(X) \Pi_j^{<b} R^p)$. Note also that in the case where the bit B is omitted in π', player j already knows the value of B, which allows the protocol π' to go on simulating π even if some bits are omitted. The set of messages sent in π' being a subset of the set of messages sent in π, for any input distribution μ, $\mathsf{PIC}_\mu(\pi') \leq \mathsf{PIC}_\mu(\pi)$.

Fix i and $j \neq i$. Write $(\Pi'_{l\to i})_{l\notin\{i,j\}}$ as a sequence of bits $U_{\sigma(0)}, \ldots, U_{\sigma(t)}$ and $\Pi'_{j\to i}$ as a sequence $V_{\tau(0)}, \ldots, V_{\tau(t')}$, in the order they are received by player j, i.e. σ and τ are increasing functions, $\sigma([\![0,t]\!]) \cup \tau([\![0,t']\!]) = [\![0, t+t'+1]\!]$ and $U_{\sigma(l)}$ is received before $V_{\tau(l')}$ if and only if $\sigma(l) < \tau(l')$.

Denoting the quantity $I((\Pi'_{l\to i})_{l\notin\{i,j\}}; \Pi'_{j\to i} \mid X_i R^p f)$ by Υ, we have:

$$
\begin{aligned}
\Upsilon &= I(U_{\sigma(0)} \dots U_{\sigma(t)}; V_{\tau(0)} \dots V_{\tau(t')} \mid X_i R^p f(X)) \\
&= \sum_{\sigma(l),\tau(l')} I(U_{\sigma(l)}; V_{\tau(l')} \mid X_i R^p f(X)\ U_{<\sigma(l)}\ V_{<\tau(l')}) \\
&\quad \text{(generalized chain rule, Proposition 1.2.10)} \\
&= \sum_{\sigma(l)<\tau(l')} I(U_{\sigma(l)}; V_{\tau(l')} \mid X_i R^p f(X)\ U_{<\sigma(l)}\ V_{<\tau(l')}) + \\
&\quad \sum_{\sigma(l)>\tau(l')} I(U_{\sigma(l)}; V_{\tau(l')} \mid X_i R^p f(X)\ U_{<\sigma(l)}\ V_{<\tau(l')}) \\
&\leq \sum_{\sigma(l)<\tau(l')} (1 - H(V_{\tau(l')} \mid X_i R^p f(X)\ U_{\leq\sigma(l)}\ V_{<\tau(l')})) + \\
&\quad \sum_{\sigma(l)>\tau(l')} (1 - H(U_{\sigma(l)} \mid X_i R^p f(X)\ U_{<\sigma(l)}\ V_{\leq\tau(l')})) \\
&\leq \sum_{\sigma(l)<\tau(l')} (1 - H(V_{\tau(l')} \mid X_i R^p f(X)\ U_{<\tau(l')}\ V_{<\tau(l')})) + \\
&\quad \sum_{\sigma(l)>\tau(l')} (1 - H(U_{\sigma(l)} \mid X_i R^p f(X)\ U_{<\sigma(l)}\ V_{<\sigma(l)})) \\
&\leq \sum_{\sigma(l)<\tau(l')} (1 - H(V_{\tau(l')} \mid X_i R^p f(X)\ \Pi_i^{<\tau(l')})) + \\
&\quad \sum_{\sigma(l)>\tau(l')} (1 - H(U_{\sigma(l)} \mid X_i R^p f(X) \Pi_i^{<\sigma(l)}) \\
&\leq 0
\end{aligned}
$$

where the last inequality, by construction of π', is a consequence of Lemma 2.5.3 which ensures that for any bit B sent to player j in π',

$$H(B \mid X_j R^p f(X) \Pi_j^{<B}) = 1.$$

⌟

We are now able to bound the public information cost of linear protocols computing the parity function Par_k. We consider here the case where all players have to compute the function.

Theorem 2.5.5. *In a setting where only linear protocols are allowed,*

$$\forall\ \epsilon \in \left[0, \frac{1}{2}\right[, \mathsf{PIC}^{\epsilon}(\mathsf{Par}_k) \geq 2(k-1).$$

Proof. Let $\epsilon \in \left[0, \frac{1}{2}\right[$. Let π be a k-player linear protocol ϵ-computing $f = \mathsf{Par}_k$. By Theorem 2.1.5, we can assume that π uses only public coins. Let μ denote the uniform distribution on $\{0,1\}^k$. The protocol π being linear, by Lemma 2.5.3,

$$\forall\ i \in [\![1,k]\!], H(f(X) \mid X_i R^p \Pi_i) \in \{0,1\}.$$

If $\exists\ i \in [\![1,k]\!] \mid H(f(X) \mid X_i R^p \Pi_i) = 1$, it is not possible for player i to ouput correctly with probability more than $\frac{1}{2}$, and thus the protocol π cannot ϵ-compute Par_k. Thus,

$$\forall\ i \in [\![1,k]\!], H(f(X) \mid X_i R^p \Pi_i) = 0$$

and

$$\forall\ i \in [\![1,k]\!], I(X_{-i}; f(X) \mid X_i R^p \Pi_i) = 0.$$

We get:

$$\begin{aligned}
\mathsf{PIC}_\mu(\pi) &= \sum_{i=1}^{k} I(X_{-i}; \Pi_i \mid X_i R^p) \\
&= \sum_{i=1}^{k} [I(X_{-i}; \Pi_i \mid X_i R^p) + I(X_{-i}; f(X) \mid X_i R^p \Pi_i)] \\
&= \sum_{i=1}^{k} I(X_{-i}; \Pi_i f(X) \mid X_i R^p) \text{ (chain rule, Proposition 1.2.9)} \\
&= \sum_{i=1}^{k} \left[I(X_{-i}; f(X) \mid X_i R^p) + I(X_{-i}; \Pi_i \mid X_i R^p f(X))\right] \text{ (idem)} \\
&= k + \sum_{i=1}^{k} I(X_{-i}; \Pi_i \mid X_i R^p f(X)) \\
&= k + \sum_{i=1}^{k} I(X_{-i}; (\Pi_{j\to i})_{j \neq i} \mid X_i R^p f(X)) \\
&= k + \sum_{i=1}^{k} \sum_{j \neq i} I(X_{-i}; \Pi_{j \to i} \mid X_i R^p f(X) (\Pi_{l \to i})_{l<j, l \neq i}) \text{ (chain rule).}
\end{aligned}$$

By Lemma 2.5.4, we can assume that

$$\forall\ i, \forall\ j \neq i,\ I\left((\Pi_{l\to i})_{l \notin \{i,j\}}; \Pi_{j \to i} \mid X_i R^p f(X)\right) = 0$$

and thus that

$$\forall\, i, \forall\, j \neq i,\ I\left((\Pi_{l\to i})_{l<j, l\neq i}; \Pi_{j\to i} \mid X_i R^p f(X)\right) = 0.$$

By Lemma 1.2.14, we can write

$$\begin{aligned}\mathsf{PIC}_\mu(\pi) &\geq k + \sum_{i=1}^{k}\sum_{j\neq i} I(X_{-i}; \Pi_{j\to i} \mid X_i R^p f(X))\\ &\geq k + \sum_{j=1}^{k}\sum_{i\neq j} I(X_{-i}; \Pi_{j\to i} \mid X_i R^p f(X)).\end{aligned}$$

We prove that

$$\exists\, T \subset [\![1,k]\!], |T| = k-2\ \&\ \forall\, j \in T, \sum_{i\neq j} I(X_{-i}; \Pi_{j\to i} \mid X_i R^p f(X)) \geq 1.$$

For this, we show $\forall\, \mathcal{S} \subseteq [\![1,k]\!]$,

$$|\mathcal{S}| = 3 \Longrightarrow \exists\, j \in S, \exists\, i \in [\![1,k]\!] \setminus \{j\}, I(X_{-i}; \Pi_{j\to i} \mid X_i R^p f(X)) \geq 1.$$

Since all the players are able to compute f, there must have been a message sent from a player j to a player i which depends on $(X_l)_{l\in\mathcal{S}}$. Let M be the first such message. Then $j \in S$, as a player from $\overline{S}$ cannot send a message depending on $(X_l)_{l\in\mathcal{S}}$ if he has only received message independent of $(X_l)_{l\in\mathcal{S}}$. The message M is a xor-combination of some elements $(X_l)_{l\in\mathcal{S}'}$ and some elements $(X_l)_{l\in T}$, where $\mathcal{S}' \subseteq \mathcal{S}$ and $T \cap \mathcal{S} = \varnothing$. Since M depends on $(X_l)_{l\in S}$, $S' \neq \emptyset$. Since m is the first message depending on $(X_l)_{l\in S}$, it can only depends on X_j, as by assumption at the time he sends message M, player j has only received messages independent of $(X_l)_{l\in S}$. This shows $S' = \{j\}$. From this we get $I(X_{-i}; M \mid X_i f(X) R^p) = H(M \mid X_i f(X) R^p)) = 1$ and thus $I(X_{-i}; \Pi_{j\to i} \mid X_i f(X) R^p) \geq 1$.

We conclude the proof: $\mathsf{PIC}_\mu(\pi) \geq k + (k-2) = 2(k-1)$, and we have thus shown $\mathsf{PIC}^\epsilon(\mathsf{Par}_k) \geq 2(k-1)$.

⌟

2.5.2 Sensitivity

In this subsection, we consider the bit-wise parity function, as defined in Subsection 2.4.5. The key concept which guides the proof is *sensitivity*.

Definition 2.5.6. *The sensitivity of a function $f : \{0,1\}^k \longrightarrow \{0,1\}$ on input x is defined as*

$$s_x(f) = |\{i, f(x) \neq f(x \oplus e_i)|,$$

where $x \oplus e_i$ denotes the input obtained from x by flipping its i^{th} bit.

The average sensitivity of f is

$$\overline{s}(f) = \frac{1}{2^k} \sum_{x \in \{0,1\}^k} s_x(f).$$

The sensitivity is a fundamental complexity measure of boolean functions, and is related to many other complexity notions (see e.g. [Shi00]). The parity function is, among all the boolean functions, the one with highest sensitivity. Even though the sensitivity notion does not appear explicitly in the proof of Theorem 2.5.7, the proof is based on the high sensitivity of the parity function.

Our bound for Par_k^n is in fact proved for a wider class of protocols, where we allow the player outputting $\bigoplus_{i=1}^{k} x_i^p$ to be different for each coordinate p and to depend on the input.

Theorem 2.5.7. *In the oblivious setting,*

$$\mathsf{PIC}_\mu^0(\mathsf{Par}_k^n) \geq n(k-1)$$

where μ is the uniform input distribution.

Proof. We use the uniform distribution μ throughout the proof. Since we are looking at zero-error protocols, the public information cost is equal to the information cost of deterministic protocols by Theorem 2.1.6. Let π be a deterministic protocol solving Par_k^n. By Theorem 2.1.8,

$$\mathsf{PIC}_\mu^0(\pi) \geq \mathsf{PIC}_\mu^{\mathsf{ext}}(\pi).$$

Showing that $\mathsf{PIC}_\mu^{\mathsf{ext}}(\pi) \geq n(k-1)$ will prove the theorem.

Let $x = (x_i^p) \in \{0,1\}^{nk}$, where x_i is the n-bit input of player i. For any p, let $q^p(x)$ be the index of the first player able to compute $\bigoplus_{i=1}^{k} x_i^p$ (formally, we say that a player is able to compute $\bigoplus_{i=1}^{k} x_i^p$ at some time if the value of $\bigoplus_{i=1}^{k} x_i^p$ is fixed given his input and his transcript until that time). For any i, define $C_i(x) = \{p, q^p(x) \neq i\}$, which represents the coordinates of his input that player i is intuitively going to leak when the players are given input x, and let $c_i(x) = |C_i(x)|$.

We show that $\forall\ i, H(X_i \mid \overleftrightarrow{\Pi}_i = \overleftrightarrow{\pi}_i(x)) \leq n - c_i(x)$. Assume towards a contradiction that for some i, $H(X_i \mid \overleftrightarrow{\Pi}_i = \overleftrightarrow{\pi}_i(x)) > n - c_i(x)$. This implies that the number of possible values for X_i consistent with $\overleftrightarrow{\Pi}_i = \overleftrightarrow{\pi}_i(x)$ is more than $2^{n-c_i(x)}$, and thus the number of coordinates of the input of the i-th player that are fixed by the transcript is strictly less than $c_i(x)$. Then there must exists x' such that

- $\overleftrightarrow{\pi}_i(x') = \overleftrightarrow{\pi}_i(x)$
- $\exists\ p \in C_i(x)$ such that $x'^p_i \neq x^p_i$.

Note that $\overleftrightarrow{\pi}_i(x') = \overleftrightarrow{\pi}_i(x)$ implies (by considering player i as Alice, and all other players together as Bob, and using arguments as those used for a similar property for 2-party protocols) that $\overleftrightarrow{\pi}_i(x) = \overleftrightarrow{\pi}_i(x'_i, x_{-i})$. As $q^p(x) \neq i$, this is a contradiction, since then player $q^p(x)$ would output the same answer on x and (x'_i, x_{-i}), while $\bigoplus_{j=1}^{k} x^p_j \neq x'^p_i \oplus \bigoplus_{j \neq i} x^p_j$.

Thus

$$\forall\ i \in [\![1, k]\!], H(X_i \mid \overleftrightarrow{\Pi}_i = \overleftrightarrow{\pi}_i(x)) \leq n - c_i(x)$$

and

$$\forall\ i \in [\![1, k]\!], H(X_i \mid \overleftrightarrow{\Pi}_i) \leq \mathop{\mathbb{E}}_x[n - c_i(x)] = n - \mathop{\mathbb{E}}_x[c_i(x)].$$

Thus

$$\forall\ i \in [\![1, k]\!], I(X_i; \overleftrightarrow{\Pi}_i) \geq \mathop{\mathbb{E}}_x[c_i(x)].$$

Summing over all i, we get

$$\mathsf{PIC}^{\mathsf{ext}}_{\mu}(\pi) \geq \sum_{i=1}^{k} \mathop{\mathbb{E}}_x[c_i(x)] = \mathop{\mathbb{E}}_x[\sum_i c_i(x)]$$

and since $\sum_{i=1}^{k} c_i(x) = n(k-1)$ for any x, we get

$$\mathsf{PIC}^{\mathsf{ext}}_{\mu}(\pi) \geq \mathop{\mathbb{E}}_x[n(k-1)] = n(k-1).$$

As $\mathsf{PIC}_{\mu}(\pi) \geq \mathsf{PIC}^{\mathsf{ext}}_{\mu}(\pi)$, we have shown that

$$\mathsf{PIC}_{\mu}(\pi) \geq n(k-1).$$

$\lrcorner$

We can now get a bound on the communication complexity of the parity function in the oblivious setting.

Theorem 2.5.8. *In the oblivious setting,*

$$\mathsf{CC}^0(\mathsf{Par}_k^n) \geq n(k-1).$$

Proof. It results from Theorems 2.5.7 and 2.3.2.

┘

Thanks to the tools we developed in Section 2.4, we can now prove a lower bound on the randomness complexity of the parity function.

Theorem (2.4.16). *There exists an input distribution μ such that*

$$\mathcal{R}_\mu(\mathsf{Par}_k^n) \geq \frac{k-2}{k-1}n.$$

Proof. For Par_k^n, where one player outputs the parity for each coordinate, we have for the uniform distribution $\sum_i H(f_i(X)) = n$. Applying Theorem 2.4.14 with Theorem 2.5.7, we get: $\mathcal{R}_\mu(\mathsf{Par}_k^n) \geq \frac{(k-2)n}{k-1}$.

┘

2.6 Compression and Direct sum

2.6.1 Compression of protocols

Background on compression of protocols

The problem of compressing communication is fundamental in computer science. In the setting of the transmission of a message, optimal compressions (or encoding) are known thanks to the work of Shannon [Sha48], Fano [Fan61] and Huffman [Huf06]. A random variable X can be transmitted with roughly $H(X)$ bits. In other words, the communication cost is equivalent to the information content of the message. This is also true in the amortized sense when the receiver already has partial information about the message [SW73].

How well one can compress communication in the interactive setting is a more complicated question. The question can be stated in terms of communication complexity and information complexity. Several results shed light on the relation between communication and information in the two-party setting. It is shown in [BBCR10] that a protocol with communication C

and information cost I can be simulated by a protocol with communication $\tilde{\mathcal{O}}(\sqrt{CI})$, and can be further compressed to a protocol with communication $\mathcal{O}(I \text{ polylog}(C))$ when the input is drawn from a product distribution. The specific case of compression under product distributions was studied in [Kol16, She16], leading to a compression to $\mathcal{O}(I \text{ polylog}(I))$. A compression to communication $O(I)$ for bounded rounds protocols is presented in [JRS03], later improved to $I + o(I)$ in [BR11]. Without any assumption, a very general compression scheme to communication $2^{\mathcal{O}(I)}$ is known [Bra12]. The case of public-coins protocols is studied in [BBK$^+$13, Pan15], where is shown a compression to communication $\mathcal{O}(I \log C)$. This was improved in [BMY15] where a compression to a protocol with communication $\mathcal{O}(H^{int} \log\log C)$, where H^{int} is the internal entropy of the protocol (which is equal to the information cost in the case of a public-coins protocol), is presented. A compression procedure for the broadcast model is described in [KOS17].

On the other hand, it is known that perfect compression is not possible. The series of articles [Bra13, GKR14, GKR15b, GKR15a] (see also [RS15]) shows, under various settings, that there exist functions for which information and communication are fundamentally different measures.

In this section, we present a compression result with regards to the average-case communication complexity and the public information cost for oblivious protocols.

Relation between direct sum and compression via the public information cost

Theorem 2.6.1. *Suppose there exists an oblivious protocol π to compute a k-variable function f over the distribution μ with distributional error probability ϵ. Then for any $\delta > 0$ there exists a public-coin protocol ρ that computes f over μ with distributional error $\epsilon + \delta$, and with average communication complexity*

$$\mathsf{AvCC}_\mu(\rho) = \mathcal{O}\left(k^2 \cdot \mathsf{PIC}_\mu(\pi) \log(\mathsf{CC}(\pi)) \log \frac{k^2 \cdot \mathsf{PIC}_\mu(\pi) \log(\mathsf{CC}(\pi))}{\delta}\right).$$

The proof of the above theorem will follow from extending the compression result presented in [BBK$^+$13, Pan15] to the case of $k > 2$ players.

Theorem 2.6.2. *Suppose there exists an oblivious public-coin protocol π to compute a k-variable function f over the distribution μ with distributional error probability ϵ. Then for any $\delta > 0$ there exists a public-coin protocol*

ρ that computes f over μ with distributional error $\epsilon + \delta$, and with average communication complexity

$$\mathsf{AvCC}_{\mu}(\rho) = \mathcal{O}\left(k^2 \cdot \mathsf{IC}_{\mu}(\pi)\log(\mathsf{CC}(\pi))\log\frac{k^2 \cdot \mathsf{IC}_{\mu}(\pi)\log(\mathsf{CC}(\pi))}{\delta}\right).$$

Theorem 2.6.2 and Theorem 2.1.5, which makes the link between the public information cost of general protocols and the information cost of public-coins protocol, imply Theorem 2.6.1.

In the two-party compression scheme of [BBK$^+$13, Pan15], the two players, given their own input, try to guess the transcript $\pi(x_1, x_2)$ of the protocol π. For this, player 1 picks a candidate t_1 from the set $\mathrm{Im}(\pi(x_1, \cdot))$ of possible transcripts knowing that it has input x_1, while player 2 picks a candidate t_2 from the set $\mathrm{Im}(\pi(\cdot, x_2))$. The two players then communicate in order to find the first bit on which t_1 and t_2 disagree. The general structure of protocols ensures that the common prefix of t_1 and t_2 (until the first bit of disagreement) is identical to the beginning of the correct transcript on inputs x_1 and x_2, i.e. identical to $\pi(x_1, x_2)$. Starting from this correct prefix, the players then pick new candidates for the transcript of the protocol $\pi(x_1, x_2)$, and so on, until they agree on the full transcript $\pi(x_1, x_2)$. Clever choices of the candidates, along with an efficient technique to find the first bit which differs between the candidates, lead to a protocol with little communication.

In extending the proof of [BBK$^+$13, Pan15] to the multi-party case, new difficulties occur. The players can no longer try to guess the full transcript, as they have little information about the conversation between the other players, but can only try to guess their partial transcript, according to their own input. Then, in order to find the first disagreement in the global transcript, every pair of players needs to find and communicate the place of the first disagreement in their partial transcripts.

We use here the notation $\overleftrightarrow{\Pi}_i$ to denote the concatenation of the messages read and sent by player i according to its local round structure, i.e. as the concatenation, local round of player i by local round of player i, of, first, the messages sent by player i and, then, the messages received by player i. Observe that since we consider here oblivious protocols, there is a one-to-one correspondence between the two interpretations of the notation $\overleftrightarrow{\Pi}_i$.

We will use a black box, call it *lcp box* (for *largest common prefix*), which can be used by two players A and B in the following way: A puts a string x in the box, B puts a string y in the box, and the box gives them back the first index j such that $x_j \neq y_j$ if $x \neq y$, or tells them that $x = y$ otherwise. The price to pay for using this black box is $\log n$ bits of communication, where $n = \max(|x|, |y|)$.

This box can be efficiently simulated if we allow error:

Lemma 2.6.3 ([FRPU94])**.** *There is a randomized public-coins protocol such that on input two n-bit strings x and y, it outputs the first index j such that $x_j = y_j$ with probability at least $1-\epsilon$ if such a j exists, and otherwise outputs that the two strings are equal, with worst case communication complexity $\mathcal{O}(\log(n/\epsilon))$.*

Corollary 2.6.4 ([BBK$^+$13])**.** *Any protocol $\tilde{\rho}$ that uses an lcp box l times on average can be simulated with error δ by a protocol ρ that does not use an lcp box, and communicates an average of $\mathcal{O}(l \log(\frac{l}{\delta}))$ extra bits.*

We will use the lcp box in the proof, and use Corollary 2.6.4 to perform the analysis at the end.

Proof of Theorem 2.6.2. For any i, define the set $\mathcal{X}_i$ to be the set of possible inputs of player i, and the set $\Pi_{(i)}(x_i)$ the set of possible transcripts of the communication between player i and the coordinator, knowing that player i has input x_i:

$$\Pi_{(i)}(x_i) = \overleftrightarrow{\pi_i}(\mathcal{X}_1, \ldots, \mathcal{X}_{i-1}, x_i, \mathcal{X}_{i+1}, \ldots, \mathcal{X}_k, r).$$

Each player i represents $\Pi_{(i)}(x_i)$ by a binary tree T_i as follows.

1. The root is the largest common prefix (lcp) of the transcripts in $\Pi_{(i)}(x_i)$, and the remaining nodes are defined inductively.
2. For each node τ,
   - the first child of τ is the lcp of the transcripts in $\Pi_{(i)}(x_i)$ beginning with $\tau \circ 0$, i.e., τ concatenated with the bit 0.
   - the second child of τ is the lcp of the transcripts in $\Pi_{(i)}(x_i)$ beginning with $\tau \circ 1$.
3. The leaves are labelled by the possible transcripts of player i, i.e. the elements of $\Pi_{(i)}(x_i)$.

We define the *weight* of a leaf f with label t_i to be

$$w(t_i) = \Pr_{(X_j)_{j\neq i}|X_i=x_i}[\overleftrightarrow{\pi_i}(X_1, \ldots, X_{i-1}, x_i, X_{i+1}, \ldots, X_k, r) = t_i].$$

The weight of a non-leaf node is defined by induction as the sum of the weights of its children. By construction, the weight of the root is 1.

We say that $(t_1, \dots, t_k) \in \Pi_{(1)}(x_1) \times \dots \times \Pi_{(k)}(x_k)$ is a *coherent profile* if, for each round l, any message which appears in t_i as sent to player j also appears in t_j as coming from player i. In fact, $(\overleftrightarrow{\pi_1}(x_1, \dots, x_k, r), \dots, \overleftrightarrow{\pi_k}(x_1, \dots, x_k, r))$ is the only coherent profile. Indeed, take a coherent profile $(t_1, \dots, t_k)$. For each i, t_i is of the form $\overleftrightarrow{\pi_i}(x^i_1, \dots, x^i_{i-1}, x_i, x^i_{i+1}, \dots, x^i_k)$ where $\forall\ j \neq i,\ x^i_j \in \mathcal{X}_j$. Since whenever a player sends a message, his choice is based only on his own input and on the messages he has received before, it implies (by induction on the lots of messages defined for oblivious protocols in Subsection 1.4.1) that

$$(t_1, \dots, t_k) = (\overleftrightarrow{\pi_1}(x_1, \dots, x_k, r), \dots, \overleftrightarrow{\pi_k}(x_1, \dots, x_k, r)).$$

We now define the protocol $\tilde{\rho}$ which allows the players to collaborate and efficiently find this coherent profile, i.e. which allows each player i to find $\overleftrightarrow{\pi_i}(x_1, \dots, x_k, r)$.

The players proceed in stages $s = 1, 2 \dots$. We will have the invariant that at the beginning of any stage s, each player i has a pointer to a node $\tau_i(s)$ of its transcript tree T_i, such that $(\tau_1(s), \dots, \tau_k(s))$ is a (term-wise) prefix of $(\overleftrightarrow{\pi_1}(x_1, \dots, x_k, r), \dots, \overleftrightarrow{\pi_k}(x_1, \dots, x_k, r))$. At every stage s, every player i furthermore has a candidate leaf $t_i(s)$ in the tree T_i (representing a candidate for its transcript), defined as follows: player i defines $\tau^1 = \tau_i(s)$, and then defines inductively τ^{j+1} to be the child of τ^j which has higher weight (breaking ties arbitrarily), until it reaches a leaf: this is the candidate $t_i(s)$. Observe that $t_i(s)$ is a descendent of $\tau_i(s)$ in T_i and that $t_i(s)$ corresponds to the transcript with highest probability conditioned on the fact that the transcript starts by $\tau_i(s)$.

At the beginning of the protocol $\tilde{\rho}$, each player i starts the protocol with a pointer to the node $\tau_i(1)$, which is the root of the tree T_i. At every stage s, the players proceed as follows:

1. Each pair of players (i, j) uses an lcp box to find the first occurrence where the transcript between i and j in $t_i(s)$ is not coherent with the transcript between i and j in $t_j(s)$. Let $q_{i,j}$ be the index of the message that includes this first occurrence, where the messages are numbered according to the global order of all messages of an oblivious protocol as defined in Subsection 1.4.1, and ∞ if no such occurrence was found. Let $Q_i = \min_j\{q_{i,j}\}$. Observe that if for all pairs of players there is no such occurrence (i.e., $Q_i = \infty$ for all i), it means that $(t_1(s), \dots, t_k(s))$ is a coherent profile, and each player i has found $\overleftrightarrow{\pi_i}(x_1, \dots, x_k, r)$.

2. Each player i now broadcasts Q_i. Each player can now find $Q = \min_i\{Q_i\}$. If $Q = \infty$, i.e. no pairwise inconsistency has been found

between any two nodes, the protocol terminates and $(t_1(s), \dots, t_k(s))$ is found as the coherent profile.

3. Let (i, j) be the pair of players such that $Q = q_{i,j}$, and assume for instance that the message number Q is sent by player i to player j. Player i, having the sender role, is considered "correct" because what player i sent in the protocol π is based on the previous rounds. Player j sets its $\tau_j(s+1)$: in T_j, starting from $t_j(s)$, it goes up the tree toward $\tau_j(s)$, until it reaches a node $\hat{\tau}_j$ which is correct (according to the result of the lcp box). Then, it defines $\tau_j(s+1)$ as the other child of $\hat{\tau}_j$.

4. Any other player $l \neq i$ defines $\tau_l(s+1) = \tau_l(s)$.

We now claim by induction on the stages that the invariant stated above is preserved for all players at all times. It clearly holds at the beginning. We claim that if it holds after stage s then it also holds after stage $s+1$. For the $k-1$ players which define $\tau_j(s+1) = \tau_j(s)$ it clearly continues to hold. For the single player, say player i, which defines a new node as $\tau_i(s+1)$ in Step (3) we proceed as follows.

We first claim, by induction on the index of the messages in the global order, that for all messages with index $\ell < Q$, where message ℓ is sent from player j to player i, it holds that the value of message number ℓ is the same in the coherent profile $(\overleftrightarrow{\pi}_1(x_1, \dots, x_k, r), \dots, \overleftrightarrow{\pi}_k(x_1, \dots, x_k, r))$ and in both $t_i(s+1)$ and $t_j(s+1)$. The basis of the induction ($\ell = 0$) clearly holds. The inductive step follows from observing that message ℓ is fully determined by the input to player j and the messages that appear before message ℓ in $\overleftrightarrow{\pi}_j$. Thus, by the induction hypothesis the value of message ℓ in $t_j(s+1)$ is as it appears in the coherent profile $(\overleftrightarrow{\pi}_1(x_1, \dots, x_k, r), \dots, \overleftrightarrow{\pi}_k(x_1, \dots, x_k, r))$. It follows from the definition of Q that the value of message ℓ is the same in $t_i(s+1)$ and $t_j(s+1)$.

For message Q, we have by similar arguments that its value according to $t_j(s+1)$ is consistent with $\overleftrightarrow{\pi}_i(x_1, \dots, x_k, r)$. The prefix of message Q as appears in the path from the root of T_i and delimited by $\tau_i(s+1)$ is consistent with $t_j(s+1)$ by the choice of $\tau_i(s+1)$ in Step (3).

Now, since the relative order of messages in a transcript $\overleftrightarrow{\Pi}_i$ and in the global order is the same, it follows that $\tau_i(s+1)$ represents a prefix of $\overleftrightarrow{\pi}_i(x_1, \dots, x_k, r)$, as required.

Let i denote the player who sets its $\tau_i(s+1)$ in Step (3). We show that $w(\tau_i(s+1)) \leq \frac{1}{2} w(\tau_i(s))$. We look at the sequence (τ^j) defined by player i when he chose his candidate leaf $t_i(s+1)$ at step s. Let τ^j be the first common ancestor of $t_i(s)$ and $\tau_i(s+1)$. By construction, $\tau_i(s+1)$ is a direct child of τ^j, and $t_i(s)$ is (a descendant of) another child of τ^j. By the candidate leaf's construction process, $w(\tau_i(s+1)) \leq \frac{1}{2}w(\tau^j) \leq \frac{1}{2}w(\tau_i(s))$.

We conclude the analysis. On inputs $(x_1, \ldots, x_k)$, let $(t_1, \ldots, t_k)$ denote the coherent profile. Each player will correct his τ_i no more than $\log \dfrac{1}{w(t_i)}$ times, because the weight of the node τ_i halves with each correction, as noticed before, and because the root has weight 1. Hence, the total number of corrections, and thus the number of stages S, is bounded by $\sum_{i=1}^{k} \log \dfrac{1}{w(t_i)}$. We now consider each t_i as a random variable (i.e. a function of X and the randomness of the protocol) and we take the average over inputs and shared randomness.

$$\begin{aligned}\mathop{\mathbb{E}}_{r,x}[S] &\leq \mathop{\mathbb{E}}_{r,x} \sum_{i=1}^{k} \log \frac{1}{w(t_i)} \\ &\leq \sum_{i=1}^{k} \mathop{\mathbb{E}}_{r,x_i} \left[\mathop{\mathbb{E}}_{(x_j)_{j\neq i}|X_i=x_i} \left[\log \frac{1}{w(t_i)} \right] \right]\end{aligned}$$

By definition of t_i, conditioned on a given x and a given r, we have:

$$\log \frac{1}{w(t_i)} = \log \frac{1}{\Pr\limits_{(X_j)_{j\neq i}|X_i=x_i} [\overleftrightarrow{\pi}_i(X_1, \ldots, X_{i-1}, x_i, X_{i+1}, \ldots, X_k, r) = \overleftrightarrow{\pi}_i(x_1, \ldots, x_k, r)]}.$$

By regrouping the $(x_j)_{j\neq i}$, we can rewrite for any fixed x_i and r

$$\mathop{\mathbb{E}}_{(x_j)_{j\neq i}|X_i=x_i} \left[\log \frac{1}{\Pr\limits_{(X_j)_{j\neq i}|X_i=x_i} [\overleftrightarrow{\pi}_i(X_1, \ldots, X_{i-1}, x_i, X_{i+1}, \ldots, X_k, r) = \overleftrightarrow{\pi}_i(x_1, \ldots, x_k, r)]} \right]$$

as

$$\mathop{\mathbb{E}}_{t_i|X_i=x_i,R=r} \left[\log \frac{1}{\Pr\limits_{(X_j)_{j\neq i}|X_i=x_i} [\overleftrightarrow{\pi}_i(X_1, \ldots, X_{i-1}, x_i, X_{i+1}, \ldots, X_k, r) = t_i]} \right].$$

Thus we have

$$
\begin{aligned}
\mathop{\mathbb{E}}_{r,x}[S] &\leq \sum_{i=1}^{k} \mathop{\mathbb{E}}_{r,x_i}\left[\mathop{\mathbb{E}}_{t_i|X_i=x_i,R=r}\left[\log \frac{1}{\Pr\limits_{(X_j)_{j\neq i}|X_i=x_i}[\overleftrightarrow{\pi_i}(X_1,\ldots,X_{i-1},x_i,X_{i+1},\ldots,X_k,r)=t_i]}\right]\right] \\
&\leq \sum_{i=1}^{k} \mathop{\mathbb{E}}_{r,x_i}\left[H(\overleftrightarrow{\Pi_i} \mid x_i r)\right] \\
&\leq \sum_{i=1}^{k} H(\overleftrightarrow{\Pi_i}|X_i R^p) \\
&\leq \sum_{i=1}^{k} I(X_{-i};\overleftrightarrow{\Pi_i}|X_i R^p) \\
&\leq \mathsf{IC}_\mu(\pi) \quad \text{(by Proposition 1.4.8).}
\end{aligned}
$$

We have shown that the average number of stages is bounded by $\mathsf{IC}_\mu(\pi)$. At each stage, the communication consists of $\frac{k(k-1)}{2}$ calls to the lcp box on strings of length at most $\mathcal{O}(\mathsf{CC}(\pi))$ (one call for each pair of players), plus $k(k-1)$ messages of broadcasts of indices at Step (2), each message of size $\mathcal{O}(\log \mathsf{CC}(\pi))$. Hence

$$\mathsf{AvCC}_\mu(\tilde{\rho}) = \mathcal{O}\left(k^2 \cdot \mathsf{IC}_\mu(\pi)\log(\mathsf{CC}(\pi))\right).$$

Using Corollary 2.6.4 we can replace each use of the lcp box with a simulation protocol, to get the protocol ρ which simulates π with distributional error $\epsilon + \delta$ and average communication:

$$
\begin{aligned}
\mathsf{AvCC}_\mu(\rho) &= \mathsf{AvCC}_\mu(\tilde{\rho}) + \mathcal{O}\left(k^2 \cdot \mathsf{IC}_\mu(\pi)\log\frac{k^2 \cdot \mathsf{IC}_\mu(\pi)}{\delta}\right) \\
&= \mathcal{O}\left(k^2 \cdot \mathsf{IC}_\mu(\pi)\log(\mathsf{CC}(\pi))\log\frac{k^2 \cdot \mathsf{IC}_\mu(\pi)\log(\mathsf{CC}(\pi))}{\delta}\right).
\end{aligned}
$$

⌟

2.6.2 The direct sum problem

Direct sums in distributed computing

The *direct sum* property is a fundamental question in complexity theory, and has been studied for many computation models. A direct sum theorem affirms that the amount of resources needed to perform t independent tasks is at least the sum of the resources needed to perform each of the t tasks. This allows one to study a complicated function by decomposing it into simpler functions.

One of the reasons making information tools powerful is that they often satisfy a direct sum property. The two-party internal information cost was shown to be additive in [Bra12]. In the study of the communication complexity of the multi-party function *Disjointness*, direct sums on information-theoretic tools proved to be useful: the *conditional information cost* of [BYJKS02], the *switched information cost* of [BEO$^+$13] and the *partial information cost* of [CM15] all satisfy a direct sum.

Direct sums for communication complexity are harder to obtain. Several direct sums theorems are known. A direct sum for the equality function in the simultaneous message model was presented in [CSWY01]. In [JRS03], a direct sum relates the bounded round randomized communication complexity of a function $f^{\otimes n}$ to the distributional communication complexity of f for product distributions, result which was strengthened in [HJMR10]. The authors then showed a direct sum for one-way public-coins communication complexity in [JSR08]. In [BBCR10], the authors showed that the dependency in n of the randomized communication complexity of $f^{\otimes n}$ is at least $\sqrt{n}$, and got a similar result for average case complexity. They also obtained a linear dependency, i.e. a full direct sum, in the case where the inputs are drawn from a product distribution. It is know, however, that a strict direct sum for the randomized communication complexity cannot hold [GKR14, GKR15b] (cf. also [RS15]). The possibility of a direct sum for zero-error average communication when the protocol is only required to be correct on the support of the distribution is ruled out in [KMSY16]

The direct sum question in communication complexity is also related to other important problems in computer science. For instance, due to the link between communication complexity and circuit complexity, it was shown that a direct sum for the communication complexity of relations would imply the separation of P and NC^1 [KRW95].

Note that information complexity has a direct sum property in the multi-party case. For PIC, it is easy to prove the following inequality.

Theorem 2.6.5. *For any k-variable functions f and g, for any distribution μ on inputs of f, for any distribution η on inputs of g, we have*

$$\mathsf{PIC}^{\epsilon}_{\mu\times\eta}(f\otimes g) \leq \mathsf{PIC}^{\epsilon}_{\mu}(f) + \mathsf{PIC}^{\epsilon}_{\eta}(g).$$

We use here the notation $f \otimes g$ to indicate that the task of computing f with error ϵ and computing g with error ϵ (by opposition to computing the couple function (f, g) with error ϵ).

A direct sum for PIC implies a direct sum for CC

Using the results from the previous subsection, we prove that under certain conditions, a direct sum for PIC would imply a direct sum for CC.

Theorem 2.6.6. *In the oblivious setting, given a k-variable function f and a distribution μ on inputs of f, if the existence of a protocol π computing $f^{\otimes t}$ with error $\epsilon \geq 0$ implies that there exists a protocol π' computing f with error ϵ and satisfying $\mathsf{PIC}_\mu(\pi') \leq \frac{1}{t}\mathsf{PIC}_{\mu^{\otimes t}}(\pi)$, $\mathsf{CC}(\pi') \leq \mathsf{CC}(\pi)$, then for any $\delta > 0$*

$$\mathsf{CC}^{2(\epsilon+\delta)}(f) = \mathcal{O}\left(\frac{k^3 \log(k)}{t(\epsilon+\delta)}\mathsf{CC}^\epsilon(f^{\otimes t})\log(\mathsf{CC}^\epsilon(f^{\otimes t}))\log\frac{k^2\mathsf{CC}^\epsilon(f^{\otimes t})\log(\mathsf{CC}^\epsilon(f^{\otimes t}))}{\delta}\right).$$

Note that the result of this theorem is meaningful when t is large with respect to k.

To prove this result, we will need the following lemma.

Lemma 2.6.7. *Given an input distribution μ, any k-party protocol with error $\frac{\epsilon}{2}$ and average communication complexity C can be turned into an oblivious protocol with distributional error ϵ and worst case communication complexity $\frac{Ck\log(k)}{\epsilon}$.*

Proof. Let π be a protocol with error $\frac{\epsilon}{2}$ and average communication complexity C. We define now define a protocol π', which is similar to π with the difference that player 1 acts as a coordinator, in addition to his original role in π, and that the other players can only communicate with player 1. The protocol π' run in rounds : in every round, player 1 sends a message to all players indicating the beginning of the round. The players answer to player 1 by sending a message of the form (m, j) to indicate that they want to send a message m to player j, or send a "no" signal indicating that they do not want to send any message. Player 1 then forwards all these messages to their destination with the form (m, i) where i indicates the origin of the message. We add the constraint that in π' all the messages m are actually single bits, thus forcing the players to decompose a long message into single bit messages. Note that the original messages in π being self-delimiting, this decomposition does not introduce any ambiguity from the point of view of the players receiving a message bit by bit. Moreover, we impose that the protocol π' always run a fixed number T of rounds, with $T = \left\lceil \frac{2C}{\epsilon} \right\rceil$.

Note that π' is oblivious. Moreover, π' fails to simulate π only in the case where T happens to be too small, thus interrupting the simulation of π before its term. As every round in π' contains at least one message from π, the probability that π' fails in simulating π is bounded by $\Pr_{x,r}(|\Pi(x)| \geq T) \leq \frac{C}{T} \leq \frac{\epsilon}{2}$ by Markov inequality. Hence, the protocol π' has error ϵ.

Last, every round in protocol π' consists in communication $\mathcal{O}(k\log(k))$, and protocol π' thus has worst case communication $\mathcal{O}(Tk\log(k)) = \mathcal{O}\left(\frac{Ck\log(k)}{\epsilon}\right)$.

⌟

Proof of Theorem 2.6.6. Let μ be a distribution on inputs of f. Consider a protocol π computing $f^{\otimes t}$ with error ϵ. By hypothesis, there exists π' computing f with error ϵ and satisfying $\mathsf{PIC}_\mu(\pi') \leq \frac{1}{t}\mathsf{PIC}_{\mu^{\otimes t}}(\pi)$, $\mathsf{CC}(\pi') \leq \mathsf{CC}(\pi)$. By Theorem 2.1.5, we can impose π' to use only public randomness.

Applying successively Theorem 2.6.1 and Lemma 2.6.7, we get a protocol ρ with distributional error $2(\epsilon + \delta)$ and such that

$$\begin{aligned}\mathsf{CC}(\rho_\mu) &= \mathcal{O}\left(\frac{1}{(\epsilon+\delta)}k^3\mathsf{PIC}_\mu(\pi')\log(\mathsf{CC}(\pi'))\log(k)\log\frac{k^2\mathsf{PIC}_\mu(\pi')\log(\mathsf{CC}(\pi'))}{\delta}\right)\\ &= \mathcal{O}\left(\frac{1}{(\epsilon+\delta)}k^3\mathsf{PIC}_\mu(\pi')\log(\mathsf{CC}(\pi))\log(k)\log\frac{k^2\mathsf{CC}(\pi)\log(\mathsf{CC}(\pi))}{\delta}\right).\end{aligned}$$

Thus

$$\begin{aligned}\mathsf{CC}(\rho_\mu) &= \mathcal{O}\left(\frac{1}{t(\epsilon+\delta)}k^3\mathsf{PIC}_{\mu^{\otimes t}}(\pi)\log(\mathsf{CC}(\pi))\log(k)\log\frac{k^2\mathsf{CC}(\pi)\log(\mathsf{CC}(\pi))}{\delta}\right)\\ &= \mathcal{O}\left(\frac{1}{t(\epsilon+\delta)}k^3\mathsf{CC}(\pi)\log(\mathsf{CC}(\pi))\log(k)\log\frac{k^2\mathsf{CC}(\pi)\log(\mathsf{CC}(\pi))}{\delta}\right).\end{aligned}$$

As this reasoning is valid for any distribution μ, Yao's minimax lemma implies

$$\begin{aligned}\mathsf{CC}_{2(\epsilon+\delta)}(f) = \mathcal{O}\Bigg(&\frac{1}{t(\epsilon+\delta)}k^3\mathsf{CC}_\epsilon(f^{\otimes t})\log(\mathsf{CC}_\epsilon(f^{\otimes t}))\log(k)\\ &\log\frac{k^2\mathsf{CC}_\epsilon(f^{\otimes t})\log(\mathsf{CC}_\epsilon(f^{\otimes t}))}{\delta}\Bigg)\end{aligned}$$

as wanted.

⌟

Chapter 3

Multi-party Information Cost

3.1 Definition and properties

3.1.1 Notations

In this chapter, we use the channel representation defined in Subsection 1.4.1. The set of possible transcripts for a protocol is usually denoted $\mathcal{T}$, and the projection of this set on the i^{th} coordinate (i.e. the set of possible transcripts of player i) is usually denoted $\mathcal{T}_i$. Observe that $\mathcal{T} \subseteq \mathcal{T}_1 \times \cdots \times \mathcal{T}_k$. By $\Pi_i^l(x, r)$ we denote $\Pi_i(x, r)$ modified such that all the messages that player i sends in local rounds $l' > l$, and all the messages that player i reads in local rounds $l' > l$ are eliminated from the transcript.

3.1.2 Definition

We now define another information-theoretic measure for multi-party peer-to-peer protocols. We note that a somewhat similar measure was proposed in [BEO$^+$13] for the coordinator model, but, to the best of our knowledge, never found an application.

Definition 3.1.1. *For any k-player protocol π and any input distribution μ, we define the multi-party information cost of π:*

$$\mathsf{MIC}_\mu(\pi) = \sum_{i=1}^{k} \left(I(X_{-i}; \Pi_i \mid X_i R_i) + I(X_i; \Pi_i \mid X_{-i} R_{-i}) \right) .$$

The second part of each of the k terms can be interpreted as the information that player i "leaks" to a virtual player formed by grouping all players except i. In the same way that PIC was taking into account the fact that private protocols need to use private randomness, we use to define MIC the intuition that private protocols must use *secret sharing* primitives, which fail when $k-1$ players collude.

Observe that the summation of each one of the two summands alone would not yield a measure useful for proving lower bounds on communication complexity. The first summand would yield a measure that would never be higher than the entropy of the computed function, due to the existence of private protocols for all functions, as discussed in Subsection 2.4.1. For the second summand, there are functions for which that measure would be far too low compared to the communication complexity: e.g. the function $f = x_1$ (i.e. the value of the function is the input of player 1); in that case the measure would be only 1, while it is clear that the communication complexity of that function is $\Omega(k)$.

We also define the multi-party information complexity of a function.

Definition 3.1.2. *For any function f and any input distribution μ, we define the quantity*

$$\mathsf{MIC}_\mu(f) = \inf_{\pi \text{ computing } f} \mathsf{MIC}_\mu(\pi).$$

Definition 3.1.3. *For any f, we define the quantity*

$$\mathsf{MIC}(f) = \inf_{\pi \text{ computing } f} \sup_{\mu} \mathsf{MIC}_\mu(\pi).$$

3.1.3 Properties

The multi-party information cost can be used as a lower bound on the communication complexity.

Theorem 3.1.4. *For any k-player function f,*

$$\mathsf{CC}(f) \geq \frac{1}{8}\mathsf{MIC}(f) - k^2.$$

Proof. Let π be any k-player communication protocol, and let μ be an arbitrary input distribution.

$$\begin{aligned}\mathsf{MIC}_\mu(\pi) &= \sum_{i=1}^{k}(I(X_i;\Pi_i \mid X_{-i}R_{-i}) + I(X_{-i};\Pi_i \mid X_iR_i)) \\ &\leq 2\sum_{i=1}^{k} H(\Pi_i) \text{ (Proposition 1.2.2).}\end{aligned}$$

We first encode Π_i, for any i, into a variable Π'_i such that the set of possible values of Π'_i is a prefix-free set of strings. Observe that the transcript Π_i is composed of a number of basic transcripts: for every $j \in [\![1,k]\!] \setminus \{i\}$, a pair of transcripts of messages, $\Pi^s_{i,j}$, $\Pi^r_{i,j}$ containing the messages sent by player i to player j, and the messages read by player i from player j, respectively. We convert Π_i into Π'_i as follows: In each one of the above $2(k-1)$ components we replace every bit $b \in \{0,1\}$ by $b \cdot b$, and then add at the end of the component the two bits 01. We then concatenate all components in order. This a one-to-one encoding, and the set of possible values of Π'_i is a prefix-free set of strings.

Defining $|\Pi_i| = \sum\limits_{j \neq i} |\Pi^s_{i,j}| + |\Pi^r_{i,j}|$ and $|\Pi| = \sum\limits_{i=1}^{k} |\Pi_i|$, we have $H(\Pi'_i) = H(\Pi_i)$, and $\mathbb{E}[|\Pi'_i|] = 2\,\mathbb{E}[|\Pi_i|] + 4(k-1)$. We get

$$\begin{aligned}
\mathsf{MIC}_\mu(\pi) &\leq 2 \sum_{i=1}^{k} H(\Pi'_i) \\
&\leq 2 \sum_{i=1}^{k} \mathbb{E}[|\Pi'_i|] \qquad \text{(by Theorem 1.2.3)} \\
&\leq 2 \sum_{i=1}^{k} (2\,\mathbb{E}[|\Pi_i|] + 4(k-1)) \\
&\leq 4 \cdot \mathbb{E}[|\Pi|] + 8k^2 \\
&\leq 8 \cdot \mathsf{CC}(\pi) + 8k^2,
\end{aligned}$$

where the last factor 2 is due to the fact that each message sent from player i to player j may appear in at most 2 basic transcripts, namely $\Pi^s_{i,j}$ and $\Pi^r_{j,i}$. As this inequality is true for any distribution μ, and for any protocol π computing f, this concludes the proof.

⌟

The multi-party information cost satisfies a direct sum property for product distributions.

Theorem 3.1.5. *For any protocol π (externally) ϵ-computing a function $f^{\otimes n}$, there exists a protocol π' (externally) ϵ-computing f such that, for any input product distribution μ of f,*

$$\mathsf{MIC}_{\mu^n}(\pi) \geq n \cdot \mathsf{MIC}_\mu(\pi').$$

Thus $\mathsf{MIC}^\epsilon(f^{\otimes n}) \geq n \cdot \mathsf{MIC}^\epsilon(f)$.

Proof. In this proof, we make the public randomness appear explicitly in the conditioning of the information terms. We denote in π' by R'^p the public randomness in π', and by R'_i the private randomness of the players in π'. We define π' on input $(Y_i)_{i\in[\![1,k]\!]}$ as follows.

The players first publicly sample a random index L uniformly in $[\![1,n]\!]$ and define $X_i^L = Y_i$. The players then publicly sample X^d for every $d < L$. Each player i then samples privately, for every $d > L$, X_i^d according to μ. The players then run π on input X. They output as the output of π' the L^{th} coordinate of the output of π. Observe that π' has error at most ϵ, and that if the input to π' is distributed according to μ, then the input of π is distributed according to μ^n. Note that there is no extra communication in π' compared to π, just some (private and public) sampling. Therefore we have $\Pi'_i = \Pi_i$ for every $i \in [\![1,k]\!]$. We further denote by R^p the random bits of R'^p beyond those used for the sampling at the start of π'. Similarly, we denote by R_i, $i \in [\![1,k]\!]$, the random bits of R'_i beyond those used for the sampling at the start of π'.

We show that $\mathsf{MIC}_\mu(\pi') = \dfrac{1}{n}\mathsf{MIC}_{\mu^n}(\pi)$. Recall that

$$\mathsf{MIC}(\pi') = \sum_{i=1}^{k} \left(I(Y_{-i};\Pi'_i \mid Y_iR'_iR^p) + I(Y_i;\Pi'_i \mid Y_{-i}R'_{-i}R'^p) \right).$$

Note that the input X on which we run the protocol π follow the product distribution μ^n, and the real input Y is thus indistinguishable from the sampled inputs X^{-L}. For this reason, we can omit the value of L from the mutual information terms which follow. We have, for every player i,

$$\begin{aligned}
I(Y_{-i};\Pi'_i \mid Y_iR'_iR'^p) &= \mathbb{E}_l[I(X^l_{-i};\Pi_i \mid X^l_iX^{>l}_iR_iX^{<l}R^p)] \\
&\quad \text{(making explicit the sampling from } R'_i\text{,}R'^p\text{)} \\
&= \mathbb{E}_l[I(X^l_{-i};\Pi_i \mid X^l_iX^{>l}_iR_iX^{<l}_iX^{<l}_{-i}R^p)] \\
&= \mathbb{E}_l[I(X^l_{-i};\Pi_i \mid X_iR_iR^pX^{<l}_{-i})] \\
&= \frac{1}{n}\sum_l[I(X^l_{-i};\Pi_i \mid X_iR_iR^pX^{<l}_{-i})] \\
&= \frac{1}{n}I(X_{-i};\Pi_i \mid X_iR_iR^p) \\
&\quad \text{(Chain rule, Proposition 1.2.9)}
\end{aligned}$$

and

$$
\begin{aligned}
I(Y_i; \Pi'_i \mid Y_{-i} R_{-i} R^p) &= \mathop{\mathbb{E}}_{l}[I(X_i^l; \Pi_i \mid X_{-i}^l X_{-i}^{>l} R_{-i} X^{<l} R^p)] \\
&\quad \text{(making explicit the sampling from } R'_{-i}, R'^p\text{)} \\
&= \mathop{\mathbb{E}}_{l}[I(X_i^l; \Pi_i \mid X_{-i}^l X_{-i}^{>l} R_{-i} X_{-i}^{<l} X_i^{<l} R^p)] \\
&= \mathop{\mathbb{E}}_{l}[I(X_i^l; \Pi_i \mid X_{-i} R_{-i} R^p X_i^{<l})] \\
&= \frac{1}{n} \sum_{l} [I(X_i^l; \Pi_i \mid X_{-i} R_{-i} R^p X_i^{<l})] \\
&= \frac{1}{n} I(X_i; \Pi_i \mid X_{-i} R_{-i} R^p) \\
&\quad \text{(Chain rule, Proposition 1.2.9).}
\end{aligned}
$$

Summing over $i \in [\![1, k]\!]$ concludes the proof.

⌟

3.2 The parity function

We are able to bound the multi-party information cost of the parity function Par_k^n defined in Section 2.5.

Theorem 3.2.1. *Let μ be the uniform distribution on $\{0,1\}^k$. For any fixed $\epsilon \in \left[0, \frac{1}{2}\right[$,*

$$\mathsf{MIC}_\mu^\epsilon(\mathsf{Par}_k) = \Omega(k).$$

Proof. Let π be a k-player protocol ϵ-computing the function $f = \mathsf{Par}_k$.

$$\begin{aligned}
\mathsf{MIC}(\pi) &= \sum_{i=1}^{k} \left(I(X_{-i}; \Pi_i \mid X_i R_i) + I(X_i; \Pi_i \mid X_{-i} R_{-i})\right) \\
&\geq \sum_{i=2}^{k} I(X_i; \Pi_i \mid X_{-i} R_{-i}) \\
&\geq \sum_{i=2}^{k} (I(X_i; \Pi_i \mid X_{-i} R_{-i}) + I(X_i; \Pi_1 \mid X_{-i} R_{-i} \Pi_i)) \\
&\quad \text{(as } H(\Pi_1 \mid X_{-i} R_{-i} \Pi_i) = 0) \\
&\geq \sum_{i=2}^{k} I(X_i; \Pi_1 \Pi_i \mid X_{-i} R_{-i}) \text{ (Chain rule, Proposition 1.2.9)} \\
&\geq \sum_{i=2}^{k} I(X_i; \Pi_1 \mid X_{-i} R_{-i}) \\
&\geq \sum_{i=2}^{k} (H(X_i \mid X_{-i} R_{-i}) - H(X_i \mid X_{-i} R_{-i} \Pi_1)) \\
&\geq \sum_{i=2}^{k} (1 - H(X_i \mid X_{-i} R_{-i} \Pi_1)) \text{ (as } \mu \text{ is the uniform distribution)} \\
&\geq \sum_{i=2}^{k} (1 - H(f(X) \mid X_{-i} R_{-i} \Pi_1)) \\
&\quad \text{(data processing inequality, as } \exists\ \Phi \mid X_i = \Phi(f(X), X_{-i})) \\
&\geq \sum_{i=2}^{k} (1 - H(f(X) \mid X_1 R_1 \Pi_1)) \\
&\geq (k-1)(1 - H(f(X) \mid X_1 R_1 \Pi_1)) \\
&\geq (k-1)(1 - h(\epsilon)) \\
&\quad \text{(by Lemma 1.4.4, as player 1 outputs } f \text{ with error } \epsilon)
\end{aligned}$$

where h is the binary entropy function.

⌟

Theorem 3.2.2. *Let μ be the uniform distribution on $\{0,1\}^k$. For any fixed $\epsilon \in \left[0, \frac{1}{2}\right[$,*

$$\mathsf{MIC}^{\epsilon}_{\mu^n}(\mathsf{Par}^n_k) = \Omega(kn).$$

Proof. It is a consequence of Theorems 3.2.1 and 3.1.5.

⌟

Theorem 3.2.3. *Given any fixed* $\epsilon \in \left[0, \frac{1}{2}\right[$*, there is a constant* α *such that for* $n \geq \frac{1}{\alpha}k$*,*

$$\mathsf{CC}^{\epsilon}(\mathsf{Par}_k^n) = \Omega(kn).$$

Proof. Let π be a protocol ϵ-computing Par_k^n. By Theorems 3.1.4 and 3.2.2, there exists a constant β such that $\mathsf{CC}(\pi) \geq \beta kn - k^2$. Let a constant $\alpha < \beta$. For $n \geq \frac{1}{\alpha}k$, we have $k^2 \leq \alpha kn$ and we get $\mathsf{CC}(\pi) \geq (\beta - \alpha)kn = \Omega(kn)$.

⌟

3.3 The disjointness function

In all this section, we work with the *return* variant of our model, which has been defined in Subsection 1.4.1.

3.3.1 The disjointness problem

In the `set-disjointness` problem (DISJ_k^n), there are k players, each having as input a subset of a base set of n elements, and the goal is to decide whether there is an element which belongs to all these subsets. This problem has been the subject of a large number of studies in communication complexity, and is often seen as a test for our ability to give lower bounds in a given model (cf. [CP10]). Its complexity in the two-party case is well understood [KS92, Raz92, BYJKS02, Bra12, BGPW13]. In the broadcast model, a promise version of the disjointness function was studied in [AMS96]. Afterwards, a sequence of improved bounds were obtained in [BYJKS02, CKS03, Gro09] through the use of information theory (cf. also [Jay09, BO17]). External information complexity was also used in [BO15] to prove a lower bound on the general disjointness function. In the coordinator model [BEO$^+$13] gave lower bounds on the disjointness problem via variants of information complexity. The latter result was extended in [CM15] to the function *Tribes*.

We consider the k-party function AND_k, where each player has an input bit x_i, and where the protocol has to compute the AND of all the input bits. In this section we will prove lower bounds on the complexity of the disjointness function DISJ_k^n, in which every player $i \in [\![1, k]\!]$ has an n-bit string $(x_i^l)_{l \in [\![1,n]\!]}$, and the players have to output 1 if and only if there exists a coordinate l where all players have the bit 1. Formally, $\mathsf{DISJ}_k^n(x) = \bigvee_{l=1}^{n} \bigwedge_{i=1}^{k} x_i^l$.

We describe the input distribution μ on $\{0,1\}^k$ that we will work with. It is similar to the definition of [BEO$^+$13], with a little twist inspired by [JKS03]. Our distribution is defined as follows. Draw a bit $M \sim \mathbf{Ber}(\frac{2}{3}, \frac{1}{3})$, and a uniformly random index $Z \in [\![1,k]\!]$. Assign 0 to X_Z. If $M = 0$, sample X_{-Z} uniformly in $\{0,1\}^{k-1}$; if $M = 1$, assign 1^{k-1} to X_{-Z}. We will then work with the product distribution μ^n. Our distribution is similar to the ones of [BEO$^+$13] in the sense that it will lead to a high information cost for the function AND_k. However, our distribution, as the one of [CM15], has an additional property: the AND of any input drawn from μ is 0. The distribution μ^n is said to be *collapsing*. As we will show in Subsection 3.3.4, this will allow us to prove a direct sum in full generality and get lower bounds for the Disjointness function without having to impose the constraint $k = \Omega(\log(n))$ (cf. the discussion on the reduction of Section 4 of [BEO$^+$13] which explains why they had to impose the constraint $k = \Omega(\log(n))$).

Given a protocol π, let $\Pi_i[x_i, m, z]$ denote the distribution of Π_i, when the input X is sampled as follows: $X \sim \mu$, conditioned on the fact that $X_i = x_i$, $M = m$ and $Z = z$.

A variant of the notion of information cost for two-party protocols for the study of the disjointness function in the coordinator model was introduced in [BEO$^+$13] under the name of *switched information cost*. We use here a similar measure adapted to our setting and distribution, the *switched multi-party information cost* (SMIC).

Definition 3.3.1. *For a k-player protocol π with inputs drawn from μ^n.*

$$\mathsf{SMIC}_{\mu^n}(\pi) = \sum_{i=1}^{k} (I(X_i; \Pi_i \mid MZ) + I(M; \Pi_i \mid X_i Z)).$$

Note that the notion of SMIC is only defined relatively to the distribution μ^n, and we may thus omit the distribution in the notation. The presence of the public randomness is implicit here. It can be materialized indifferently either as part of the transcript or in the conditioning of the information-theoretic expressions.

We first prove some technical tools in Subsection 3.3.2. In Subsection 3.3.3, we will prove a lower bound on the switched multi-party information cost of the function AND_k. While the general structure of the proof of this lower bound is similar to the one in the coordinator model, given in

[BEO$^+$13],[1] we do have to overcome a number of difficulties, both technical and more fundamental, that require new ideas and new proofs: The very basic *rectangularity* property of communication protocols is, in the multi-party setting, very sensitive to the definition of the model and the notion of a transcript. Then, in Subsection 3.3.4, we will prove a direct sum theorem which will allow us to prove a lower bound on the switched multi-party information cost of the disjointness function DISJ_k^n. The fact that our model does not have a "coordinator" requires one to define a more elaborate reduction protocol compared to [BEO$^+$13], together with a more complicated proof for the direct-sum argument. Indeed, since we do not have a coordinator that can sample privately "dummy inputs", we need to use "distributed sampling", inspired by classic secret-sharing primitives, and prove that nevertheless a direct-sum property holds with respect to the information. In Subsection 3.3.5, we will show that SMIC provides a lower bound on MIC. Last, in Subsection 3.3.6, we will see how this bound translates to the public information cost of the function DISJ_k^n.

3.3.2 Technical lemmas

Rectangularity The *rectangularity property* (or *Markov property*) is one of the key properties that follow from the structure and definition of a protocol. For randomized protocols it was introduced in the two-party setting and in the multi-party blackboard model in [BYJKS02], and in the coordinator model in [BEO$^+$13]. We prove a similar rectangularity property adapted to the peer-to-peer model that we consider in the present paper.

To define this property, for any transcript $\overline{\tau} \in \mathcal{T}_i$, let

$$\mathcal{A}_i(\overline{\tau}) = \{(x, r) \mid \Pi_i(x, r) = \overline{\tau}\},$$

and define the projection of $\mathcal{A}_i(\overline{\tau})$ on coordinate i as

$$\mathcal{I}_i(\overline{\tau}) = \{(x', r'), \exists\ (x, r) \in \mathcal{A}_i(\overline{\tau}), x' = x_i \ \&\ r' = r_i\},$$

and the projection of $\mathcal{A}_i(\overline{\tau})$ on the complement of coordinate i as

$$\mathcal{J}_i(\overline{\tau}) = \{(x', r'), \exists\ (x, r) \in \mathcal{A}_i(\overline{\tau}), x' = x_{-i} \ \&\ r' = r_{-i}\}.$$

Similarly, for any transcript $\tau \in \mathcal{T}$, let

$$\mathcal{B}(\tau) = \{(x, r) \mid \Pi(x, r) = \tau)\},$$

[1] The lower bound in [BEO$^+$13] would yield an $\Omega(\frac{1}{\log k} \cdot nk)$ lower bound for *Disjointness* in the peer-to-peer setting.

and for any player i,

$$\mathcal{H}_i(\tau) = \{(x', r'), \exists\ (x, r) \in \mathcal{B}(\tau), x' = x_{-i}\ \&\ r' = r_{-i}\}.$$

We start by proving a combinatorial property of transcripts of communication protocols, which intuitively follows from the fact that each player has access to only its own input. The proof of this property is technically more involved compared to the analogous property in other settings, since the structure of protocols and the manifestation of the transcripts in the peer-to-peer setting are more flexible than in the other settings.

Lemma 3.3.2. *Let π be a private-coins k-player protocol with inputs from $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_k$. $\forall\ i \in [\![1, k]\!]$:*

- $\forall\ \overline{\tau} \in \mathcal{T}_i,\ \ \mathcal{A}_i(\overline{\tau}) = \mathcal{I}_i(\overline{\tau}) \times \mathcal{J}_i(\overline{\tau})$.
- $\forall\ \tau \in \mathcal{T},\ \ \mathcal{B}(\tau) = \mathcal{I}_i(\tau_i) \times \mathcal{H}_i(\tau)$.

Proof. We start by proving the first claim. Since the other inclusion is immediate from the definition, we only need to show that

$$\forall\ \overline{\tau} \in \mathcal{T}_i,\ \mathcal{I}_i(\overline{\tau}) \times \mathcal{J}_i(\overline{\tau}) \subseteq \mathcal{A}_i(\overline{\tau}).$$

To this end take an arbitrary $(x_i, r_i) \in \mathcal{I}_i(\overline{\tau})$ and an arbitrary $(x_{-i}, r_{-i}) \in \mathcal{J}_i(\overline{\tau})$. Since $(x_i, r_i) \in \mathcal{I}_i(\overline{\tau})$, we have that

$$\exists\ (\tilde{x}, \tilde{r}) \in \mathcal{A}_i(\overline{\tau}) \mid x_i = \tilde{x}_i\ \&\ r_i = \tilde{r}_i.$$

Similarly, since $(x_{-i}, r_{-i}) \in \mathcal{J}_i(\overline{\tau})$,

$$\exists\ (\hat{x}, \hat{r}) \in \mathcal{A}_i(\overline{\tau}) \mid x_{-i} = \hat{x}_{-i}\ \&\ r_{-i} = \hat{r}_{-i}.$$

Let (x, r) be $((x_i, r_i), (x_{-i}, r_{-i})) \in \mathcal{I}_i(\overline{\tau}) \times \mathcal{J}_i(\overline{\tau})$. We will now show that

$$(x, r) \in \mathcal{A}_i(\overline{\tau}).$$

Let L be the number of local rounds of player i in the run of π on input (x, r). We will show by induction on the index of the local round of player i that for any $\ell \leq L$, $\Pi_i^\ell(x, r) = \Pi_i^\ell(\tilde{x}, \tilde{r})$. Observe that whether or not a player stops and returns its output at a given round is a function of its input, its private randomness and its transcript until that round. Therefore, since player i stops and returns its value at local round L if the input is (x, r), it will follow from $\Pi_i^L(x, r) = \Pi_i^L(\tilde{x}, \tilde{r})$ that player i stops and returns its output at local round L also when the input is $(\tilde{x}, \tilde{r})$. We will thus get that $\Pi_i(x, r) = \overline{\tau}$, and hence $(x, r) \in \mathcal{A}_i(\overline{\tau})$.

The base of the induction, for $l = 0$, follows since the transcript is empty. We now prove the claim for $l + 1 \leq L$, based on the induction hypothesis that the claim holds for l.[2]

The messages that player i sends at local round $l + 1$ are a function of x_i, r_i and $\Pi_i^l(x, r)$. As $x_i = \tilde{x}_i$ & $\tilde{r}_i = r_i$, and using the induction hypothesis, we get that the messages sent by player i at local round $l + 1$ are the same in $\pi_i(x, r)$ and in $\pi_i(\tilde{x}, \tilde{r})$.

For the same reason we also get that the set of players from which player i waits for a message at round $l+1$ is the same when π is run on input (x, r) and on input $(\tilde{x}, \tilde{r})$.

We now claim that the messages read by player i at round $l + 1$ are the same when π is run on input (x, r) and on input $(\tilde{x}, \tilde{r})$. To this end we define an imaginary "protocol" ψ where player i sends in its first local round all the messages that it sends in $\overline{\tau}$ and the players in $Q_i = \llbracket 1, k \rrbracket \setminus \{i\}$ run π.[3] Player i sends the messages on each link according to the order in $\overline{\tau}$.[4] The messages that the players in Q_i send in each of their local rounds are a function of their inputs, their local randomness, and the messages they read from the links that connect to player i. Since $\Pi(\hat{x}, \hat{r}) = \overline{\tau}$ we can conclude that in ψ (when the input is $(\hat{x}, \hat{r})$) the messages sent by the players in Q_i (in particular, to player i) are the same as those sent in π on input $(\hat{x}, \hat{r})$.

Recall that we have proved above that when π is run on (x, r), the messages player i sends up to round $l+1$ are consistent with $\overline{\tau}$. We therefore can consider now a "protocol" ψ' which is the same as ψ with the only difference that player i sends (in its first local round) only the messages of $\overline{\tau}$ it would have sent in $\pi(x, r)$ until (and including) round $l+1$ (and not all the message it sends in $\overline{\tau}$). It follows that in ψ', when run on input $(\hat{x}, \hat{r})$, the sequences of messages sent from the players in Q_i to i are a prefix of the sequences they send in ψ. Since $x_{-i} = \hat{x}$ and $r_{-i} = \hat{r}$, the same claim holds when ψ' is run on (x, r). Observe now that when π is run on (x, r), at the time where player i is waiting at local round $l + 1$ for incoming messages, it has sent exactly the messages that player i sends in ψ'.

Using the induction hypothesis $\Pi_i^l(x, r) = \Pi_i^l(\tilde{x}, \tilde{r})$, that $x_i = \tilde{x}_i$, $r_i = \tilde{r}_i$ and the fact that the set of players from which player i waits for a message at

[2] Note that $\Pi_i(x, r)$ by itself does not define which messages are sent/read in which local round.

[3] Technically speaking, this is not a protocol according to our definition as more than one message may be sent in a single round on a single link.

[4] Recall that a transcript of a player is a $2(k - 1)$-tuple of transcripts, one for each of its $2(k - 1)$ directed links.

local round $l+1$ is the same for input (x,r) and $(\tilde{x},\tilde{r})$, we can conclude that the messages that player i reads while waiting for messages at local round $l+1$ when π is run on (x,r) are consistent with the messages it would read when π is run on $(\tilde{x},\tilde{r})$. Since player i running π must, by the definition of a protocol, reach its "return" statement, it must receive messages from all the players it is waiting for. We therefore conclude that the messages read by player i in local round $l+1$ when π is run on (x,r) are the same as those it reads when run on $(\tilde{x},\tilde{r})$.

Together with the induction hypothesis, and the fact (proved above) that the messages sent by player i at local round $l+1$ are the same when π is run on in (x,r) and on $(\tilde{x},\tilde{r})$, we have that $\Pi_i^{l+1}(x,r) = \Pi_i^{l+1}(\tilde{x},\tilde{r})$.

We now prove the second claim. We only need to show that

$$\forall\ \tau \in \mathcal{T},\ \mathcal{I}_i(\tau_i) \times \mathcal{H}_i(\tau) \subseteq \mathcal{B}(\tau),$$

the other inclusion being immediate from the definitions, since $\mathcal{B}(\tau) \subseteq \mathcal{A}_i(\tau_i)$.

Take an arbitrary $(x_i, r_i) \in \mathcal{I}_i(\tau_i)$ and an arbitrary $(x_{-i}, r_{-i}) \in \mathcal{H}_i(\tau)$. Since $(x_{-i}, r_{-i}) \in \mathcal{H}_i(\tau)$, $\exists\ (\hat{x},\hat{r})$ such that $\pi(\hat{x},\hat{r}) = \tau$, $x_{-i} = \hat{x}_{-i}$ and $r_{-i} = \hat{r}_{-i}$. Let $(x,r) = ((x_i,r_i),(x_{-i},r_{-i}))$. Since $\mathcal{B}(\tau) \subseteq \mathcal{A}(\tau_i)$, we have $\mathcal{H}_i(\tau) \subseteq \mathcal{J}_i(\tau_i)$. Thus, using the first claim,

$$\mathcal{I}_i(\tau_i) \times \mathcal{H}_i(\tau) \subseteq \mathcal{I}_i(\tau_i) \times \mathcal{J}_i(\tau_i) \subseteq \mathcal{A}_i(\tau_i),$$

and $\Pi_i(x,r) = \tau_i$. It remains to show that $\forall\ j \neq i,\ \Pi_j(x,r) = \tau_j$.

Consider the two runs of protocol π on the input (x,r) and on the input $(\hat{x},\hat{r})$. We have that $\Pi(\hat{x},\hat{r}) = \tau$, and that $\Pi_i(x,r) = \tau_i$. Since $x_{-i} = \hat{x}_{-i}$ and $r_{-i} = \hat{r}_{-i}$, we have that also for all $j \neq i$ $\Pi_j(x,r) = \Pi_j(\hat{x},\hat{r}) = \tau_j$. It follows that $(x,r) \in \mathcal{B}(\tau)$ as needed.

⌟

We now prove the *rectangularity property of randomized protocols* in the peer-to-peer setting.

Lemma 3.3.3. *Let π be a private-coins k-player protocol with inputs from $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_k$. For every $i \in [\![1,k]\!]$, there exist functions $q_i : \mathcal{X}_i \times \mathcal{T}_i \to [0,1]$, $q_{-i} : \mathcal{X}_{-i} \times \mathcal{T}_i \to [0,1]$ and $p_{-i} : \mathcal{X}_{-i} \times \mathcal{T} \to [0,1]$ such that*

$$\forall\ x \in \mathcal{X}, \forall\ \tau = (\tau_1, \ldots, \tau_k) \in \mathcal{T}, \Pr[\Pi_i(x) = \tau_i] = q_i(x_i,\tau_i) q_{-i}(x_{-i},\tau_i),$$

and

$$\forall\ x \in \mathcal{X}, \forall\ \tau = (\tau_1, \ldots, \tau_k) \in \mathcal{T}, \Pr[\Pi(x) = \tau] = q_i(x_i,\tau_i) p_{-i}(x_{-i},\tau).$$

Proof. We prove the claim for an arbitrary player $i \in [\![1,k]\!]$. Define, for $\tilde{x} \in \mathcal{X}_i$ and $\overline{\tau} \in \mathcal{T}_i$,

$$q_i(\tilde{x}, \overline{\tau}) = \Pr[(\tilde{x}, R_i) \in \mathcal{I}_i(\overline{\tau})],$$

and for $\hat{x} \in \mathcal{X}_{-i}$ and $\overline{\tau} \in \mathcal{T}_i$,

$$q_{-i}(\hat{x}, \overline{\tau}) = \Pr[(\hat{x}, R_{-i}) \in \mathcal{J}_i(\overline{\tau})].$$

We have, for $x \in \mathcal{X}$ and $\overline{\tau} \in \mathcal{T}_i$,

$$\begin{aligned} \Pr[\Pi_i(x) = \overline{\tau}] &= \Pr[(x, R) \in \mathcal{A}_i(\overline{\tau})] \\ &= \Pr[(x_i, R_i) \in \mathcal{I}_i(\overline{\tau}) \ \& \ (x_{-i}, R_{-i}) \in \mathcal{J}_i(\overline{\tau})] \\ &\quad \text{(by Lemma 3.3.2)} \\ &= \Pr[(x_i, R_i) \in \mathcal{I}_i(\overline{\tau})] \times \Pr[(x_{-i}, R_{-i}) \in \mathcal{J}_i(\overline{\tau})] \\ &= q_i(x_i, \overline{\tau}) q_{-i}(x_{-i}, \overline{\tau}). \end{aligned}$$

We now prove the second claim. Define, for $\hat{x} \in \mathcal{X}_{-i}$ and $\tau \in \mathcal{T}$,

$$p_{-i}(\hat{x}, \tau) = \Pr[(\hat{x}, R_{-i}) \in \mathcal{H}_i(\tau)].$$

We have, for $x \in \mathcal{X}$ and $\tau \in \mathcal{T}$,

$$\begin{aligned} \Pr[\Pi(x) = \tau] &= \Pr[(x, R) \in \mathcal{B}(\tau))] \\ &= \Pr[(x_i, R_i) \in \mathcal{I}_i(\tau_i) \ \& \ (x_{-i}, R_{-i}) \in \mathcal{H}_i(\tau))] \\ &\quad \text{(by Lemma 3.3.2)} \\ &= \Pr[(x_i, R_i) \in \mathcal{I}_i(\tau_i)] \times \Pr[(x_{-i}, R_{-i}) \in \mathcal{H}_i(\tau))] \\ &= q_i(x_i, \tau_i) p_{-i}(x_{-i}, \tau). \end{aligned}$$

⌟

The following lemma is an application of Lemma 3.3.3 to the specific case of the distribution μ that we have defined above.

Lemma 3.3.4. *Let* π *be a private-coins protocol. There exists a function* $c : \{0,1\} \times [\![1,k]\!] \times \mathcal{T} \to [0,1]$*, and for every* $i \in [\![1,k]\!]$ *there is a function* $c_i : \{0,1\} \times [\![1,k]\!] \times \mathcal{T}_i \to [0,1]$*, such that* $\forall\ i \in [\![1,k]\!]$*,* $\forall\ x' \in \{0,1\}$*,* $\forall\ m \in \{0,1\}$*,* $\forall\ z \in [\![1,k]\!] \setminus \{i\}$*,* $\forall\ \tau = (\tau_1, \ldots, \tau_k) \in \mathcal{T}$*,*

$$\Pr[\Pi_i = \tau_i \mid X_i = x', M = m, Z = z] = q_i(x', \tau_i) c_i(m, z, \tau_i)$$

and

$$\Pr[\Pi = \tau \mid X_i = x', M = m, Z = z] = q_i(x', \tau_i) c(m, z, \tau).$$

Proof. The term $\Pr[\Pi_i = \tau_i \mid X_i = x', M = m, Z = z]$ is equal to

$$\sum_{x\in\{0,1\}^k} (\Pr[X = x \mid X_i = x', M = m, Z = z]\times$$
$$\Pr[\Pi_i = \tau_i \mid X = x, X_i = x', M = m, Z = z]).$$

Note that

$$\Pr[X = x \mid X_i = x', M = m, Z = z] = \delta_{x_i,x'} \Pr[X_{-i} = x_{-i} \mid M = m, Z = z],$$

since, conditioned on $M = m, Z = z$, X_i and X_{-i} are independent. Further note that for x such that $x_i = x'$,

$$\Pr[\Pi_i = \tau_i \mid X = x, X_i = x', M = m, Z = z] = \Pr[\Pi_i(x) = \tau_i].$$

By Lemma 3.3.3, there exist functions q_i and q_{-i} such that

$$\forall\ x \in \{0,1\}^k,\ \ \Pr[\Pi_i(x) = \tau_i] = q_i(x_i, \tau_i)q_{-i}(x_{-i}, \tau_i).$$

Therefore we can write

$$\begin{aligned}\Pr[\Pi_i = \tau_i \mid X_i = x, M = m, Z = z] &= \sum_{x\in\{0,1\}^k} (\delta_{x_i,x'} q_i(x_i, \tau_i)q_{-i}(x_{-i}, \tau_i)\times \\ &\qquad \Pr[X_{-i} = x_{-i} \mid M = m, Z = z]) \\ &= q_i(x', \tau_i) \sum_{\hat{x}\in\{0,1\}^{k-1}} (q_{-i}(\hat{x}, \tau_i)\times \\ &\qquad \Pr[X_{-i} = \hat{x} \mid M = m, Z = z]) \\ &= q_i(x', \tau_i)c_i(m, z, \tau_i)\end{aligned}$$

where $c_i(m, z, \tau_i) = \sum_{\hat{x}\in\{0,1\}^{k-1}} q_{-i}(\hat{x}, \tau_i) \Pr[X_{-i} = \hat{x} \mid M = m, Z = z]$.

The proof of the second statement is similar:
the term $\Pr[\Pi = \tau \mid X_i = x, M = m, Z = z]$ is equal to

$$\sum_{x\in\{0,1\}^k} (\Pr[X = x \mid X_i = x', M = m, Z = z]\times$$
$$\Pr[\Pi = \tau \mid X = x, X_i = x', M = m, Z = z]).$$

Note that

$$\Pr[X = x \mid X_i = x', M = m, Z = z] = \delta_{x_i,x'} \Pr[X_{-i} = x_{-i} \mid M = m, Z = z],$$

since, conditioned on $M = m, Z = z,$, X_i and X_{-i} are independent. Further note that for x such that $x_i = x'$,

$$\Pr[\Pi = \tau \mid X = x, X_i = x', M = m, Z = z] = \Pr[\Pi(x) = \tau].$$

By Lemma 3.3.3, there exist functions q_i and p_{-i} such that

$$\forall\ x \in \{0,1\}^k,\ \ \Pr[\Pi(x) = \tau] = q_i(x_i, \tau_i)p_{-i}(x_{-i}, \tau).$$

Therefore we can write

$$\begin{aligned}\Pr[\Pi = \tau \mid X_i = x', M = m, Z = z] &= \sum_{x \in \{0,1\}^k} (\delta_{x_i,x'} q_i(x_i, \tau_i) p_{-i}(x_{-i}, \tau) \times \\ &\qquad \Pr[X_{-i} = x_{-i} \mid M = m, Z = z]) \\ &= q_i(x', \tau_i) \sum_{\hat{x} \in \{0,1\}^{k-1}} (p_{-i}(\hat{x}, \tau) \times \\ &\qquad \Pr[X_{-i} = \hat{x} \mid M = m, Z = z]) \\ &= q_i(x', \tau_i) c(m, z, \tau)\end{aligned}$$

where $c(m, z, \tau) = \sum_{\hat{x} \in \{0,1\}^{k-1}} p_{-i}(\hat{x}, \tau) \Pr[X_{-i} = \hat{x} \mid M = m, Z = z]$.

┘

The Diagonal Lemma The following lemma is often called the *diagonal lemma.* It was proved in [BYJKS02] for the two-party setting under the name of *Pythagorean lemma*, and in [BEO$^+$13] for the coordinator model. We show here that is also holds in the peer-to-peer model. The proof of this lemma does not (directly) use the properties of a protocol, and in fact follows from Lemma 3.3.3 and Proposition 1.2.20 in the same way that its two-party analogue follows from the analogous lemma and proposition.

Lemma 3.3.5. *Let π be a private-coins protocol taking input in $\{0,1\}^k$. $\forall\ x \in \{0,1\}^k$, $\forall\ y \in \{0,1\}^k$, $\forall\ i \in [\![1, k]\!]$,*

$$h(\Pi(x), \Pi(y))^2 \geq \frac{1}{2}\left[h(\Pi(x), \Pi(y_{[i \leftarrow x_i]}))^2 + h(\Pi(x_{[i \leftarrow y_i]}), \Pi(y))^2\right].$$

Proof. In what follows we simplify notation and write $\sum_{\tau}$ instead of $\sum_{\tau \in \mathcal{T}}$.

Using Proposition 1.2.20,

$$
\begin{aligned}
1 - h^2(\Pi(x), \Pi(y)) &= \sum_\tau \sqrt{\Pr[\Pi(x) = \tau] \Pr[\Pi(y) = \tau]} \\
&= \sum_\tau \sqrt{q_i(x_i, \tau_i) p_{-i}(x_{-i}, \tau) q_i(y_i, \tau_i) p_{-i}(y_{-i}, \tau)} \\
&\text{(using Lemma 3.3.3)} \\
&= \sum_\tau \sqrt{q_i(x_i, \tau_i) q_i(y_i, \tau_i)} \sqrt{p_{-i}(x_{-i}, \tau) p_{-i}(y_{-i}, \tau)} \\
&\leq \sum_\tau \frac{q_i(x_i, \tau_i) + q_i(y_i, \tau_i)}{2} \sqrt{p_{-i}(x_{-i}, \tau) p_{-i}(y_{-i}, \tau)} \\
&\leq \frac{1}{2} \left(\sum_\tau \sqrt{q_i(x_i, \tau_i) p_{-i}(x_{-i}, \tau) q_i(x_i, \tau_i) p_{-i}(y_{-i}, \tau)} + \right. \\
&\qquad \left. \sum_\tau \sqrt{q_i(y_i, \tau_i) p_{-i}(x_{-i}, \tau) q_i(y_i, \tau_i) p_{-i}(y_{-i}, \tau)} \right) \\
&\leq \frac{1}{2} \left(\sum_\tau \sqrt{\Pr[\Pi(x) = \tau] \Pr[\Pi(y_{[i \leftarrow x_i]}) = \tau]} + \right. \\
&\qquad \left. \sum_\tau \sqrt{\Pr[\Pi(x_{[i \leftarrow y_i]}) = \tau] \Pr[\Pi(y) = \tau]} \right) \\
&\leq \frac{1}{2} \left[1 - h^2(\Pi(x), \Pi(y_{[i \leftarrow x_i]})) + 1 - h^2(\Pi(x_{[i \leftarrow y_i]}), \Pi(y)) \right] \\
&\leq 1 - \frac{1}{2} \left[h^2(\Pi(x), \Pi(y_{[i \leftarrow x_i]})) + h^2(\Pi(x_{[i \leftarrow y_i]}), \Pi(y)) \right].
\end{aligned}
$$

⌟

The following lemma is a version of the Lemma 3.3.5 adapted to our distribution.

Lemma 3.3.6. *Let π be a private-coins protocol.*

$$\forall\ i \in [\![1, k]\!], \forall\ j \in [\![1, k]\!] \setminus \{i\}, h^2(\Pi_i[0, 0, j], \Pi_i[1, 1, j]) \geq \frac{1}{2} h^2(\Pi_i(\overline{e}_{i,j}), \Pi_i(\overline{e}_j)).$$

Proof. Using Lemma 3.3.4, we write $\Pr[\Pi_i[0, 0, j] = \overline{\tau}] = q_i(0, \overline{\tau}) c_i(0, j, \overline{\tau})$ and $\Pr[\Pi_i[1, 1, j] = \overline{\tau}] = q_i(1, \overline{\tau}) c_i(1, j, \overline{\tau})$.

Using Lemma 3.3.3, we write $\Pr[\Pi_i(\overline{e}^k_{i,j}) = \overline{\tau}] = q_i(0, \overline{\tau}) q_{-i}(\overline{e}^{k-1}_j, \overline{\tau})$ and $\Pr[\Pi_i(\overline{e}^k_j) = \overline{\tau}] = q_i(1, \overline{\tau}) q_{-i}(\overline{e}^{k-1}_j, \overline{\tau})$.

Note that $\Pi_i[1,1,j] = \Pi_i(\overline{e}^k_j)$, and thus

$$q_i(1,\overline{\tau}) \neq 0 \Rightarrow c_i(1,j,\overline{\tau}) = q_{-i}(\overline{e}^{k-1}_j,\overline{\tau}).$$

By Proposition 1.2.20,

$$\begin{aligned}
1 - h^2(\Pi_i[0,0,j],\Pi_i[1,1,j]) &= \sum_{\overline{\tau}} \sqrt{\Pr[\Pi_i[0,0,j]=\overline{\tau}]\Pr[\Pi_i[1,1,j]=\overline{\tau}]} \\
&= \sum_{\overline{\tau}} \sqrt{q_i(0,\overline{\tau})c_i(0,j,\overline{\tau})q_i(1,\overline{\tau})c_i(1,j,\overline{\tau})} \\
&\leq \sum_{\overline{\tau}} \sqrt{q_i(0,\overline{\tau})q_i(1,\overline{\tau})} \left(\frac{c_i(0,j,\overline{\tau})+c_i(1,j,\overline{\tau})}{2}\right) \\
&\leq \frac{1}{2}\left(\sum_{\overline{\tau}} \sqrt{q_i(0,\overline{\tau})c_i(0,j,\overline{\tau})q_i(1,\overline{\tau})c_i(0,j,\overline{\tau})} \;+\right. \\
&\qquad \left.\sum_{\overline{\tau}} \sqrt{q_i(0,\overline{\tau})c_i(1,j,\overline{\tau})q_i(1,\overline{\tau})c_i(1,j,\overline{\tau})}\right) \\
&\leq \frac{1}{2}\left(\sum_{\overline{\tau}} \sqrt{q_i(0,\overline{\tau})c_i(0,j,\overline{\tau})q_i(1,\overline{\tau})c_i(0,j,\overline{\tau})} \;+\right. \\
&\qquad \left.\sum_{\overline{\tau}\mid q_i(1,\overline{\tau})\neq 0} \sqrt{q_i(0,\overline{\tau})c_i(1,j,\overline{\tau})q_i(1,\overline{\tau})c_i(1,j,\overline{\tau})}\right) \\
&\leq \frac{1}{2}\left(\sum_{\overline{\tau}} \sqrt{q_i(0,\overline{\tau})c_i(0,j,\overline{\tau})q_i(1,\overline{\tau})c_i(0,j,\overline{\tau})} \;+\right. \\
&\qquad \left.\sum_{\overline{\tau}\mid q_i(1,\overline{\tau})\neq 0} \sqrt{q_i(0,\overline{\tau})q_{-i}(\overline{e}^{k-1}_j,\overline{\tau})q_i(1,\overline{\tau})q_{-i}(\overline{e}^{k-1}_j,\overline{\tau})}\right) \\
&\leq \frac{1}{2}\left(\sum_{\overline{\tau}} \sqrt{q_i(0,\overline{\tau})c_i(0,j,\overline{\tau})q_i(1,\overline{\tau})c_i(0,j,\overline{\tau})} \;+\right. \\
&\qquad \left.\sum_{\overline{\tau}} \sqrt{q_i(0,\overline{\tau})q_{-i}(\overline{e}^{k-1}_j,\overline{\tau})q_i(1,\overline{\tau})q_{-i}(\overline{e}^{k-1}_j,\overline{\tau})}\right) \\
&\leq \frac{1}{2}\left(\sum_{\overline{\tau}} \sqrt{\Pr[\Pi_i[0,0,j]=\overline{\tau}]\Pr[\Pi_i[1,0,j]=\overline{\tau}]} \;+\right. \\
&\qquad \left.\sum_{\overline{\tau}} \sqrt{\Pr[\Pi_i(\overline{e}^k_{i,j})=\overline{\tau}]\Pr[\Pi_i(\overline{e}^k_j)=\overline{\tau}]}\right)
\end{aligned}$$

$$\begin{aligned}1 - h^2(\Pi_i[0,0,j], \Pi_i[1,1,j]) &\leq \frac{1}{2}(1 - h^2(\Pi_i[0,0,j], \Pi_i[1,0,j]) + \\ &\quad 1 - h^2(\Pi_i(\overline{e}^k_{i,j}), \Pi_i(\overline{e}^k_j))) \\ &\leq 1 - h^2(\Pi_i(\overline{e}^k_{i,j}), (\Pi_i(\overline{e}^k_j)).\end{aligned}$$

⌟

Localization. The following lemma formalizes the fact that if changing the input of a player changes the transcript of the protocol, then this change necessarily appears in the partial transcript of that player. For randomized protocols, this change is observed and quantified by the Hellinger distance between the distributions of the transcripts.

Lemma 3.3.7. *Let π be a private-coins protocol.*

$$\forall\ i \in [\![1,k]\!], \forall\ j \in [\![1,k]\!] \setminus \{i\}, \quad h(\Pi_i(\overline{e}_{i,j}), \Pi_i(\overline{e}_j)) = h(\Pi(\overline{e}_{i,j}), \Pi(\overline{e}_j)).$$

Proof. Using Lemma 3.3.3, we write

$$\Pr[\Pi_i(\overline{e}^k_{i,j}) = \overline{\tau}] = q_i(0,\overline{\tau}) q_{-i}(\overline{e}^{k-1}_j, \overline{\tau})$$

and

$$\Pr[\Pi(\overline{e}^k_{i,j}) = \tau] = q_i(0,\tau_i) p_{-i}(\overline{e}^{k-1}_j, \tau).$$

As $\Pr[\Pi_i(\overline{e}^k_{i,j}) = \overline{\tau}] = \sum_{\tau|\tau_i=\overline{\tau}} \Pr[\Pi(\overline{e}^k_{i,j}) = \tau]$ we have

$$q_i(0,\overline{\tau}) q_{-i}(\overline{e}^{k-1}_j, \overline{\tau}) = \sum_{\tau|\tau_i=\overline{\tau}} q_i(0,\tau_i) p_{-i}(\overline{e}^{k-1}_j, \tau) = q_i(0,\overline{\tau}) \sum_{\tau|\tau_i=\overline{\tau}} p_{-i}(\overline{e}^{k-1}_j, \tau),$$

and thus

$$q_i(0,\overline{\tau}) \neq 0 \Rightarrow q_{-i}(\overline{e}^{k-1}_j, \overline{\tau}) = \sum_{\tau|\tau_i=\overline{\tau}} p_{-i}(\overline{e}^{k-1}_j, \tau).$$

Using Proposition 1.2.20, we can write

$$\begin{aligned}
1 - h^2(\Pi_i(\overline{e}^k_{i,j}), \Pi_i(\overline{e}^k_j)) &= \sum_{\overline{\tau}} \sqrt{\Pr[\Pi_i(\overline{e}^k_{i,j}) = \overline{\tau}] \Pr[\Pi_i(\overline{e}^k_j) = \overline{\tau}]} \\
&= \sum_{\overline{\tau}} \sqrt{q_i(0,\overline{\tau}) q_{-i}(\overline{e}^{k-1}_j, \overline{\tau}) q_i(1,\overline{\tau}) q_{-i}(\overline{e}^{k-1}_j, \overline{\tau})} \\
&= \sum_{\overline{\tau}} \sqrt{q_i(0,\overline{\tau}) q_i(1,\overline{\tau})}\, q_{-i}(\overline{e}^{k-1}_j, \overline{\tau}) \\
&= \sum_{\overline{\tau} \mid q_i(0,\overline{\tau}) \neq 0} \sqrt{q_i(0,\overline{\tau}) q_i(1,\overline{\tau})}\, q_{-i}(\overline{e}^{k-1}_j, \overline{\tau}) \\
&= \sum_{\overline{\tau} \mid q_i(0,\overline{\tau}) \neq 0} \left(\sqrt{q_i(0,\overline{\tau}) q_i(1,\overline{\tau})} \sum_{\tau \mid \tau_i = \overline{\tau}} p_{-i}(\overline{e}^{k-1}_j, \tau) \right) \\
&= \sum_{\overline{\tau}} \left(\sqrt{q_i(0,\overline{\tau}) q_i(1,\overline{\tau})} \sum_{\tau \mid \tau_i = \overline{\tau}} p_{-i}(\overline{e}^{k-1}_j, \tau) \right) \\
&= \sum_{\tau} \left(\sqrt{q_i(0,\tau_i) q_i(1,\tau_i)}\, p_{-i}(\overline{e}^{k-1}_j, \tau) \right) \\
&= \sum_{\tau} \sqrt{\Pr[\Pi(\overline{e}^k_{i,j}) = \tau] \Pr[\Pi(\overline{e}^k_j) = \tau]} \\
&= 1 - h^2(\Pi(\overline{e}^k_{i,j}), \Pi(\overline{e}^k_j)).
\end{aligned}$$

⌟

3.3.3 Switched multi-party information cost of AND_k

We can now prove a lower bound on the switched multi-party information cost of the function AND_k.

Proposition 3.3.8. *For any* $\epsilon \in \left[0, \frac{1}{2}\right[$*, for any protocol* π *externally* ϵ*-computing* AND_k*,*

$$\mathsf{SMIC}_\mu(\pi) = \Omega(k).$$

Proof. We prove below the claim for an arbitrary private-coins protocol π. The claim for general protocols (i.e. with public randomness) then follows from averaging over all possible assignments of the public randomness.

Observe that for any $i \in \llbracket 1, k \rrbracket$, if $M = 0$ and $Z = z \neq i$ then $X_i \sim \mathbf{Ber}(\frac{1}{2}, \frac{1}{2})$. We therefore get by Lemma 1.2.21 that $\forall\ i \in \llbracket 1, k \rrbracket, \forall\ z \in \llbracket 1, k \rrbracket \setminus \{i\}$,

$$I(X_i; \Pi_i \mid M = 0, Z = z) \geq h^2(\Pi_i[0, 0, z], \Pi_i[1, 0, z]).$$

By definition of μ, we have that for any $i \in [\![1,k]\!]$, that if $X_i = 1$ and $Z = z \neq i$ then $M \sim \mathbf{Ber}(\frac{1}{2}, \frac{1}{2})$.
We get by Lemma 1.2.21: $\forall\ i \in [\![1,k]\!], \forall\ z \in [\![1,k]\!] \setminus \{i\}$,

$$I(M; \Pi_i \mid X_i = 1, Z = z) \geq h^2(\Pi_i[1,0,z], \Pi_i[1,1,z]).$$

Let us define $\mathsf{SMIC}_i(\pi) = I(X_i; \Pi_i \mid MZ) + I(M; \Pi_i \mid X_i Z)$ so that

$$\mathsf{SMIC}(\pi) = \sum_{i=1}^{k} \mathsf{SMIC}_i(\pi).$$

We get

$$\begin{aligned}
\mathsf{SMIC}_i(\pi) &= I(X_i; \Pi_i \mid MZ) + I(M; \Pi_i \mid X_i Z) \\
&= \mathop{\mathbb{E}}_{z} \left[I(X_i; \Pi_i \mid M, Z = z) + I(M; \Pi_i \mid X_i, Z = z) \right] \\
&\geq \frac{1}{k} \sum_{z \neq i} \left[I(X_i; \Pi_i \mid M, Z = z) + I(M; \Pi_i \mid X_i, Z = z) \right] \\
&\geq \frac{1}{k} \sum_{z \neq i} \big[\Pr[M = 0 \mid Z = z] I(X_i; \Pi_i \mid M = 0, Z = z) + \\
&\qquad\qquad \Pr[X_i = 1 \mid Z = z] I(M; \Pi_i \mid X_i = 1, Z = z) \big].
\end{aligned}$$

By the definition of μ, $\Pr[M = 0 \mid Z = z] = \frac{2}{3}$ for any z. Also, for any $i \neq z$,

$$\begin{aligned}
\Pr[X_i = 1 \mid Z = z] &= \Pr[M = 0 \mid Z = z] \Pr[X_i = 1 \mid M = 0, Z = z] + \\
&\quad\ \Pr[M = 1 \mid Z = z] \Pr[X_i = 1 \mid M = 1, Z = z] \\
&= \frac{2}{3} \times \frac{1}{2} + \frac{1}{3} \times 1 = \frac{2}{3}.
\end{aligned}$$

Thus,

$$\begin{aligned}
\mathsf{SMIC}_i(\pi) &\geq \frac{1}{k} \sum_{z \neq i} \left[\frac{2}{3} h^2(\Pi_i[0,0,z], \Pi_i[1,0,z]) + \frac{2}{3} h^2(\Pi_i[1,0,z], \Pi_i[1,1,z]) \right] \\
&\geq \frac{1}{3k} \sum_{z \neq i} \left[h(\Pi_i[0,0,z], \Pi_i[1,0,z]) + h(\Pi_i[1,0,z], \Pi_i[1,1,z]) \right]^2 \\
&\geq \frac{1}{3k} \sum_{z \neq i} h^2(\Pi_i[0,0,z], \Pi_i[1,1,z]) \quad \text{(by triangular inequality).}
\end{aligned}$$

We have

$$\begin{aligned}
\mathsf{SMIC}(\pi) &= \sum_{i=1}^{k} \mathsf{SMIC}_i(\pi)\\
&\geq \frac{1}{3k} \sum_{i,z|i\neq z} h^2(\Pi_i[0,0,z], \Pi_i[1,1,z])\\
&\geq \frac{1}{3k} \sum_{\{i,z\}} [h^2(\Pi_i[0,0,z], \Pi_i[1,1,z]) + h^2(\Pi_z[0,0,i], \Pi_z[1,1,i])]\\
&\geq \frac{1}{6k} \sum_{\{i,z\}} [h^2(\Pi_i(\overline{e}_{i,z}), \Pi_i(\overline{e}_z)) + h^2(\Pi_z(\overline{e}_{i,z}), \Pi_z(\overline{e}_i))]\\
&\quad \text{(by Lemma 3.3.6)}\\
&\geq \frac{1}{6k} \sum_{\{i,z\}} [h^2(\Pi(\overline{e}_{i,z}), \Pi(\overline{e}_z)) + h^2(\Pi(\overline{e}_{i,z}), \Pi(\overline{e}_i))] \quad \text{(by Lemma 3.3.7)}\\
&\geq \frac{1}{12k} \sum_{\{i,z\}} [h(\Pi(\overline{e}_{i,z}), \Pi(\overline{e}_z)) + h(\Pi(\overline{e}_{i,z}), \Pi(\overline{e}_i))]^2\\
&\geq \frac{1}{12k} \sum_{\{i,z\}} h^2(\Pi(\overline{e}_i), \Pi(\overline{e}_z)) \quad \text{(by triangular inequality)}\\
&\geq \frac{1}{24k} \sum_{\{i,z\}} h^2(\Pi(\overline{e}_i), \Pi(\overline{1}))\\
&\quad \text{(by Lemma 3.3.5, omitting part of the right-hand term)}\\
&\geq \frac{1}{24k} \sum_{\{i,z\}} \frac{(1-2\epsilon)^2}{2} \quad \text{(by Lemma 1.4.3)}\\
&\geq \frac{(k-1)(1-2\epsilon)^2}{96} = \Omega(k).
\end{aligned}$$

⌟

3.3.4 Switched multi-party information cost of DISJ_k^n

We first prove a direct-sum property which will allow us to make the link between the functions AND_k and DISJ_k^n. We observe that while a similar property was proved in [BEO$^+$13] in the coordinator model, our peer-to-peer model requires a different, somewhat more involved, construction, since, on the one hand we do not have the coordinator, and on the other hand no player can act as the coordinator as it would get too much information. Since the function DISJ_k^n is the disjunction of n AND_k functions, we will analyze the switched multi-party information cost of DISJ_k^n using the input distribution μ^n.

Proposition 3.3.9. *Let $k > 3$. For any protocol π externally ϵ-computing DISJ_k^n, there exists a protocol π' externally ϵ-computing AND_k such that*

$$\mathsf{SMIC}_{\mu^n}(\pi) \geq n \cdot \mathsf{SMIC}_{\mu}(\pi').$$

Proof. Given an arbitrary protocol π for DISJ_k^n, we define a protocol π' for AND_k, and then analyze $\mathsf{SMIC}_{\mu^n}(\pi)$ and $\mathsf{SMIC}_{\mu}(\pi')$.

Let $u \in \{0,1\}^k$ be the input to π' such that u_i is given to player i. The protocol π' is defined as follows.

1. The players first sample publicly an index L uniformly in $[\![1,n]\!]$, and then sample publicly Z^t, for $t \in [\![1,n]\!] \setminus \{L\}$, independently and uniformly in $[\![1,k]\!]$.

2. They then proceed to sample M^t, for $t \in [\![1,n]\!] \setminus \{L\}$, as follows. The set of players is partitioned into two subsets, $\{1,2\}$ and $\{3,\dots,k\}$. Player 1 samples $M^1 \dots M^{L-1}$ and sends the sampled values to player 2 (player 3 samples $M^{L+1} \dots M^n$, see below).

3. Then player 1 samples $X_1^1 \dots X_1^{L-1}$ according to the distribution μ, and player 2 samples $X_2^1 \dots X_2^{L-1}$, according to the distribution μ. Observe that they can do this as they know $M^1, \dots, M^{L-1}, Z^1, \dots, Z^{L-1}$.

4. Players 1 and 2 then apply the following procedure to communicate X_j^t to player j, for $j > 2$ and $t < l$: player 1 sends a bit p_j^t to player j, and player 2 sends a bit q_j^t to player j. Player j then defines $X_j^t = p_j^t \oplus q_j^t$. The bits p_j^t and q_j^t are generated in the following way.

   - If $Z^t = j$, player 1 privately samples a random bit p_j^t, sends it to player 2, who defines $q_j^t = p_j^t$. Player j thus defines $X_j^t = 0$.
   - If $Z^t \neq j$ and $M^t = 0$, player 1 privately samples a random bit p_j^t, and player 2 privately samples a random bit q_j^t. The bit X_j^t defined by player j is in this case a uniform random bit.
   - If $Z^t \neq j$ and $M^t = 1$, player 1 privately samples a random bit p_j^t, sends it to player 2, who defines $q_j^t = p_j^t \oplus 1$. Player j thus defines $X_j^t = 1$.

5. Player 3 samples $M^{L+1} \dots M^n$ and sends the sampled values to players 4 to k. Every player $i \geq 3$ privately samples $X_i^{L+1} \dots X_i^n$.

6. Players 3 and 4 (or any two other players from the set $\{3, \dots, n\}$) then apply the previous procedure to communicate X_j^t to player j, for $j \leq 2$ and $t > L$. We denote by p_1^t and by p_2^t the bits sent by player 3 to player 1 and to player 2, and by q_1^t and by q_2^t the bits sent by player 4 to player 1 and to player 2.

7. Now all the players run the protocol π, on the input composed of (1) the values defined above for x_i^t, $i \in [\![1, k]\!]$, $t \in [\![1, n]\!] \setminus \{L\}$, and (2) $x_i^L = u_i$, for $i \in [\![1, k]\!]$.

8. The output of the protocol π' is the output of the protocol π.

First observe that if π computes DISJ_k^n with error ϵ, then π' computes AND_k with error ϵ, and this is regardless of the values of the random bits used in the construction of the input to π. In other words, the distribution of the input to π is *collapsing* on coordinate l.

Now observe that if the input to protocol π', denote it U, is distributed according to μ (as defined above) then the definition of π' guarantees that the input X to protocol π is distributed according to μ^n. Using the notation we use for μ we can write that if $(U, N, S) \sim \mu$ then $(X, M, Z) \sim \mu^n$.

We now give an upper bound on $\mathsf{SMIC}_\mu(\pi')$ in terms of $\mathsf{SMIC}_{\mu^n}(\pi)$. To this end we first express the transcripts of protocol π', Π_i', $1 \leq i \leq k$, in terms of the transcripts (Π_i) of the protocol π, run in Step 7.

Let us take player 2 and express Π_2' as a function of Π_2. Given the preliminary sampling procedure, we can write Π_2' in four parts.

1. The public randomness that comes from the definition of π': L, Z^{-L}

2. - Read by player 2 (and sent by player 1), $M^{<L}$.
   - Read by player 2 (and sent by player 1), all the p_j^t for $j > 2$ and $t < L$ such that $M^t = 1$ or $j = Z_t$.
   - All the q_j^t, $j > 2$ and $t < L$ sent by player 2.

3. Player 2 also receives $p_2^{L+1} \dots p_2^n, q_2^{L+1} \dots q_2^n$ from players 3 and 4.

4. The last part is the transcript of player 2 when running π.

Note that thanks to the way we realize the distributed sampling procedure, given $Z^{-L}, M^{t<L}$, the p_j^t and the q_j^t from point (2) above are independent from the X_j^t, $j > 2$ and $t < L$ (even conditioned on the transcript of the protocol), and thus we are allowed not to make p_j^t and the q_i^t from point (2) appear in the transcript Π_2' in the manipulations of $\mathsf{SMIC}_\mu(\pi')$ which follow.

Similarly, as the $p_2^{L+1} \dots p_2^n, q_2^{L+1} \dots q_2^n$ from point (3) above are independent from the transcript of the protocol given $X_2^{L+1} \dots X_2^n$, we can replace them in Π'_2 by $X_2^{>L}$. Thus, we will write Π'_2 as $Z^{-L}M^{<L}X_2^{>L}\Pi_2$ (we do not put the random index L here, since as the players are running the protocol π with input $(X, M, Z) \sim \mu^n$ following a product distribution, the index L is independent from the input even conditioned on the transcript: the real input is indistinguishable from the sampled inputs). Similarly, we can consider Π'_1 as $Z^{-L}M^{<L}X_1^{>L}\Pi_1$, and for $i \geq 3$, Π'_i as $Z^{-L}M^{>L}X_i^{<L}\Pi_i$.

We have

$$\begin{aligned}
\mathsf{SMIC}_\mu(\pi') &= \sum_{i=1}^{k} \left(I(U_i; \Pi'_i \mid NS) + I(N; \Pi'_i \mid U_i S)\right) \\
&= \mathop{\mathbb{E}}_{l} \Bigg[\sum_{i=1}^{2} \big(I(X_i^l; Z^{-l}M^{<l}X_i^{>l}\Pi_i \mid M^l Z^l) + I(M^l; Z^{-l}M^{<l}X_i^{>l}\Pi_i \mid X_i^l Z^l) \big) + \\
&\qquad \sum_{i=3}^{k} \left(I(X_i^l; Z^{-l}M^{>l}X_i^{<l}\Pi_i \mid M^l Z^l) + I(M^l; Z^{-l}M^{>l}X_i^{<l}\Pi_i \mid X_i^l Z^l) \right) \Bigg] \\
&= \mathop{\mathbb{E}}_{l} \Bigg[\sum_{i=1}^{2} \Big(I(X_i^l; \Pi_i \mid X_i^{>l} M^{\leq l} Z) + I(M^l; \Pi_i \mid X_i^{\geq l} M^{<l} Z) \Big) + \\
&\qquad \sum_{i=3}^{k} \Big(I(X_i^l; \Pi_i \mid X_i^{<l} M^{\geq l} Z) + I(M^l; \Pi_i \mid X_i^{\leq l} M^{>l} Z) \Big) \Bigg].
\end{aligned}$$

Now, applying Lemma 1.2.14, we have that for any l,

since $I(X_i^l; M^{>l} \mid X_i^{>l} M^{\leq l} Z) = 0$,

$$I(X_i^l; \Pi_i \mid X_i^{>l} M^{\leq l} Z) \leq I(X_i^l; \Pi_i \mid X_i^{>l} M Z),$$

since $I(M^l; X_i^{<l} \mid X_i^{\geq l} M^{<l} Z) = 0$,

$$I(M^l; \Pi_i \mid X_i^{\geq l} M^{<l} Z) \leq I(M^l; \Pi_i \mid X_i M^{<l} Z),$$

since $I(X_i^l; M^{<l} \mid X_i^{<l} M^{\geq l} Z) = 0$,

$$I(X_i^l; \Pi_i \mid X_i^{<l} M^{\geq l} Z) \leq I(X_i^l; \Pi_i \mid X_i^{<l} M Z),$$

since $I(M^l; X_i^{>l} \mid X_i^{\leq l} M^{>l} Z) = 0$,

$$I(M^l; \Pi_i \mid X_i^{\leq l} M^{>l} Z) \leq I(M^l; \Pi_i \mid X_i M^{>l} Z).$$

Thus

$$
\begin{aligned}
\mathsf{SMIC}_\mu(\pi') &\le \mathop{\mathbb{E}}_{l}\Bigg[\sum_{i=1}^{2}\big(I(X_i^l;\Pi_i \mid X_i^{>l}MZ) + I(M^l;\Pi_i \mid X_iM^{<l}Z)\big) + \\
&\qquad \sum_{i=3}^{k}\big(I(X_i^l;\Pi_i \mid X_i^{<l}MZ) + I(M^l;\Pi_i \mid X_iM^{>l}Z)\big)\Bigg] \\
&\le \frac{1}{n}\sum_{l=1}^{n}\Bigg[\sum_{i=1}^{2}\big(I(X_i^l;\Pi_i \mid X_i^{>l}MZ) + I(M^l;\Pi_i \mid X_iM^{<l}Z)\big) + \\
&\qquad \sum_{i=3}^{k}\big(I(X_i^l;\Pi_i \mid X_i^{<l}MZ) + I(M^l;\Pi_i \mid X_iM^{>l}Z)\big)\Bigg] \\
&\le \frac{1}{n}\Bigg[\sum_{i=1}^{2}\left(\sum_{l=n}^{1} I(X_i^l;\Pi_i \mid X_i^{>l}MZ) + \sum_{l=1}^{n} I(M^l;\Pi_i \mid X_iM^{<l}Z)\right) + \\
&\qquad \sum_{i=3}^{k}\left(\sum_{l=1}^{n} I(X_i^l;\Pi_i \mid X_i^{<l}MZ) + \sum_{l=n}^{1} I(M^l;\Pi_i \mid X_iM^{>l}Z)\right)\Bigg] \\
&\le \frac{1}{n}\Bigg[\sum_{i=1}^{2}(I(X_i;\Pi_i \mid MZ) + I(M;\Pi_i \mid X_iZ)) + \\
&\qquad \sum_{i=3}^{k}(I(X_i;\Pi_i \mid MZ) + I(M;\Pi_i \mid X_iZ))\Bigg] \\
&\le \frac{1}{n}\sum_{i=1}^{k}(I(X_i;\Pi_i \mid MZ) + I(M;\Pi_i \mid X_iZ)) \\
&\le \frac{1}{n}\mathsf{SMIC}_{\mu^n}(\pi).
\end{aligned}
$$

⌟

This direct sum, coupled with the lower bound on $\mathsf{SMIC}(\pi')$ for any protocol π' that computes AND_k (Proposition 3.3.8), gives us a lower bound on $\mathsf{SMIC}(\pi)$ for any protocol that computes the function DISJ_k^n.

Theorem 3.3.10. *Assume $k > 3$. Given any fixed $\epsilon \in \left[0, \frac{1}{2}\right[$, for any protocol π externally ϵ-computing DISJ_k^n it holds that*

$$\mathsf{SMIC}_{\mu^n}(\pi) = \Omega(kn).$$

Proof. By applying Propositions 3.3.9 and 3.3.8.

⌟

3.3.5 Multi-party information cost

We will now relate SMIC and MIC, which will allow us to obtain a bound on the MIC of the disjointness function. We also obtain a bound on the communication complexity of the disjointness function.

Proposition 3.3.11. *For any k-player protocol π, $\mathsf{SMIC}_{\mu^n}(\pi) \leq \mathsf{MIC}_{\mu^n}(\pi)$.*

Proof. We first prove that

$$\forall\ i \in [\![1,k]\!], I(M;\Pi_i \mid X_iR_iZ) \leq I(X_{-i};\Pi_i \mid X_iR_i).$$

$$\begin{aligned}
I(M;\Pi_i \mid X_iR_iZ) &\leq I(MX_{-i};\Pi_i \mid X_iR_iZ)\\
&\leq I(X_{-i};\Pi_i \mid X_iR_iZ) + I(M;\Pi_i \mid XR_iZ)\\
&\quad \text{(Chain rule, Proposition 1.2.9)}\\
&\leq I(X_{-i};\Pi_i \mid X_iR_iZ) + I(M;R_{-i}\Pi_i \mid XR_iZ)\\
&\leq I(X_{-i};\Pi_i \mid X_iR_iZ) + I(M;R_{-i} \mid XR_iZ) + I(M;\Pi_i \mid XRZ)\\
&\quad \text{(Chain rule)}\\
&\leq I(X_{-i};\Pi_i \mid X_iR_iZ) + I(M;\Pi_i \mid XRZ)\\
&\leq I(X_{-i};\Pi_i \mid X_iR_iZ) + H(\Pi_i \mid XRZ)\\
&\leq I(X_{-i};\Pi_i \mid X_iR_iZ)\\
&\leq H(\Pi_i \mid X_iR_iZ) - H(\Pi_i \mid XR_iZ)\\
&\leq H(\Pi_i \mid X_iR_iZ) - H(\Pi_i \mid XR_iZ) - I(Z;\Pi_i \mid XR_i)\\
&\leq H(\Pi_i \mid X_iR_iZ) - H(\Pi_i \mid XR_iZ) -\\
&\quad (H(\Pi_i \mid XR_i) - H(\Pi_i \mid XR_iZ))\\
&\leq H(\Pi_i \mid X_iR_iZ) - H(\Pi_i \mid XR_i)\\
&\leq H(\Pi_i \mid X_iR_i) - H(\Pi_i \mid XR_i)\\
&\leq I(X_{-i};\Pi_i \mid X_iR_i)
\end{aligned}$$

We now prove that

$$\forall\ i \in [\![1,k]\!], I(X_i;\Pi_i \mid MZ) \leq I(X_i;\Pi_i \mid X_{-i}R_{-i}).$$

As by definition of μ, $I(X_i;X_{-i}R_{-i} \mid MZ) = 0$, we get by Lemma 1.2.14

$$I(X_i;\Pi_i \mid MZ) \leq I(X_i;\Pi_i \mid X_{-i}R_{-i}MZ)$$

and

$$\begin{aligned}
I(X_i;\Pi_i \mid X_{-i}R_{-i}MZ) &= H(\Pi_i \mid X_{-i}R_{-i}MZ) - H(\Pi_i \mid XR_{-i}MZ) \\
&= H(\Pi_i \mid X_{-i}R_{-i}MZ) - H(\Pi_i \mid XR_{-i}MZ) - \\
&\quad I(MZ;\Pi_i \mid XR_{-i}) \\
&= H(\Pi_i \mid X_{-i}R_{-i}MZ) - H(\Pi_i \mid XR_{-i}MZ) - \\
&\quad (H(\Pi_i \mid XR_{-i}) - H(\Pi_i \mid XR_{-i}MZ)) \\
&= H(\Pi_i \mid X_{-i}R_{-i}MZ) - H(\Pi_i \mid XR_{-i}) \\
&\leq H(\Pi_i \mid X_{-i}R_{-i}) - H(\Pi_i \mid XR_{-i}) \\
&\leq I(X_i;\Pi_i \mid X_{-i}R_{-i})
\end{aligned}$$

and thus

$$I(X_i;\Pi_i \mid MZ) \leq I(X_i;\Pi_i \mid X_{-i}R_{-i}).$$

Summing over $i \in [\![1,k]\!]$ concludes the proof.

$\lrcorner$

Theorem 3.3.12. *Assuming $k > 3$, for any fixed $\epsilon \in \left[0,\frac{1}{2}\right[$, for any protocol π externally ϵ-computing DISJ_k^n, it holds*

$$\mathsf{MIC}_{\mu^n}(\pi) = \Omega(kn).$$

Proof. It is a consequence of Theorem 3.3.10 and Proposition 3.3.11.

$\lrcorner$

Theorem 3.3.13. *Given any fixed $\epsilon \in \left[0,\frac{1}{2}\right[$, there is a constant α such that for $n \geq \frac{1}{\alpha}k$,*

$$\mathsf{CC}^\epsilon(\mathsf{DISJ}_k^n) = \Omega(kn).$$

Proof. The case $k = 3$ can be reduced to the case $k = 2$ for which an $\Omega(n)$ bound is already known (cf. [CP10]). Assume now that $k > 3$. Let π be a protocol ϵ-computing DISJ_k^n. We first convert π into a protocol π' which *externally* ϵ-computes DISJ_k^n, by having player 1 send his output to player 2 before halting. Since in π player 1 ϵ-computes the function DISJ_k^n, π' externally ϵ-computes DISJ_k^n. Observe that $\mathsf{CC}(\pi') = \mathsf{CC}(\pi) + 1$.

By Theorems 3.1.4 and 3.3.12, there exists a constant β such that $\mathsf{CC}(\pi') \geq \beta kn - k^2$. Let a constant $\alpha < \beta$. For $n \geq \frac{1}{\alpha}k$, we have $k^2 \leq \alpha kn$ and we get $\mathsf{CC}(\pi') \geq (\beta - \alpha)kn = \Omega(kn)$, and $\mathsf{CC}(\pi) = \Omega(kn)$.

$\lrcorner$

We note that this lower bound holds also for protocols where only one player is required to output the value of the function.

3.3.6 Back to the public information cost

We now prove that the switched multi-party information cost lower bounds the public information cost. In this subsection we will make the randomness appear explicitly in information terms such as SMIC. We will follow the convention to make the public randomness appear in the conditioning.

Proposition 3.3.14. *For any public-coins oblivious k-player protocol π where the players have n-bit inputs X from $(X, M, Z) \sim \mu^n$,*

$$\mathsf{PIC}_{\mu^n}(\pi) \geq \frac{1}{2}\mathsf{SMIC}_{\mu^n}(\pi).$$

The notation for the transcripts Π_i that we have been using for SMIC differs from the one we have used for PIC in Chapter 2. The above proposition deals with oblivious protocols. In the oblivious setting, the two notations are completely equivalent in terms of information. Thus we will in this subsection use the notation for transcript that we have defined at the beginning of this chapter, both for PIC and for SMIC.[5]

We start by defining two variants of the information cost, which we will use as intermediate quantities in the proof of Proposition 3.3.14. These measures are defined only with respect to the input distribution μ^n, and thus we do not indicate the distribution in the notation of these measures.

Definition 3.3.15.

$$\widehat{\mathsf{IC}}(\pi) = \sum_{i=1}^{k} I(X_{-i}; \overleftarrow{\Pi_i} \mid X_i R^p M Z).$$

Definition 3.3.16.

$$\widetilde{\mathsf{IC}}(\pi) = \sum_{i=1}^{k} I(X_{-i}; \Pi_i \mid X_i R^p Z).$$

We now start the proof with two lemmas that relate the intermediate measures that we just defined to the measure PIC.

Lemma 3.3.17. *For any public-coins protocol π,*

$$\widehat{\mathsf{IC}}(\pi) \leq \mathsf{PIC}_{\mu^n}(\pi).$$

[5] In fact, the two transcript notations would be equivalent in the definition of PIC even if we were not restricting ourself to oblivious protocols. This is because of the presence of X_i, R_i in the conditioning.

Proof. For any $i \in [\![1,k]\!]$,

$$\begin{aligned} I(X_{-i}; \overleftarrow{\Pi_i} \mid X_i R^p M Z) &= H(\overleftarrow{\Pi_i} \mid X_i R^p M Z) \\ &\leq H(\overleftarrow{\Pi_i} \mid X_i R^p) \\ &\leq H(\overleftarrow{\Pi_i} \mid X_i R^p) - H(\overleftarrow{\Pi_i} \mid X R^p) \\ &\leq I(X_{-i}; \overleftarrow{\Pi_i} \mid X_i R^p). \end{aligned}$$

Summing over $i \in [\![1,k]\!]$ concludes the proof.

⌟

Lemma 3.3.18. *For any public-coins protocol π,*

$$\widetilde{\mathsf{IC}}(\pi) \leq \mathsf{PIC}_{\mu^n}(\pi).$$

Proof. The proof is similar to the one of Lemma 3.3.17. For any $i \in [\![1,k]\!]$,

$$\begin{aligned} I(X_{-i}; \Pi_i \mid X_i R^p Z) &= H(\Pi_i \mid X_i R^p Z) \\ &\leq H(\Pi_i \mid X_i R^p) \quad \text{(by Proposition 1.2.2)} \\ &\leq I(X_{-i}; \Pi_i \mid X_i R^p). \end{aligned}$$

By Proposition 1.4.8, summing over $i \in [\![1,k]\!]$ concludes the proof.

⌟

The next two lemmas together relate SMIC to the intermediate measures that we defined.

Lemma 3.3.19. *For any public-coins protocol π,*

$$\sum_{i=1}^{k} I(M; \Pi_i \mid X_i R^p Z) \leq \widetilde{\mathsf{IC}}(\pi).$$

Proof. Let $i \in [\![1,k]\!]$.

$$\begin{aligned} I(M; \Pi_i \mid X_i R^p Z) &\leq I(M X_{-i}; \Pi_i \mid X_i R^p Z) \\ &\leq I(X_{-i}; \Pi_i \mid X_i R^p Z) + I(M; \Pi_i \mid X R^p Z) \\ &\quad \text{(using the chain rule, Proposition 1.2.9)} \\ &\leq I(X_{-i}; \Pi_i \mid X_i R^p Z) + H(\Pi_i \mid X R^p Z) \\ &\leq I(X_{-i}; \Pi_i \mid X_i R^p Z). \end{aligned}$$

Summing over $i \in [\![1,k]\!]$ concludes the proof.

⌟

The ideas behind the proof of the next lemma are similar to the ones developed in the proof of Theorem 2.1.8. However, the distribution and the quantities involved being different, a careful analysis is required.

Lemma 3.3.20. *For any public-coins oblivious protocol π,*

$$\sum_{i=1}^{k} I(X_i;\Pi_i \mid R^p MZ) \leq \widehat{\mathsf{IC}}(\pi).$$

Proof. Let $i \in [\![1,k]\!]$. Using the chain rule (Proposition 1.2.9) and splitting the set of terms in two subsets,

$$I(X_i;\Pi_i \mid R^p MZ) = \sum_l I(X_i;T_i^{\overrightarrow{l}} \mid T_i^{\overrightarrow{<l}} R^p MZ) + \sum_l I(X_i;T_i^{\overleftarrow{l}} \mid T_i^{\overleftarrow{<l}} R^p MZ).$$

We show that every term of the second sum is 0. Let us organize the messages of Π_i as a sequence of messages (B^d), ordered by local round and inside each round, letting first the messages sent by player i, ordered by index of the recipient, and letting then the messages read by player i, ordered by index of the sender. We show by induction that
$\forall\ d, I(X_i;X_{-i} \mid MZR^pB^0\dots B^d) = 0$. We have $I(X_i;X_{-i} \mid MZR^p) = 0$. Suppose that for some d, $I(X_i;X_{-i} \mid MZR^pB^0\dots B^d) = 0$. If the message B^{d+1} is sent by player i, then B^{d+1} is a function of X_i, R^p and $B^0\dots B^d$ and thus

$$\begin{aligned}
I(X_i;X_{-i} \mid MZR^pB^0\dots B^{d+1}) &= H(X_{-i} \mid MZR^pB^0\dots B^{d+1}) - \\
&\qquad H(X_{-i} \mid MZR^pB^0\dots B^{d+1}X_i) \\
&\leq H(X_{-i} \mid MZR^pB^0\dots B^{d}) - \\
&\qquad H(X_{-i} \mid MZR^pB^0\dots B^{d}X_i) \\
&\leq I(X_i;X_{-i} \mid MZR^pB^0\dots B^d) = 0
\end{aligned}$$

Similarly, if the message B^{d+1} is received by player i, then B^{d+1} is a function of X_{-i}, R^p and $B^0\dots B^d$ and thus

$$\begin{aligned}
I(X_i;X_{-i} \mid MZR^pB^0\dots B^{d+1}) &= H(X_i \mid MZR^pB^0\dots B^{d+1}) - \\
&\qquad H(X_i \mid MZR^pB^0\dots B^{d+1}X_{-i}) \\
&\leq H(X_i \mid MZR^pB^0\dots B^{d}) - \\
&\qquad H(X_i \mid MZR^pB^0\dots B^{d}X_{-i}) \\
&\leq I(X_i;X_{-i} \mid MZR^pB^0\dots B^d) = 0
\end{aligned}$$

Thus $\forall\, d, I(X_i; X_{-i} \mid MZR^pB^0 \dots B^d) = 0$, and choosing the relevant d, $I(X_i; X_{-i} \mid \overleftarrow{T}_i^{<l} R^p MZ) = 0$. Applying Lemma 1.2.15 with $A = X_i$, $B = X_{-i}$, $C = (\overleftarrow{T}_i^{<l}, R^p)$, $D = (M, Z)$ and $\phi = \overleftarrow{T}_i^{l} = \varphi(\overleftarrow{T}_i^{<l}, R^p, X_{-i})$ leads to $I(X_i; \overleftarrow{T}_i^{l} \mid \overleftarrow{T}_i^{<l} R^p MZ) = 0$. We have shown that

$$I(X_i; \Pi_i \mid R^p MZ) = \sum_l I(X_i; \overrightarrow{T}_i^{l} \mid \overrightarrow{T}_i^{<l} R^p MZ).$$

Starting from the definition of $\widehat{\mathsf{IC}}$ and using the chain rule (Proposition 1.2.9), we can decompose it as a sum over all messages received in the protocol:

$$\widehat{\mathsf{IC}}(\pi) = \sum_i \sum_l I(X_{-i}; \overleftarrow{T}_i^{l} \mid \overleftarrow{T}_i^{<l} X_i R^p MZ).$$

Note that in $\overleftarrow{T}_i^{<l}$ we also included the messages sent by player i here, but as these are a function of X_i, R^p and of the messages received in $\overleftarrow{T}_i^{<l}$, we can safely include them in the conditioning.

We rearrange the sum by considering the messages from the point of view of the sender rather than the receiver.

$$\widehat{\mathsf{IC}}(\pi) = \sum_i \sum_l I(X_{-j}; \overrightarrow{T}_i^{l} \mid \overleftarrow{T}_j^{<l'} X_j R^p MZ).$$

Our objective is now to show that for any message $\overrightarrow{T}_i^{l}$,

$$I(X_i; \overrightarrow{T}_i^{l} \mid \overrightarrow{T}_i^{<l} R^p MZ) \leq I(X_{-j}; \overrightarrow{T}_i^{l} \mid \overleftarrow{T}_j^{<l'} X_j R^p MZ).$$

As $\overrightarrow{T}_i^{l}$ is determined by X_i, R^p and $\overrightarrow{T}_i^{<l}$, $H(\overrightarrow{T}_i^{l} \mid X_i \overrightarrow{T}_i^{<l} R^p MZ) = 0$, and we have $I(X_i; \overrightarrow{T}_i^{l} \mid \overrightarrow{T}_i^{<l} R^p MZ) = H(\overrightarrow{T}_i^{l} \mid \overrightarrow{T}_i^{<l} R^p MZ)$, and similarly $I(X_{-j}; \overrightarrow{T}_i^{l} \mid \overleftarrow{T}_j^{<l'} X_j R^p MZ) = H(\overrightarrow{T}_i^{l} \mid \overleftarrow{T}_j^{<l'} X_j R^p MZ)$. Thus

$$\begin{gathered}
I(X_i; \overrightarrow{T}_i^{l} \mid \overrightarrow{T}_i^{<l} R^p MZ) \leq I(X_{-j}; \overrightarrow{T}_i^{l} \mid \overleftarrow{T}_j^{<l'} X_j R^p MZ) \\
\Updownarrow \\
H(\overrightarrow{T}_i^{l} \mid \overrightarrow{T}_i^{<l} R^p MZ) \leq H(\overrightarrow{T}_i^{l} \mid \overleftarrow{T}_j^{<l'} X_j R^p MZ) \\
\Updownarrow \\
I(\overrightarrow{T}_i^{l}; \overrightarrow{T}_i^{<l} R^p MZ) \geq I(\overrightarrow{T}_i^{l}; \overleftarrow{T}_j^{<l'} X_j R^p MZ)
\end{gathered}$$

We show that $I(\overrightarrow{T}_i^{l}; \overrightarrow{T}_i^{<l} R^p MZ) = I(\overrightarrow{T}_i^{l}; \overrightarrow{T}_i^{<l} \overleftarrow{T}_j^{<l'} X_j R^p MZ)$, which implies that the last inequality is true. For this we just need to show that

$I(T_i^{\overrightarrow{l}}; T_j^{<\overleftarrow{l'}} X_j \mid T_i^{<\overrightarrow{l}} R^p MZ) = 0$. Notice that given the value of $T_i^{<\overrightarrow{l}} R^p MZ$, $T_i^{\overrightarrow{l}}$ is determined by X_i and thus by data processing inequality 1.2.12

$$I(X_i; T_j^{<\overleftarrow{l'}} X_j \mid T_i^{<\overrightarrow{l}} R^p MZ) \geq I(T_i^{\overrightarrow{l}}; T_j^{<\overleftarrow{l'}} X_j \mid T_i^{<\overrightarrow{l}} R^p MZ),$$

and so we just have to show that $I(X_i; T_j^{<\overleftarrow{l'}} X_j \mid T_i^{<\overrightarrow{l}} R^p MZ) = 0$, which we now do.

Note that $(T_j^{<\overleftarrow{l'}}, X_j)$ is a function of $(X_{-i}, T_i^{<\overrightarrow{l}})$. The data processing inequality 1.2.12 implies that

$$I(X_i; T_j^{<\overleftarrow{l'}} X_j \mid T_i^{<\overrightarrow{l}} R^p MZ) \leq I(X_i; X_{-i} T_i^{<\overrightarrow{l}} \mid T_i^{<\overrightarrow{l}} R^p MZ)$$

and thus

$$I(X_i; T_j^{<\overleftarrow{l'}} X_j \mid T_i^{<\overrightarrow{l}} R^p MZ) \leq I(X_i; X_{-i} \mid T_i^{<\overrightarrow{l}} R^p MZ).$$

We have already shown above that $\forall\ d, I(X_i; X_{-i} \mid MZR^p B^0 \ldots B^d) = 0$, and choosing the relevant d, $I(X_i; X_{-i} \mid T_i^{<\overrightarrow{l}} R^p MZ) = 0$. Hence we have proved that for any message $T_i^{\overrightarrow{l}}$,

$$I(X_i; T_i^{\overrightarrow{l}} \mid T_i^{<\overrightarrow{l}} R^p MZ) \leq I(X_{-j}; T_i^{\overrightarrow{l}} \mid T_j^{<\overleftarrow{l'}} X_j R^p MZ),$$

which implies that $\sum_{i=1}^{k} I(X_i; \Pi_i \mid R^p MZ) \leq \widehat{\mathsf{IC}}(\pi)$.

┘

We are now able to prove Proposition 3.3.14.

Proof of Proposition 3.3.14. Let π be an oblivious public-coins protocol ϵ-computing DISJ_k^n. By Lemmas 3.3.19 and 3.3.20 we have that

$$\mathsf{SMIC}_{\mu^n}(\pi) \leq \widetilde{\mathsf{IC}}(\pi) + \widehat{\mathsf{IC}}(\pi).$$

Using Lemmas 3.3.18 and 3.3.17,

$$\mathsf{SMIC}_{\mu^n}(\pi) \leq 2 \cdot \mathsf{PIC}_{\mu^n}(\pi).$$

┘

We can now lower bound the public information cost of the disjointness function.

Theorem 3.3.21. *Assume $k > 3$. In the oblivious setting,*

$$\mathsf{PIC}_{\mu^n}^{\epsilon}(\mathsf{DISJ}_k^n) = \Omega(kn).$$

Proof. Let π be an oblivious protocol ϵ-computing DISJ_k^n. By Theorem 2.1.5, we only have to consider public-coins protocols. Observe that, by adding a single bit message, we can convert π into a protocol π' externally computing DISJ_k^n. By Theorem 3.3.10 and Proposition 3.3.14, it holds $\mathsf{PIC}_{\mu^n}(\pi') = \Omega(kn)$. As $\mathsf{PIC}_{\mu^n}(\pi') \leq \mathsf{PIC}_{\mu^n}(\pi) + 1$, we get that $\mathsf{PIC}_{\mu^n}(\pi) = \Omega(kn)$.

⌟

We now give a lower bound of $\Omega(n)$ on the randomness complexity of the function DISJ_k^n. The importance of this result lies in that it is the first such lower bound that grows with the size of the input while the output remains a single bit. Note that the theorem which follows and its proof can be translated to the setting of epsilon-error randomness complexity, that we did not consider here.

Theorem 3.3.22. *Assume $k > 3$. There exists an input distribution μ such that*

$$\mathcal{R}_{\mu^n}(\mathsf{DISJ}_k^n) = \Omega(n).$$

Proof. Theorem 3.3.21 provides a distribution μ such that $\mathsf{PIC}_{\mu^n}(\mathsf{DISJ}_k^n) = \Omega(kn)$. Moreover, $H_{\mu^n}(\mathsf{DISJ}_k^n) = 0$. Applying Theorem 2.4.14, we get

$$\mathcal{R}_{\mu^n}(\mathsf{DISJ}_k^n) \geq \frac{\Omega(kn)}{k-1} = \Omega(n).$$

⌟

We can also get a stronger bound on the communication complexity of the disjointness function in the oblivious setting.

Theorem 3.3.23. *In the oblivious setting,*

$$\mathsf{CC}^{\epsilon}(\mathsf{DISJ}_k^n) = \Omega(kn).$$

Proof. The case $k = 3$ can be reduced to the case $k = 2$ for which an $\Omega(n)$ bound is already known (cf. [CP10]). For $k > 3$, the result comes from the combination of Theorems 3.3.21 and 2.3.2.

⌟

Conclusion

In this thesis, we presented a multi-party peer-to-peer number-in-hand communication model which, while being general enough to allow most protocols considered in the literature, has good properties which allowed us to introduce an information-theoretic framework for the study of multi-party communication protocols. We introduced two main new information-theoretic measures: the public information cost PIC, and the multi-party information cost MIC, and showed that these measures have interesting properties which make them suitable to the study of classic distributed functions such as *Parity* and *Disjointness*.

One of the fundamental properties of the public information cost is its relation to the number of random coins required in order to run private protocols. As an illustration, we obtained tight lower bounds on the number of coins required for privacy for the n-bit parity and disjointness functions. Another interesting result, which is the consequence of a communication compression procedure, is that the existence of a direct sum for the public information cost would imply a certain direct sum for communication complexity.

We showed that the multi-party information cost exhibits good properties, among which a direct sum, which make it suitable to the study of multi-party communication complexity. It allowed us to give tight communication complexity lower bounds in a fully distributed setting for the parity and disjointness functions.

Some important questions remain open. For both the public information cost and the multi-party information cost, it is necessary to develop more techniques to obtain lower bounds. In particular, for the public information cost, the existence of a direct sum result is worth investigating, not only as it would lead to an easier way to get lower bounds, but also because of its relation to the existence of a direct sum for communication complexity. It would also be interesting to introduce a generalisation of the public information cost to study the number of random coins required to run private protocols which are secure against collusions of players. Last, our framework is not sufficient to allow for the study of fully asynchronous protocols, and discovering novel tools for the study of fully asynchronous protocols remains necessary.

Bibliography

[AMS96] Noga Alon, Yossi Matias, and Mario Szegedy. The space complexity of approximating the frequency moments. In *Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing*, STOC '96, pages 20–29, New York, NY, USA, 1996. ACM.

[BBCR10] Boaz Barak, Mark Braverman, Xi Chen, and Anup Rao. How to compress interactive communication. In *Proceedings of the 42nd ACM symposium on Theory of computing*, STOC '10, pages 67–76, New York, NY, USA, 2010. ACM.

[BBK$^+$13] 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 *Proceedings of the 28th Conference on Computational Complexity, CCC 2013, K.lo Alto, California, USA, 5-7 June, 2013*, pages 24–33. IEEE, 2013.

[BCKO93] Reuven Bar-Yehuda, Benny Chor, Eyal Kushilevitz, and Alon Orlitsky. Privacy, additional information and communication. *IEEE Transactions on Information Theory*, 39(6):1930–1943, 1993.

[BDSPV99] C. Blundo, A. De Santis, G. Persiano, and U. Vaccaro. Randomness complexity of private computation. *computational complexity*, 8(2):145–168, 1999.

[BEO$^+$13] Mark Braverman, Faith Ellen, Rotem Oshman, Toniann Pitassi, and Vinod Vaikuntanathan. A tight bound for set disjointness in the message-passing model. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 668–677. IEEE Computer Society, 2013.

[BG14] Mark Braverman and Ankit Garg. Public vs private coin in bounded-round information. In Javier Esparza, Pierre Fraigniaud, Thore Husfeldt, and Elias Koutsoupias, editors, *Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part I*, volume 8572 of *Lecture Notes in Computer Science*, pages 502–513. Springer, 2014.

[BGPW13] Mark Braverman, Ankit Garg, Denis Pankratov, and Omri Weinstein. From information to exact communication. In *Proceedings of the 45th annual ACM symposium on Symposium on theory of computing*, STOC '13, pages 151–160, New York, NY, USA, 2013. ACM.

[BMY15] Balthazar Bauer, Shay Moran, and Amir Yehudayoff. Internal compression of protocols to entropy. In Naveen Garg, Klaus Jansen, Anup Rao, and José D. P. Rolim, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2015, August 24-26, 2015, Princeton, NJ, USA*, volume 40 of *LIPIcs*, pages 481–496. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015.

[BO15] Mark Braverman and Rotem Oshman. On information complexity in the broadcast model. In Chryssis Georgiou and Paul G. Spirakis, editors, *Proceedings of the 2015 ACM Symposium on Principles of Distributed Computing, PODC 2015, Donostia-San Sebastián, Spain, July 21 - 23, 2015*, pages 355–364. ACM, 2015.

[BO17] Mark Braverman and Rotem Oshman. A rounds vs. communication tradeoff for multi-party set disjointness. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 144–155. IEEE Computer Society, 2017.

[BOGW88] Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In *Proceedings of the twentieth annual ACM symposium on Theory of computing*, STOC '88, pages 1–10, New York, NY, USA, 1988. ACM.

[BR11] Mark Braverman and Anup Rao. Information equals amortized communication. In *Proceedings of the 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, FOCS '11, pages 748–757, Washington, DC, USA, 2011. IEEE Computer Society.

[Bra12] Mark Braverman. Interactive information complexity. In *Proceedings of the 44th symposium on Theory of Computing*, STOC '12, pages 505–524, New York, NY, USA, 2012. ACM.

[Bra13] Mark Braverman. A hard-to-compress interactive task? In *51st Annual Allerton Conference on Communication, Control, and Computing, Allerton 2013, Allerton Park & Retreat Center, Monticello, IL, USA, October 2-4, 2013*, pages 8–12. IEEE, 2013.

[BYJKS02] Ziv Bar-Yossef, T. S. Jayram, Ravi Kumar, and D. Sivakumar. An information statistics approach to data stream and communication complexity. In *Proceedings of the 43rd Symposium on Foundations of Computer Science*, FOCS '02, pages 209–218, Washington, DC, USA, 2002. IEEE Computer Society.

[CCD88] David Chaum, Claude Crépeau, and Ivan Damgard. Multiparty unconditionally secure protocols. In *Proceedings of the twentieth annual ACM symposium on Theory of computing*, STOC '88, pages 11–19, New York, NY, USA, 1988. ACM.

[CDvdG88] David Chaum, Ivan B. Damgård, and Jeroen van de Graaf. *Multiparty Computations Ensuring Privacy of Each Party's Input and Correctness of the Result*, pages 87–119. Springer Berlin Heidelberg, Berlin, Heidelberg, 1988.

[CK93] Benny Chor and Eyal Kushilevitz. A communication-privacy tradeoff for modular addition. *Information Processing Letters*, 45(4):205 – 210, 1993.

[CK16] Amit Chakrabarti and Sagar Kale. Strong fooling sets for multiplayer communication with applications to deterministic estimation of stream statistics. In Dinur [Din16], pages 41–50.

[CKS03] Amit Chakrabarti, Subhash Khot, and Xiaodong Sun. Near-optimal lower bounds on the multi-party communication complexity of set disjointness. In *In IEEE Conference on Computational Complexity*, pages 107–117, 2003.

[CM15] Arkadev Chattopadhyay and Sagnik Mukhopadhyay. Tribes is hard in the message passing model. In Mayr and Ollinger [MO15], pages 224–237.

[CP10] Arkadev Chattopadhyay and Toniann Pitassi. The story of set disjointness. *SIGACT News*, 41(3):59–85, September 2010.

[CR15] Arkadev Chattopadhyay and Atri Rudra. The range of topological effects on communication. In Magnús M. Halldórsson, Kazuo Iwama, Naoki Kobayashi, and Bettina Speckmann, editors, *Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part II*, volume 9135 of *Lecture Notes in Computer Science*, pages 540–551. Springer, 2015.

[CRR14] Arkadev Chattopadhyay, Jaikumar Radhakrishnan, and Atri Rudra. Topology matters in communication. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 631–640, 2014.

[CSWY01] Amit Chakrabarti, Yaoyun Shi, Anthony Wirth, and Andrew Chi-Chih Yao. Informational complexity and the direct sum problem for simultaneous message complexity. In *FOCS*, pages 270–278, 2001.

[CT06] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing).* Wiley-Interscience, 2006.

[DF89] Danny Dolev and Tomás Feder. Multiparty communication complexity. In *30th Annual Symposium on Foundations of Computer Science, Research Triangle Park, North Carolina, USA, 30 October - 1 November 1989*, pages 428–433. IEEE Computer Society, 1989.

[Din16] Irit Dinur, editor. *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*. IEEE Computer Society, 2016.

[EOPV13] Faith Ellen, Rotem Oshman, Toniann Pitassi, and Vinod Vaikuntanathan. Brief Announcement: Private Channel

Models in Multi-party Communication Complexity. In *27th International Symposium on Distributed Computing (DISC), Jerusalem, Israel*, pages 575–576, 2013.

[Fan61] Robert M Fano. Transmission of information: A statistical theory of communications. *American Journal of Physics*, 29:793–794, 1961.

[FHW12] Silvio Frischknecht, Stephan Holzer, and Roger Wattenhofer. Networks cannot compute their diameter in sublinear time. In Yuval Rabani, editor, *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pages 1150–1162. SIAM, 2012.

[FKN94] Uri Feige, Joe Killian, and Moni Naor. A minimal model for secure computation (extended abstract). In *Proceedings of the Twenty-sixth Annual ACM Symposium on Theory of Computing*, STOC '94, pages 554–563, New York, NY, USA, 1994. ACM.

[FKNN95] Tomás Feder, Eyal Kushilevitz, Moni Naor, and Noam Nisan. Amortized communication complexity. *SIAM J. Comput.*, 24(4):736–750, 1995.

[FRPU94] Uriel Feige, Prabhakar Raghavan, David Peleg, and Eli Upfal. Computing with noisy information. *SIAM J. Comput.*, 23(5):1001–1018, October 1994.

[FY92] Matthew Franklin and Moti Yung. Communication complexity of secure computation (extended abstract). In *Proceedings of the Twenty-fourth Annual ACM Symposium on Theory of Computing*, STOC '92, pages 699–710, New York, NY, USA, 1992. ACM.

[GG10] Anna Gál and Parikshit Gopalan. Lower bounds on streaming algorithms for approximating the length of the longest increasing subsequence. *SIAM J. Comput.*, 39(8):3463–3479, 2010.

[GKR14] Anat Ganor, Gillat Kol, and Ran Raz. Exponential separation of information and communication. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 176–185, 2014.

[GKR15a] Anat Ganor, Gillat Kol, and Ran Raz. Exponential separation of communication and external information. *Electronic Colloquium on Computational Complexity (ECCC)*, 22:88, 2015.

[GKR15b] Anat Ganor, Gillat Kol, and Ran Raz. Exponential separation of information and communication for boolean functions. In Rocco A. Servedio and Ronitt Rubinfeld, editors, *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 557–566. ACM, 2015.

[GMW87] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game. In *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*, STOC '87, pages 218–229, New York, NY, USA, 1987. ACM.

[GR05] Anna Gál and Adi Rosén. Omega(log n) lower bounds on the amount of randomness in 2-private computation. *SIAM J. Comput.*, 34(4):946–959, 2005.

[Gro09] Andre Gronemeier. Asymptotically optimal lower bounds on the nih-multi-party information complexity of the and-function and disjointness. In Susanne Albers and Jean-Yves Marion, editors, *26th International Symposium on Theoretical Aspects of Computer Science, STACS 2009, February 26-28, 2009, Freiburg, Germany, Proceedings*, volume 3 of *LIPIcs*, pages 505–516. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany, 2009.

[GU18] François Le Gall and Florent Urrutia. Improved rectangular matrix multiplication using powers of the coppersmith-winograd tensor. In *SODA*, 2018.

[HJMR10] 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.

[HRVZ15] Zengfeng Huang, Bozidar Radunovic, Milan Vojnovic, and Qin Zhang. Communication complexity of approximate matching in distributed graphs. In Mayr and Ollinger [MO15], pages 460–473.

[Huf06] David A. Huffman. A method for the construction of minimum-redundancy codes. *Resonance*, 11(2):91–99, 2006.

[Jai15] Rahul Jain. New strong direct product results in communication complexity. *J. ACM*, 62(3):20, 2015.

[Jay09] T. S. Jayram. Hellinger strikes back: A note on the multi-party information complexity of and. In *Proceedings of the 12th International Workshop and 13th International Workshop on Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, APPROX '09 / RANDOM '09, pages 562–573, Berlin, Heidelberg, 2009. Springer-Verlag.

[JKS03] T. S. Jayram, Ravi Kumar, and D. Sivakumar. Two applications of information complexity. In *Proceedings of the Thirty-fifth Annual ACM Symposium on Theory of Computing*, STOC '03, pages 673–682, New York, NY, USA, 2003. ACM.

[JRS03] Rahul Jain, Jaikumar Radhakrishnan, and Pranab Sen. A direct sum theorem in communication complexity via message compression. In Jos C. M. Baeten, Jan Karel Lenstra, Joachim Parrow, and Gerhard J. Woeginger, editors, *Automata, Languages and Programming, 30th International Colloquium, ICALP 2003, Eindhoven, The Netherlands, June 30 - July 4, 2003. Proceedings*, volume 2719 of *Lecture Notes in Computer Science*, pages 300–315. Springer, 2003.

[JSR08] Rahul Jain, Pranab Sen, and Jaikumar Radhakrishnan. Optimal direct sum and privacy trade-off results for quantum and classical communication complexity. *CoRR*, abs/0807.1267, 2008.

[Kla10] Hartmut Klauck. A strong direct product theorem for disjointness. In Leonard J. Schulman, editor, *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, pages 77–86. ACM, 2010.

[Kle02] Stephen Cole Kleene. *Mathematical Logic*. Dover, 2002. Reprint of the John Wiley & Sons, Inc., New York, 1967 edition.

[KM97] Eyal Kushilevitz and Yishay Mansour. Randomness in private computations. *SIAM J. Discrete Math.*, 10(4):647–661, 1997.

[KMSY16] Gillat Kol, Shay Moran, Amir Shpilka, and Amir Yehudayoff. Direct sum fails for zero-error average communication. *Algorithmica*, 76(3):782–795, 2016.

[KN97] Eyal Kushilevitz and Noam Nisan. *Communication complexity.* Cambridge University Press, 1997.

[Kol16] Gillat Kol. Interactive compression for product distributions. In *Proceedings of the Forty-eighth Annual ACM Symposium on Theory of Computing*, STOC '16, pages 987–998, New York, NY, USA, 2016. ACM.

[KOR96] Eyal Kushilevitz, Rafail Ostrovsky, and Adi Rosén. Characterizing linear size circuits in terms of privacy. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, STOC '96, pages 541–550, New York, NY, USA, 1996. ACM.

[KOR98] Eyal Kushilevitz, Rafail Ostrovsky, and Adi Rosén. Amortizing randomness in private multiparty computations. In *Proceedings of the Seventeenth Annual ACM Symposium on Principles of Distributed Computing*, PODC '98, pages 81–90, New York, NY, USA, 1998. ACM.

[KOS17] Gillat Kol, Rotem Oshman, and Dafna Sadeh. Interactive Compression for Multi-Party Protocol. In Andréa W. Richa, editor, *31st International Symposium on Distributed Computing (DISC 2017)*, volume 91 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 31:1–31:15, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[Koz15] Alexander Kozachinskiy. *Computer Science – Theory and Applications: 10th International Computer Science Symposium in Russia, CSR 2015, Listvyanka, Russia, July 13-17, 2015, Proceedings*, chapter Making Randomness Public in Unbounded-Round Information Complexity, pages 296–309. Springer International Publishing, Cham, 2015.

[KR98] Eyal Kushilevitz and Adi Rosén. A randomness-rounds tradeoff in private computation. *SIAM Journal on Discrete Mathematics*, 11(1):61–80, 1998.

[KRU16] Iordanis Kerenidis, Adi Rosén, and Florent Urrutia. Multi-party protocols, information complexity and privacy. In Piotr Faliszewski, Anca Muscholl, and Rolf Niedermeier, editors, *41st International Symposium on Mathematical Foundations of Computer Science, MFCS 2016, August 22-26, 2016 - Kraków, Poland*, volume 58 of *LIPIcs*, pages 57:1–57:16. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016.

[KRW95] Mauricio Karchmer, Ran Raz, and Avi Wigderson. Super-logarithmic depth lower bounds via the direct sum in communication complexity. *computational complexity*, 5(3):191–204, 1995.

[KS92] Bala Kalyanasundaram and Georg Schintger. The probabilistic communication complexity of set intersection. *SIAM J. Discret. Math.*, 5(4):545–557, November 1992.

[KY76] Donald E. Knuth and Andrew C. Yao. The complexity of nonuniform random number generation. pages 375–428, 1976.

[Lov14] Shachar Lovett. Recent advances on the log-rank conjecture in communication complexity. *CoRR*, abs/1403.8106, 2014.

[MNSW95] Peter Bro Miltersen, Noam Nisan, Shmuel Safra, and Avi Wigderson. On data structures and asymmetric communication complexity. In *Proceedings of the Twenty-seventh Annual ACM Symposium on Theory of Computing*, STOC '95, pages 103–111, New York, NY, USA, 1995. ACM.

[MO15] Ernst W. Mayr and Nicolas Ollinger, editors. *32nd International Symposium on Theoretical Aspects of Computer Science, STACS 2015, March 4-7, 2015, Garching, Germany*, volume 30 of *LIPIcs*. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015.

[Pan15] Denis Pankratov. *Communication complexity and information complexity.* PhD thesis, The university of Chicago, 2015.

[PVZ12] Jeff M. Phillips, Elad Verbin, and Qin Zhang. Lower bounds for number-in-hand multiparty communication complexity, made easy. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '12, pages 486–501. SIAM, 2012.

[Raz92] A. A. Razborov. On the distributional complexity of disjointness. *Theor. Comput. Sci.*, 106(2):385–390, December 1992.

[RS15] Anup Rao and Makrand Sinha. Simplified separation of information and communication. *Electronic Colloquium on Computational Complexity (ECCC)*, 22:57, 2015.

[RU17] Adi Rosén and Florent Urrutia. A new approach to multi-party communication complexity. 2017.

[Sha48] C. E. Shannon. A mathematical theory of communication. *Bell system technical journal*, 27, 1948.

[Sha79] Adi Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, November 1979.

[Sha03] Ronen Shaltiel. Towards proving strong direct product theorems. *Computational Complexity*, 12(1-2):1–22, 2003.

[She16] Alexander A. Sherstov. Compressing interactive communication under product distributions. In Dinur [Din16], pages 535–544.

[Shi00] Yaoyun Shi. Lower bounds of quantum black-box complexity and degree of approximating polynomials by influence of boolean variables. *Inf. Process. Lett.*, 75(1-2):79–83, 2000.

[SHK$^+$10] Atish Das Sarma, Stephan Holzer, Liah Kor, Amos Korman, Danupon Nanongkai, Gopal Pandurangan, David Peleg, and Roger Wattenhofer. Distributed verification and hardness of distributed approximation. *CoRR*, abs/1011.3049, 2010.

[SW73] David S. Slepian and Jack K. Wolf. Noiseless coding of correlated information sources. *IEEE Trans. Information Theory*, 19(4):471–480, 1973.

[WZ11] David P. Woodruff and Qin Zhang. Tight bounds for distributed functional monitoring. *CoRR*, abs/1112.5153, 2011.

[WZ13] David P. Woodruff and Qin Zhang. When distributed computation does not help. *CoRR*, abs/1304.4636, 2013.

[WZ14] David P. Woodruff and Qin Zhang. An optimal lower bound for distinct elements in the message passing model. In Chandra Chekuri, editor, *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 718–733. SIAM, 2014.

[Yao79] Andrew Chi-Chih Yao. Some complexity questions related to distributive computing(preliminary report). In *Proceedings of the eleventh annual ACM symposium on Theory of computing*, STOC '79, pages 209–213, New York, NY, USA, 1979. ACM.

[Yao82] Andrew C. Yao. Protocols for secure computations. In *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science*, SFCS '82, pages 160–164, Washington, DC, USA, 1982. IEEE Computer Society.