Information Complexity and Data Stream Algorithms for Basic Problems

Dissertation

zur Erlangung des Grades eines
Doktors der Naturwissenschaften
der Technischen Universität Dortmund
an der Fakultät für Informatik

von

André Gronemeier

Dortmund

2010

Tag der mündlichen Prüfung:	21. Oktober 2010
Dekan:	Prof. Dr. Peter Buchholz
Gutachter:	PD Dr. Martin Sauerhoff
	Prof. Dr. Christian Sohler

Summary

Data stream algorithms obtain their input as a stream of data elements that have to be processed immediately as they arrive using only a very limited amount of memory. They solve a new class of algorithmic problems that emerged recently with the growing importance of computer networks and the ever-increasing size of the data sets that are processed algorithmically. In this thesis data stream algorithms for basic problems under extreme space restrictions are developed, namely counting and random sampling. Then we apply these algorithms to improve the space complexity of the celebrated data stream algorithm for the computation of frequency moments by Alon, Matias, and Szegedy for very long data streams.

Lower bounds on the space complexity of data stream algorithms are usually proved by using communication complexity arguments. Information complexity is a related field that applies Shannon's information theory to obtain lower bounds on the communication complexity of functions. The development of information complexity is closely linked to the recent interest in data stream algorithms since important parts of this theory have been developed to prove a lower bound on the space complexity of data stream algorithms for the frequency moments. In this thesis we prove an optimal lower bound on the multi-party information complexity of the disjointness function, the underlying communication problem in the proof of the lower bound on the space complexity of data stream algorithms for the frequency moments. Additionally, we generalize and simplify known lower bounds on the one-way communication complexity of the index function by using information complexity and we present the first attempt to apply information complexity to multi-party one-way protocols in the number on the forehead model by Chandra, Furst, and Lipton.

Acknowledgments

I am grateful for the direct and indirect contribution of many friends and colleagues to this thesis. First and foremost I thank Ingo Wegener (∗4 December 1950, †26 November 2008). His passionate enthusiasm for theoretical computer science inspired me to become a theoretical computer scientist myself and his values, attitudes, and behavior inspired me afterwards in my daily work as a theoretical computer scientist and in my daily life as a member of the human race. I am grateful for the steady support of Martin Sauerhoff and for the numerous discussions with Martin that contributed to this text. Furthermore I thank Martin Sauerhoff and Christian Sohler for applying "the right dose of motivational carrots and sticks", actually mainly carrots, to help me finish this thesis. I am grateful to my current and former colleagues at the chair for *Efficient Algorithms and Complexity Theory* for their cooperativeness and for the pleasant and friendly working atmosphere. Finally and most importantly, I thank my parents for always supporting me.

Danksagungen

Ich bin für die direkten und indirekten Beiträge vieler Freunde und Kollegen zu dieser Dissertation dankbar. Zuallererst danke ich Ingo Wegener (∗4. Dezember 1950, †26. November 2008), dessen leidenschaftliche Begeisterung für die theoretische Informatik mich dazu motiviert hat, selbst ein theoretischer Informatiker zu werden, und dessen Werte, Einstellung und Taten mich anschließend in meiner täglichen Arbeit als theoretischer Informatiker und in meinem täglichen Leben als Mitglied der Menschheit inspiriert haben. Ich danke Martin Sauerhoff für seine stetige Unterstützung und für die zahlreichen Diskussionen, durch die Martin viel zu diesem Text beigetragen hat. Darüber hinaus danke ich Martin Sauerhoff und Christian Sohler für "die richtige Dosierung von Zuckerbrot und Peitsche", eigentlich gab es meistens Zuckerbrot, um mich beim Schreiben dieser Arbeit zu motivieren. Meinen derzeitigen und ehemaligen Kollegen am *Lehrstuhl für effiziente Algorithmen und Komplexitätstheorie* danke ich für ihre Hilfsbereitschaft und für die angenehme Arbeitsatmosphäre. Der größte Dank gilt aber meinen Eltern, die mich immer und bei allem unterstützt haben.

Contents

Chapter 1

Introduction and Overview

The two main topics of this thesis are *information complexity*, an information theoretical proof method for lower bounds on the communication complexity of functions, and the design of *data stream algorithms* for basic problems.

1.1 Data Stream Algorithms

With the growing importance of the Internet – and computer networks in general – a new type of algorithmic problems emerged. Network devices like routers must process incoming packets immediately and they do not have enough memory to store a lot of information about each packet that is processed. Nevertheless, information about these packets is valuable and can be put to good use, for example to optimize the routing of packets or for an early detection of denial of service attacks. Generally, an algorithm for this type of problem obtains its input sequentially as a stream of data elements, for example network packets, that have to be processed immediately as they arrive. The space complexity of the algorithm should be significantly smaller than the length of the input stream. Algorithms in this scenario are usually called *data stream algorithms*.

The second contribution to the growing interest in data stream algorithms besides network problems is the ever-increasing size of the data sets that have to be processed algorithmically. Even if an algorithm can access the input in an arbitrary order in principle, a sequential access of the input might be preferable or even the only solution that is practically feasible for efficiency reasons. For example, modern hard disks allow a random access of the stored data, but a sequential access to the data is faster by orders of magnitude due to the long access times that are caused by the slow mechanical components of a hard disks. This problem is multiplied if the data is stored on dozens, hundreds, or even thousands of hard disks that are accessed via a relatively slow computer network. Hence, in this situation a data stream algorithm may be a better solution than an algorithm that accesses the input randomly. Google introduced *MapReduce*, a practical distributed programming model for large data sets [33]. This model is closely related to data stream algorithms [37]. Google use MapReduce to handle the huge data sets that are needed for the operation of their search engine [1], today's largest Internet search engine.

A third contributor to the success of data stream algorithms are databases. The updates to a database can be considered as a data stream. Data stream algorithms are used to compute

[1] `http://www.google.com/`

small "summaries" of a database that fit easily into the main memory of a computer and can be used for query optimization in relational databases.

The recent increase in theoretical research on data stream algorithms was sparked by the seminal work of Alon, Matias, and Szegdy [3]. Their Gödel Prize-winning paper describes space-efficient data stream algorithms for the computation of the frequency moments of data streams. The algorithmic techniques that were introduced in this paper found numerous applications to data stream problems beyond frequency moments.

In Chapter 4 we consider data stream algorithms for basic problems under extreme memory restrictions, namely counting and sampling. Then we will apply these data stream algorithms to the computation of frequency moments of very long data streams.

1.2 Information Complexity

Communication complexity, introduced by Yao [76] in 1979 as a simple model of distributed computations, measures the amount of communication that is needed for the computation of a function if the arguments of the function are distributed among several parties. Nowadays, communication complexity is an elaborate theory that has found applications in many fields of complexity theory beyond distributed computation. Lower bounds on the space complexity of data stream algorithms are usually proved using communication complexity. Information complexity is a related field that uses information theoretical methods to obtain lower bounds on the communication complexity of functions. Although information complexity was conceived by Chakrabarti, Shi, Wirth, and Yao [26] independently of the developments in the field of data stream algorithms, today information complexity is closely tied to data stream algorithms. This is due to the fact that important refinements of this theory were introduced by Bar-Yossef, Jayram, Kumar, and Sivakumar [13] to prove a lower bound on the space complexity of data stream algorithms for the computation of frequency moments.

Chapter 3 contains our results on the information complexity of some functions. In a short introductory section we will use information complexity to simplify a lower bound on the size of OBDDs that approximate the hidden weighted bit function, a well-known benchmark function in the branching program literature. The space complexity of data stream algorithms for the computation of frequency moments is closely related to the information complexity of the AND function and the so-called disjointness function. Lower bounds on the information complexity of the AND function and the disjointness function were improved in a series of papers by different authors. This thesis contains the last result in this series, an asymptotically optimal lower bound on the information complexity of the AND function and the disjointness function. Finally, we present the first attempt to extend information complexity to one-way multi-party protocols in the number on the forehead model. Communication complexity in the number on the forehead model by Chandra, Furst, and Lipton [27] measures the amount of communication that is needed for the distributed computation of a function $f(x_1, \ldots, x_k)$ by k parties where the ith party knows all arguments of f except x_i. We prove a lower bound on the information complexity of a pointer jumping function for a restricted class of one-way protocols in this model.

1.3 Information Statistics

Communication complexity and algorithm design are two distinct fields of theoretical computer science that often utilize different mathematical ideas and tools. Nevertheless there is a common mathematical thread that connects the two topics of this text. This common thread is best described by the term "information statistics" that was coined by Bar-Yossef, Jayram, Kumar, and Sivakumar [13] to "loosely describe the interplay between information theory and distances between probability distributions." Distances between probability distributions are measured by so-called *statistical divergences*. Our lower bounds on the information complexity of the AND function and the disjointness function in the first part of this thesis combine information theory with the more general concept of statistical divergences. But statistical divergences also play a major role in the analysis of the data stream algorithms that are designed in the second part of the thesis. We think that the potential of information statistics for the analysis of combinatorial problems has not yet been fully realized and that this technique will find more applications in the future. Chapter 2 contains a self-contained introduction to information theory and statistical divergences.

1.4 Publications and Contributions of the Author

The material in this thesis is based on the following publications: Chapter 3 is based on

1. A. Gronemeier. Approximating Boolean functions by OBDDs. *Discrete Applied Mathematics*, 155(2): pp. 194–209, 2007.

2. A. Gronemeier. Asymptotically optimal lower bounds on the NIH-multi-party information complexity of the AND-function and disjointness. In *Proceedings of the 26th International Symposium on Theoretical Aspects of Computer Science*, pp. 505–516, 2009.

3. A. Gronemeier. NOF-multiparty information complexity bounds for pointer jumping. In *Mathematical Foundations of Computer Science 2006*, volume 4162 of *LNCS*, pp. 459–470, 2006.

Chapter 4 is based on

4. A. Gronemeier and M. Sauerhoff. Applying approximate counting for computing the frequency moments of long data streams. *Theory of Computing Systems*, 44(3): pp. 332–348, 2009.

Both authors contributed equally to this publication. This thesis contains major improvements on some of the result of this work.

Chapter 2

Mathematical Tools

This section contains the unavoidable preliminaries that are needed in the following. We will first define our notation for some basic mathematical concepts, then we will give a self-contained introduction to information theory and statistical divergences.

2.1 Notation

In this section we will define some basic notation that is used throughout the whole text. Additional application specific notation will be defined in the sections where it is needed. Note that the most important notation and symbols are listed in Appendix B.1 for reference. This list also provides references to the definitions where the symbols are defined.

2.1.1 Sets

The sets of the natural and real numbers are denoted by the symbols $\mathbb{N}$ and $\mathbb{R}$, respectively. The closed interval of real numbers $x \in \mathbb{R}$ such that $a \leq x \leq b$ is denoted by $[a, b]$, the open interval of real numbers $x \in \mathbb{R}$ such that $a < x < b$ is denoted by (a, b), and half-open intervals are denoted by $(a, b]$ and $[a, b)$, respectively. In general, we use capital letters for sets. Often we use calligraphic capital letters to aid the distinction from random variables that are also denoted by capital letters. Conditions that must be satisfied by the elements of a set are preceded by a colon, for example $S = \{n \in \mathbb{N} : n \text{ is odd}\}$ is the set of all odd natural numbers. Several conditions that must hold simultaneously are usually separated by commas, hence $T = \{n \in \mathbb{N} : n \text{ is even}, n < 42\}$ is the set of all even natural numbers that are smaller than 42.

2.1.2 Probabilities and Random Variables

Let Ω be a discrete sample space and let $\mu \colon \Omega \longrightarrow [0, 1]$ be a probability mass function on Ω. For $A, B \subseteq \Omega$ the probability of the event A is denoted by $\Pr\{A\}$ and the conditional probability of A given B is denoted by $\Pr\{A|B\}$. If we need to emphasize that probabilities are with respect to the probability mass function μ then we write $\Pr_\mu\{A\}$ instead of $\Pr\{A\}$. Let $X : \Omega \longrightarrow \mathcal{S}$ be a random variable that takes values in the set $\mathcal{S}$. In general, we will use capital letters for random variables. Then $\text{range}(X) = \mathcal{S}$ denotes the range of the random variable X and $\text{supp}(X) = \{x \in \text{range}(X) : \Pr\{X = x\} > 0\}$ denotes the support set of X. For random variables X where $\text{range}(X) \subseteq \mathbb{R}$ we use the notation $\mathrm{E}[X]$ and $\mathrm{Var}[X]$ for the

expectation and variance of X, respectively. Note that we mainly consider finite random variables in this text, hence in this case the expectation and variance always exist. For two random variables X and Y the notation $X \sim Y$ is an abbreviation for the fact that X and Y have the same distribution. If μ is a probability mass function on the set range(X) then $X \sim \mu$ means that X is distributed with respect to the probability mass function μ. If $E \subseteq \Omega$ is an event in the underlying probability space of the random variable X then $(X|E)$ denotes the conditional distribution of X given that the event E happened. For example, if the conditional distribution of the random variables X and Y given the event E happened is identical, this can be written briefly as $(X|E) \sim (Y|E)$. For events that involve random variables we will use rather informal notation to describe the underlying subsets of the sample space, for example $\Pr\{X \text{ odd}, X \geq 0\}$ denotes the probability of the event that the random variable X is nonnegative and odd. Note that a colon denotes the conjunction of events in this context.

2.1.3 *O*-Notation

We use the standard O-notation to hide asymptotically irrelevant constants in our upper and lower bounds on the space complexity of algorithms. A definition of O, o, Ω, ω, and Θ can be found in many introductory texts on algorithms and complexity, for example [74, 67, 5]. Usually, the O-notation is used to characterize the asymptotic growth of some resource with respect to a single parameter that is understood from the context, most of the time the size of the input. For data stream algorithms it has become a frequent practice to use several parameters in a single application of the O-Notation. For example, the space complexity of some randomized algorithm may be $O(\log(n)/\epsilon)$ where n denotes the size of the input and ϵ denotes the adjustable error probability of the algorithm. In this case the asymptotic bound holds simultaneously for n and ϵ as n approaches infinity and ϵ approaches zero. More formally, there are constants n_0, ϵ_0, and c such that for all $n \geq n_0$ and all $\epsilon \leq \epsilon_0$ the algorithm uses at most $c \log(n)/\epsilon$ bits of memory. The asymptotic limits of the parameters are usually evident from the context. In most cases input sizes approach infinity whereas error parameters approach zero.

2.1.4 Miscellaneous

If the bounds of the index of summation in a sum can be inferred from the context we will sometimes drop the bounds in our notation and simply write $\sum_i x_i$. For example, the expectation of the random variable X can be written as $\sum_x \Pr\{X = x\} \cdot x$ since in this case it is obvious that the sum should be taken over all values x in the range of X.

We call $\widetilde{x} \in \mathbb{R}$ an ϵ-approximation of the value $x \in \mathbb{R}$ if $|\widetilde{x} - x| \leq \epsilon x$. Note that this is the case if and only if $\widetilde{x} \geq (1 - \epsilon)x$ and $\widetilde{x} \leq (1 + \epsilon)x$.

Finally, in this text the symbol e denotes the base of the exponential function which is sometimes called Euler's number, hence $e = \exp(1)$.

2.2 A Self-Contained Introduction to Information Theory

In Section 3.2 we will describe a proof method for lower bounds on the communication complexity of functions that is based on information theoretical arguments. Although information theory is an established mathematical theory that is covered by some excellent textbooks, for example the monograph by Cover and Thomas [29], we believe that information theory is not

so widely known in the computer science community such that an introduction to information theory could be dispensed with. Therefore this section contains a self-contained introduction to information theory and the related topic of statistical divergences. Readers with a sufficient background in this area can skim this section to pick up our notation for information theoretical quantities and statistical divergences.

2.2.1 Introduction

Information theory is a mathematical theory that quantifies "information", a rather elusive term, and explores the properties of this concept. This branch of mathematics was essentially established by a single 1948 publication, namely "A Mathematical Theory of Communication" by Claude Elwood Shannon [69], that was republished as a book [70] with contributions by Warren Weaver in 1949. In this publication Shannon solved two important open problems in communication theory: How can information be encoded efficiently and how much information can be transmitted over a given communication channel. The solution to these problems was the cornerstone for a unifying theory of information that has found numerous applications to communication, mathematics, probability theory, statistics, computer science, physics, economy, and many more fields of science (see [29] for an overview). One of the key insights of Shannon was to separate the quantitative aspects of communication from the semantic aspects (Shannon [70], p. 31):

> "The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point. Frequently the messages have *meaning*; that is they refer to or are correlated according to some system with certain physical or conceptual entities. These semantic aspect of communication are irrelevant to the engineering problem. The significant aspect is that the actual message is one *selected from a set* of possible messages."

In Shannon's information theory the messages are selected at random from a set of possible messages, hence the main subject-matter of information theory are certain functionals of probability distributions. In the following sections we give a self-contained introduction to the mathematical basics of information theory that is sufficient for the aims of this thesis and we try to build some intuition about information theory. A more comprehensive introduction to information theory can be found, for instance, in the books by Cover and Thomas [29] and by Fano [36]. Warren Weaver's chapter in [70] discusses the implications of Shannon's quantitative information theory for the semantics of communication.

2.2.2 Entropy

One of the most important concepts of information theory is the *entropy* of a random variable which measures the uncertainty about the value of the random variable. The entropy of a variable is measured in *bit*, a contraction of the words *binary digit* that was coined by John W. Tukey while he was working at Bell Labs.

Definition 2.2.1 (Entropy)**.** The entropy $\mathrm{H}(X)$ of a finite random variable $X \in \mathcal{X}$ with the probability mass function $p(x)$ is defined by

$$\mathrm{H}(X) = -\sum_{x \in \mathcal{X}} p(x) \log p(x)$$

with the convention that $0 \log 0 = 0$. We also write $\mathrm{H}(p)$ instead of $\mathrm{H}(X)$.

This definition can be justified axiomatically as the only measure of information (up to constant factors) that has certain desirable properties, but it was already pointed out by Shannon [69] that the real justification for this definition resides in its implications, because entropy emerges in the answers to many natural communication problems. The convention that $0 \log 0 = 0$ is based on a continuity argument since $\lim_{x \to 0} x \log x = 0$. Note that the definition of entropy does not depend on the domain or the range of the distribution function. Hence we could also define the entropy of any finite probability space.

Intuition about entropy is best gained by looking at some examples:

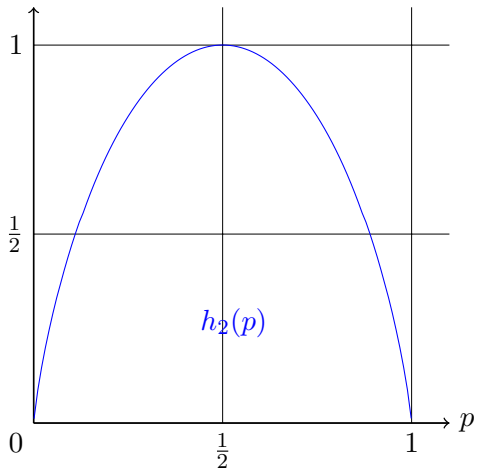
Example 2.2.2. The simplest nontrivial random variable is a binary random variable. Let the random variable X be defined as follows:

$$X = \begin{cases} 0 & \text{with probability } p \text{ and} \\ 1 & \text{with probability } 1 - p. \end{cases}$$

In this case the entropy of X is a function $h_2(p)$ of the parameter p

$$\mathrm{H}(X) = h_2(p) = -p \log p - (1 - p) \log(1 - p)$$

and can be visualized easily by the plot of the function $h_2(p)$:



Note that the function $h_2(p)$ matches our intuition about the uncertainty of the value of X. If $p = 0$ or $p = 1$ then X is a constant and the uncertainty about the value of X is 0. The uncertainty about X is maximal if $p = 1/2$ because in this case neither $X = 0$ nor $X = 1$ is more likely, and it decreases monotonically if one of the values 0 or 1 becomes more likely than the other.

Since binary random variables occur frequently, we will reserve the name $h_2(p)$ for the entropy of binary random variables with the parameter p as seen above:

Definition 2.2.3 (Binary entropy function)**.** The binary entropy function $h_2\colon [0,1] \longrightarrow [0,1]$ is defined by

$$h_2(p) = -p\log(p) - (1-p)\log(1-p)\ .$$

The last example shows that the entropy of a binary random variable lies in the unit interval. General lower and upper bound on the entropy of random variables are proved in the following proposition.

Proposition 2.2.4 (Bounds on entropy)**.** *Let $X \in \mathcal{X}$ be a finite random variable. Then*

$$0 \leq \mathrm{H}(X) \leq \log|\operatorname{supp}(X)|$$

with $\mathrm{H}(X) = 0$ if and only if X is a constant and $\mathrm{H}(X) = \log|\operatorname{supp}(X)|$ if and only if X is uniformly distributed in $\operatorname{supp}(X)$.

Proof. The lower bound follows immediately from the fact that $\log 1/p(x) \geq 0$ for probabilities $p(x)$ with equality if and only if $p(x) = 1$. The upper bound follows from the strict concavity of the log function and Jensen's inequality (see Thm. A.2.1 in Appendix A.2):

$$\mathrm{H}(X) = \sum_{x\in\operatorname{supp}(X)} p(x)\log\frac{1}{p(x)} \ \leq\ \log\sum_{x\in\operatorname{supp}(X)}\frac{p(x)}{p(x)} \ =\ \log|\operatorname{supp}(X)|\ . \tag{2.1}$$

Here Jensen's inequality holds with equality if and only if $p(x)$ is a constant that is independent of x. This implies the claim for the case of equality. □

The entropy of a random variable is closely related to the minimal average length of a binary encoding of the random variable. One of Shannon's main results in [69] was the *source coding theorem* which states that the entropy of a random variable X is a lower bound for the average length of a prefix-free binary encoding of X. In a prefix-free encoding no codeword is a prefix of another codeword.

Example 2.2.5. Let $X \in \{1,\dots,n+1\}$ be a random variable with the probability mass function $p(x) = 2^{-x}$ for $x \in \{1,\dots,n\}$ and $p(n+1) = 2^{-n}$. Then

$$\mathrm{H}(X) = -\sum_{i=1}^{n} 2^{-i}\log 2^{-i} - 2^{-n}\log 2^{-n} = \sum_{i=1}^{n} 2^{-i}i + 2^{-n}n = 2 - 2^{-(n-1)}\ .$$

Now suppose that we want to transmit the value of X using a binary encoding. Clearly, we can encode the value of X as a $\lceil\log n\rceil$ digit binary number. But if we take the distribution of X into account then we can do better on average. Consider the following encoding of X. Note that the encoding of $X = n+1$ is a special case that breaks the pattern of the preceding cases:

value of X	encoding
1	0
2	10
3	110
$\vdots$	$\vdots$
n	$\underbrace{1\ldots1}_{n-1}0$
$n+1$	$\underbrace{1\ldots1}_{n-1}1$

Clearly, the *average* message length of this encoding is $\mathrm{H}(X)$ since for all $x \in \{1,\ldots,n\}$ the length of the message for $X = x$ is exactly $x = -\log 2^{-x} = -\log p(x)$ and the length of the message for $X = n+1$ is also $-\log p(X)$. Instead of $\lceil \log n \rceil$ bits on average that are used by the simple encoding as binary numbers with uniform length, the average length of this encoding is less than two bits. This encoding, a so called *Rice-code*, is a special case of an encoding that was introduced by Golomb [42]. By Shannon's source coding theorem, this encoding is optimal with respect to the average message length.

The last example explains the choice of the word "bit" as the unit of entropy. The entropy of a random variable is the average number of binary digits that is needed for a binary representation of the value of a random variable. We repeat the main argument for the optimality of the encoding in the example as an explicit observation, since the interpretation of the entropy as the expectation of a random variable can be useful in other situations. Note that the expression $\log p(X)$ for a random variable X with probability mass function p in the observation might look a little self-referential at first sight, but it is a well defined random variable which takes the value $\log p(x)$ given that the event $X = x$ occurs.

Observation 2.2.6. Let $X \in \mathcal{X}$ be a random variable with probability mass function $p(x)$. Then the entropy of X is the expected value of $-\log p(X)$, thus $\mathrm{H}(X) = -\mathrm{E}_p[\log p(X)]$.

The explicit definition of probability mass functions for every random variable in this section would become too tedious. Therefore we will use an abbreviated notation for probabilities concerning random variables in this part of the text.

Definition 2.2.7. We use the notation $p(x)$ as a short form for $\Pr\{X = x\}$. The relationship between the value and the random variable is established by the letter that is used, for example $p(x,y) = \Pr\{X = x, Y = y\}$. Similarly, $p(x|y)$ is the short form of $\Pr\{X = x | Y = y\}$.

The definition of entropy can be generalized to tuples of random variables.

Definition 2.2.8 (Joint entropy)**.** The joint entropy $\mathrm{H}(X_1,\ldots,X_n)$ of the finite random variables $(X_1,\ldots,X_n) \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_n$ is defined by

$$\mathrm{H}(X_1,\ldots,X_n) = -\sum_{(x_1,\ldots,x_n)\in\mathcal{X}_1\times\cdots\times\mathcal{X}_n} p(x_1,\ldots,x_n)\log p(x_1,\ldots,x_n)$$

with the convention that $0\log 0 = 0$.

Note that this does not add anything new to Definition 2.2.1. A tuple of n random variables $X_i \in \mathcal{X}_i$ for $i \in \{1,\ldots,n\}$ can always be considered as a single random variable $X = (X_1,\ldots,X_n)$ with the range $\mathcal{X}_1 \times \cdots \times \mathcal{X}_n$. The reader should keep in mind

that, by the same reasoning, all statements about the entropy of a single random variable can also be applied to the joint entropy of several random variables. From now on we will only mention joint entropy explicitly, if it is essential for the statement of a claim.

2.2.3 Conditional Entropy

We have seen that the definition of joint entropy does not differ substantially from the definition of entropy, but the situation of several – possibly dependent – random variables becomes more interesting, if we consider the relations between these variables. For example, how does the knowledge of one random variable affect the uncertainty about another related random variable? To answer this question, we need to define the conditional entropy of a random variable X given another random variable Y, a measure of the average uncertainty about X if Y is known.

Definition 2.2.9 (Conditional entropy)**.** Let $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$ be random variables and let E be an event in the underlying probability space. Then $\mathrm{H}(X|E)$ denotes the entropy of X with respect to the conditional distribution of X given that the event E occurred and the conditional entropy $\mathrm{H}(X|Y)$ of X given Y is defined by

$$\mathrm{H}(X|Y) = \sum_{y \in \mathcal{Y}} p(y)\, \mathrm{H}(X|Y=y)$$

with the convention that $p(y)\,\mathrm{H}(X|Y=y) = 0$ if $p(y) = 0$.

Note that we have to distinguish carefully between *conditioning on events* and *conditioning on variables*. The conditional entropy of X given an event E is the entropy of X with respect to the conditional distribution of X given that the event E occurred, while the conditional entropy of X given another random variable Y is the weighted average of the conditional entropy of X with respect to the events $Y = y$ for all $y \in \mathcal{Y}$. Since the two different uses of conditional entropy are not distinguished by notation, the reader has to infer the meaning from context. If we condition on random variables and events then we will first list the variables and then the events to aid the reader in distinguishing variables and events.

Example 2.2.10. Let the random variable $X \in \{1, \ldots, 6\}$ be the result of rolling a fair six-sided die and let the random variable Y indicate whether the result is at most 2, say $Y = 0$ if X is one or two and $Y = 1$, otherwise. Then $\mathrm{H}(X) = \log 6$ and

$$\mathrm{H}(X|Y) = \Pr\{Y=0\}\,\mathrm{H}(X|Y=0) + \Pr\{Y=1\}\,\mathrm{H}(X|Y=1) = \frac{1}{3}\log 2 + \frac{2}{3}\log 4 = \frac{5}{3} < \log 6\,.$$

Note that, like our intuition suggests, knowing Y reduces our uncertainty about X. If on the other hand $Z \in \{0, 1\}$ is obtained by throwing a fair coin independently from the result of X then, by using the independence of X and Z, it is easy to verify that

$$\mathrm{H}(X|Z) = \log 6 = \mathrm{H}(X)\,.$$

This matches our intuition. Knowing the result of the coin does not reduce our uncertainty about the result of the die because the random experiments are independent.

Our observations from the last example are generalized in the following proposition.

Proposition 2.2.11 (Bounds on conditional entropy)**.** *Let $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$ be finite random variables. Then*

$$0 \leq \mathrm{H}(X|Y) \leq \mathrm{H}(X)$$

with $\mathrm{H}(X|Y) = 0$ if and only if X is a function of Y and $\mathrm{H}(X|Y) = \mathrm{H}(X)$ if and only if X and Y are independent.

Proof. The upper bound follows from the concavity of the log function and Jensen's inequality:

$$\begin{aligned}
\mathrm{H}(X|Y) &= \sum_{y\in\mathcal{Y}} p(y)\,\mathrm{H}(X|Y=y) && (2.2)\\
&= \sum_{y\in\mathcal{Y}} p(y) \sum_{x\in\mathcal{X}} -p(x|y)\log p(x|y) && (2.3)\\
&= \sum_{x\in\mathcal{X}} p(x) \sum_{y\in\mathcal{Y}} p(y|x)\log\frac{p(y)}{p(x)\cdot p(y|x)} && (2.4)\\
&\leq \sum_{x\in\mathcal{X}} p(x)\log\sum_{y\in\mathcal{Y}}\frac{p(y)}{p(x)} && (2.5)\\
&= -\sum_{x\in\mathcal{X}} p(x)\log p(x) && (2.6)\\
&= \mathrm{H}(X)\,. && (2.7)
\end{aligned}$$

The log function is strictly concave, thus Jensen's inequality (2.5) holds with equality if and only if

$$\log\frac{p(y)}{p(x)\cdot p(y|x)} = -\log p(x|y) \tag{2.8}$$

is independent of y for every fixed value of x. This is the case if and only if X and Y are independent. The lower bound follows immediately from the definition of conditional entropy (Def. 2.2.9) and the lower bound for entropy (Prop. 2.2.4). Since $\mathrm{H}(X|Y) = 0$ if and only if $\mathrm{H}(X|Y=y) = 0$ for every $y \in \mathrm{supp}(Y)$, in this case the value of X must be uniquely determined by the value of Y, hence X must be a function of Y. □

The upper bound in this proposition is often paraphrased as "conditioning reduces entropy". Note that only conditioning on *variables* reduces the entropy whereas conditioning on *events* can either reduce or increase the entropy. For example, if Y is a random bit and the random variable X is the constant 0 given that $Y = 0$ and the value of X is chosen uniformly at random from 0 and 1 given that $Y = 1$, then $\mathrm{H}(X) = h_2(3/4)$, $\mathrm{H}(X|Y=0) = 0 < \mathrm{H}(X)$, and $\mathrm{H}(X|Y=1) = h_2(1/2) > \mathrm{H}(X)$.

The last proposition has an additional interpretation: The entropy $\mathrm{H}(X)$ is a functional of the probability mass function $p(x)$. By the law of total probability, $p(x) = \sum_{y\in\mathcal{Y}} p(y)p(x|y)$, hence $p(x)$ can be considered as a convex combination of several conditional probability mass functions. On the other hand, the conditional entropy $\mathrm{H}(X|Y)$ is a convex combination of the entropy of the conditional probability mass functions $p(x|y)$ for all $y \in \mathcal{Y}$. In this light the last proposition states that $\mathrm{H}(X)$ is a *concave* functional of $p(x)$.

In general, the conditional entropy $\mathrm{H}(X|Y)$ is reduced if we add an additional variable to the condition, thus we have $\mathrm{H}(X|Y) \geq \mathrm{H}(X|Y,Z)$. Now we will consider two special cases, in which the entropy is not reduced by an additional variable in the condition. In the first case the variable Z is a function of the variable Y.

Proposition 2.2.12 (Functions of conditions)**.** *Let $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$ be finite random variables, let $\mathcal{S}$ be a set, and let $f\colon \mathcal{Y} \longrightarrow \mathcal{S}$ be a function. Then*

$$\mathrm{H}(X|Y) = \mathrm{H}(X|Y, f(Y)) \ .$$

Proof. Observe that $\Pr\{Y=y, f(Y)=s\} = 0$ if $s \neq f(y)$. Then

$$\begin{aligned}
\mathrm{H}(X|Y, f(Y)) &= \sum_{y\in\mathcal{Y}, s\in\mathcal{S}} \Pr\{Y=y, f(Y)=s\}\, \mathrm{H}(X|Y=y, f(Y)=s) && (2.9)\\
&= \sum_{y\in\mathcal{Y}} \Pr\{Y=y, f(Y)=f(y)\}\, \mathrm{H}(X|Y=y, f(Y)=f(y)) && (2.10)\\
&= \sum_{y\in\mathcal{Y}} \Pr\{Y=y\}\, \mathrm{H}(X|Y=y) && (2.11)
\end{aligned}$$

where the last equation is due to the fact that $f(Y) = f(y)$ is implied by $Y = y$. □

In the second case we consider random variables that are conditionally independent. A short introduction to conditional independence can be found in Appendix A.1.

Proposition 2.2.13 (Conditional independence)**.** *Let X, Y, and Z be finite random variables. Then X and Y are conditionally independent given Z if and only if*

$$\mathrm{H}(X|Y, Z) = \mathrm{H}(X|Z) \ .$$

Proof. Note that X and Y are conditionally independent if and only if X and Y are independent with respect to the conditional distribution of X and Y given that $Z = z$ for all $z \in \mathrm{supp}(Z)$. Then it suffices to apply Proposition 2.2.11 to each term of the expansion

$$\mathrm{H}(X|Y, Z) = \sum_{z} \Pr\{Z=z\} \cdot \mathrm{H}(X|Y, Z=z) \ . \tag{2.12}$$

□

The next proposition, the chain rule for entropy, describes a property that seems very natural for a useful measure of uncertainty. We mentioned before that the definition of entropy can be justified axiomatically. The chain rule is one of the desirable properties of measures of uncertainty that are postulated in the axiomatic definition of entropy.

Proposition 2.2.14 (Chain rule for entropy)**.** *Let $X_1, \ldots, X_n$ be finite random variables. Then*

$$\mathrm{H}(X_1, \ldots, X_n) = \sum_{i=1}^{n} \mathrm{H}(X_i|X_1, \ldots, X_{i-1}) \ .$$

Proof. By using the fact that $p(x_1, \ldots, x_n) = \prod_{i=1}^{n} p(x_i|x_1, \ldots, x_{i-1})$ in the definition of

entropy (Def. 2.2.1) we obtain

$$\begin{aligned}
\mathrm{H}(X_1,\dots,X_n) &= -\sum_{x_1,\dots,x_n} p(x_1,\dots,x_n)\log p(x_1,\dots,x_n) && (2.13)\\
&= -\sum_{x_1,\dots,x_n} p(x_1,\dots,x_n)\log\prod_{i=1}^{n} p(x_i|x_1,\dots,x_{i-1}) && (2.14)\\
&= -\sum_{x_1,\dots,x_n} p(x_1,\dots,x_n)\sum_{i=1}^{n}\log p(x_i|x_1,\dots,x_{i-1}) && (2.15)\\
&= -\sum_{i=1}^{n}\sum_{x_1,\dots,x_n} p(x_1,\dots,x_n)\log p(x_i|x_1,\dots,x_{i-1}) && (2.16)\\
&= -\sum_{i=1}^{n}\sum_{x_1,\dots,x_i} p(x_1,\dots,x_i)\log p(x_i|x_1,\dots,x_{i-1}) && (2.17)\\
&= \sum_{i=1}^{n}\mathrm{H}(X_i|X_1,\dots,X_{i-1})\,. && (2.18)
\end{aligned}$$

□

The chain rule can be weakened to an inequality if we use that entropy is reduced by conditions.

Proposition 2.2.15 (Subadditivity of entropy)**.** *Let $X_1,\dots,X_n$ be finite random variables. Then*

$$\mathrm{H}(X_1,\dots,X_n)\le\sum_{i=1}^{n}\mathrm{H}(X_i)$$

with equality if and only if $X_1,\dots,X_n$ are independent.

Proof. The inequality follows immediately from the chain rule for entropy (Prop. 2.2.14) and the fact that conditioning reduces entropy (Prop. 2.2.11). The claim for the case of equality follows by induction if the case for equality in Prop. 2.2.11 is applied to every term in the chain rule sum. □

The chain rule $\mathrm{H}(X,Y)=\mathrm{H}(Y)+\mathrm{H}(X|Y)$ also offers an alternative definition of conditional entropy in terms of joint entropy.

Remark 2.2.16 (Alternative definition of conditional entropy). Let X and Y be finite random variables. Then $\mathrm{H}(X|Y)=\mathrm{H}(X,Y)-\mathrm{H}(Y)$.

Finally, note that the preceding propositions about entropy also hold for conditional entropy. All claims in the following corollary can be proved by applying the corresponding propositions for entropy to each term in the expansion of the conditional entropy according to Def. 2.2.9.

Corollary 2.2.17 (Properties of conditional entropy)**.** *Let X, Y, Z, and $X_1, \dots, X_n$ be finite random variables. All propositions in this section do also apply to conditional entropy:*

- *Lower and upper bound* (Prop. 2.2.4)*:* $0 \leq \mathrm{H}(X|Z) \leq \log|\operatorname{supp}(X)|$.
- *Multiple conditions* (Def. 2.2.9)*:* $\mathrm{H}(X|Y,Z) = \sum_y p(y)\,\mathrm{H}(X|Z, Y=y)$.
- *Conditioning reduces entropy* (Prop. 2.2.11)*:* $\mathrm{H}(X|Y,Z) \leq \mathrm{H}(X|Z)$.
- *Chain rule* (Prop. 2.2.14)*:* $\mathrm{H}(X_1,\dots,X_n|Z) = \sum_{i=1}^n \mathrm{H}(X_i|X_1,\dots,X_{i-1},Z)$.
- *Subadditivity* (Prop. 2.2.15)*:* $\mathrm{H}(X_1,\dots,X_n|Z) \leq \sum_{i=1}^n \mathrm{H}(X_i|Z)$.

If we condition on Z in these propositions, then independence of random variables has to be replaced by conditional independence given Z in all propositions.

2.2.4 Mutual Information

In the last section we have seen that knowing a random variable Y can reduce the uncertainty about a related random variable X. Intuitively, in this case Y contains information about the value of X. The following definition quantifies the concept of information.

Definition 2.2.18 (Mutual information)**.** The mutual information of the finite random variables X and Y is defined by

$$\mathrm{I}(X:Y) = \mathrm{H}(X) - \mathrm{H}(X|Y)\ .$$

Viewed together with our interpretation of $\mathrm{H}(X)$ as the uncertainty about X and $\mathrm{H}(X|Y)$ as the average uncertainty about X if we know Y, this definition matches our intuition about information and uncertainty. The mutual information of X and Y is the average reduction of the uncertainty about X if we learn the value of the random variable Y.

Remark 2.2.19. Let X be a finite random variable. Then $\mathrm{H}(X) = \mathrm{I}(X:X)$. Because of this equality, the entropy of X is also called the *self information of X*.

Again, we first look at a simple example to build some intuition about mutual information.

Example 2.2.20. Suppose that X and X' are chosen independently, uniformly at random from the set $\{0,1\}$ and that $Y = X + X'$. Then Y contains information about X. For example, if $Y = 0$ then we are certain that $X = 0$ and if $Y = 2$ then certainly $X = 1$, thus $\mathrm{H}(X|Y=0) = \mathrm{H}(X|Y=2) = 0$. Only if $Y = 1$, this happens with probability $1/2$, we are uncertain about the value of X. Since $\Pr\{X=1|Y=1\} = 1/2$ we have $\mathrm{H}(X|Y=1) = 1$. Then, by the definition of mutual information (Def. 2.2.18),

$$\mathrm{I}(X:Y) = \mathrm{H}(X) - \mathrm{H}(X|Y) = 1 - \frac{1}{2} = \frac{1}{2}\ .$$

Conversely, knowing X also reduces our uncertainty about Y. If $X = 0$ then we can conclude that $Y = X'$ and therefore $Y \neq 2$, if on the other hand $X = 1$ then $Y = 1 + X'$ and $Y \neq 0$. Hence $\mathrm{H}(Y|X=0) = \mathrm{H}(Y|X=1) = \mathrm{H}(X') = 1$. Note that $\mathrm{H}(Y) = 3/2$. Then Def. 2.2.18 yields

$$\mathrm{I}(Y:X) = \mathrm{H}(Y) - \mathrm{H}(Y|X) = \frac{3}{2} - 1 = \frac{1}{2}\ .$$

In the last example the variable X contains as much information about the variable Y, as the variable Y contains information about the variable X. This is not a coincidence.

Proposition 2.2.21 (Symmetry of mutual information)**.** *Let X and Y be finite random variables. Then*

$$\mathrm{I}(X:Y) = \mathrm{I}(Y:X) \ .$$

Proof. Apply the chain rule for entropy (Prop. 2.2.14) in different orders to get

$$\mathrm{H}(X) + \mathrm{H}(Y|X) = \mathrm{H}(X,Y) = \mathrm{H}(Y) + \mathrm{H}(X|Y) \tag{2.19}$$

and subtract $\mathrm{H}(Y|X) + \mathrm{H}(X|Y)$ from this equation. □

Intuitively, the information that can be gained about a random variable X should be non-negative and bounded from above by $\mathrm{H}(X)$. This intuition is confirmed in the following proposition.

Proposition 2.2.22 (Bounds on mutual information)**.** *Let X and Y be finite random variables such that $\mathrm{H}(X) \leq \mathrm{H}(Y)$. Then*

$$0 \leq \mathrm{I}(X:Y) \leq \mathrm{H}(X)$$

with $\mathrm{I}(X:Y) = 0$ if and only if X and Y are independent and $\mathrm{I}(X:Y) = \mathrm{H}(X)$ if and only if X is a function of Y.

Proof. Follows immediately from the definition of mutual information (Def. 2.2.18) and the bounds on conditional entropy (Prop. 2.2.11). □

Note that in general $\mathrm{I}(X:Y) \leq \min\{\mathrm{H}(X), \mathrm{H}(Y)\}$ by the symmetry of mutual information.

Analogously to conditional entropy, one can define conditional mutual information. The conditional mutual information of X and Y given Z is the average mutual information of X and Y if the value of Z is known.

Definition 2.2.23 (Conditional mutual information)**.** Let X, Y, and Z be finite random variables and let E be an event in the underlying probability space. Then $\mathrm{I}(X:Y|E)$ denotes the mutual information of X and Y with respect to the joint conditional distribution of X and Y given that the event E occurred and the conditional mutual information $\mathrm{I}(X:Y|Z)$ of X and Y given Z is defined by

$$\mathrm{I}(X:Y|Z) = \sum_z p(z)\, \mathrm{I}(X:Y|Z{=}z) \ .$$

An alternative equivalent definition of conditional mutual information is stated as a proposition in the following. Note the similarity to Definition 2.2.18.

Proposition 2.2.24. *Let X, Y, and Z be finite random variables. Then*

$$\mathrm{I}(X:Y|Z) = \mathrm{H}(X|Z) - \mathrm{H}(X|Y,Z) \ .$$

Proof. Follows immediately from the definitions of conditional mutual information (Def. 2.2.23), mutual information (Def. 2.2.18) and conditional entropy (Def. 2.2.9). □

For mutual information there is also a chain rule that resembles the chain rule for entropy.

Proposition 2.2.25 (Chain rule for mutual information)**.** *Let $X_1, \dots, X_n$, and Y be finite random variables. Then*

$$\mathrm{I}(X_1, \dots, X_n : Y) = \sum_{i=1}^{n} \mathrm{I}(X_i : Y | X_1, \dots, X_{i-1}) \ .$$

Proof. Follows immediately from the definition of mutual information (Def. 2.2.18) and the chain rule for entropy (Prop. 2.2.14):

$$\begin{aligned} \mathrm{I}(X_1, \dots, X_n : Y) &= \mathrm{H}(X_1, \dots, X_n) - \mathrm{H}(X_1, \dots, X_n | Y) && (2.20) \\ &= \sum_{i=1}^{n} \left(\mathrm{H}(X_i | X_1, \dots, X_{i-1}) - \mathrm{H}(X_i | X_1, \dots, X_{i-1}, Y) \right) && (2.21) \\ &= \sum_{i=1}^{n} \mathrm{I}(X_i : Y | X_1, \dots, X_{i-1}) \ . && (2.22) \end{aligned}$$

□

The following Proposition, a simple consequence of the chain rule, states that additional variables increase the mutual information.

Proposition 2.2.26. *Let X, Y, and Z be finite random variables. Then*

$$\mathrm{I}(X, Y : Z) \geq \mathrm{I}(X : Z) \ .$$

Proof. By the chain rule and non-negativity of mutual information we have

$$\mathrm{I}(X, Y : Z) = \mathrm{I}(X : Z) + \mathrm{I}(Y : Z | X) \geq \mathrm{I}(X : Z) \ .$$

□

Just like the chain rule for entropy, the chain rule for mutual information can also be weakened to an inequality, but note carefully that in this case we need stronger assumptions than for the chain rule for entropy, namely the independence of the variables.

Proposition 2.2.27 (Superadditivity of mutual information)**.** *Let $X_1, \dots, X_n$, and Y be finite random variables such that the variables $X_1, \dots, X_n$ are independent. Then*

$$\mathrm{I}(X_1, \dots, X_n : Y) \geq \sum_{i=1}^{n} \mathrm{I}(X_i : Y)$$

with equality if and only if $X_1, \dots, X_n$ are conditionally independent given Y.

Proof. Follows immediately from the definition of mutual information (Def. 2.2.18) and the

subadditivity of entropy (Prop. 2.2.15):

$$\begin{aligned} \mathrm{I}(X_1, \ldots, X_n : Y) &= \mathrm{H}(X_1, \ldots, X_n) - \mathrm{H}(X_1, \ldots, X_n|Y) && (2.23)\\ &= \sum_{i=1}^{n} \mathrm{H}(X_i) - \mathrm{H}(X_1, \ldots, X_n|Y) && (2.24)\\ &\geq \sum_{i=1}^{n} \mathrm{H}(X_i) - \sum_{i=1}^{n} \mathrm{H}(X_i|Y) && (2.25)\\ &= \sum_{i=1}^{n} \mathrm{I}(X_i : Y)\ . && (2.26) \end{aligned}$$

In line (2.24) the independence of $X_1, \ldots, X_n$ is used. The inequality (2.25) holds with equality if and only if $X_1, \ldots, X_n$ are conditionally independent given Y. □

Finally, all propositions about mutual information in this section can also be extended to conditional mutual information. The results are summarized in the following corollary.

Corollary 2.2.28 (Properties of conditional mutual information)**.** *Let X, Y, Z, W, and $X_1, \ldots, X_n$ be finite random variables. All propositions in this section do also apply to conditional mutual information:*

- *Symmetry* (Prop. 2.2.21)*:* $\mathrm{I}(X : Y|Z) = \mathrm{I}(Y : X|Z)$.
- *Lower and upper bound* (Prop. 2.2.22)*:* $0 \leq \mathrm{I}(X : Y|Z) \leq \min\{\mathrm{H}(X|Z), \mathrm{H}(Y|Z)\}$.
- *Multiple conditions* (Def. 2.2.23)*:* $\mathrm{I}(X : Y|W, Z) = \sum_w p(w)\, \mathrm{I}(X : Y|Z, W = w)$.
- *Chain rule* (Prop. 2.2.25)*:* $\mathrm{I}(X_1, \ldots, X_n : Y|Z) = \sum_{i=1}^{n} \mathrm{I}(X_i : Y|X_1, \ldots, X_{i-1}, Z)$.
- *Additional variables* (Prop. 2.2.26)*:* $\mathrm{I}(X, Y : Z|W) \geq \mathrm{I}(X : Z|W)$.
- *Superadditivity* (Prop. 2.2.27)*: If $X_1, \ldots, X_n$ are conditionally independent given Z, then $\mathrm{I}(X_1, \ldots, X_n : Y|Z) \geq \sum_{i=1}^{n} \mathrm{I}(X_i : Y|Z)$ with equality if and only if $X_1, \ldots, X_n$ are conditionally independent given Y, Z.*

All claims of the corollary can be proved by applying the corresponding proposition for mutual information to each term in the sum expansion of the conditional mutual information according to Definition 2.2.18.

2.2.5 Fano's Inequality

So far, we have quantified the concepts of uncertainty and information and we have appealed to the reader's intuition about these concepts. To make these concepts more tangible, we next look at a practical consequence of uncertainty: The larger the uncertainty about a random variable is, the harder it is to predict the actual value of the random variable. Suppose that we are interested in the outcome of a random experiment that is described by the random variable X, but we can only observe a related random variable Y and we need to predict the value of X based on our observation Y. This setting appears frequently in statistics. Here we assume that our prediction is a function $f(Y)$ of Y and that the joint distribution of X and Y is known. We are interested in the error $\epsilon = \Pr\{f(Y) \neq X\}$ of the prediction.

Certainly, this error is related to the mutual information of X and Y. We will first look at the extreme cases. If X is uniquely determined by Y then we can predict X from Y with zero error. If, on the other hand, X and Y are independent, then Y does not help us at all in predicting X. We could create a random variable Y' by ourselves such that X, Y' and X, Y have the same joint distribution. In the former case $\mathrm{H}(X|Y) = 0$, in the latter case $\mathrm{H}(X|Y) = \mathrm{H}(X)$. In between these extremes, we would expect lower errors of prediction for lower values of $\mathrm{H}(X|Y)$. Fano [35] made this precise.

Theorem 2.2.29 (Fano's inequality)**.** *Let $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$ be random variables, let $f\colon \mathcal{Y} \longrightarrow \mathcal{X}$ be a function, and let $\epsilon = \Pr\{f(Y) \neq X\}$. Then*

$$h_2(\epsilon) + \epsilon \log(|\mathcal{X}| - 1) \geq \mathrm{H}(X|Y) \ .$$

Proof. Let E be the following indicator variable:

$$E = \begin{cases} 1 & \text{if } f(Y) \neq X, \\ 0 & \text{if } f(Y) = X \end{cases} \tag{2.27}$$

and note that $\Pr\{E{=}1\} = \epsilon$ and $\mathrm{H}(E) = h_2(\epsilon)$. Now expand $\mathrm{H}(X, E|Y)$ using the chain rule (Cor. 2.2.17) in two different ways: First use that

$$\mathrm{H}(X, E|Y) = \mathrm{H}(X|Y) + \mathrm{H}(E|X, Y) = \mathrm{H}(X|Y) \ . \tag{2.28}$$

The last equality follows from the fact that E is completely determined by X and Y, hence E is a constant for each fixed assignment $X = x$ and $Y = y$ and $H(E|X{=}x, Y{=}y) = 0$ by Prop. 2.2.4. Next use that

$$\mathrm{H}(X, E|Y) = \mathrm{H}(E|Y) + \mathrm{H}(X|Y, E) \ . \tag{2.29}$$

Since conditioning reduces entropy (Prop. 2.2.11), we have $\mathrm{H}(E|Y) \leq \mathrm{H}(E) = h_2(\epsilon)$ and, by the properties of conditional entropy (Cor. 2.2.17), we have

$$\begin{aligned} \mathrm{H}(X|Y, E) &= \Pr\{E{=}0\} \cdot \mathrm{H}(X|Y, E{=}0) + \Pr\{E{=}1\} \cdot \mathrm{H}(X|Y, E{=}1) && (2.30)\\ &= \Pr\{E{=}1\} \cdot \mathrm{H}(X|Y, E{=}1) && (2.31)\\ &\leq \Pr\{E{=}1\} \cdot \mathrm{H}(X|E{=}1) && (2.32)\\ &\leq \epsilon \log(|\mathcal{X}| - 1) \ . && (2.33) \end{aligned}$$

In the second line we used that $\mathrm{H}(X|Y, E{=}0) = 0$ since X is determined by Y given that $E = 0$, for the first inequality we used that conditioning reduces entropy, and for the second inequality we applied the upper bound from Prop. 2.2.4 using the fact that $X \in \mathcal{X} - \{f(Y)\}$ if $E = 1$. By plugging this into (2.29) we obtain

$$\mathrm{H}(X, E|Y) \leq h_2(\epsilon) + \epsilon \log(|\mathcal{X}| - 1) \ . \tag{2.34}$$

The claimed result follows by combining (2.28) and (2.34). □

2.2.6 Statistics and Information Theory

In many fields of statistics there is a common need for measures of the dissimilarity of probability distributions. These measure are usually called *dissimilarity coefficients*, *separation measures*, or *statistical divergences*, depending on the subfield of statistics and the preferences of authors. Introductions to statistical divergences and their statistical applications can be found, for example, in publications by Csiszár and Shields [31], Liese and Vajda [57], Sgarro [68], and Le Cam and Yang [23].

We will see in the following sections that information theoretical quantities like entropy and mutual information can be regarded as special cases of statistical divergences. For our purposes, it is sufficient to look at a special class of statistical divergences, the so called *f-divergences* that were introduced and studied independently by Csiszár [30] and Ali and Silvey [2]. Moreover, we can restrict our discussion to f-divergences of discrete distributions on finite sets. We will soon see that many widely used statistical divergences are f-divergences. The properties of f-divergences that are explored in this section will apply to all of these special cases.

Definition 2.2.30 (f-divergence)**.** Let p and q be probability mass function on the finite sample space Ω and let $f\colon \mathbb{R}_+ \longrightarrow \mathbb{R}$ be a convex function such that $f(1) = 0$. The f-divergence $\mathrm{D}_f(p, q)$ of p and q is defined by

$$\mathrm{D}_f(p, q) = \sum_{\omega\in\Omega} q(\omega) f\left(\frac{p(\omega)}{q(\omega)}\right) .$$

We take $0 \cdot f(0/0) = 0$, $f(0) = \lim_{x\longrightarrow 0} f(x)$, and $0 \cdot f(a/0) = \lim_{x\longrightarrow 0} x \cdot f(a/x)$ for $a \neq 0$.

The convexity of f in this definition suggests the application of Jensen's inequality to obtain a lower bound for $\mathrm{D}_f(p, q)$.

Proposition 2.2.31 (Jensen's inequality for f-divergences)**.** *Suppose that $a_1, \ldots, a_n$ and $b_1, \ldots, b_n$ are nonnegative numbers and let $A = \sum_{i=1}^n a_i$ and $B = \sum_{i=1}^n b_i$. Then*

$$\sum_{i=1}^n b_i f\left(\frac{a_i}{b_i}\right) \geq B f\left(\frac{A}{B}\right) .$$

If f is strictly convex then this inequality holds with equality if and only if there is a constant c such that $a_i/b_i = c$ for all $i \in \{1, \ldots, n\}$.

Proof. By the convexity of f and Jensen's inequality

$$\sum_{i=1}^n \frac{b_i}{B} f\left(\frac{a_i}{b_i}\right) \geq f\left(\sum_{i=1}^n \frac{a_i}{B}\right) = f\left(\frac{A}{B}\right) . \tag{2.35}$$

The claim for equality follows immediately from the case of equality in Jensen's inequality. □

Recall that f-divergences are a measure for the dissimilarity of probability distributions, in this sense they measure the distance of probability distributions and share some properties of metrics. In fact there are f-divergences that are proper metrics, but there are also f-divergences that are neither symmetric nor satisfy the triangle inequality. A common property of metrics and f-divergences for strictly convex functions f is stated in the following proposition.

Proposition 2.2.32. *Let p and q be probability mass functions on the finite sample space Ω and let $\mathrm{D}_f(p,q)$ be an f-divergence of p and q for a strictly convex function f. Then*

$$\mathrm{D}_f(p,q) \geq 0$$

with equality if and only if $p = q$.

Proof. By Prop. 2.2.31 and by the property $f(1) = 0$

$$\mathrm{D}_f(p,q) = \sum_{\omega\in\Omega} p(\omega) f\left(\frac{p(\omega)}{q(\omega)}\right) \geq \left(\sum_{\omega\in\Omega} p(\omega)\right) f\left(\frac{\sum_{\omega\in\Omega} p(\omega)}{\sum_{\omega\in\Omega} q(\omega)}\right) = 1 \cdot f\left(\frac{1}{1}\right) = 0\,. \tag{2.36}$$

Clearly, if $p_i = q_i$ for all $i \in \{1,\ldots,n\}$ then $\mathrm{D}_f(p,q) = 0$. Conversely, if the inequality in the equation above holds with equality then, by Prop. 2.2.31, there is a constant c such that $p_i/q_i = c$ for all $i \in \{1,\ldots,n\}$. Since p and q are probability mass functions, there must be indices i and j where $i \neq j$ such that $p_i \geq q_i$ and $p_j \leq q_j$. Hence $p_i/q_i = c \geq 1$ and $p_j/q_j = c \leq 1$ and therefore $c = 1$ and consequently $p_i = q_i$ for all $i \in \{1,\ldots,n\}$. □

In the next sections we will look at three well-known f-divergences of probability distributions. Since historically these divergences have been introduced before f-divergences were formally defined, the standard notation for these divergences differs slightly from Def. 2.2.30. Here we generally try to use standard notation or at least notation that is consistent and similar to standard notation.

Kullback-Leibler Distance

A well-known statistical divergence is obtained from Def. 2.2.30 by choosing $f(x) = x\log x$. In this case the corresponding f-divergence, introduced by Kullback and Leibler [55] in 1951, is called *Kullback-Leibler distance*, *informational divergence*, or *relative entropy*.

Definition 2.2.33 (Kullback-Leibler distance)**.** Let p and q be probability mass functions on the finite sample space Ω. The Kullback-Leibler distance or relative entropy $\mathrm{D}(p,q)$ of p and q is defined by

$$\mathrm{D}(p,q) = \sum_{\omega\in\Omega} p(\omega)\log\frac{p(\omega)}{q(\omega)}$$

with the convention that $0\cdot\log(0/q(\omega)) = 0$, $0\cdot\log(0/0) = 0$, and $p(\omega)\cdot\log(p(\omega)/0) = \infty$ if $p(\omega) \neq 0$. For finite random variables X and Y we briefly write $\mathrm{D}(X,Y)$ for the Kullback-Leibler distance $\mathrm{D}(p_X,p_Y)$ of the corresponding probability mass functions p_X and p_Y.

Note that the Kullback-Leibler distance of p and q is often denoted by $D(p \parallel q)$ in the literature. We do not see the need for an additional syntactic element like $\parallel$, hence we separate p and q by a comma in accordance with Def. 2.2.30 and the standard notation for other f-divergences.

According to Sgarro [68] it has been frequently pointed out in the literature that the Kullback-Leibler distance is a rather natural statistical measure of distinguishability between probability distributions. Unfortunately, the Kullback-Leibler distance can be inconvenient and difficult to apply in applications. In general it is neither symmetric nor does it satisfy the triangle inequality [41]. In addition, if there is a single $\omega \in \Omega$ such that $q(\omega) = 0$, but $p(\omega) \neq 0$, then $\mathrm{D}(p,q) = \infty$ independently of the values of p and q on the remaining elements.

For us it is important that this f-divergence is closely related to the information theoretical quantities that are defined in the previous sections: The mutual information of the random variables X and Y is the Kullback-Leibler distance of their joint distribution $p_{XY}(x, y)$ and the product distribution of their marginal distributions $p_X(x)$ and $p_Y(y)$.

Proposition 2.2.34. *Let $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$ be finite random variables with the joint probability mass function $p_{XY}(x, y)$ and the marginal probability mass functions $p_X(x)$ and $p_Y(y)$ and let $p_X \otimes p_Y$ denote the product distribution of p_X and p_Y. Then*

$$\mathrm{I}(X : Y) = \sum_{(x,y)\in\mathcal{X}\times\mathcal{Y}} p_{XY}(x, y) \log \frac{p_{XY}(x, y)}{p_X(x)p_Y(y)} = \mathrm{D}(p_{XY}, p_X \otimes p_Y) \ .$$

Proof. Follows immediately from the definition of mutual information (Def. 2.2.18), the definition of entropy (Def. 2.2.1), and the definition of conditional entropy (Def. 2.2.1):

$$\begin{aligned} \mathrm{I}(X : Y) &= \mathrm{H}(X) - \mathrm{H}(X|Y) && (2.37)\\ &= -\sum_{(x,y)\in\mathcal{X}\times\mathcal{Y}} p_{XY}(x, y) \log p_X(x) + \sum_{(x,y)\in\mathcal{X}\times\mathcal{Y}} p_{XY}(x, y) \log \frac{p_{XY}(x, y)}{p_Y(y)} && (2.38)\\ &= \sum_{(x,y)\in\mathcal{X}\times\mathcal{Y}} p_{XY}(x, y) \log \frac{p_{XY}(x, y)}{p_X(x)p_Y(y)} \ . && (2.39) \end{aligned}$$

□

By Remark 2.2.19, the entropy of a random variable is just the self information of the variable. Hence, all information theoretical quantities that we have discussed so far can be expressed in terms of the Kullback-Leibler distance and Shannon's information theory can be regarded as a special case of a broader theory of statistical divergences. The last proposition also offers a new intuitive interpretation of mutual information: The mutual information measures how far the joint distribution of two random variables is from a product distribution.

Because of the importance of the Kullback-Leibler distance for information theory, we state Jensen's inequality for the Kullback-Leibler distance explicitly. In this case Prop. 2.2.31 is usually called *the log sum inequality*.

Corollary 2.2.35 (Log sum inequality)**.** *If $a_1, \ldots, a_n$ and $b_1, \ldots, b_n$ are nonnegative numbers, then*

$$\sum_{i=1}^{n} a_i \log \frac{a_i}{b_i} \geq \left(\sum_{i=1}^{n} a_i\right) \log \frac{\sum_{i=1}^{n} a_i}{\sum_{i=1}^{n} b_i}$$

with equality if and only if $a_i/b_i = a_j/b_j$ for all $i, j \in \{1, \ldots, n\}$ where we use the convention that $0 \cdot \log(0/b) = 0$ for all b including $b = 0$ and that $a \log(a/0) = \infty$ for all $a \neq 0$.

Total Variation Distance

The total variation distance is a well-known measure for the dissimilarity of probability distributions. In fact, it is also an f-divergence, namely the f-divergence for $f(x) = \frac{1}{2}|x - 1|$.

Definition 2.2.36 (Total variation distance)**.** Let p and q be probability mass functions on the finite sample space Ω. The total variation distance $\mathrm{V}(p,q)$ of p and q is defined by

$$\mathrm{V}(p,q) = \frac{1}{2}\sum_{\omega\in\Omega} |p(\omega) - q(\omega)| \ .$$

For finite random variables X and Y we briefly write $\mathrm{V}(X,Y)$ for the total variation distance $\mathrm{V}(p_X, p_Y)$ of the corresponding probability mass functions p_X and p_Y.

Clearly, the total variation distance is closely related to the L^1-norm: $\mathrm{V}(p,q) = \frac{1}{2}\|p-q\|_1$. The importance of the total variation distance stems from the fact that the probability of a given event for two distributions on the same sample space differs at most by the total variation distance of the distributions. This facilitates the direct comparison of probabilities for given events, a very practical measure of dissimilarity that is needed frequently in applications.

Proposition 2.2.37. *Let p and q be probability mass functions on the finite set Ω. Then*

$$\mathrm{V}(p,q) = \max\{|p(A) - q(A)| \mid A \subseteq \Omega\} \ .$$

Proof. Let $A^* = \{\omega \in \Omega | p(\omega) > q(\omega)\}$. Then $p(A^*) - q(A^*) = \max\{|p(A) - q(A)| \mid A \subseteq \Omega\}$ since both removing elements from A^* and adding elements from $\Omega - A^*$ to A^* decreases $|p(A^*) - q(A^*)|$. On the other hand we have

$$\begin{aligned}
\mathrm{V}(p,q) &= \frac{1}{2}\sum_{\omega\in\Omega} |p(\omega) - q(\omega)| && (2.40)\\
&= \frac{1}{2}\left(\sum_{\omega\in A^*} p(\omega) - q(\omega) + \sum_{\omega\notin A^*} q(\omega) - p(\omega)\right) && (2.41)\\
&= \frac{1}{2}(\ p(A^*) - q(A^*) + (1 - q(A^*)) - (1 - p(A^*))\ ) && (2.42)\\
&= p(A^*) - q(A^*) && (2.43)
\end{aligned}$$

which completes the proof. □

The following proposition shows that the total variation distance is subadditive with respect to product distributions.

Proposition 2.2.38. *For $i \in \{1,\dots,n\}$ let p_i and q_i be a probability mass function on the finite sample space Ω_i. Furthermore let $p(\omega_1,\dots,\omega_n) = p_1(\omega_1)\cdot\dots\cdot p_n(\omega_n)$ and $q(\omega_1,\dots,\omega_n) = q_1(\omega_1)\cdot\dots\cdot q_n(\omega_n)$ denote the probability mass functions of the corresponding product distributions, respectively. Then*

$$\mathrm{V}(p,q) \le \sum_{i=1}^{n} \mathrm{V}(p_i,q_i) \ .$$

Proof. First we will prove the claim for $n = 2$. Define $\delta_i(\omega_i) = p_i(\omega_i) - q_i(\omega_i)$ for all $i \in \{1,2\}$ and all $\omega_i \in \Omega_i$. Then

$$\mathrm{V}(p_1,q_1) = \frac{1}{2}\sum_{\omega_1\in\Omega_1} |\delta_1(\omega_1)| \ \text{ and } \ \mathrm{V}(p_2,q_2) = \frac{1}{2}\sum_{\omega_2\in\Omega_2} |\delta_2(\omega_2)| \ . \qquad (2.44)$$

On the other hand, by the triangle inequality,

$$\mathrm{V}(p,q) = \frac{1}{2} \sum_{\omega_1\in\Omega_1,\omega_2\in\Omega_2} |(q_1(\omega_1)+\delta_1(\omega_1))(q_2(\omega_2)+\delta_2(\omega_2)) - q_1(\omega_1)q_2(\omega_2)| \tag{2.45}$$

$$= \frac{1}{2} \sum_{\omega_1\in\Omega_1,\omega_2\in\Omega_2} \left|\delta_1(\omega_1)\left(q_2(\omega_2)+\frac{1}{2}\delta_2(\omega_2)\right) + \delta_2(\omega_2)\left(q_1(\omega_1)+\frac{1}{2}\delta_1(\omega_1)\right)\right| \tag{2.46}$$

$$\begin{aligned} &\leq \frac{1}{2} \sum_{\omega_1\in\Omega_1,\omega_2\in\Omega_2} \left|\delta_1(\omega_1)\left(q_2(\omega_2)+\frac{1}{2}\delta_2(\omega_2)\right)\right| \\ &\quad + \frac{1}{2} \sum_{\omega_1\in\Omega_1,\omega_2\in\Omega_2} \left|\delta_2(\omega_2)\left(q_1(\omega_1)+\frac{1}{2}\delta_1(\omega_1)\right)\right| \end{aligned} \tag{2.47}$$

Let S_1 and S_2 denote the first and the second sum in (2.47), respectively. Then we have

$$S_1 = \frac{1}{2} \sum_{\omega_1\in\Omega_1,\omega_2\in\Omega_2} \left|\delta_1(\omega_1)\left(q_2(\omega_2)+\frac{1}{2}\delta_2(\omega_2)\right)\right| \tag{2.48}$$

$$= \frac{1}{2} \sum_{\omega_1\in\Omega_1} |\delta_1(\omega_1)| \sum_{\omega_2\in\Omega_2} \left|q_2(\omega_2)+\frac{1}{2}\delta_2(\omega_2)\right| \tag{2.49}$$

$$= \mathrm{V}(p_1,q_1) \sum_{\omega_2\in\Omega_2} \left(\frac{1}{2}p_2(\omega_2)+\frac{1}{2}q_2(\omega_2)\right) \tag{2.50}$$

$$= \mathrm{V}(p_1,q_1)\ . \tag{2.51}$$

By the same line of arguments we also obtain $S_2 = \mathrm{V}(p_2,q_2)$. This completes the proof of the claim for $n = 2$. The claim for $n > 2$ follows from this immediately by induction. □

Additionally, the total variation distance is a proper metric, hence it is symmetric and satisfies the triangle inequality [41]. These strong properties can be very useful in applications.

Hellinger Distance

The Hellinger distance is a well-known statistical divergence that, surprisingly, was not used by Hellinger according to [23]. The introduction of Hellinger distance and especially Hellinger affinity is mainly credited to Kakutani [53].

Definition 2.2.39 (Hellinger distance, Hellinger affinity)**.** Let p and q be probability mass functions on the finite sample space Ω. The Hellinger distance $\mathrm{h}(p,q)$ of p and q is defined by

$$\mathrm{h}^2(p,q) = 1 - \sum_{\omega\in\Omega} \sqrt{p(\omega)q(\omega)}\ .$$

Note that this equation defines the square of the Hellinger distance. The term $1-\mathrm{h}^2(p,q)$ is called the Hellinger affinity. For finite random variables X and Y we briefly write $\mathrm{h}(X,Y)$ for the Hellinger distance $\mathrm{h}(p_X,p_Y)$ of the corresponding probability mass functions.

The square of the Hellinger distance, as it is defined above, is the f-divergence for the function $f(x) = 1-\sqrt{x}$. This definition of Hellinger distance is taken from Bar-Yossef *et al.* [13], different definitions are used frequently in the literature. Sometimes Hellinger distance is

defined as the square of $\mathrm{h}(p,q)$, for example in [31], sometimes it is defined as $\sqrt{2}\,\mathrm{h}(p,q)$, e.g. in [41].

Note that the Hellinger distance $\mathrm{h}(p,q)$ is a metric whereas the square of the Hellinger distance $\mathrm{h}^2(p,q)$ is not a metric [41], but has nevertheless interesting geometric properties that were explored and used, for instance, by Bar-Yossef *et al.* [12] and Jayram [51]. Hellinger affinity has the nice property that it is separable for product distributions, that is, the Hellinger affinity of two product distributions is the product of the Hellinger affinities of the corresponding marginal distributions [41].

Inequalities Between f-Divergences

In the last sections we have seen examples of f-divergences with vastly different properties. Due to these differences, some f-divergences may be more suitable for a given application than others. In some applications even a single f-divergence may not be sufficient at all, in this case it is useful if one can pass from one f-divergence to a different one. This is usually done via inequalities between f-divergences, or more generally, inequalities between different measures of dissimilarity.

Since divergences are an important tool in statistics it comes as no surprise that many inequalities between different statistical divergences are known. We are mainly interested in the comparison of the Kullback-Leibler distance and the total variation distance of distributions, a classical result by Kullback.

Theorem 2.2.40 (Kullback [54])**.** *Let p and q be probability mass functions. Then*

$$2\,\mathrm{V}^2(p,q) \leq \mathrm{D}(p,q)\ .$$

For further inequalities between statistical divergences we refer the reader to the statistical literature. For example, a useful "map" of inequalities between divergences and an overview of the literature can be found in a survey by Gibbs and Su [41].

2.2.7 Information Statistics and the Probabilistic Method

The probabilistic method, which is mainly attributed to Paul Erdős, is a proof method that can be used to prove the existence of combinatorial objects. To this end one constructs an appropriate probability space and shows that a randomly chosen element from this space is the sought-after combinatorial object with a non-zero probability. Introductions to this subject can be found in Alon and Spencer [4] or Jukna [52]. Although information theoretical arguments have been used in the probabilistic method, overall information theory has only played a marginal role in this field.

Recently information theory has been used in communication complexity as the main tool to prove results that are essentially combinatorial, for example by Chakrabarti *et al.* [26, 25, 24], by Bar-Yossef *et al.* [12, 13], by Jayram [51], and by Gronemeier [43, 45]. The main idea of these results, that was first used by Chakrabarti, Shi, Wirth, and Yao [26], is similar to the probabilistic method: Here a lower bound on the size of a set is shown by constructing a random variable X such that the support set of X is the set of interest. Then, by Prop. 2.2.4, the size of this set is bounded from below by $2^{\mathrm{H}(X)}$. So far this approach has only been applied to communication complexity. But we believe that the general idea that is stated above and the combination of classical information theory and the more general statistical divergences

in the cited results, a technique for which Bar-Yossef *et al.* [13] coined the term *information statistics*, is a useful and new addition to the probabilistic method.

Chapter 3

Complexity

In this part of the thesis we will first give an introduction to communication complexity and information complexity. Then we will prove lower bounds on the information complexity of functions for two important models of communication, namely the index function, the AND function, and the disjointness function in the number in the hand model and a pointer jumping function for a restricted variant of the number on the forehead model.

3.1 Communication Complexity

3.1.1 Introduction

Computation can also be regarded as a communication process. In distributed computations several computers need to communicate via a network to jointly perform a task that cannot be fulfilled by a single computer, possibly due to the fact that the input of the computation is distributed among several computers or that a single computer lacks the resources to solve the problem. But also in other models of computation communication is an essential part of computation, although it is sometimes less obvious. For example, even in a single computer the CPU communicates with the memory over the system bus and then again different parts of the CPU are connected by internal buses of the CPU. Finally, the components of a computer are binary circuits. In a circuit the gates literally communicate over the wires that connect the output of a gate to the input of another gate. In 1979 Yao [76] introduced a simple model that captures the essence of all of these diverse communication processes. In the following we will describe a generalization of Yao's model. The initial two-player communication game by Yao will be treated in Section 3.1.3. A thorough introduction to communication complexity can be found in the monograph by Kushilevitz and Nisan [56].

Suppose that k players jointly compute a function $f(x_1, \ldots, x_k)$ of k inputs, but each player only knows a proper subset of the inputs. Clearly, if f depends on all inputs then the players need to communicate to fulfill this task since no player can compute the value of $f(x_1, \ldots, x_k)$ on his own. Communication is carried out via a shared blackboard that is seen by all players, each player can append binary messages to the current inscription on blackboard. We assume that all players know the function f and that the players can agree on a *communication protocol* in advance. A communication protocol governs the communication of the players, for example it determines whether the protocol continues or terminates, whose turn it is to append the next message to the blackboard in the former case, and the output of the protocol in the latter case. The protocol computes the function f if in the end all

players know $f(x_1, \ldots, x_k)$. When the protocol terminates, the value of $f(x_1, \ldots, x_k)$ should be determined by the contents of the blackboard, the so called *transcript* of the protocol. The only resource we care about in this model is communication, the cost of a protocol is the worst-case length of the transcript. We are not interested in the cost of the individual computations by the players, we even assume that the players have unlimited computational power. The communication complexity of a function f is the cost of a cheapest protocol that computes the function f. Since communication is an essential component of every computation, lower bounds on the communication complexity of a function can be used to obtain lower bounds on the resources that are needed for the computation of the function in a variety of different models of computation. Because of its versatility and the simplicity of the underlying model, nowadays communication complexity is an ubiquitous tool in complexity theory. Applications range from the time complexity of Turing machines to the space complexity of data stream algorithms. Various applications of communication complexity are described in [56].

3.1.2 Communication Protocols

Deterministic Communication Complexity

Historically, different models of communication with regard to the number of players, the distribution of the inputs among the players, and the rules of communication have been defined in the literature. Here we will first give a general formal definition that encompasses the shared properties of the most important models of communication.

Definition 3.1.1 (Deterministic k-party protocol)**.** Let $\mathcal{X}_1, \ldots, \mathcal{X}_k$, and $\mathcal{Y}$ be finite sets, let $f\colon \mathcal{X}_1 \times \cdots \times \mathcal{X}_k \longrightarrow \mathcal{Y}$ be a function, and let $x = (x_1, \ldots, x_k) \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_k$. Furthermore let $A_i \subseteq \{1, \ldots, k\}$ for $i \in \{1, \ldots, k\}$ be a family of subsets and let $A = (A_1, \ldots, A_k)$. We call A a *variable allocation*, the variables x_j subject to $j \in A_i$ are called the *variables seen by the ith player*. A deterministic k-party (or k-player) protocol P with respect to the variable allocation A is a game that is played by k players to jointly compute $f(x_1, \ldots, x_k)$. During this game the players compute the *transcript* $T(x) \in \{0,1\}^*$ and *output* $P(x) \in \mathcal{Y}$ of the protocol using a binary tree G, the so-called *protocol tree*, such that

- internal nodes of G have two children,
- each leaf node v of G is labeled by an element $t_v \in \mathcal{Y}$, and
- each internal node v of G is labeled by a number $p_v \in \{1, \ldots, k\}$ and a function

$$T_v : \prod_{i \in A_{p_v}} \mathcal{X}_i \longrightarrow \{0,1\} .$$

The transcript $T(x)$ and output $P(x)$ of the protocol are defined inductively by a path in G: The first node of the path is the root and initially the transcript $T(x)$ is the empty string. Let v be the last internal node of the path that has been defined so far and let $A_{p_v} = \{i_1, \ldots, i_\ell\}$ be the variables that are seen by player p_v. Then player p_v appends $t_v = T_v(x_{i_1}, \ldots, x_{i_\ell})$ to the transcript of the protocol. If $t_v = 0$ then the left child of v is the next node of the path, otherwise the right child of v is the next node. The output $P(x) \in \mathcal{Y}$ is the label of the leaf that is reached by this path. The protocol P computes the function f if $P(x) = f(x)$ for all x. The cost of the protocol $\mathrm{cost}(P)$ is the height of the tree G.

Note that each input x_i for $i \in \{1, \ldots, k\}$ must be seen by at least one player – otherwise the players cannot compute $f(x_1, \ldots, x_k)$ if f depends on all inputs – and that nontrivial protocols are only needed if no player sees all inputs – otherwise a single player can compute $f(x_1, \ldots, x_k)$ and write the result to the blackboard using only one bit of communication. In general, all variable allocations subject to these constraints are interesting, but only a few particular variable allocations have been studied in the literature. These variable allocations are described in Section 3.1.3, Section 3.1.4, and Section 3.1.5.

Once we have chosen a variable allocation A, the communication complexity of a function f with respect to A can be defined in the canonical way.

Definition 3.1.2 (Deterministic communication complexity)**.** Let $f\colon \mathcal{X}_1 \times \cdots \times \mathcal{X}_k \longrightarrow \mathcal{Y}$ be a function and let A be a variable allocation. Then the *deterministic communication complexity* $\mathrm{C}^A(f)$ of f with respect to A is defined by

$$\mathrm{C}^A(f) = \min\{\mathrm{cost}(P) : \text{deterministic protocol } P \text{ computes } f\} .$$

In general, the communication complexity of a function is trivially bounded from above by $\sum_{i=1}^{k} \lceil \log_2 |\mathcal{X}_i| \rceil$: The value of f can always be determined by the transcript of a trivial protocol in which each input $x_i \in \mathcal{X}_i$ is written to the blackboard by a player who sees this input.

Randomized Communication Complexity

The definition of communication protocols can be extended in several ways, for example the definition of nondeterministic protocols is straightforward (see [56] for more details on nondeterministic protocols). In the following we will define randomized protocols. Here the transcript and the output of the protocol may also depend on random inputs in addition to the inputs of the function f. The requirements on the protocol are weakened with respect to the correctness of the output. In a randomized ϵ-error protocol P for the function f we only require that the output $P(x)$ of the protocol is equal to $f(x)$ with the probability $1 - \epsilon$ over the choice of the random input. Analogously to the inputs of the function, the random inputs have to be allocated to the players. We will consider three natural choices for this allocation.

Definition 3.1.3 (Randomized k-party protocol)**.** A randomized k-party protocol P is defined similarly to a deterministic k-party protocol. The function f and the variable allocation A are defined as in Definition 3.1.1. In addition to the inputs $x = (x_1, \ldots, x_k)$ of $f\colon \mathcal{X}_1 \times \cdots \times \mathcal{X}_k \longrightarrow \mathcal{Y}$ a randomized protocol has also random inputs $r = (r_1, \ldots, r_k)$. The ith player sees the inputs x_j for $j \in A_i$ and a subset of the random inputs that depends on the type of randomization. The different types of randomization are described below. Like deterministic protocols, randomized protocols are defined by a protocol tree G, but for an inner node v of G the function t_v may also depend on the random variables seen by player p_v in addition to the variables that are specified by A_{p_v}. The protocol P computes the function f with error ϵ if $\Pr\{P(x) \neq f(x)\} \leq \epsilon$ for all $x \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_k$ where the probability is over the random choice of r. Here we consider the following types of randomization:

- In a *private coin* protocol the ith player sees r_i.
- In a *public coin* protocol the complete random input r is seen by all players.
- In a *canonical coin* protocol the ith player sees all r_j subject to $j \in A_i$.

The public coin model and the private coin model are well-known and have been used before. The canonical coin model is, to the best of our knowledge, new and will be useful in the number on the forehead model which is described later. Note that private coin protocols and canonical coin protocols can be simulated by public coin protocols without any modifications because in a public coin protocol each player sees a superset of the random variables which he would see in a private coin or canonical coin protocol.

The randomized communication complexity of a function can be defined analogously to its deterministic communication complexity. Our notation for randomized communication complexity does not include any indication of the type of randomization (private, public, or canonical coins). To simplify the notation, we will define a standard type of randomization for each variable allocation that is defined in the following sections and it will be mentioned explicitly, if we deviate from this standard.

Definition 3.1.4 (Randomized Communication complexity)**.** Let $f\colon \mathcal{X}_1 \times \cdots \times \mathcal{X}_k \longrightarrow \mathcal{Y}$ be a function and let A be a variable allocation. Then the ϵ-error *randomized communication complexity* $\mathrm{R}_\epsilon^A(f)$ of f with respect to A is defined by

$$\mathrm{R}_\epsilon^A(f) = \min\{\mathrm{cost}(P) : \text{randomized protocol } P \text{ computes } f \text{ with error } \epsilon\} .$$

Distributional Communication Complexity

In a different form of randomization the inputs of the function f are chosen at random. Here the error ϵ of a protocol for f is defined with respect to the random choice of the inputs of f. In contrast to randomized protocols, in which the error of the protocol is bounded for every input, the protocol may always compute the wrong result for some inputs. But this may happen only for an ϵ-fraction of the inputs with respect to a given probability distribution on the inputs. Hence the protocol only computes an approximation of the function f, in this case the error of approximation ϵ is usually called distributional error and the ϵ-error distributional communication complexity of a function can be defined in the obvious way.

Definition 3.1.5 (Distributional communication complexity)**.** Let $f\colon \mathcal{X}_1 \times \cdots \times \mathcal{X}_k \longrightarrow \mathcal{Y}$ be a function, let μ be a probability distribution on the finite set $\mathcal{X}_1 \times \cdots \times \mathcal{X}_k$, and let A be a variable allocation. A deterministic k-party protocol P computes the function f with distributional error ϵ if $\Pr_{X\sim\mu}\{P(X) \neq f(X)\} \leq \epsilon$. The ϵ-error *distributional communication complexity* $\mathrm{D}_{\mu,\epsilon}^A(f)$ of f with respect to the distribution μ is defined by

$$\mathrm{D}_{\mu,\epsilon}^A(f) = \min\{\mathrm{cost}(P) : \text{det. protocol } P \text{ computes } f \text{ with distributional error } \epsilon\} .$$

Randomized and distributional communication complexity are closely related. The following proposition is an application of Yao's minimax principle to randomized communication protocols. Details about Yao's minimax principle can be found in [60], for a proof of the proposition we refer the reader to [56].

Proposition 3.1.6 (Yao's minimax principle)**.** *Let $f\colon \mathcal{X}_1 \times \cdots \times \mathcal{X}_k \longrightarrow \mathcal{Y}$ be a function and let A be a variable allocation. The following holds in the public coin model of randomization:*

$$\mathrm{R}_\epsilon^A(f) = \max\{\mathrm{D}_{\mu,\epsilon}^A(f) : \mu \textit{ is a distribution on } \mathcal{X}_1 \times \cdots \times \mathcal{X}_k\} .$$

Lower bounds on $\mathrm{R}_\epsilon^A(f)$ in the public coin model can be shown by choosing an appropriate probability distribution μ on the inputs of the function and by bounding the distributional

communication complexity $\mathrm{D}^{A}_{\mu,\epsilon}(f)$ from below. By Yao's minimax principle this yields a lower bound on the randomized communication complexity of f. Since public coin protocols can simulate private coin and canonical coin protocols, lower bounds on the communication complexity in the public coin model do also apply to the private and canonical coin model. For some time Yao's minimax principle was the only known way of proving lower bounds on the randomized communication complexity of a function. Today randomized communication complexity can be also bounded by using the information complexity of a function. This will be described in section 3.2.

Rounds, One-Way Protocols, and Simultaneous Message Protocols

According to Definition 3.1.1, in a k-party protocol the label p_v of the current node v determines whose turn it is to append the next bit to the transcript of the protocol. It is possible that one player appends a sequence of several bits to the transcript. We will call a maximal consecutive subsequence of the transcript that was written by a single player a *communication round*. In a general protocol the number of alternations between the players is not restricted, the number of communication rounds is only bounded by the cost of the protocol. Sometimes we will limit the interaction of protocols by imposing constraints on the number and order of communication rounds.

Definition 3.1.7 (r-round protocols, one-way protocols)**.** In an r-round protocol P the transcript T contains at most $r-1$ alternations between the k players. A k-party one-way protocol is a k-round protocol where player i may only write to the blackboard in the ith round of the protocol (each player may also pass his round without writing to the blackboard).

Naturally, we can also define the r-round communication complexity and the one-way communication complexity of a function. Here we will only define the one-way communication complexity of a function.

Definition 3.1.8 (One-way communication complexity)**.** Let $f\colon \mathcal{X}_1 \times \cdots \times \mathcal{X}_k \longrightarrow \mathcal{Y}$ be a function and let A be a variable allocation. Then the *deterministic one-way communication complexity* $\mathrm{C}^{A,\text{one-way}}(f)$ of f with respect to A is defined by

$$\mathrm{C}^{A,\text{one-way}}(f) = \min\{\mathrm{cost}(P) : \text{deterministic one-way protocol } P \text{ computes } f\} .$$

The randomized one-way communication complexity $\mathrm{R}^{A,\text{one-way}}_{\epsilon}(f)$ and distributional one-way communication complexity $\mathrm{D}^{A,\text{one-way}}_{\mu,\epsilon}(f)$ are defined analogously.

Finally, in a simultaneous message protocol the players do not interact at all. Here each player computes a message that only depends on the inputs seen by him. The players simultaneously send their messages to a referee who does not see the inputs. The output of the protocol is computed by the referee as a function of the messages. Formally, simultaneous message protocols can be defined as a special case of one-way protocols.

Definition 3.1.9 (Simultaneous message protocol)**.** A k-party simultaneous message protocol is k-party one-way protocol such that the part of the transcript that is written by the ith player does not depend on the parts that are written by the players $1, \ldots, i-1$ for all $i \in \{1, \ldots, k\}$.

Note that, by Definition 3.1.1, the output of a protocol is uniquely determined by the transcript of the protocol, hence the referee can compute the output as a function of the transcript.

Promise Problems

Sometimes it is useful to restrict the inputs of a k-party protocol for a given function f to a subset of the domain of f, this can be seen as an extension of communication complexity to partially defined functions. Then the restricted function is called a promise problem and we only require that the output of the protocol is correct if the input is from the restricted domain. Hence, it is promised to the protocol that the input x is from the restricted domain. If this promise holds then the protocol must compute $f(x)$, but if the promise is broken then the protocol may compute an arbitrary output.

Definition 3.1.10 (Promise problem)**.** Let $f\colon \mathcal{X}_1 \times \cdots \times \mathcal{X}_k \longrightarrow \mathcal{Y}$ be a function and let $\mathcal{S} \subseteq \mathcal{X}_1 \times \cdots \times \mathcal{X}_k$. We call the restriction $f|_S$ of f to the domain S a *promise problem.* A deterministic k-party protocol P computes the function $f|_S$ if $P(x_1, \ldots, x_k) = f(x_1, \ldots, x_k)$ for all $x \in S$. For inputs $x \notin S$ the output of P may be arbitrary.

Remark 3.1.11. We will use the notation $f|_S$ only if we talk about general promise problems where the function f and the set S are not specified. For concrete promise problems we will use a more concise notation that will be defined specifically for each promise problem.

Promise problems for randomized and distributional protocols are defined analogously by changing the meaning of "the protocol P computes the function f" accordingly. Note that the altered meaning of "P computes f" in the definition of promise problems affects the communication complexity of promise problems: A protocol for the function f also computes the promise problem $f|_S$ for every subset S of the range of f. In contrast, a protocol for $f|_S$ does not necessarily compute the function f since potentially $P(x_1, \ldots, x_k) \neq f(x_1, \ldots, x_k)$ for inputs $(x_1, \ldots, x_k) \notin S$. This implies that the communication complexity of $f|_S$ is not larger than the communication complexity of f or, put the opposite way, that the communication complexity of $f|_S$ is a lower bound on the communication complexity of f for every subset S of the range of f. This can be a useful tool to prove lower bounds on the communication complexity of f. The subset S can be chosen freely such that the proof of a lower bound on the communication complexity of $f|_S$ is easy. Moreover, if communication complexity is applied to prove lower bounds on the complexity of functions in different models of computation then sometimes lower bounds on the communication complexity of promise problems $f|_S$ yield stronger lower bounds. An example of this will be given in section 4.4.3.

3.1.3 Yao's Two-Player Model

Historically, communication complexity was introduced by Yao [76] in 1979 mainly as a two-player game. For $k = 2$ our definition of deterministic k-party protocols (Def. 3.1.1) and randomized k-party private coin protocols (Def. 3.1.3) coincides with Yao's model. Note that for two players, up to symmetry, there is only one nontrivial variable allocation: One player sees the input x_1, the other player sees the input x_2. Additionally, for $k = 2$ the private coin model and the canonical coin model of randomization are identical. Like Yao, we will use private coin protocols as the standard model of randomization for $k = 2$. Because of the historical relevance of the two-player model, we will slightly deviate from the notation in the previous definitions of communication complexity and stick to the traditional notation.

Definition 3.1.12 (Two-player communication protocols and complexity)**.** Deterministic and randomized 2-party protocols with respect to the variable allocation $A = (\{1\}, \{2\})$ are called two-player protocols for short. This variable allocation is dropped in the notation for deterministic and randomized communication complexity, hence $\mathrm{C}(f)$, $\mathrm{R}_\epsilon(f)$, and $\mathrm{D}_{\mu,\epsilon}(f)$ denote the deterministic, randomized, and distributional communication complexity of a function f with respect to the variable allocation A, respectively. The corresponding one-way communication complexity is denoted by $\mathrm{C}^{\mathrm{A}\to\mathrm{B}}(f)$, $\mathrm{R}_\epsilon^{\mathrm{A}\to\mathrm{B}}(f)$, and $\mathrm{D}_{\mu,\epsilon}^{\mathrm{A}\to\mathrm{B}}(f)$, respectively. Randomized two-player communication complexity is defined with respect to private coins.

The two-player model is both a special case of the number in the hand model and the number on the forehead model which will be treated in the following sections. Therefore at this point a detailed discussion of the two-player model is not needed.

3.1.4 The NIH Multi-Party Model

The number in the hand model is a straightforward generalization of Yao's two-player model. Here the ith player sees the input x_i, figuratively each player hides his input in his hand. Early studies of this input allocation include, for example, the work of Dolev and Feder [34], who mainly studied the relation of determinism and nondeterminism in a similar model.

Definition 3.1.13 (NIH communication protocols and complexity)**.** Deterministic and randomized k-party protocols with respect to the variable allocation $\mathrm{NIH} = (\{1\}, \{2\}, \ldots, \{k\})$ are called k-party number in the hand protocols or NIH protocols for short. Randomized k-party NIH communication complexity is defined with respect to private coins.

The NIH model for $k = 2$ parties is equivalent to Yao's two-player model. Note that, like in Yao's two-player model, for this variable allocation the private coin model and the canonical coin model are identical. This is not the only similarity of Yao's 2-player model and the k-party NIH model. The basic combinatorial properties of the NIH model are very similar to the properties of two-player protocols. Most proof methods for lower bounds on the communication complexity of functions in Yao's model can be adapted easily to the k-party NIH model. The basic combinatorial properties which underlie this similarity are described in the following section.

The Combinatorial Structure of NIH Protocols

Clearly, the set of the inputs of a deterministic NIH protocol is partitioned by the transcripts of the protocol: For each fixed transcript there is a subset of the inputs that generate this transcript. The fundamental combinatorial properties of these subsets are the foundation of all proof methods for lower bounds on the NIH communication complexity of functions.

Definition 3.1.14 (k-box, combinatorial rectangle)**.** Let $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_k$ be a set. A subset $\mathcal{S} \subseteq \mathcal{X}$ is called *a k-box* if there are subsets $\mathcal{S}_i \subseteq \mathcal{X}_i$ for $i \in \{1, \ldots, n\}$ such that

$$\mathcal{S} = \mathcal{S}_1 \times \cdots \times \mathcal{S}_k \,.$$

If $k = 2$ then k-boxes are also called *combinatorial rectangles* or *rectangles* for short.

The simple combinatorial structure of the subset of inputs that corresponds to a given transcript is due to the fact that the ith player does only see the ith coordinate of the input. The implications of this restriction are explored in the following proposition.

Proposition 3.1.15. *Suppose that P is a deterministic k-party NIH protocol with inputs from the set $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_k$ and that $G = (V, E)$ is the corresponding protocol tree according to Definition 3.1.1. Each input $x = (x_1, \ldots, x_k) \in \mathcal{X}$ defines a path from the root of G to a leaf of G. For every $v \in V$ let $\mathcal{X}(v)$ denote the set of all inputs x such that v lies on the path that is defined by x. Then $\mathcal{X}(v)$ is a k-box, hence for every $v \in V$*

$$\mathcal{X}(v) = \mathcal{X}_1(v) \times \cdots \times \mathcal{X}_k(v)$$

where $\mathcal{X}_i(v) \subseteq \mathcal{X}_i$ for all $i \in \{1, \ldots, k\}$.

Proof. We prove the proposition by induction on the depth of the node v. The claim of the proposition is obviously true for the root v of the tree G, in this case $\mathcal{X}(v) = \mathcal{X}_1 \times \cdots \times \mathcal{X}_k$. We will now prove the following: If the claim is true for an inner node v, then it also holds for the left child w_0 and right child w_1 of v. By the induction hypothesis, we have $\mathcal{X}(v) = \mathcal{X}_1(v) \times \cdots \times \mathcal{X}_k(v)$. We assume w.l.o.g. that $p_v = 1$, the other cases can be shown analogously. Since $p_v = 1$, the function T_v only depends on the input $x_1 \in \mathcal{X}_1$. Let $\mathcal{X}_1(v, t) = \{x_1 \in \mathcal{X}_1(v) : t_v(x_1) = t\}$. Then, by the definition of NIH protocols, the child node w_t is reached by the inputs in $\mathcal{X}(w_t) = \mathcal{X}_1(v, t) \times \mathcal{X}_2(v) \times \cdots \times \mathcal{X}_k(v)$ and the claim also holds for w_0 and w_1. □

The leafs of the tree G for a given protocol correspond to the transcripts of the protocol. If we apply the previous proposition to the leafs of G then we obtain the following corollary.

Corollary 3.1.16. *Let P be a deterministic k-party NIH protocol with inputs from the set $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_k$ and let $\mathcal{T}$ denote the set of all possible transcripts of P. Then P partitions $\mathcal{X}$ into $|\mathcal{T}|$ subsets*

$$\mathcal{X}(t) = \mathcal{X}_1(t) \times \cdots \times \mathcal{X}_k(t)$$

where $t \in \mathcal{T}$ and $\mathcal{X}_i(t) \subseteq \mathcal{X}_i$ for all $i \in \{1, \ldots, k\}$ such that $T(x) = t$ for all $x \in \mathcal{X}(t)$.

Remark 3.1.17. Note that, with respect to the transcript of the protocol, a randomized private coin protocol with the input $x = (x_1, \ldots, x_k)$ and the random input $r = (r_1, \ldots, r_k)$ can also be seen as a deterministic protocol where the ith player sees the input (x_i, r_i). Therefore Proposition 3.1.15 and Corollary 3.1.16 are also applicable to randomized protocols.

3.1.5 The NOF Multi-Party Model

The number in the hand model is probably the most natural generalization of Yao's two-player model to k players. Here the power of k-party protocols, in a sense, decreases as the number of players k increases. The more players are involved, the more communication is needed. Chandra, Furst, and Lipton [27] took a different route and defined a model where the power of k-party protocols increases as the number of players grows, the so called *number on the forehead model*. Here the ith player sees all inputs except the ith input, figuratively the input is written on the players foreheads.

Definition 3.1.18 (NOF communication protocols and complexity)**.** Deterministic and randomized k-party protocols with respect to the variable allocation NOF $= (A_1, \ldots, A_k)$ such that $A_i = \{1, \ldots, k\} - \{i\}$ for all $i \in \{1, \ldots, k\}$ are called k-party number on the forehead protocols or NOF protocols for short. Randomized k-party NOF communication complexity is defined with respect to canonical coins.

Note that the two-party NOF model is essentially equivalent to the two-party NIH model since each player exclusively sees one of the two coordinates of the input.

The Combinatorial Structure of NOF Protocols

As it is the case for NIH protocols, the set of the inputs of a deterministic k-party NOF protocol is also partitioned into subsets by the transcripts of the protocol. But for $k > 2$ the combinatorial structure of the subset that corresponds to a fixed transcript is substantially more complicated than the simple k-box that corresponds to a transcript of a NIH protocol.

Definition 3.1.19 (Cylinder, cylinder intersection)**.** Let $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_k$ be a set. A subset $\mathcal{S} \subseteq \mathcal{X}$ is called *a cylinder in the ith dimension* if for all $(x_1, \ldots, x_k) \in \mathcal{X}$ and all $x_i' \in \mathcal{X}_i$

$$(x_1, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_k) \in \mathcal{S} \Leftrightarrow (x_1, \ldots, x_{i-1}, x_i', x_{i+1}, \ldots, x_k) \in \mathcal{S} \, .$$

A subset $\mathcal{S} \subseteq \mathcal{X}$ is called *a cylinder intersection* if there are sets $\mathcal{S}_i$ for $i \in \{1, \ldots, k\}$ such that $\mathcal{S}_i$ is a cylinder in the ith dimension and $\mathcal{S} = \bigcap_{i=1}^{k} \mathcal{S}_i$.

Given the definition of cylinder intersections, it is surprisingly easy to show that the inputs of a NOF protocol are partitioned into cylinder intersections by the transcripts of a protocol.

Proposition 3.1.20. *Suppose that P is a deterministic k-party NOF protocol with inputs from the set $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_k$ and that $G = (V, E)$ is the corresponding protocol tree according to Definition 3.1.1. Each input $x = (x_1, \ldots, x_k) \in \mathcal{X}$ defines a path from the root of G to a leaf of G. For every $v \in V$ let $\mathcal{X}(v)$ denote the set of all inputs x such that v lies on the path that is defined by x. Then $\mathcal{X}(v)$ is a cylinder intersection for every $v \in V$.*

Proof. The proof follows the same line of arguments as the proof of Proposition 3.1.15. Here we are using the fact that $\mathcal{X}$ is a cylinder intersection for the base case of the induction. For the inductive step we use that for a node $w \in V$ and its parent node $v \in V$ in the protocol tree G we have that $\mathcal{X}(w)$ is the intersection of $\mathcal{X}(v)$ and a cylinder in the p_wth dimension, and that the intersection of a cylinder intersection and a cylinder in any dimension is once again a cylinder intersection. □

The following corollary follows immediately from the last proposition.

Corollary 3.1.21. *Let P be a deterministic k-party NOF protocol with inputs from the set $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_k$ and let $\mathcal{T}$ denote the set of all possible transcripts of P. Then $\mathcal{X}$ is partitioned into $|\mathcal{T}|$ cylinder intersections by P.*

Remark 3.1.22. Similarly to NIH protocols, the last results can be generalized to randomized NOF protocols with canonical coins. This is due to the fact that the random inputs of a canonical coin protocol can be regarded as ordinary inputs of a deterministic protocol.

3.2 Information Complexity

In the preface of their seminal monograph [56] on communication complexity Kushilevitz and Nisan contrast Shannon's classical information theory with communication complexity: The premise of information theory is that a certain predetermined communication needs to be carried out. Information theory deals with the details of the *transmission of information*,

for example the efficient encoding of the information and reliable communication over faulty communication channels. On the other hand, in communication complexity the transmission of information is only a means to solve a problem, namely to compute a function of some arguments that are distributed among several parties. Then communication complexity is about the *contents of the communication* that is needed in order to solve the problem. Up to the present, the mathematical tools that were mainly used in information theory and communication complexity were as different as the roles of communication in both disciplines: While information theory is based on the work of Shannon that is described in Section 2.2, communication complexity was mainly based on pure combinatorics. This changed recently with a new concept, the so-called *information cost* of a protocol that was introduced by Chakrabarti, Shi, Wirth, and Yao [26], although similar ideas were used implicitly in earlier publications by Ablayev [1]. The information cost of a protocol is the mutual information of the input and the transcript of the protocol. If this mutual information is large, then the support set of the transcripts must also be large and a lower bound on the worst case length of the transcript can be obtained. Once the information cost of a protocol is defined, the information complexity of a function can be defined in the canonical way and it can be used to obtain lower bounds on the communication complexity of a function. The concept of information cost was further refined to the *conditional information cost* of a protocol by Bar-Yossef, Jayram, Kumar, and Sivakumar [13]. Here the conditional mutual information of the input and the transcript of a protocol is used. The main use of the conditioning variable is the decomposition of non-product distributions into a mixture of product distributions. This approach will be exemplified in Section 3.3.3.

3.2.1 Information Complexity in the NIH model

First, we will define the information cost of protocols in the number in the hand model. The information cost of a protocol and the information complexity of a function depend on a distribution of the inputs. Our notation for information cost and information complexity tries to mimic the usual notation from information theory. The inputs of a protocol or function will be specified as random variables with distributions that have to be fixed in advance. The following definition applies to deterministic and randomized NIH protocols.

Definition 3.2.1 (NIH information cost)**.** Let P be a k-party NIH protocol with inputs from the set $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_k$ and let $T(x)$ denote the transcript of P for an input $x \in \mathcal{X}$. Suppose that X and D are random variables such that $X \in \mathcal{X}$. Then the information cost $\mathrm{icost}(P; X)$ of P with respect to X is defined by

$$\mathrm{icost}(P; X) = \mathrm{I}(T(X) : X) \; .$$

The conditional information cost $\mathrm{icost}(P; X|D)$ of P with respect to X given D is defined by

$$\mathrm{icost}(P; X|D) = \mathrm{I}(T(X) : X|D) \; .$$

Note that the information cost is a special case of the conditional information cost. If we condition on a variable D that is independent of the computation of P then we have $\mathrm{icost}(P; X) = \mathrm{icost}(P; X|D)$. Therefore we will not distinguish between information cost and conditional information cost in the following.

The information cost of randomized protocols depends on the joint distribution of the input $X = (X_1, \ldots, X_k)$ and the random input $R = (R_1, \ldots, R_k)$. The transcript $T(X)$ is

a function of X and R. Unless otherwise noted, in the following we will assume that R is independent of X and that the random variables R_i for $i \in \{1, \ldots, k\}$ are independent whenever we consider the information cost of a NIH protocol.

Once the information cost of a given NIH protocol is defined, the NIH information complexity of a function can be defined in the canonical way. The information complexity of a function depends on the type of protocols that we consider (deterministic, distributional, or randomized) and the joint distribution of the inputs and the conditioning variable.

Definition 3.2.2 (Conditional NIH information complexity)**.** Let $f\colon \mathcal{X}_1 \times \cdots \times \mathcal{X}_k \longrightarrow \mathcal{Y}$ be a function and let $0 \leq \epsilon \leq 1$. Suppose that $X \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_k$ and D are random variables. Then the ϵ-error distributional conditional information complexity $\mathrm{DIC}^{\mathrm{NIH}}_{\epsilon}(f; X|D)$ of f with respect to X given D in the NIH model is defined by

$$\mathrm{DIC}^{\mathrm{NIH}}_{\epsilon}(f; X|D) = \min\{\mathrm{icost}(P; X|D) : P \text{ computes } f \text{ with distributional error } \epsilon\}$$

where the protocols P in the minimum are deterministic NIH protocols. The deterministic conditional information complexity of a function can be defined as a special case of the distributional conditional information complexity for the error $\epsilon = 0$. In this case we drop the error ϵ from the notation and simply write $\mathrm{DIC}^{\mathrm{NIH}}(f; X|D)$. The ϵ-error randomized conditional information complexity $\mathrm{IC}^{\mathrm{NIH}}_{\epsilon}(f; X|D)$ of f with respect to X given D in the NIH model is defined by

$$\mathrm{IC}^{\mathrm{NIH}}_{\epsilon}(f; X|D) = \min\{\mathrm{icost}(P; X|D) : \text{randomized protocol } P \text{ computes } f \text{ with error } \epsilon\}\ .$$

Sometimes we omit the variable allocation NIH from the notation if it is evident from the context.

We also consider the information complexity of functions with respect to one-way protocols.

Definition 3.2.3 (One-way information complexity)**.** For the information complexity with respect to one-way protocols the same notation as for the communication complexity is used, for example $\mathrm{IC}^{\mathrm{NIH,one\text{-}way}}_{\epsilon}(f; X|D)$ and $\mathrm{IC}^{\mathrm{A}\to\mathrm{B}}_{\epsilon}(f; X|D)$ for two-players.

Remark 3.2.4. The information complexity of a function (without a condition) could be defined analogously to Definition 3.2.2 if we use the information cost instead of the conditional information cost. But the information complexity of f is only a special case of the conditional information complexity of f for the case that the conditioning variable D is independent of X for deterministic protocols and independent of (X, R) for randomized protocols with the additional random input R. Hence we abstain from a separate definition and simply omit the condition from the notation if it is not needed. In this case we briefly write $\mathrm{DIC}^{\mathrm{NIH}}_{\epsilon}(f; X)$ and $\mathrm{IC}^{\mathrm{NIH}}_{\epsilon}(f; X)$ mimicking the notation for entropy and mutual information.

The following theorem shows that the information complexity is a lower bound on the communication complexity of a function, irrespective of the joint distribution of X and D. The joint distribution of the inputs and the conditioning variable can be chosen freely to facilitate simple proofs of strong lower bounds.

Theorem 3.2.5. *Let $f\colon \mathcal{X}_1 \times \cdots \times \mathcal{X}_k \longrightarrow \mathcal{Y}$ be a function and let $0 \leq \epsilon \leq 1$. Suppose that $X \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_k$ and D are random variables such that $X \sim \mu$. Then*

(i) $\mathrm{C}^{\mathrm{NIH}}(f) \geq \mathrm{DIC}^{\mathrm{NIH}}(f; X|D)$,

(ii) $\mathrm{D}^{\mathrm{NIH}}_{\mu,\epsilon}(f) \geq \mathrm{DIC}^{\mathrm{NIH}}_{\epsilon}(f; X|D)$, *and*

(iii) $\mathrm{R}^{\mathrm{NIH}}_{\epsilon}(f) \geq \mathrm{IC}^{\mathrm{NIH}}_{\epsilon}(f; X|D)$.

Proof. We will only prove claim (iii) of the theorem. The proofs of the other claims follow the same line of arguments. Let P be an optimal ϵ-error randomized NIH protocol for f with respect to the communication cost of the protocol, hence $\mathrm{cost}(P) = \mathrm{R}^{\mathrm{NIH}}_{\epsilon}(f)$, and let $T(X)$ denote the transcript of P for the input X. Clearly, we have $\log(|\operatorname{supp}(T(X))|) \leq \mathrm{cost}(P)$ since at least $\log(|\operatorname{supp}(T(X))|)$ bits are needed to encode $|\operatorname{supp}(T(X))|$ different transcripts. On the other hand, we have

$$\begin{aligned}
\mathrm{IC}^{\mathrm{NIH}}_{\epsilon}(f; X|D) &\leq \mathrm{icost}(P; X|D) && (3.1)\\
&= \mathrm{I}(T(X) : X|D) && (3.2)\\
&= \mathrm{H}(T(X)|D) - \mathrm{H}(T(X)|X, D) && (3.3)\\
&\leq \mathrm{H}(T(X)|D) && (3.4)\\
&\leq \mathrm{H}(T(X)) && (3.5)\\
&\leq \log(|\operatorname{supp}(T(X))|)\ . && (3.6)
\end{aligned}$$

Here we used the definition of information complexity and information cost, then the non-negativity of entropy and the fact that conditioning reduces entropy (Prop. 2.2.4), and finally the upper bound on the entropy (Prop. 2.2.4). By combining this inequality with our first observation, we obtain claim (iii) of the theorem. □

Note that, analogously to Theorem 3.2.5, similar result can be shown for the information complexity and communication complexity with respect to one-way protocols. We omit a separate treatment of this case to avoid a tedious repetition of Theorem 3.2.5.

Remark 3.2.6. The information complexity of a function can also be used to obtain lower bounds on the communication complexity of promise problems $f|_S$. Analogously to Section 3.1.2, we just need to redefine the meaning of "the protocol P computes the function $f|_S$" in Definition 3.2.2 such that it is only required that the output of P agrees with f on the subset S of the inputs. In this case it is required that the input variable X satisfies $\operatorname{supp}(X) \subseteq S$ since $f|_S(x)$ is not well-defined for inputs $x \notin S$. Also note that proofs of lower bounds on the information complexity of a function f usually use properties of the function f. For promise problems $f|_S$ we have to ensure that we only use properties of f that remain valid for the promise problem since the output of a protocol for $f|_S(x)$ may differ from $f(x)$ for inputs $x \notin S$.

3.2.2 Information Complexity in the NOF model

In the number on the forehead model several sensible definitions of information cost and information complexity are conceivable. Some of the problems that have to be considered for a meaningful definition of information cost in the NOF model will be discussed in Section 3.4.1. Here we will only define the information cost of deterministic one-way protocols. We will use the following notation for the inputs and the transcript of a one-way protocol.

Definition 3.2.7 (Notation)**.** Let $X = (X_1, \ldots, X_k)$ be a random input for a deterministic k-party one-way NOF protocol P and let $T(X) = (T_1, \ldots, T_k)$ be the transcript of P for the input X where T_i is the part of the transcript that was written by the ith player. Then

- let $X_{-i} = (X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_k)$ denote the vector of all X_ℓ where $\ell \neq i$ and
- let $T_{1,i} = (T_1, \ldots, T_i)$ denote the first i messages.

The information cost of one-way protocols in the NOF model is defined as follows.

Definition 3.2.8 (NOF information cost)**.** Let P be a deterministic k-party one-way NOF protocol, let X be a random input for P, and let $T(X) = (T_1, \ldots, T_k)$ be the transcript of P for the input X where T_i denotes the part of the transcript that is written by the ith player for $i \in \{1, \ldots, k\}$. Then the information cost $\mathrm{icost}(P; X)$ of P with respect to X is

$$\mathrm{icost}(P; X) = \max\{\mathrm{I}(T_i : X_{i+1} | X_{-(i+1)}, T_{1,i-1}) \,:\, 1 \leq i < k\} \,.$$

Given the definition of the NOF information cost for one-way protocols, the information complexity of a function in the one-way NOF model can be defined analogously to Definition 3.2.2. The following definition summarizes our notation for the one-way information complexity of a function in the NOF model avoiding a detailed repetition of 3.2.2.

Definition 3.2.9 (NOF one-way information complexity)**.** Let $f \colon \mathcal{X}_1 \times \cdots \times \mathcal{X}_k \longrightarrow \mathcal{Y}$ be a function, let $0 \leq \epsilon \leq 1$, and let $X \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_k$ be a random variable. Then the ϵ-error distributional one-way information complexity of f with respect to X in the NOF model is denoted by $\mathrm{DIC}_\epsilon^{\mathrm{NOF,one\text{-}way}}(f; X)$.

Analogously to the NIH model, the one-way information complexity of a function in the NOF model is a lower bound on its one-way communication complexity for every distribution on the inputs of the protocol.

Theorem 3.2.10. *Let $f \colon \mathcal{X}_1 \times \cdots \times \mathcal{X}_k \longrightarrow \mathcal{Y}$ be a function, let $0 \leq \epsilon \leq 1$, and let $X \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_k$ be a random variable such that $X \sim \mu$. Then*

$$\mathrm{D}_{\mu,\epsilon}^{\mathrm{NOF,one\text{-}way}}(f) \geq \mathrm{DIC}_\epsilon^{\mathrm{NOF,one\text{-}way}}(f; X) \,.$$

Proof. The length of the longest message that is written to the blackboard by any player of a one-way protocol is usually called the *maximum communication* of the protocol. The length of the complete transcript is called the *total communication* of a one-way protocol. By using the same arguments as in the proof of Theorem 3.2.5 it is easy to see that the information cost of a one-way NOF protocol is a lower bound on its maximum communication. Since the maximum communication of a one-way protocol is a lower bound on its total communication, we obtain the claim of the theorem. □

3.3 The NIH Information Complexity of Selected Problems

In this section we will prove lower bounds on the information complexity of some problems in the NIH model. First, as a simple introduction, we will consider the two-player one-way information complexity of the index function, a simple model of memory access that has many applications to OBDD complexity. In the main part of this section, we will describe the

properties of NIH protocols that are the basis for the proof of lower bounds on the information complexity of functions in this model. Then a lower bound on the information cost of protocols for the conjunction of k bits is shown if each player of a k-party NIH protocol sees exactly one of the input bits. In the final part of this section, we use this result to prove an optimal lower bound on the k-party NIH information complexity of the disjointness function.

3.3.1 A Warm-Up: One-Way Protocols for the Index Function

As a first introduction to information complexity we will use information theory to prove a lower bound on the distributional communication complexity of the so called *index function* in the two-player one-way model of communication.

Definition 3.3.1 (Index function)**.** The index function IND_n for inputs of size n such that $x = (x_1, \ldots, x_n) \in \{0,1\}^n$ and $y \in \{1, \ldots, n\}$ is defined by

$$\mathrm{IND}_n(x, y) = x_y \ .$$

Sometimes this function is also called *multiplexer function* MUX_n or *direct storage access function* DSA_n. Clearly, the computation of this function in the two-player communication model is easy if the second player who holds the input y is allowed to communicate before the first player communicates. In this case the second player can just announce y using $\lceil \log n \rceil$ bits and then the first player can output x_y using one additional bit resulting in an overall communication of $\lceil \log n \rceil + 1$ bits. The situation is different if the first player who holds x has to send the first message and the second player who holds y has to compute the output of the protocol by using only the message and y, as it is the case in the two-player one-way model.

Theorem 3.3.2. *Let X_i for $i \in \{1, \ldots, n\}$ be chosen independently at random from $\{0,1\}$ subject to $\Pr\{X_i = 1\} = q$, let $X = (X_1, \ldots, X_n)$, and let $Y \in \{1, \ldots, n\}$ be a random variable that is independent of X.*

- *If Y is distributed uniformly in $\{1, \ldots, n\}$ then*

$$\mathrm{DIC}_\epsilon^{\mathrm{A}\to\mathrm{B}}(\mathrm{IND}_n; (X, Y)) = \Omega\left(n(h_2(q) - h_2(\epsilon))\right) \ .$$

- *If $Y - 1$ is distributed binomially with the parameters $n - 1$ and $1/2$ then*

$$\mathrm{DIC}_\epsilon^{\mathrm{A}\to\mathrm{B}}(\mathrm{IND}_n; (X, Y)) = \Omega\left(\sqrt{n}(h_2(q) - h_2(\epsilon))\right) \ .$$

By Theorem 3.2.5, these lower bounds on the distributional information complexity of IND_n immediately imply lower bounds on the distributional communication complexity of IND_n and, by Yao's minimax principle (Prop. 3.1.6), we also obtain lower bounds on the randomized communication complexity of IND_n.

The index function may look like a toy problem that is complicated artificially by the severe limitations of the one-way model. But, in fact, the communication complexity of IND_n is an essential tool in the proofs of many lower bounds on the size of Boolean branching programs, especially OBDDs. Numerous applications of the communication complexity of IND_n can be found in the monograph on branching programs by Wegener [75], one particular application is discussed briefly in Section 3.3.1.

Proof of the Lower Bound

The proof of Theorem 3.3.2 extends the arguments in a similar result by Bar-Yossef, Jayram, Kumar, and Sivakumar [12].

Lemma 3.3.3. *Let X_i for $i \in \{1,\dots,n\}$ be chosen independently at random from $\{0,1\}$ subject to $\Pr\{X_i{=}1\} = q$ and let $X = (X_1,\dots,X_n)$. Furthermore let $Y \in \{1,\dots,n\}$ be a random variable that is independent of X and has the probability mass function $p(y)$ and let $p_{\max} = \max\{p(y) : y \in \{1,\dots,n\}\}$. If $T(X,Y)$ is the transcript of a deterministic two-player one-way protocol P that computes $\mathrm{IND}_n(X,Y)$ with distributional error ϵ, then*

$$\mathrm{I}(T(X,Y) : X|Y) \geq \frac{h_2(q) - h_2(\epsilon)}{p_{\max}} \,.$$

Proof. Let $T(X)$ and $T(Y)$ denote the part of the transcript that is written by the first and the second player, respectively. The main property of one-way protocols that is used in the proof is the fact that the random variables $(X,T(X))$ and Y are independent. This is easily verified: We assumed that the inputs X and Y of IND_n are independent. Since $T(X)$ is a function of X, the only input that is known by the first player, $T(X)$ is also independent of Y. Hence the random variables $(X,T(X))$ and Y are independent. By using this, we obtain

$$\begin{aligned}
\mathrm{I}(T(X) : \mathrm{IND}_n(X,Y)|Y) &= \sum_y \Pr\{Y=y\}\,\mathrm{I}(T(X) : \mathrm{IND}_n(X,Y)|Y=y) && (3.7)\\
&= \sum_y \Pr\{Y=y\}\,\mathrm{I}(T(X) : X_y|Y=y) && (3.8)\\
&= \sum_y \Pr\{Y=y\}\,\mathrm{I}(T(X) : X_y|Y) && (3.9)\\
&\leq p_{\max} \sum_y \mathrm{I}(T(X) : X_y|Y) && (3.10)\\
&\leq p_{\max}\,\mathrm{I}(T(X) : X|Y)\,. && (3.11)
\end{aligned}$$

In the third line we used the fact that the joint distribution of X_y and $T(X)$ is the same irrespective of the value of Y. This fact follows immediately from the independence of $(X,T(X))$ and Y. In the last line we used the superadditivity of conditional mutual information for conditionally independent random variables.

On the other hand, by the independence of the variables X_i and Y for all i,

$$\begin{aligned}
\mathrm{H}(\mathrm{IND}_n(X,Y)|Y) &= \sum_y \Pr\{Y=y\}\,\mathrm{H}(X_y|Y=y) && (3.12)\\
&= \sum_y \Pr\{Y=y\}\,\mathrm{H}(X_y) && (3.13)\\
&= \sum_y \Pr\{Y=y\}\,h_2(q) && (3.14)\\
&= h_2(q)\,. && (3.15)
\end{aligned}$$

The second player of P computes the output $P(X,Y)$ of the protocol as a function of his input Y and the message $T(X)$ of the first player. Since P is an ϵ-error protocol, the output

is a correct prediction of $\mathrm{IND}_n(X,Y)$ with the probability ϵ with respect to the random choice of X and Y. Then, by Fano's inequality (Thm. 2.2.29), we obtain

$$\mathrm{H}(\mathrm{IND}_n(X,Y)|Y,T(X)) \leq h_2(\epsilon) \tag{3.16}$$

and the definition of conditional mutual information yields

$$\begin{aligned} \mathrm{I}(T(X) : \mathrm{IND}_n(X,Y)|Y) &= \mathrm{H}(\mathrm{IND}_n(X,Y)|Y) - \mathrm{H}(\mathrm{IND}_n(X,Y)|T(X),Y) && (3.17)\\ &\geq h_2(q) - h_2(\epsilon)\ . && (3.18) \end{aligned}$$

Finally, by combining the upper and lower bound on $\mathrm{I}(T(X) : \mathrm{IND}_n(X,Y)|Y)$, we obtain

$$p_{\max}\, \mathrm{I}(T(X) : X|Y) \geq h_2(q) - h_2(\epsilon) \tag{3.19}$$

and the claimed results follows from the fact that mutual information is increased by additional variables (Cor. 2.2.28):

$$\mathrm{I}(T(X,Y) : X|Y) = \mathrm{I}(T(X), T(Y) : X|Y) \geq \mathrm{I}(T(X) : X|Y)\ . \tag{3.20}$$

□

Theorem 3.3.2 follows from Lemma 3.3.3 since $\mathrm{icost}(P; X, Y) \geq \mathrm{I}(T(X,Y) : X|Y)$. If $p(y)$ is the uniform distribution then $p_{\max} = 1/n$. If $p(y)$ is the binomial distribution on $\{1, \ldots, n\}$ for the parameter $1/2$ then $p_{\max} = p(n/2)$. Asymptotic approximations of this probability are well known [67]. For example, if $n = 2N$ then

$$p(n/2) = p(N) = 2^{-2N}\binom{2N}{N} = (\pi N)^{-\frac{1}{2}}\left(1 + O\left(\frac{1}{N}\right)\right)\ .$$

Lemma 3.3.3 does not yield large lower bounds if $p_{\max}$ is relatively large. For example, if $p_{\max}$ is a constant that does not decrease in n then the lower bound that is obtained by Lemma 3.3.3 is only a constant. But even in this case the information complexity of IND_n can grow linearly in n if ϵ is sufficiently small and Y is distributed appropriately. If this happens, it can be useful to exclude values of Y that have large probabilities explicitly by applying the lemma to a conditional distribution of Y: Let $\mathcal{Y}_\alpha = \{y \in \{1, \ldots, n\} : p(y) \leq \alpha\}$. Then Lemma 3.3.3 can be applied to the conditional distribution of Y given that $Y \in \mathcal{Y}_\alpha$ under the assumption that the error of the protocol is 0 given that $Y \notin \mathcal{Y}_\alpha$. This approach can succeed if $\alpha/p(\mathcal{Y}_\alpha)$ is sufficiently small.

Applications of the Lower Bound to OBDDs

Bryant [20, 21] introduced the use of ordered binary decision diagrams or OBDDs for short, a special class of binary branching programs, as a data structure for the representation of Boolean functions. He has shown that ordered binary decision diagrams can be minimized efficiently and that minimized OBDDs are a canonical form for the representation of Boolean functions. Additionally, he devised efficient algorithms for the most important operations on Boolean functions that are represented by OBDDs. OBDDs have found numerous practical applications in the design, synthesis, and verification of integrated circuits. Bryant's hidden weighted bit function [22], or HWB function for short, is a benchmark function that is often used to examine the computational power of restricted branching programs

like OBDDs (see [75]). For the input $x = (x_1, \ldots, x_n) \in \{0,1\}^n$ let $|x|$ denote the hamming weight of x. Then the hidden weighted bit function is defined by $\mathrm{HWB}_n(x) = x_{|x|}$ with $x_0 = 0$. Note that the hidden weighted bit function is very similar to the index function (Def. 3.3.1), except for the fact that the index y is the hamming weight of the input vector x instead of a separate parameter. If the input bits x_i are chosen independently such that $\Pr\{x_i{=}1\} = 1/2$ then the index $y = |x|$ is distributed binomially with the parameters n and $1/2$. Bollig *et al.* [18] used the similarity of HWB_n and IND_n to prove a lower bound on the size of OBDDs that approximate the HWB_n function with a constant error. Their proof is based on the distributional communication complexity of the index function for a binomially distributed index y. Gronemeier [44] improved the lower bound and proved a matching upper bound. In the proof of the lower bound the distributional communication complexity of IND_n is analyzed by combinatorial means, the improvement in [44] is partially due to an improved combinatorial analysis of the index function. The lower bound for the binomial distribution in Theorem 3.3.2 matches the combinatorial lower bound by Gronemeier, but compared to the lengthy and opaque combinatorial proofs in [18] and [44] our information theoretical proof is simpler and more intuitive. Moreover, Lemma 3.3.3 is more general than the result in [44] since it can be easily applied to different distributions of the index.

3.3.2 The Statistical Structure of NIH Protocols

In the proof of the lower bound for two-party one-way protocols in the previous section we used the fact that the first player's part of the transcript is independent of the second player's input in one-way protocols. Clearly, this property does not hold for unrestricted two-player protocols. Here, in general, the communication of the first player depends on his input and the previous messages which may depend on the second players input. Hence, for unrestricted two-player protocols, and more generally for unrestricted k-party NIH protocols, we can at most hope for a weaker property that can be used to prove lower bounds on the information complexity of functions: It turns out that the inputs of the players in NIH protocols are conditionally independent given the transcript of the protocol. This property is stated in the following proposition, which is essentially a restatement of Corollary 3.1.16 in statistical terms.

Proposition 3.3.4. *Let $X = (X_1, \ldots, X_k)$ be a random variable such that the random variables X_i for $i \in \{1, \ldots, k\}$ are independent. If $T(X)$ is the transcript of a deterministic k-party NIH protocol P for the random input X, then the random variables X_i for $i \in \{1, \ldots, k\}$ are also conditionally independent given $T(X)$, thus for all $x = (x_1, \ldots, x_k) \in \mathrm{range}(X)$ and all $t \in \mathrm{range}(T(X))$ the following equalities hold:*

(i) $\Pr\{X{=}x | T(X){=}t\} = \prod_{i=1}^{k} \Pr\{X_i{=}x_i | T(X){=}t\}$

(ii) $\Pr\{X_i{=}x_i | X_{-i}{=}x_{-i}, T(X){=}t\} = \Pr\{X_i{=}x_i | T(X){=}t\}$

The same equalities also hold for randomized protocols P.

Proof. The condition $T(X) = t$ is equivalent to the condition that the input X is contained

in a k-box $\mathcal{X}(t) = \mathcal{X}_1(t) \times \cdots \times \mathcal{X}_k(t)$ by Corollary 3.1.16. Therefore we have

$$\Pr\{X{=}x|T(X){=}t\} = \frac{\Pr\{X_1{=}x_1, \ldots, X_k{=}x_k, T(X){=}t\}}{\Pr\{T(X){=}t\}} \tag{3.21}$$

$$= \frac{\Pr\{X_1{=}x_1, \ldots, X_k{=}x_k, X_1 \in \mathcal{X}_1(t), \ldots, X_k \in \mathcal{X}_k(t)\}}{\Pr\{X_1 \in \mathcal{X}_1(t), \ldots, X_k \in \mathcal{X}_k(t)\}} \tag{3.22}$$

$$= \prod_{i=1}^{k} \frac{\Pr\{X_i{=}x_i, X_i \in \mathcal{X}_i(t)\}}{\Pr\{X_i \in \mathcal{X}_i(t)\}} \tag{3.23}$$

$$= \prod_{i=1}^{k} \Pr\{X_i{=}x_i|X_i \in \mathcal{X}_i(t)\} \tag{3.24}$$

$$= \prod_{i=1}^{k} \Pr\{X_i{=}x_i|X \in \mathcal{X}(t)\} \tag{3.25}$$

$$= \prod_{i=1}^{k} \Pr\{X_i{=}x_i|T(X){=}t\} \ . \tag{3.26}$$

In the third line and in the second to last line we used the independence of the random variables X_i. Notice that the same argument actually applies to arbitrary subsets of the variables $X_1, \ldots, X_k$, hence the variables are conditionally independent given that $T(X) = t$. Then the second claim follows immediately from the first claim by Proposition A.1.3.

By Remark 3.1.17, we can see randomized NIH protocols as deterministic protocols where the random input of the protocol is considered as an ordinary input of the protocol. By this reasoning, for randomized protocols P with the random input $r = (r_1, \ldots, r_k)$ we could strengthen our claims and replace the events $X_i = x_i$ by the events $(X_i, R_i) = (x_i, r_i)$ for $i \in \{1, \ldots, k\}$, respectively. Then the actual claims of the lemma for randomized protocols follow from the strengthened claims if we sum the equations of the strengthened claims over all choices of r. □

3.3.3 The AND_k Function

Next, we will investigate the k-party information complexity of one of the simplest nontrivial functions: The AND function of k bits. We will show a lower bound on the information complexity of a promise variant that will turn out to be useful later on.

Definition 3.3.5 (AND_k and $\mathrm{AND}_k^{\mathrm{unique}}$ function)**.** Let $x_i \in \{0, 1\}$ for $i \in \{1, \ldots, k\}$. Then the k-player AND function AND_k is defined by

$$\mathrm{AND}_k(x_1, \ldots, x_k) = \bigwedge_{i=1}^{k} x_i \ .$$

In the promise problem $\mathrm{AND}_k^{\mathrm{unique}}$ the domain of AND_k is restricted to inputs $x \in \{0, 1\}^k$ such that either $x = (1, \ldots, 1)$ or that at most one bit in x has the value 1.

Note that the deterministic NIH k-party communication complexity of AND_k is obviously k since AND_k depends on all inputs and therefore each player has to send at least one bit. The deterministic communication complexity of the promise problem $\mathrm{AND}_k^{\mathrm{unique}}$ is 2 since

the value of $\mathrm{AND}_k^{\mathrm{unique}}$ is uniquely determined by every projection of the input x to two coordinates and a single coordinate does not suffice to determine $\mathrm{AND}_k^{\mathrm{unique}}(x)$ in general. Even the information complexity of AND_k and $\mathrm{AND}_k^{\mathrm{unique}}$ for uniformly distributed inputs is not very interesting. For uniformly distributed inputs X the random variable $\mathrm{AND}_k(X)$ is not constant. Since the output $\mathrm{AND}_k(X)$ of a protocol is completely determined by the transcript of the protocol, the transcript must at least reveal some information about the input. Here, due to reasons that will become clear in Section 3.3.4, we will consider random inputs $Z = (Z_1, \ldots, Z_k) \in \{0,1\}^k$ for a distribution such that $\Pr\{\mathrm{AND}_k(Z)=1\} = 0$. In this case it is not at all clear that the transcript has to reveal information about the input since it is not needed to determine the constant output 0 of the protocol. The fact that the transcript nevertheless reveals information about the input in this case is due to the statistical properties of NIH protocols that are described in the previous section.

Our Result

We will prove a lower bound on the conditional information complexity of $\mathrm{AND}_k^{\mathrm{unique}}$ for the following distribution on the inputs:

Definition 3.3.6. Let $Z = (Z_1, \ldots, Z_k) \in \{0,1\}^k$ and $D \in \{1, \ldots, k\}$ be random variables such that their joint distribution has the following properties: The random variable D is uniformly distributed in the set $\{1, \ldots, k\}$ and for all $i \in \{1, \ldots, k\}$ the conditional distribution of Z given D satisfies $\Pr\{Z_i=0|D \neq i\} = 1$ and $\Pr\{Z_i=0|D=i\} = \Pr\{Z_i=1|D=i\} = \frac{1}{2}$.

We will see in Section 3.3.4 that, by a result of Bar-Yossef *et al.* [13], a lower bound on the information complexity of $\mathrm{AND}_k^{\mathrm{unique}}$ for this input distribution is a useful building block for information complexity lower bounds of functions that are more complicated. Now we will prove the following result:

Theorem 3.3.7 (The information complexity of $\mathrm{AND}_k^{\mathrm{unique}}$)**.** *Suppose that ϵ is a constant such that $0 \leq \epsilon < \frac{3}{10}\left(1 - \sqrt{\frac{1}{2}\log\frac{4}{3}}\right) \approx 0.163$. Then there is a constant $c(\epsilon) > 0$ that only depends on ϵ such that*

$$\mathrm{IC}_\epsilon^{\mathrm{NIH}}(\mathrm{AND}_k^{\mathrm{unique}}; Z|D) \geq \frac{c(\epsilon)}{k} .$$

Theorem 3.3.7 is asymptotically optimal with respect to k. To see this, consider the following trivial one-way protocol P for AND_k: For the input $x = (x_1, \ldots, x_k)$ the players, in turn, write their input x_i to the blackboard until the first input bit with the value zero is written to the blackboard. If a bit with the value zero is written to the blackboard then all players know that $\mathrm{AND}_k(x) = 0$. If the protocol terminates with k ones on the blackboard then all players know that $\mathrm{AND}_k(x) = 1$. Let $T(Z)$ denote the transcript of P for the input Z. If $D \neq 1$ then $Z_1 = 0$ and P stops after Z_1 has been written to the blackboard. Since Z_1 is constant given $D \neq 1$, we get

$$\mathrm{I}(T(Z) : Z|D \neq 1) = \mathrm{I}(T(Z) : Z_1|D \neq 1) = \mathrm{H}(Z_1|D \neq 1) = 0 . \tag{3.27}$$

If $D = 1$ then $Z_1 \in \{0,1\}$ is written to the blackboard and the protocol stops at the latest after $Z_2 = 0$ has been written to the blackboard. Since Z_2 is constant and Z_1 is an unbiased random bit given that $D = 1$ we have

$$\mathrm{I}(T(Z) : Z|D=1) = \mathrm{I}(T(Z) : Z_1, Z_2|D=1) = \mathrm{H}(Z_1|D=1) = 1 . \tag{3.28}$$

By combining the previous observations we get

$$\begin{aligned} \operatorname{icost}(P; Z|D) &= \mathrm{I}(T(Z) : Z|D) && (3.29)\\ &= \frac{1}{k}\,\mathrm{I}(T(Z) : Z|D{=}1) + \frac{k-1}{k}\,\mathrm{I}(T(Z) : Z|D \neq 1) && (3.30)\\ &= \frac{1}{k}\,. && (3.31) \end{aligned}$$

Clearly, a protocol for AND_k also solves the promise problem $\mathrm{AND}_k^{\mathrm{unique}}$. Thus our lower bound on the randomized information complexity of $\mathrm{AND}_k^{\mathrm{unique}}$ differs only by a constant factor from a trivial upper bound on the deterministic information complexity of $\mathrm{AND}_k^{\mathrm{unique}}$.

Related Work

The investigation of the conditional information complexity of $\mathrm{AND}_k^{\mathrm{unique}}$ with respect to the distribution that is described in Definition 3.3.6 started with the work of Bar-Yossef, Jayram, Kumar, and Sivakumar [13]. Their work introduced the combination of information theoretical methods and the use of statistical divergences to prove lower bounds on the communication complexity of functions. The term "information statistics" was also coined in this paper. By combining the idea of information complexity from [26] with a novel application of the Hellinger distance, they proved a $\Omega(1/k^2)$ lower bound on the conditional information complexity of $\mathrm{AND}_k^{\mathrm{unique}}$. The lower bound was improved to $\Omega(1/(k \log k))$ by Chakrabarti, Khot and Sun [25]. Noting that Bar-Yossef *et al.* [13] attribute the weakness of their lower bound to the limitations of the properties of the statistical divergences that were used in the proof, they used a direct analytical approach that relies on the analytical properties of the information cost. Chakrabarti *et al.* also proved an optimal $\Omega(1/k)$ lower bound for one-way protocols and thereby raised the question whether the $\Omega(1/(k \log k))$ bound is tight for unrestricted protocols. Finally, an optimal $\Omega(1/k)$ lower bound (Thm. 3.3.7) for unrestricted protocols was proved by Gronemeier [45]. The proof is based on the information statistics approach using the Kullback-Leibler distance. In the meantime Jayram also found a proof of the $\Omega(1/k)$ lower bound that, like the first result of Bar-Yossef *et al.*, is based on the Hellinger distance [51]. His paper is aptly titled "Hellinger strikes back".

Some Definitions and Basic Observations

Our lower bound on the conditional information complexity of $\mathrm{AND}_k^{\mathrm{unique}}$ will be based on the simple observation that the distribution of the transcript of a randomized k-party NIH protocol that computes $\mathrm{AND}_k^{\mathrm{unique}}$ with small error must be sufficiently dissimilar for the all-zero input and the all-one input. This observation is evident from the fact that the value of the function $\mathrm{AND}_k^{\mathrm{unique}}$ differs on these inputs and that the output of the protocol is uniquely determined by the transcript. Unfortunately, the all-one input is not contained in the support set of our input Z according to Definition 3.3.6, hence it is not clear how we can apply this observation to the input Z. Therefore we define an auxiliary input variable X such that the two inputs of interest are contained in the support set of X.

Definition 3.3.8. Let P be a randomized k-party NIH protocol for the promise problem $\mathrm{AND}_k^{\mathrm{unique}}$ such that the error of P is bounded from above by the constant ϵ. Let $X = (X_1, \ldots, X_k) \in \{0,1\}^k$ be a random variable that is uniformly distributed in the set $\{0,1\}^k$ and recall the definition of Z and D from Definition 3.3.6. Then

- let the transcript of P for input X be denoted by T and
- let the transcript of P for input Z be denoted by T'.

Note that even under the condition $X = x$ and $Z = z$ for constants x and z the random variables T and T' are not constant since the transcript also depends on the random inputs of the randomized protocol P. For the transition from the input X to the input Z we need to relate the distribution of the transcript T for the input X to that of the transcript T' for the input Z. To this end, we need to define some additional notation.

Definition 3.3.9. For any vector $v = (v_1, \ldots, v_k) \in \{0,1\}^k$ let v_{-i} denote the projection of v on the $k-1$ coordinates $\{1, \ldots, k\} - \{i\}$, hence $v_{-i} = (v_1, \ldots, v_{i-1}, v_{i+1}, \ldots, v_k) \in \{0,1\}^{k-1}$. Furthermore let $\vec{0}$ and $\vec{1}$ denote the all-zero and all-one vector, respectively. The size of $\vec{0}$ and $\vec{1}$ is not specified explicitly, it must be inferred from the context.

Now observe that the conditional distribution of Z given $D = i$ and the conditional distribution of X given $X_{-i} = \vec{0}$ are identical. If we apply this observation to the conditional information cost of the protocol P then we obtain the following proposition.

Proposition 3.3.10. *Let $i \in \{1, \ldots, k\}$. Then* $\mathrm{I}(T' : Z | D{=}i) = \mathrm{I}(T : X_i | X_{-i}{=}\vec{0})$.

Proof. Let R denote the random inputs of the randomized protocol P. We observed that the conditional distribution of Z given $D = i$ and the conditional distribution of X given $X_{-i} = \vec{0}$ are identical, hence the conditional joint distribution of (X, R) given $X_{-i} = \vec{0}$ is identical to the conditional joint distribution of (Z, R) given that $D = i$ since the random inputs R are independent of X and Z by our assumptions in Section 3.2.1. The transcript of P is a function of the inputs and the random inputs of P, therefore the joint conditional distribution of (X, T) given $X_{-i} = \vec{0}$ and the joint conditional distribution of (Z, T') given that $D = i$ are identical and the claim of the proposition is true. □

Overview of the Proof

Our main objective is to use the dissimilarity of the conditional distribution of T given $X = \vec{0}$ and $X = \vec{1}$ to show that, on average over the choice of the index i, the mutual information of T and X_i must be large under the condition that $X_{-i} = \vec{0}$. By Proposition 3.3.10, this yields a lower bound on the conditional information cost of the protocol P for $\mathrm{AND}_k^{\mathrm{unique}}$ with respect to Z given D. The dissimilarity of the conditional distributions of T given $X = \vec{0}$ and $X = \vec{1}$ is quantified using the Kullback-Leibler distance of $(T|X{=}\vec{0})$ and $(T|X{=}\vec{1})$. Our first intermediate goal will be a lower bound on the Kullback-Leibler distance in terms of the error ϵ of the randomized protocol P for $\mathrm{AND}_k^{\mathrm{unique}}$. Then we need to link this result to the conditional distribution of X_i and T given that $X_{-i} = \vec{0}$ for $i \in \{1, \ldots, n\}$. To this end, Proposition 3.3.4 is used to decompose the Kullback-Leibler distance of $(T|X{=}\vec{0})$ and $(T|X{=}\vec{1})$ into functionals of the conditional distributions $(X_i|T{=}t, X_{-i}{=}\vec{0})$. Note that, by the conditional independence of the variables X_i given T, the conditional distributions $(X_i|T{=}t)$

and $(X_i|T=t, X_{-i}=\vec{0})$ are the same. We will rewrite the Kullback-Leibler distance as a sum of the form

$$\sum_{t\in S} p(t)\cdot g(\Pr\{X_i=0|T=t\}) \tag{3.32}$$

for a subset S of the transcripts of P and functions p and g that are chosen appropriately. In the next main step of the proof we will lower bound the information cost of P for the input Z by a sum of the form

$$\sum_{t\in S} p(t)\cdot f(\Pr\{X_i=0|T=t\}) \tag{3.33}$$

for an appropriately chosen function f. Then, to obtain a lower bound on the information cost of P, it remains to lower bound the functions f in terms of g. This will only work under certain conditions. Finally, we will show that a lower bound on the information cost of P is obtained easily by different means, if these conditions do not hold.

The Error of the Protocol P and $g(x)$

Starting from our initial observation that the distribution of the transcript T for the inputs $X=\vec{0}$ and $X=\vec{1}$ must be dissimilar, we will quantify this dissimilarity by a measure of dissimilarity that is closely related to the Kullback-Leibler distance of the distributions.

Definition 3.3.11. Let V_1 and V_2 be a random variables such that $\text{range}(V_1)=\text{range}(V_2)$ and let $S\subseteq\text{range}(V_1)$ be a set. Then

$$\mathrm{D}_S(V_1,V_2)=\sum_{v\in S}\Pr\{V_1=v\}\log\frac{\Pr\{V_1=v\}}{\Pr\{V_2=v\}}$$

with the convention that $0\cdot\log(0/x)=0$, $0\cdot\log(0/0)=0$, and $x\cdot\log(x/0)=\infty$ if $x\neq 0$.

Note that if $S=\text{range}(V_1)$ then $D_S(V_1,V_2)$ is the Kullback-Leibler distance of V_1 and V_2 according to Definition 2.2.33. The restriction to a subset of the transcripts will turn out to be useful later on. In our proof we will estimate the Kullback-Leibler distance of the conditional distribution of T given that $T\in S$ and $X=\vec{0}$ and the conditional distribution of T given that $T\in S$ and $X=\vec{1}$. The relation of this distance to the quantity D_S is expressed in the following proposition.

Proposition 3.3.12. *Let S be subset of the transcripts of P and let $q=\frac{\Pr\{T\in S|X=\vec{1}\}}{\Pr\{T\in S|X=\vec{0}\}}$. Then*

$$\frac{\mathrm{D}_S((T|X=\vec{0}),(T|X=\vec{1}))}{\Pr\{T\in S|X=\vec{0}\}}=\mathrm{D}((T|T\in S,X=\vec{0}),(T|T\in S,X=\vec{1}))-\log q\ .$$

Proof. It is easy to verify that

$$\frac{\mathrm{D}_S((T|X=\vec{0}),(T|X=\vec{1}))}{\Pr\{T\in S|X=\vec{0}\}}=\sum_{t\in S}\frac{\Pr\{T=t|X=\vec{0}\}}{\Pr\{T\in S|X=\vec{0}\}}\log\frac{\frac{\Pr\{T=t|X=\vec{0}\}}{\Pr\{T\in S|X=\vec{0}\}}}{\frac{\Pr\{T=t|X=\vec{1}\}}{\Pr\{T\in S|X=\vec{1}\}}}-\log q\ . \tag{3.34}$$

Since we are only summing over $t\in S$ we can replace $\Pr\{T=t|X=\vec{0}\}$ and $\Pr\{T=t|X=\vec{1}\}$ by $\Pr\{T=t, t\in S|X=\vec{0}\}$ and $\Pr\{T=t, t\in S|X=\vec{1}\}$, respectively, without changing the value of the sum. Then the claim follows immediately from the definition of conditional probabilities and the definition of the Kullback-Leibler distance (Def. 2.2.33). □

Now we are ready to lower bound $\mathrm{D}_S((T|X=\vec{0}),(T|X=\vec{1}))$ in terms of the error ϵ of P. This is possible if ϵ is sufficiently small and if T is contained in S with a high probability given that $X=\vec{0}$. If these conditions, especially the second condition, do not hold then we will have to resort to a proof method that is different from the ideas that are sketched in this section. Note that the following lemma is the only place in the proof of Theorem 3.3.7 where the properties of the function $\mathrm{AND}_k^{\mathrm{unique}}$ are used. Here we have to verify carefully that we only use properties of the AND_k function that remain valid for the promise problem $\mathrm{AND}_k^{\mathrm{unique}}$.

Lemma 3.3.13. *Let S be subset of all possible transcripts of the protocol P. If the error of P satisfies $\epsilon \leq \frac{3}{10}$ and $\Pr\{T \in S|X=\vec{0}\} \geq \frac{3}{4}$ then*

$$\frac{\mathrm{D}_S((T|X=\vec{0}),(T|X=\vec{1}))}{\Pr\{T \in S|X=\vec{0}\}} \geq \min\left\{\log\frac{3}{2},\, 2\left(1-\frac{10}{3}\epsilon\right)^2 - \log\frac{4}{3}\right\}\,.$$

Proof. Let L denote the left hand side of the inequality in the lemma. In the proof of the lemma we will distinguish two cases. For the first case assume that $\Pr\{T \in S|X=\vec{1}\} < \frac{1}{2}$. Then, by the log sum inequality (Cor. 2.2.35), we get

$$\begin{aligned} L &= \sum_{t\in S} \frac{\Pr\{T=t|X=\vec{0}\}}{\Pr\{T \in S|X=\vec{0}\}} \log \frac{\Pr\{T=t|X=\vec{0}\}}{\Pr\{T=t|X=\vec{1}\}} && (3.35)\\ &\geq \left(\sum_{t\in S} \frac{\Pr\{T=t|X=\vec{0}\}}{\Pr\{T \in S|X=\vec{0}\}}\right) \log \frac{\sum_{t\in S}\Pr\{T=t|X=\vec{0}\}}{\sum_{t\in S}\Pr\{T=t|X=\vec{1}\}} && (3.36)\\ &= \log \frac{\Pr\{T \in S|X=\vec{0}\}}{\Pr\{T \in S|X=\vec{1}\}} && (3.37)\\ &\geq \log \frac{3/4}{1/2} = \log\frac{3}{2}\,. && (3.38) \end{aligned}$$

For the second case assume that $\Pr\{T \in S|X=\vec{1}\} \geq \frac{1}{2}$. In this case we first apply Proposition 3.3.12 and then Theorem 2.2.40. Note that $q = \frac{\Pr\{T\in S|X=\vec{1}\}}{\Pr\{T\in S|X=\vec{0}\}} \leq \frac{4}{3}$ by our assumption that $\Pr\{T \in S|X=\vec{0}\} \geq \frac{3}{4}$. Then we obtain

$$\begin{aligned} L &\geq \mathrm{D}((T|T \in S, X=\vec{0}),(T|T \in S, X=\vec{1})) - \log\frac{4}{3} && (3.39)\\ &\geq 2\cdot \mathrm{V}^2((T|T \in S, X=\vec{0}),(T|T \in S, X=\vec{1})) - \log\frac{4}{3}\,. && (3.40) \end{aligned}$$

The protocol P is a randomized ϵ-error protocol for $\mathrm{AND}_k^{\mathrm{unique}}$, the error bound holds for every input that fulfills the promise of Definition 3.3.5. The input $x=\vec{1}$ is a valid input for the promise problem, therefore the assumption $\Pr\{T \in S|X=\vec{1}\} \geq \frac{1}{2}$ implies that the conditional error probability given that $T \in S$ and $X=\vec{1}$ is bounded by 2ϵ. Otherwise the error for the input $X=\vec{1}$ would be too large. Similarly, $x=\vec{0}$ is a valid input for the promise problem and the conditional error given that $T \in S$ and $X=\vec{0}$ is bounded by $\frac{4}{3}\epsilon$ since $\Pr\{T \in S|X=\vec{0}\} \geq \frac{3}{4}$ by the assumptions of the lemma. Let S_0 be the set of all transcripts $t \in S$ of P such that the output of P is 0 for the transcript t. Then, by our conditional error bounds, we have $\Pr\{T \in S_0|T \in S, X=\vec{1}\} \leq 2\epsilon$ and $\Pr\{T \in S_0|T \in S, X=\vec{0}\} \geq 1-\frac{4}{3}\epsilon$.

Note also that $\Pr\{T \in S_0 | T \in S, X = \vec{0}\} - \Pr\{T \in S_0 | T \in S, X = \vec{1}\} \geq 0$ by our assumption that $\epsilon \leq \frac{3}{10}$. The total variation distance of two distributions on the same set is bounded from below by the absolute difference of the probabilities of any event with respect to the given distributions (Prop. 2.2.37), hence

$$L \geq 2 \cdot \mathrm{V}^2((T|T \in S, X = \vec{0}), (T|T \in S, X = \vec{1})) - \log \frac{4}{3} \tag{3.41}$$

$$\geq 2 \left| \Pr\{T \in S_0 | T \in S, X = \vec{0}\} - \Pr\{T \in S_0 | T \in S, X = \vec{1}\} \right|^2 - \log \frac{4}{3} \tag{3.42}$$

$$\geq 2 \left(1 - \frac{4}{3}\epsilon - 2\epsilon\right)^2 - \log \frac{4}{3} \tag{3.43}$$

$$= 2 \left(1 - \frac{10}{3}\epsilon\right)^2 - \log \frac{4}{3} \,. \tag{3.44}$$

The claim of the lemma follows by taking the minimum of the lower bounds for the two cases in our case distinction. Note that the lower bound on L is positive if ϵ is sufficiently small. □

Ultimately, we will use this lower bound to prove a lower bound on the conditional information cost $\mathrm{I}(T' : Z|D)$ of the protocol P. For the comparison of $\mathrm{I}(T' : Z|D)$ and $\mathrm{D}_S((T|X = \vec{0}), (T|X = \vec{1}))$ we will now rewrite the latter expression in terms of a function g as it was laid out in the overview of the proof. The function g is defined as follows.

Definition 3.3.14. The function $g \colon [0,1] \to \mathbb{R}$ is defined by $g(x) = x \log \frac{x}{1-x}$.

But before we proceed, we will state a simple technical observation that will be used repeatedly in the following as a proposition.

Proposition 3.3.15. *For all $x \in \{0,1\}$ and $t \in \mathrm{supp}(T)$ we have*

(i) $\Pr\{T = t, X_i = x | X_{-i} = \vec{0}\} = \Pr\{T = t | X_{-i} = \vec{0}\} \cdot \Pr\{X_i = x | T = t\}$ *and*

(ii) $\frac{\Pr\{T=t, X_i=x | X_{-i}=\vec{0}\}}{\Pr\{T=t | X_{-i}=\vec{0}\} \cdot \Pr\{X_i=x | X_{-i}=\vec{0}\}} = 2 \Pr\{X_i = x | T = t\}$.

Proof. This proposition is an immediate consequence of the independence of the random variables X_i and Proposition 3.3.4. For the proof of claim (i) observe that

$$\Pr\{T = t, X_i = x | X_{-i} = \vec{0}\} = \Pr\{T = t | X_{-i} = \vec{0}\} \cdot \Pr\{X_i = x | T = t, X_{-i} = \vec{0}\} \tag{3.45}$$

$$= \Pr\{T = t | X_{-i} = \vec{0}\} \cdot \Pr\{X_i = x | T = t\} \,. \tag{3.46}$$

In the last line we used claim (ii) of Proposition 3.3.4. The proof of claim (ii) follows immediately from claim (i) and the observation that $\Pr\{X_i = x | X_{-i} = \vec{0}\} = \Pr\{X_i = x\}$. □

So far, we observed that the distributions $(T|X = \vec{0})$ and $(T|X = \vec{1})$ must be sufficiently dissimilar if ϵ is small. The fact that a protocol for $\mathrm{AND}_k^{\mathrm{unique}}$ must reveal information about some input X_i even if $X_{-i} = \vec{0}$ is due to the observation that the coordinates X_i of the input X are conditionally independent given the transcript T (see Sect. 3.3.2). The first observation is a statement about the conditional distribution of the transcripts given the inputs, the second observation is about the conditional distribution of the inputs given the transcript. We need a "passage" between these conditional distributions to combine our observations. This passage is provided by the following simple observation:

Observation 3.3.16. Note that $\Pr\{X=x\} = \Pr\{X=x'\}$ for all $x, x' \in \{0,1\}^k$. Then, by the definition of conditional probabilities, we have

$$\frac{\Pr\{T=t|X=x\}}{\Pr\{T=t|X=x'\}} = \frac{\Pr\{T=t, X=x\} \cdot \Pr\{X=x'\}}{\Pr\{T=t, X=x'\} \cdot \Pr\{X=x\}} \tag{3.47}$$

$$= \frac{\Pr\{T=t, X=x\}}{\Pr\{T=t, X=x'\}} \tag{3.48}$$

$$= \frac{\Pr\{T=t, X=x\} \cdot \Pr\{T=t\}}{\Pr\{T=t, X=x'\} \cdot \Pr\{T=t\}} \tag{3.49}$$

$$= \frac{\Pr\{X=x|T=t\}}{\Pr\{X=x'|T=t\}} \,. \tag{3.50}$$

The application of this observation in the following lemma is our main motivation for the use of the Kullback-Leibler distance in the proof of a lower bound on $\mathrm{IC}^{\mathrm{NIH}}_{\epsilon}(\mathrm{AND}^{\mathrm{unique}}_k; Z|D)$.

Lemma 3.3.17. *Let S be a subset of all transcripts of the protocol P. Then*

$$\mathrm{D}_S((T|X=\vec{0}), (T|X=\vec{1})) = 2\sum_{i=1}^{k}\sum_{t\in S} \Pr\{T=t|X_{-i}=\vec{0}\} \cdot g(\Pr\{X_i=0|T=t\}) \,.$$

Proof. For brevity let $D = \mathrm{D}_S((T|X=\vec{0}), (T|X=\vec{1}))$. Then, by Observation 3.3.16,

$$D = \sum_{t\in S} \Pr\{T=t|X=\vec{0}\} \cdot \log \frac{\Pr\{T=t|X=\vec{0}\}}{\Pr\{T=t|X=\vec{1}\}} \tag{3.51}$$

$$= \sum_{t\in S} \Pr\{T=t|X=\vec{0}\} \cdot \log \frac{\Pr\{X=\vec{0}|T=t\}}{\Pr\{X=\vec{1}|T=t\}} \,. \tag{3.52}$$

By using the conditional independence of the variables X_i given T (Prop. 3.3.4), we get

$$D = \sum_{t\in S} \Pr\{T=t|X=\vec{0}\} \cdot \log \prod_i \frac{\Pr\{X_i=0|T=t\}}{\Pr\{X_i=1|T=t\}} \tag{3.53}$$

$$= \sum_i \sum_{t\in S} \Pr\{T=t|X=\vec{0}\} \cdot \log \frac{\Pr\{X_i=0|T=t\}}{\Pr\{X_i=1|T=t\}} \tag{3.54}$$

$$= \sum_i \sum_{t\in S} \frac{\Pr\{T=t, X_i=0|X_{-i}=\vec{0}\}}{\Pr\{X_i=0|X_{-i}=\vec{0}\}} \cdot \log \frac{\Pr\{X_i=0|T=t\}}{\Pr\{X_i=1|T=t\}} \tag{3.55}$$

$$= 2\sum_i \sum_{t\in S} \Pr\{T=t, X_i=0|X_{-i}=\vec{0}\} \cdot \log \frac{\Pr\{X_i=0|T=t\}}{\Pr\{X_i=1|T=t\}} \,. \tag{3.56}$$

Now claim (i) of Proposition 3.3.15 can be applied to the first factor of each term to obtain

$$D = 2\sum_i \sum_{t\in S} \Pr\{T=t|X_{-i}=\vec{0}\} \cdot \Pr\{X_i=0|T=t\} \cdot \log \frac{\Pr\{X_i=0|T=t\}}{\Pr\{X_i=1|T=t\}} \tag{3.57}$$

$$= 2\sum_i \sum_{t\in S} \Pr\{T=t|X_{-i}=\vec{0}\} \cdot \Pr\{X_i=0|T=t\} \cdot \log \frac{\Pr\{X_i=0|T=t\}}{1-\Pr\{X_i=0|T=t\}} \tag{3.58}$$

$$= 2\sum_i \sum_{t\in S} \Pr\{T=t|X_{-i}=\vec{0}\} \cdot g(\Pr\{X_i=0|T=t\}) \,. \tag{3.59}$$

□

The Information Cost of the Protocol P and $f(x)$

Next, we will work on the "information cost end" of the proof. We will rewrite the information cost of P in terms of a function f. The definition of f may look somewhat opaque at this point, but the rationale behind the following definition will be explained in the next section.

Definition 3.3.18. The function $f\colon [0,1] \to \mathbb{R}$ is defined by $f(x) = x \log 2x + \frac{1-x}{2} \log 2(1-x)$.

Like in the last section, we will also restrict our analysis to subsets of the transcripts of P, but here we need to consider subsets that have certain useful properties.

Definition 3.3.19. Let $S(\alpha)$ denote the set of all transcripts t of the protocol P such that $\Pr\{X_i{=}0|T{=}t\} < \alpha$ for all $i \in \{1, \ldots, k\}$.

Now we are ready for the main result of this section, a lower bound on the information cost of P that is expressed in terms of the function f.

Lemma 3.3.20. *Let $\alpha \geq \frac{1}{2}$ be a constant. Then*

$$\mathrm{I}(T' : Z|D) \geq \frac{1}{k} \sum_{i=1}^{k} \sum_{t \in S(\alpha)} \Pr\{T{=}t|X_{-i}{=}\vec{0}\} \cdot f(\Pr\{X_i{=}0|T{=}t\}) \ .$$

Proof. Define the functions $f_1(x) = x \log 2x + (1-x) \log 2(1-x)$ and $f_2(x) = x \log 2x$. Then we have $f(x) = \frac{1}{2}(f_1(x) + f_2(x))$ and Lemma 3.3.20 follows immediately from these claims:

Claim 3.3.21. *Let $\alpha \geq \frac{1}{2}$ be a constant. Then*

$$\mathrm{I}(T' : Z|D) \geq \frac{1}{k} \sum_{i} \sum_{t \in S(\alpha)} \Pr\{T{=}t|X_{-i}{=}\vec{0}\} \cdot f_1(\Pr\{X_i{=}0|T{=}t\}) \ . \tag{3.60}$$

Claim 3.3.22. *Let $\alpha \geq \frac{1}{2}$ be a constant. Then*

$$\mathrm{I}(T' : Z|D) \geq \frac{1}{k} \sum_{i} \sum_{t \in S(\alpha)} \Pr\{T{=}t|X_{-i}{=}\vec{0}\} \cdot f_2(\Pr\{X_i{=}0|T{=}t\}) \ . \tag{3.61}$$

It remains to prove that the claims are true. Let L denote the left hand side of the inequality in the lemma. By using Proposition 3.3.10 and by the definition of the mutual information in terms of the Kullback-Leibler distance (Prop. 2.2.34) we obtain

$$L = \mathrm{I}(T' : Z|D) = \frac{1}{k} \sum_{i} \mathrm{I}(T' : Z_i|D{=}i) = \frac{1}{k} \sum_{i} \mathrm{I}(T : X_i|X_{-i}{=}\vec{0}) \tag{3.62}$$

$$= \frac{1}{k} \sum_{i} \sum_{t,x} \Pr\{T{=}t, X_i{=}x|X_{-i}{=}\vec{0}\} \cdot \log \frac{\Pr\{T{=}t, X_i{=}x|X_{-i}{=}\vec{0}\}}{\Pr\{T{=}t|X_{-i}{=}\vec{0}\} \Pr\{X_i{=}x|X_{-i}{=}\vec{0}\}} \ . \tag{3.63}$$

Then, by applying both claims of Proposition 3.3.15 and by using the fact that $\Pr\{X_i=1|T=t\} = 1 - \Pr\{X_i=0|T=t\}$ in the last line, we obtain

$$L = \frac{1}{k}\sum_i \sum_{t,x} \Pr\{T=t, X_i=x|X_{-i}=\vec{0}\} \cdot \log \frac{\Pr\{T=t, X_i=x|X_{-i}=\vec{0}\}}{\Pr\{T=t|X_{-i}=\vec{0}\}\Pr\{X_i=x|X_{-i}=\vec{0}\}} \tag{3.64}$$

$$= \frac{1}{k}\sum_i \sum_t \Pr\{T=t|X_{-i}=\vec{0}\} \sum_x \Pr\{X_i=x|T=t\} \cdot \log\left(2\Pr\{X_i=x|T=t\}\right) \tag{3.65}$$

$$= \frac{1}{k}\sum_i \sum_t \Pr\{T=t|X_{-i}=\vec{0}\} \cdot f_1\left(\Pr\{X_i=0|T=t\}\right) \ . \tag{3.66}$$

Now, for the proof of the Claim 3.3.21, it is sufficient to observe that $f_1(x) = 1 - h_2(x) \geq 0$. Hence, each term of the last sum is nonnegative and restricting the range of summation to the subset $S(\alpha)$ can only decrease the sum.

For the proof of Claim 3.3.22 we first observe that $f_1(x) = f_2(x) + f_2(1-x)$. Then we observe that $f_2(x) \geq 0$ for all $x \in [\frac{1}{2}, 1]$ and that $\Pr\{X_i=0|T=t\} \geq \alpha > \frac{1}{2}$ for all $t \notin S(\alpha)$, hence $f_2(\Pr\{X_i=0|T=t\}) \geq 0$ for all $t \notin S(\alpha)$. By using theses observations, we obtain

$$L = \frac{1}{k}\sum_i \sum_t \Pr\{T=t|X_{-i}=\vec{0}\} \cdot f_1\left(\Pr\{X_i=0|T=t\}\right) \tag{3.67}$$

$$= \frac{1}{k}\sum_i \sum_t \Pr\{T=t|X_{-i}=\vec{0}\} \cdot \left(f_2\left(\Pr\{X_i=0|T=t\}\right) + f_2\left(\Pr\{X_i=1|T=t\}\right)\right) \tag{3.68}$$

$$\begin{aligned} &\geq \frac{1}{k}\sum_i \sum_{t\in S(\alpha)} \Pr\{T=t|X_{-i}=\vec{0}\} \cdot f_2\left(\Pr\{X_i=0|T=t\}\right) \\ &\quad + \frac{1}{k}\sum_i \sum_t \Pr\{T=t|X_{-i}=\vec{0}\} \cdot f_2\left(\Pr\{X_i=1|T=t\}\right) \ . \end{aligned} \tag{3.69}$$

Now it suffices to show that $R_i = \sum_t \Pr\{T=t|X_{-i}=\vec{0}\} \cdot f_2\left(\Pr\{X_i=1|T=t\}\right)$ is nonnegative for all $i \in \{1, \ldots, k\}$. To this end we apply Proposition 3.3.15 to obtain

$$R_i = \sum_t \Pr\{T=t|X_{-i}=\vec{0}\} \cdot \Pr\{X_i=1|T=t\} \cdot \log\left(2\Pr\{X_i=1|T=t\}\right) \tag{3.70}$$

$$= \sum_t \Pr\{T=t, X_i=1|X_{-i}=\vec{0}\} \cdot \log \frac{\Pr\{X_i=1|T=t, X_{-i}=\vec{0}\}}{\Pr\{X_i=1|X_{-i}=\vec{0}\}} \tag{3.71}$$

$$= \sum_t \Pr\{T=t, X_i=1|X_{-i}=\vec{0}\} \cdot \log \frac{\Pr\{T=t, X_i=1|X_{-i}=\vec{0}\}}{\Pr\{T=t|X_{-i}=\vec{0}\} \cdot \Pr\{X_i=1|X_{-i}=\vec{0}\}} \ . \tag{3.72}$$

Now we can apply the log sum inequality (Cor. 2.2.35) to this result. Note that

$$\sum_t \Pr\{T=t, X_i=1|X_{-i}=\vec{0}\} = \Pr\{X_i=1|X_{-i}=\vec{0}\} \tag{3.73}$$

and that

$$\sum_t \Pr\{T=t|X_{-i}=\vec{0}\} \cdot \Pr\{X_i=1|X_{-i}=\vec{0}\} = \Pr\{X_i=1|X_{-i}=\vec{0}\} \ . \tag{3.74}$$

Hence the log sum inequality yields

$$R_i \geq \Pr\{X_i=1|X_{-i}=\vec{0}\} \cdot \log \frac{\Pr\{X_i=1|X_{-i}=\vec{0}\}}{\Pr\{X_i=1|X_{-i}=\vec{0}\}} = 0 \tag{3.75}$$

and Claim 3.3.22 follows immediately. □

Comparing $f(x)$ and $g(x)$

First, we summarize our progress so far: Let $\alpha \geq \frac{1}{2}$ be a constant. Then

$$\mathrm{I}(T' : Z|D) \geq \frac{1}{k} \sum_{i=1}^{k} \sum_{t \in S(\alpha)} \Pr\{T=t|X_{-i}=\vec{0}\} \cdot f(\Pr\{X_i=0|T=t\}) \tag{3.76}$$

by Lemma 3.3.20 and

$$\frac{1}{2k} \mathrm{D}_{S(\alpha)}((T|X=\vec{0}),(T|X=\vec{1})) = \frac{1}{k} \sum_{i=1}^{k} \sum_{t \in S(\alpha)} \Pr\{T=t|X_{-i}=\vec{0}\} \cdot g(\Pr\{X_i=0|T=t\}) \tag{3.77}$$

by Lemma 3.3.17. Note that the sums on the right hand sides are almost identical, except for the fact that the first sum uses the function f whereas the second sum uses the function g. Additionally, we can lower bound $\mathrm{D}_{S(\alpha)}((T|X=\vec{0}),(T|X=\vec{1}))$ in terms of the error ϵ of the protocol P if $\Pr\{T \in S(\alpha)|X=\vec{0}\} \geq \frac{3}{4}$ by Lemma 3.3.13. Hence, proving a lower bound on the information cost of P essentially boils down to a simple comparison of the functions f and g. Unfortunately, this simple idea seems to be doomed to fail: On the unit interval the function $f(x)$ is bounded from above whereas the function $g(x)$ is not, therefore bounding $f(x)$ from below in terms of $g(x)$ should be a futile attempt. But this is only true if we compare f and g on the whole unit interval, the ratio of $f(x)$ and $g(x)$ is bounded for every interval $[0,\alpha]$ such that $\alpha < 1$. If we can find a constant $\frac{1}{2} \leq \alpha < 1$ and a strictly positive constant c_α such that $f(x) \geq c_\alpha \cdot g(x)$ for all $x \in [0,\alpha]$ then our initial plan of comparing f and g does work. We can use the set $S(\alpha)$ (see Def. 3.3.19) in the two sums to restrict the index of summation to transcripts t such that $\Pr\{X_i=0|T=t\} < \alpha$. In this case we have $f(\Pr\{X_i=0|T=t\}) \geq c_\alpha \cdot f(\Pr\{X_i=0|T=t\})$ for each $t \in S(\alpha)$. The requirement $\alpha \geq \frac{1}{2}$ is due to Lemma 3.3.20. For reasons that will become clear later we actually need that $\alpha > \frac{1}{2}$. The following proposition shows that appropriate constants α and c_α exist.

Proposition 3.3.23. *There is a constant $\beta > \frac{1}{2}$ such that $4f(x) \geq g(x)$ for all $x \in [0,\beta]$.*

Proof. First observe that

$$\begin{aligned} 4 \cdot f(x) &= 4x \log 2x + 2(1-x) \log 2(1-x) && (3.78)\\ &= x \log \frac{x}{1-x} + 3x \log 2x + (2-x) \log 2(1-x) && (3.79)\\ &= g(x) + 3x \log 2x + (2-x) \log 2(1-x)\ . && (3.80) \end{aligned}$$

Thus it is sufficient to show that there is a constant $\beta > \frac{1}{2}$ such that

$$r(x) = 3x \log 2x + (2-x) \log 2(1-x) \geq 0 \tag{3.81}$$

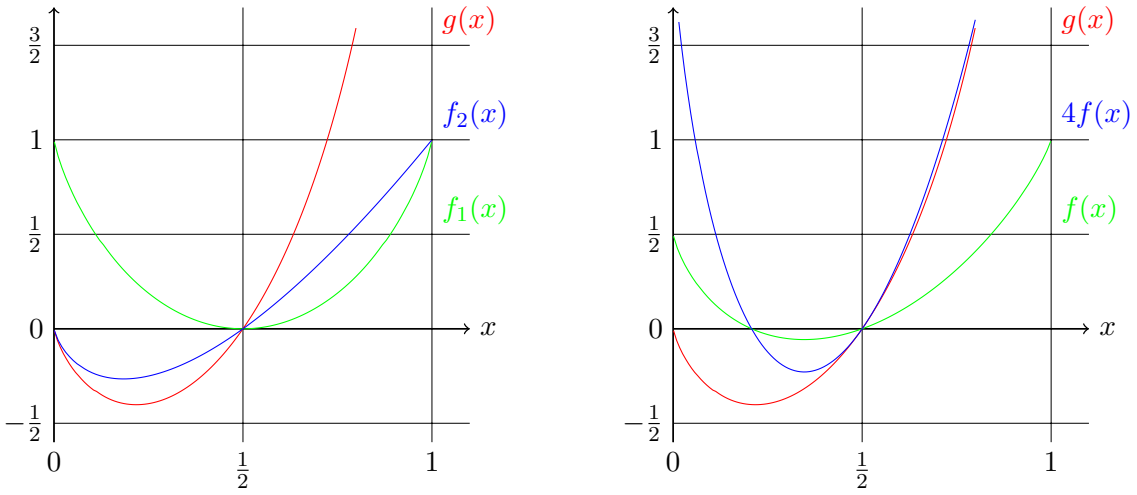


Figure 3.1: Comparison of the functions $g(x)$, $f(x)$, $f_1(x)$, and $f_2(x)$

for all $x \in [0, \beta]$. The first derivative of r is

$$r'(x) = 3\log 2x - \log(2-2x) + \frac{1-2x}{(1-x)\ln 2} \tag{3.82}$$

and the second derivative of r is

$$r''(x) = \frac{2x^2 - 6x + 3}{x(1-x)^2 \ln 2}\,. \tag{3.83}$$

It is easy to verify that the roots of $r''(x)$ are $\frac{3}{2} - \frac{1}{2}\sqrt{3}$ and $\frac{3}{2} + \frac{1}{2}\sqrt{3}$, that $r''(\frac{1}{2}) = \frac{4}{\ln 2} > 0$, and that $\frac{3}{2} - \frac{1}{2}\sqrt{3} \approx 0.634 > \frac{1}{2}$. Hence $r''(x) \geq 0$ for all $x \in [0, \frac{3}{2} - \frac{1}{2}\sqrt{3}]$ and r is convex in this interval. Since $r(\frac{1}{2}) = r'(\frac{1}{2}) = 0$, this implies that $r(x) \geq 0$ in this interval and the claim of the lemma holds for $\beta = \frac{3}{2} - \frac{1}{2}\sqrt{3}$. □

We still have to solve the problem that the approach outlined above only works under the condition that $\Pr\{T \in S(\beta) | X = \vec{0}\} \geq \frac{3}{4}$ where β is the constant from the last Proposition. This is due to the requirements of Lemma 3.3.13. Before we approach this problem in the next section, we will give some rationale for our definition of the function f.

Our choice of the function f is closely tied to the comparison of f and g. Recall that we bounded the information cost of P from below in terms of the functions $f_1(x)$ and $f_2(x)$ in the proof of Lemma 3.3.20. As we observed in Claim 3.3.21 and Claim 3.3.22 of this proof, these functions are closely related to the mutual information of the transcript T and the input X_i for $i \in \{1, \ldots, k\}$, hence they are natural candidates for the choice of the function f in the proof strategy that was outlined in the overview of the proof. However, for the choice $f(x) = f_1(x)$ or $f(x) = f_2(x)$ this proof strategy would break down at the comparison of $f(x)$ and $g(x)$ that was carried out for the actual choice of $f(x)$ according to Definition 3.3.18 in this chapter. The problems that are caused by the choice $f(x) = f_1(x)$ or $f(x) = f_2(x)$ can be easily observed in the left plot of Figure 3.1:

For $x \in [0, \frac{1}{2}]$ we have $f_1(x) \geq c \cdot g(x)$ for every positive constant c, but unfortunately for every $\alpha < 1$ we have $f_1(x) < g(x)$ for $x \in (\frac{1}{2}, \alpha]$. Additionally, there is no constant c such

that $f_1(x) \geq c \cdot g(x)$ for all $x \in (\frac{1}{2}, \alpha]$ since $\lim_{x \to 0} f_1(\frac{1}{2}+x)/g(\frac{1}{2}+x) = \infty$. Thus for the choice $f(x) = f_1(x)$ the comparison of $f(x)$ and $g(x)$ would always fail on the interval $(\frac{1}{2}, 1]$.

The situation is slightly different for the choice $f(x) = f_2(x)$. Here we can find constants α and c such that $f_2(x) \geq c \cdot g(x)$ for all $x \in [\frac{1}{2}, \alpha]$, but whenever $f_2(\frac{1}{2}+x) \geq c \cdot g(\frac{1}{2}+x)$ for some positive x, then there exists also some positive x' such that $f(\frac{1}{2}-x') < c \cdot g(\frac{1}{2}-x')$. This can be verified mathematically by inspecting the slope of the functions $g(x)$ and $f_2(x)$ near $x = \frac{1}{2}$. Intuitively, close to $x = \frac{1}{2}$ these functions locally behave like linear functions that intersect in the point $(\frac{1}{2}, 0)$. Consequently, the comparison of $f_2(x)$ and $g(x)$ works in the interval $(\frac{1}{2}, \alpha]$ at the expense of a failure in the interval $[0, \frac{1}{2}]$.

In summary, for the choice $f(x) = f_1(x)$ the comparison of $f(x)$ and $g(x)$ would fail on the interval $(\frac{1}{2}, \alpha]$ and for the choice $f(x) = f_2(x)$ it would fail on the interval $[0, \frac{1}{2}]$. Our actual choice of $f(x)$ is the average of $f_1(x)$ and $f_2(x)$. It turns out that the particular deficiencies of the functions f_1 and f_2 are compensated by the other function of the average, respectively. The fact that $f(x)/g(x)$ is bounded in the interval $(\frac{1}{2}, \alpha]$ is due to the contribution of $f_2(x)$, the contribution of $f_1(x)$ ensures that $f(x)$ does not get smaller than $c \cdot g(x)$ in the interval $[0, \frac{1}{2}]$ if we multiply $g(x)$ by an appropriate constant c such that $f(x) \geq c \cdot g(x)$ for all $x \in (\frac{1}{2}, \alpha]$. The right plot in Figure 3.1 illustrates this property of the function $f(x)$.

Combining the Partial Results

We have seen in the last sections that our proof strategy of comparing the functions $f(x)$ and $g(x)$ works under the condition that $\Pr\{T \in S(\beta) | X = \vec{0}\} \geq \frac{3}{4}$ where β is the constant from Proposition 3.3.23. We cannot guarantee that this condition always holds. For example, we will see in the proof of Corollary 3.3.26 below that $\Pr\{T \in S(1) | X = \vec{0}\} = 0$ if P is a zero-error protocol. Therefore we need a different proof strategy for the case that $\Pr\{T \in S(\beta) | X = \vec{0}\} < \frac{3}{4}$. Fortunately, proving a lower bound on the information cost of P under this condition is easy. If $t \notin S(\beta)$ then $\Pr\{X_i = 0 | T = t\} \geq \beta > \frac{1}{2}$ and $\mathrm{H}(X_i | T = t) < 1$ for at least one $i \in \{1, \ldots, k\}$. In this case the entropy of X_i is reduced significantly by the knowledge of the fact that $T = t$. If this happens with a sufficiently large probability with respect to the distribution of T, then the mutual information of T and X must be large. This idea is quantified in the following lemma.

Lemma 3.3.24. *Let α be a constant subject to $\frac{1}{2} < \alpha \leq 1$. Then*

$$\mathrm{I}(T' : Z | D) \geq \frac{1 - h_2(\alpha)}{2k} \Pr\{T \notin S(\alpha) | X = \vec{0}\} .$$

Proof. By using Proposition 3.3.10, the conditional independence of the random variables X_i

given T (Prop. 3.3.4), and the fact that $1 - \mathrm{H}(X_i|T{=}t) \geq 0$, we obtain

$$\mathrm{I}(T' : Z|D) = \frac{1}{k}\sum_i \mathrm{I}(T' : Z_i|D{=}i) \tag{3.84}$$

$$= \frac{1}{k}\sum_i \mathrm{I}(T : X_i|X_{-i}{=}\vec{0}) \tag{3.85}$$

$$= \frac{1}{k}\sum_i \Big(\mathrm{H}(X_i|X_{-i}{=}\vec{0}) - \mathrm{H}(X_i|T, X_{-i}{=}\vec{0})\Big) \tag{3.86}$$

$$= \frac{1}{k}\sum_i \sum_t \Pr\{T{=}t|X_{-i}{=}\vec{0}\}\Big(1 - \mathrm{H}(X_i|T{=}t, X_{-i}{=}\vec{0})\Big) \tag{3.87}$$

$$= \frac{1}{k}\sum_i \sum_t \Pr\{T{=}t|X_{-i}{=}\vec{0}\}\,(1 - \mathrm{H}(X_i|T{=}t)) \tag{3.88}$$

$$\geq \frac{1}{k}\sum_i \sum_{t\notin S(\alpha)} \Pr\{T{=}t|X_{-i}{=}\vec{0}\}\,(1 - \mathrm{H}(X_i|T{=}t))\ . \tag{3.89}$$

By the fact that $\Pr\{X_i = 0|X_{-i}{=}\vec{0}\} = \frac{1}{2}$, we have

$$\Pr\{T{=}t|X_{-i}{=}\vec{0}\} \geq \Pr\{T{=}t, X_i = 0|X_{-i}{=}\vec{0}\} = \frac{1}{2}\Pr\{T{=}t|X{=}\vec{0}\} \tag{3.90}$$

and by the definition of $S(\alpha)$, for every $t \notin S(\alpha)$ there is an i such that $\Pr\{X_i{=}0|T{=}t\} \geq \alpha$, hence $\mathrm{H}(X_i|T{=}t) \leq h_2(\alpha)$. Then our claim follows from these observations

$$\mathrm{I}(T' : Z|D) \geq \frac{1}{k}\sum_i \sum_{t\notin S(\alpha)} \Pr\{T{=}t|X_{-i}{=}\vec{0}\}\,(1 - \mathrm{H}(X_i|T{=}t)) \tag{3.91}$$

$$\geq \frac{1}{2k}\sum_{t\notin S(\alpha)} \Pr\{T{=}t|X{=}\vec{0}\}\sum_i (1 - \mathrm{H}(X_i|T{=}t)) \tag{3.92}$$

$$\geq \frac{1}{2k}\sum_{t\notin S(\alpha)} \Pr\{T{=}t|X{=}\vec{0}\}(1 - h_2(\alpha))\ . \tag{3.93}$$

□

So far we have shown partial results that yield lower bounds on the information complexity of $\mathrm{AND}_k^{\mathrm{unique}}$ under certain conditions that do not necessarily hold. For a full proof of Theorem 3.3.7 we will finally combine our previous partial results into a lower bound on the information cost of P that holds unconditionally.

Theorem 3.3.25. *Suppose that the randomized k-party protocol P computes $\mathrm{AND}_k^{\mathrm{unique}}$ with an error that is bounded from above by the constant ϵ. If $\epsilon < \frac{3}{10}\left(1 - \sqrt{\frac{1}{2}\log\frac{4}{3}}\right)$ then there is a constant $c(\epsilon) > 0$ that only depends on the constant ϵ such that*

$$\mathrm{I}(T' : Z|D) \geq \frac{c(\epsilon)}{k}\ .$$

Proof. Let β be the constant from Proposition 3.3.23. For the proof of the lemma we will consider two cases. For the first case, assume that $\Pr\{T \in S(\beta) | X = \vec{0}\} < \frac{3}{4}$. In this case we apply Lemma 3.3.24 for $\alpha = \beta$ and we get

$$\mathrm{I}(T' : Z|D) \geq \frac{1}{2k} \Pr\{T \notin S(\beta) | X = \vec{0}\}(1 - h_2(\beta)) \geq \frac{1}{8k}(1 - h_2(\beta)) \,. \tag{3.94}$$

Note that, since $\beta > 1/2$, there is a constant $c_1 > 0$ such that the right hand side of the last inequality is bounded from below by c_1/k.

For the second case, assume that $\Pr\{T \in S(\beta) | X = \vec{0}\} \geq \frac{3}{4}$. In this case we first apply Lemma 3.3.20 for $\alpha = \beta$ to obtain

$$\mathrm{I}(T' : Z|D) \geq \frac{1}{k} \sum_i \sum_{t \in S(\beta)} \Pr\{T = t | X_{-i} = \vec{0}\} \cdot f(\Pr\{X_i = 0 | T = t\}) \,. \tag{3.95}$$

Note that $\Pr\{X_i = 0 | T = t\} < \beta$ for all $i \in \{1, \ldots, k\}$ if $t \in S(\beta)$. Hence for all $t \in S(\beta)$ and all $i \in \{1, \ldots, k\}$ we have $4f(\Pr\{X_i = 0 | T = t\}) \geq g(\Pr\{X_i = 0 | T = t\})$ by Proposition 3.3.23 and therefore

$$\mathrm{I}(T' : Z|D) \geq \frac{1}{4k} \sum_i \sum_{t \in S(\beta)} \Pr\{T = t | \vec{X}_{-i} = \vec{0}\} \cdot g(\Pr\{X_i = 0 | T = t\}) \,. \tag{3.96}$$

Now we can apply Lemma 3.3.17 to obtain

$$\mathrm{I}(T' : Z|D) \geq \frac{1}{8k} \mathrm{D}_{S(\beta)}(\, (T|X = \vec{0}) \, , \, (T|X = \vec{1}) \,) \,. \tag{3.97}$$

Finally, we apply Lemma 3.3.13 to get

$$\mathrm{I}(T' : Z|D) \geq \frac{1}{8k} \cdot \Pr\{T \in S(\beta) | X = \vec{0}\} \cdot \min\left\{ \log\frac{3}{2}, 2\left(1 - \frac{10}{3}\epsilon\right)^2 - \log\frac{4}{3} \right\} \tag{3.98}$$

$$\geq \frac{3}{32k} \cdot \min\left\{ \log\frac{3}{2}, 2\left(1 - \frac{10}{3}\epsilon\right)^2 - \log\frac{4}{3} \right\} \,. \tag{3.99}$$

For $\epsilon < \frac{3}{10}\left(1 - \sqrt{\frac{1}{2}\log\frac{4}{3}}\right)$ the minimum in the last inequality is a positive constant that only depends on the constant ϵ. Hence, there is a constant $c_2(\epsilon) > 0$ that only depends on the constant ϵ such that the right hand side is bounded from below by $\frac{c_2(\epsilon)}{k}$. The claim of the lemma follows from the two cases if we choose $c(\epsilon) = \min\{c_1, c_2(\epsilon)\}$. □

This completes the proof of Theorem 3.3.7 since it follows immediately from the last theorem.

A Simple Lower Bound for Zero-Error Protocols

For zero-error protocols the lower bound on the information complexity of $\mathrm{AND}_k^{\mathrm{unique}}$ can be strengthened while the proof is simplified significantly. The following corollary shows that a lower bound for zero-error protocols can be obtained using only Lemma 3.3.24.

Corollary 3.3.26. *The zero-error randomized information complexity of* $\mathrm{AND}_k^{\mathrm{unique}}$ *satisfies*

$$\mathrm{IC}_0^{\mathrm{NIH}}(\mathrm{AND}_k^{\mathrm{unique}}; Z|D) \geq \frac{1}{2k} \,.$$

Proof. Let P be a randomized k-party zero-error protocol for $\mathrm{AND}_k^{\mathrm{unique}}$ and let T' denote the transcript of P for the input Z. For a proof of the corollary it is sufficient to show that $\mathrm{I}(T' : Z|D) \geq \frac{1}{2k}$. Recall that T is the transcript of P for the input X. Then the corollary follows immediately from Lemma 3.3.24 for $\alpha = 1$: By Definition 3.1.3, the output of the protocol is uniquely determined by the transcript. Let $P(t)$ denote the output of the protocol P for transcript t. Suppose that $\Pr\{T=t\} > 0$ and $P(t) = 0$. Then $\Pr\{X_i=0|T=t\} = 1$ for at least one $i \in \{1, \ldots, k\}$ since if this was not the case then, by Proposition 3.3.4, we would have $\Pr\{X=\vec{1}|T=t\} > 0$. This is not possible for a zero error protocol because then the event that $X=\vec{1}$ and $T=t$ would have a nonzero probability and consequently the output of the protocol would be wrong with a nonzero probability by our assumption that $P(t) = 0$. This implies that $\Pr\{T \notin S(1)|P(T)=0\} = 1$. Under the condition $X=\vec{0}$ the output of P is 0 with probability 1, again by the zero-error property and the fact that the input $X=\vec{0}$ is valid for the promise problem $\mathrm{AND}_k^{\mathrm{unique}}$, therefore the last observation implies that $\Pr\{T \notin S(1)|X=\vec{0}\} = 1$ and obviously $1 - h_2(1) = 1$. Plugging this into Lemma 3.3.24 yields the claimed result. □

3.3.4 The Information Complexity of Direct Sum Problems

In a *direct sum problem* we wish to evaluate a given function $h(x_1, \ldots, x_k)$ on n independent inputs $a_i = (a_{i,1}, \ldots, a_{i,k})$ for $i \in \{1, \ldots, n\}$ using a k-party NIH-protocol. As usual, the ith player sees the ith coordinate of each input, hence player i sees the inputs $(a_{1,i}, \ldots, a_{n,i})$. An obvious solution of this problem is to compute the value of $h(a_i)$ separately for each i using an optimal protocol for h. Clearly, for deterministic protocols this solution has the cost $n \cdot \mathrm{D}^{\mathrm{NIH}}(h)$, for randomized protocols it has the cost $n \cdot \mathrm{R}_\epsilon^{\mathrm{NIH}}(h)$ if we require that for each input the output is correct with a probability of at least $1 - \epsilon$, and so on. However, it is not at all clear whether this solution is optimal. This question addresses a fundamental property of computation and has been investigated for different models of communication complexity as well as other models of computation. It is known that the obvious solution described above is optimal for nondeterministic two-player protocols, whereas the optimality of this solution for mostly all other models of communication complexity is an open problem [56]. A related problem is the computation of simple functions of the n independent copies of h, for example the computation of $\bigvee_{i=1}^n h(a_i)$ where we assume that the range of h is $\{0, 1\}$. This type of problem is usually also called a direct sum problem. Note that, even if direct sum lower bounds do not hold in general for a given model of computation, direct sum lower bounds may nevertheless be obtained for proof methods that only yield strong lower bounds for a subset of problems (e.g. [65, 14]) or, at least, direct sum lower bounds may be obtained for individual problems (e.g. [39]).

Remark 3.3.27. In this section we will consider direct sums of n copies of a given function $f\colon \{0,1\}^k \longrightarrow \{0,1\}$. This results in functions of the form $f\colon \left(\{0,1\}^k\right)^n \longrightarrow \{0,1\}$. The ith player of a NIH protocol sees the ith coordinate of each input, hence each player sees an input from the set $\{0,1\}^n$. In the corresponding communication problem we therefore formally have to consider functions of the form $f\colon (\{0,1\}^n)^k \longrightarrow \{0,1\}$. To avoid needlessly cluttered notation for the transition from direct sums to communication problems, in this section we will not distinguish between functions $f\colon \left(\{0,1\}^k\right)^n \longrightarrow \{0,1\}$ and $f\colon (\{0,1\}^n)^k \longrightarrow \{0,1\}$.

In the following we will describe a direct sum result for the conditional information complexity of functions due to Bar-Yossef, Jayram, Kumar, and Sivakumar [13]. Together with

our lower bound on the conditional information complexity of $\mathrm{AND}_k^{\mathrm{unique}}$ this result will imply an optimal lower bound on the conditional information complexity for a useful promise variant of the well-known disjointness function (see Sect. 3.3.5). Bar-Yossef *et al.* consider the computation of a function g of the n independent copies of the function h.

Definition 3.3.28 (g-h-direct sum)**.** Let $f\colon \left(\{0,1\}^k\right)^n \longrightarrow \{0,1\}$ be a function. If there are functions $g\colon \{0,1\}^n \longrightarrow \{0,1\}$ and $h\colon \{0,1\}^k \longrightarrow \{0,1\}$ such that

$$f(x_1,\ldots,x_n) = g(\,h(x_1),\ldots,h(x_n)\,)$$

for all $x = (x_1,\ldots,x_n)$ where $x_i \in \{0,1\}^k$ then f is called a g-h-direct sum.

The definition of g-h-direct sums is very general, we have not put any restrictions on the choice of the functions g and h. To obtain a direct sum result in this very general setting, we will have to restrict the inputs of g-h-direct sums to certain subsets.

Definition 3.3.29 (collapsing sets)**.** Let $g\colon \{0,1\}^n \longrightarrow \{0,1\}$ and $h\colon \{0,1\}^k \longrightarrow \{0,1\}$ be functions. The set $S \subseteq \{0,1\}^k$ is called collapsing for g and h, if the following holds for all $i \in \{1,\ldots,n\}$ and all $a \in \{0,1\}^k$: If $x_j \in S$ for all $j \in \{1,\ldots,n\}$ then

$$g(\,h(x_1),h(x_2),\ldots,h(x_{i-1}),h(a),h(x_{i+1}),h(x_{i+2}),\ldots,h(x_n)\,) = h(a)\;.$$

Using the notion of collapsing sets, Bar-Yossef *et al.* have shown the following direct sum lower bound for the conditional information complexity of g-h-direct sums.

Theorem 3.3.30 (Bar-Yossef *et al.* [13])**.** *Suppose that* $f\colon \left(\{0,1\}^k\right)^n \longrightarrow \{0,1\}$ *is a* g-h*-direct sum and that* $\mathcal{D}$ *is a finite set. Let* $X = (X_i,\ldots,X_k) \in \{0,1\}^k$ *and* $D \in \mathcal{D}$ *be random variables such that the random variables* X_i *for* $i \in \{1,\ldots,k\}$ *are conditionally independent given* D *and* $\mathrm{supp}(X)$ *is collapsing for* g *and* h*. If* $Y = (Y_1,\ldots,Y_n)$ *and* $E = (E_1,\ldots,E_n)$ *are random variables such that* (Y_i,E_i) *is an independent copy of* (X,D) *for* $i = 1,\ldots,n$ *then*

$$\mathrm{IC}_\epsilon^{\mathrm{NIH}}(f;Y|E) \geq n \cdot \mathrm{IC}_\epsilon^{\mathrm{NIH}}(h;X|D)\;.$$

Proof. Let P be a randomized ϵ-error protocol for f and let $T(Y)$ denote the transcript of P for the input Y. The assumptions of the theorem imply that the random variables Y_i for $i \in \{1,\ldots,n\}$ are conditionally independent given E. Hence, by the superadditivity of conditional mutual information for conditionally independent random variables (Cor. 2.2.28), we obtain

$$\begin{aligned}
\mathrm{icost}(P;Y|E) &= \mathrm{I}(T(Y):Y|E) && (3.100)\\
&\geq \sum_{i=1}^{n} \mathrm{I}(T(Y):Y_i|E) && (3.101)\\
&= \sum_{i=1}^{n} \sum_{e} \Pr\{E_{-i}=e\}\,\mathrm{I}(T(Y):Y_i|E_i,E_{-i}=e)\;. && (3.102)
\end{aligned}$$

For the proof of the theorem it is sufficient to show that for every i and every e there is a randomized ϵ-error protocol $P_{i,e}$ for the function h such that

$$\mathrm{I}(T(Y):Y_i|E_i,E_{-i}=e) = \mathrm{icost}(P_{i,e};X|D) \qquad (3.103)$$

since this would immediately imply the claim of the theorem:

$$\operatorname{icost}(P;Y|E) \geq \sum_{i=1}^{n} \sum_{e} \Pr\{E_{-i}=e\} \operatorname{icost}(P_{i,e};X|D) \tag{3.104}$$

$$\geq \sum_{i=1}^{n} \sum_{e} \Pr\{E_{-i}=e\} \operatorname{IC}_{\epsilon}(h;X|D) \tag{3.105}$$

$$= n \cdot \operatorname{IC}_{\epsilon}(h;X|D) \,. \tag{3.106}$$

The protocol $P_{i,e}$ uses the random inputs of P and an additional random input R such that $R \sim (Y_{-i}|E_{-i}=e)$. Notice that under the condition $E_{-i}=e$ for each $j \neq i$ all but one coordinate of Y_j is fixed to the constant 0, hence the random input of each player may also contain constant bits. The pth player sees the coordinates of R that correspond to the coordinates of Y_{-i} seen by him. For an input $a \in \{0,1\}^k$ the protocol $P_{i,e}$ simulates P on the input $y \in \left(\{0,1\}^k\right)^n$ such that $y_i = a$ and $y_{-i} = R$. Here it is important to observe that the random choice of $y_{-i} = R$ can be carried out without any additional communication: For each $j \neq i$ the coordinates of Y_j are conditionally independent given that $E_{-i} = e$, therefore $y_{j,p}$ can be chosen by the pth player independently of the other players. Now, since $\operatorname{supp}(Y_j) = \operatorname{supp}(X)$ is collapsing for g and h and $R \sim (Y_{-i}|E_{-i}=e)$, we have $\Pr\{P_{i,e}(a) \neq h(a)\} \leq \epsilon$ by the error bound of P, thus $P_{i,e}$ is an ϵ-error protocol for the function h. Let $T_{i,e}(a)$ denote the transcript of $P_{i,e}$ for the input a. Clearly, by the construction of $P_{i,e}$, we have

$$(T_{i,e}(X), X, D) \sim (T(Y), Y_i, E_i|E_{-i}=e) \tag{3.107}$$

and therefore

$$\operatorname{icost}(P_{i,e};X|D) = \operatorname{I}(T_{i,e}(X) : X|D) = \operatorname{I}(T(Y) : Y_i|E_i, E_{-i}=e) \tag{3.108}$$

which completes the proof of the theorem. □

Next we consider the application of Theorem 3.3.30 to promise problems. Although the theorem has only been applied to promise problems in the literature [13, 25, 45, 51], surprisingly, the conditions under which this result is applicable to promise problems have not been stated explicitly in these publications.

Corollary 3.3.31. *Suppose that* $f\colon \left(\{0,1\}^k\right)^n \longrightarrow \{0,1\}$ *is a* g*-*h*-direct sum and that* $\mathcal{D}$ *is a finite set. Let* $X = (X_i, \ldots, X_k) \in \{0,1\}^k$ *and* $D \in \mathcal{D}$ *be random variables such that the random variables* X_i *for* $i \in \{1,\ldots,k\}$ *are conditionally independent given* D *and* $\operatorname{supp}(X)$ *is collapsing for* g *and* h*. Let* $S \subseteq \{0,1\}^k$ *be a set such that* $\operatorname{supp}(X) \subseteq S$ *and define*

$$T = \bigcup_{i=1}^{n} \left\{ (x_1,\ldots,x_n) \in \left(\{0,1\}^k\right)^n : x_i \in S \,,\, x_j \in \operatorname{supp}(X) \text{ if } j \neq i \right\} \,.$$

If $Y = (Y_1,\ldots,Y_n)$ *and* $E = (E_1,\ldots,E_n)$ *are random variables such that* (Y_i, E_i) *is an independent copy of* (X, D) *for* $i \in \{1,\ldots,n\}$*, then*

$$\operatorname{IC}_{\epsilon}^{\mathrm{NIH}}(f|_T;Y|E) \geq n \cdot \operatorname{IC}_{\epsilon}^{\mathrm{NIH}}(h|_S;X|D) \,.$$

Proof. The proof uses the same line of arguments as the proof of Theorem 3.3.30. Here we just need to observe that the protocols $P_{i,e}$ (compare with the proof of Theorem 3.3.30) are randomized ϵ-error protocols for the promise problem $h|_S$. To this end recall that the protocol $P_{i,e}$ on the input a simulates the protocol P for $f|_T$ on the input $y \in \left(\{0,1\}^k\right)^n$ such that $y_i = a$ and $y_j \in \mathrm{supp}(X)$ if $j \neq i$. Then, by the definition of T and the fact that $\mathrm{supp}(X)$ is collapsing for g and h, the protocol $P_{i,e}$ computes the function $h(a)$ for all $a \in S$. The output of $P_{i,e}$ for inputs $a \notin S$ is undefined. The condition $\mathrm{supp}(X) \subseteq S$ ensures that $h|_S$ is well-defined on the set $\mathrm{supp}(X)$ and that $f|_T$ is well defined on $\mathrm{supp}(Y)$, hence the information cost of $P_{i,e}$ with respect to X and the information cost of P with respect to Y is well-defined. □

3.3.5 The Disjointness Function

In this section we will apply the direct sum approach of Bar-Yossef *et al.* from the last section and our lower bound on the conditional information complexity of AND_k from Section 3.3.3 to a well known problem in communication complexity theory, the so-called disjointness function.

Definition 3.3.32 (k-party disjointness function)**.** Let $x_i \in \{0,1\}^n$ for $i \in \{1,\dots,k\}$. Then the k-party disjointness function $\mathrm{DISJ}_{k,n}$ is defined by

$$\mathrm{DISJ}_{k,n}(x_1,\dots,x_k) = \bigvee_{j=1}^{n} \bigwedge_{i=1}^{k} x_{i,j} \ .$$

Note that often (for example in [56]) the disjointness function is defined as the complement of the function $\mathrm{DISJ}_{k,n}$, but the definition according to Def. 3.3.32 seems to be more popular in the context of direct sum problems (see [51, 45, 25, 13]). Clearly, the k-party disjointness function is an OR_n-AND_k-direct sum where OR_n denotes the disjunction of n bits. Therefore we can obtain a lower bound on the conditional information complexity of $\mathrm{DISJ}_{k,n}$ by using Theorem 3.3.30. More precisely, we will use Corollary 3.3.31 to prove a lower bound on the information complexity of a promise variant of $\mathrm{DISJ}_{k,n}$ that is related to the promise problem $\mathrm{AND}_k^{\mathrm{unique}}$, the so-called *disjointness function with the unique intersection promise* which will turn out to be useful in applications (see Sect. 4.4.3).

Definition 3.3.33 (Unique intersection promise)**.** For the $\mathrm{DISJ}_{k,n}$ function with the *unique intersection promise*, $\mathrm{DISJ}_{k,n}^{\mathrm{unique}}$ for short, it is promised that either

(i) for all $j \in \{1,\dots,n\}$ there is at most one $i \in \{1,\dots,k\}$ such that $x_{i,j} = 1$, or

(ii) there is exactly one $j^* \in \{1,\dots,n\}$ such that $x_{i,j^*} = 1$ for all $i \in \{1,\dots,k\}$ and for all $j \in \{1,\dots,n\} - \{j^*\}$ there is at most one $i \in \{1,\dots,k\}$ such that $x_{i,j} = 1$.

In our lower bound on the information complexity of $\mathrm{DISJ}_{k,n}^{\mathrm{unique}}$ we will use the following joint distribution of the input Y and the variable E on which we condition.

Definition 3.3.34. Suppose that $Z = (Z_1,\dots,Z_k) \in \{0,1\}^k$ and $D \in \{1,\dots,k\}$ are random variables such that their joint distribution has the following properties: The random variable D is uniformly distributed in the set $\{1,\dots,k\}$ and for all $i \in \{1,\dots,k\}$ the conditional distribution of Z given D satisfies $\Pr\{Z_i{=}0|D \neq i\} = 1$ and $\Pr\{Z_i{=}0|D{=}i\} = \Pr\{Z_i{=}1|D{=}i\} = \frac{1}{2}$. Then define random variables $Y = (Y_1,\dots,Y_n)$ and $E = (E_1,\dots,E_n)$ such that (Y_i, E_i) is an independent copy of (X, D) for all $i \in \{1,\dots,n\}$.

The joint distribution of Z and D in the definition above is identical to the distribution that is used in the lower bound on the conditional information complexity of $\mathrm{AND}_k^{\mathrm{unique}}$. The random variables Z_i for $i \in \{1, \ldots, k\}$ are conditionally independent given D and $\mathrm{supp}(Z)$ is collapsing for OR_n and AND_k. Every element from $\mathrm{supp}(Z)$ satisfies the promise of $\mathrm{AND}_k^{\mathrm{unique}}$ (Def. 3.3.5) and if for any $i \in \{1, \ldots, n\}$ the coordinate Y_i of $Y = (Y_1, \ldots, Y_n)$ is replaced by an input that honors the promise of $\mathrm{AND}_k^{\mathrm{unique}}$ then the resulting input also honors the promise of $\mathrm{DISJ}_{k,n}^{\mathrm{unique}}$. Therefore a lower bound on the information complexity of $\mathrm{DISJ}_{k,n}^{\mathrm{unique}}$ with respect to Y and E follows immediately from Corollary 3.3.31 and the lower bound on the information complexity of $\mathrm{AND}_k^{\mathrm{unique}}$ in Theorem 3.3.7.

Corollary 3.3.35. *Suppose that the random variables (Y, E) are distributed according to Definition 3.3.34 and let $0 \leq \epsilon < \frac{3}{10}\left(1 - \sqrt{\frac{1}{2}\log\frac{4}{3}}\right) \approx 0.163$ be a constant. Then there is a constant $c(\epsilon) > 0$ that only depends on ϵ such that*

$$\mathrm{IC}_\epsilon^{\mathrm{NIH}}(\mathrm{DISJ}_{k,n}^{\mathrm{unique}}; Y|E) \geq \frac{c(\epsilon)n}{k} \ .$$

Note that lower bounds on the *communication complexity* of $\mathrm{DISJ}_{k,n}^{\mathrm{unique}}$ by Theorem 3.2.5 for errors that are larger than 0.163 are easily obtained from this result by using probability amplification.

Related Work

The communication complexity of the two-player disjointness function has been studied extensively in communication complexity theory. We refer the reader to the book by Kushilevitz and Nisan [56] and the references therein for details. Here we only focus on recent results on the k-party communication complexity of the disjointness function with the unique intersection promise in the NIH model, and especially on lower bounds that use information complexity arguments. Alon, Matias, and Szegedy [3] introduced the unique intersection promise for the disjointness function to prove lower bounds on the space complexity of algorithms for the frequency moments of data streams (see Section 4.4). They proved an $\Omega(n/k^4)$ lower bound on the randomized communication complexity of $\mathrm{DISJ}_{k,n}^{\mathrm{unique}}$ for randomized protocols with a constant error. Bar-Yossef, Jayram, Kumar, and Sivakumar [13] introduced the direct sum paradigm for the conditional information complexity of the disjointness function that is described in Section 3.3.4 and improved the lower bound to $\Omega(n/k^2)$. Building on the direct sum paradigm of Bar-Yossef *et al.*, the lower bound was improved to $\Omega(n/(k \log k))$ by Chakrabarti, Khot, and Sun [25]. Their improvement is due to an improved analysis of the conditional information complexity of the AND_k function (see Sect. 3.3.3). They also described a one-way protocol with the communication cost $O(n/k + \log n)$ for the k-party disjointness function with the unique intersection promise and proved an optimal $\Omega(n/k)$ lower bound for one-way protocols. We already mentioned that the conditions under which Theorem 3.3.30 is applicable to promise problems are not stated explicitly in [13]. Consequently, in the application to the promise problem $\mathrm{DISJ}_{k,n}^{\mathrm{unique}}$ it is not stated explicitly that a lower bound for a promise variant of AND_k is needed. We therefore note that all results [13, 25, 45, 51] which prove lower bounds on the information complexity of $\mathrm{DISJ}_{k,n}^{\mathrm{unique}}$ by using Theorem 3.3.30 actually use lower bounds on the information complexity of the promise problem $\mathrm{AND}_k^{\mathrm{unique}}$ although they only claim to show lower bounds on the information complexity of AND_k. Chakrabarti *et al.* [25] mention that the only property that is needed in

their proof (and the other results) is the fact that $\mathrm{AND}_k(\vec{0}) \neq \mathrm{AND}_k(\vec{1})$. This property is shared by AND_k and the promise problem $\mathrm{AND}_k^{\mathrm{unique}}$.

3.4 The NOF Information Complexity of Pointer Jumping

In this section we will apply information complexity to prove lower bounds on the communication complexity of functions in the number on the forehead model (see Sect. 3.1.5). In this model we are less successful than in the NIH model. We will only obtain lower bounds for one-way protocols that have additional artificial restrictions. First, we will discuss the new difficulties that arise in the NOF model.

3.4.1 What is so Difficult About the NOF Model?

The main difficulty in the proof of lower bounds on the communication complexity of functions in the NOF model lies in the large amount of information that is shared by the players. In a k-party NOF protocol each player sees almost all inputs. Since the fraction of the inputs that is seen by each player increases in k, NOF protocols get more powerful as the number of players increases. The strongest lower bounds that are currently known in the NOF model only work for $o(\log n)$ players where n denotes the length of the input. Proving a superpolylogarithmic lower bound on the communication complexity of a function for a polylogarithmic number of players in the NOF model would be a major breakthrough that could solve some long-standing open problems in circuit complexity (see [56, 5]).

The immediate consequences of the large amount of shared information for the information complexity approach are the strong dependencies between the messages of the players that are caused by shared information. Even for simultaneous message protocols, where the players do not interact at all, in the NOF model the messages of the players are not independent since for each pair of players there are input variables that are seen by both players. In contrast, the messages of a simultaneous message protocol in the NIH model are independent if the inputs of the protocol are independent since here the message of the ith player only depends on the ith input.

For NIH protocols our lower bounds used the fact that the inputs of the players are conditionally independent given the transcript of the protocol. Our observations on the combinatorial structure of NOF protocols in Section 3.1.5 reveal that we do not have similar properties in the NOF model. In fact, here we may introduce strong dependencies between the inputs of the players if we condition on the transcript of the protocol. Independence and conditional independence are our strongest tools for the proof of lower bounds in the NIH model. Apparently, in the NOF model these tools are not immediately applicable.

The following example illustrates the problems that are caused by shared information: Consider the following randomized k-party NOF protocol for the function $f(X_1, \ldots, X_k)$ where $X_1, \ldots, X_k$ are random variables and the random variable $R = (R_1, \ldots, R_k)$ is the random input of the randomized protocol: The random input R_3 is seen by the first and second player and the input X_2 is seen by the first player. The first player encrypts X_2 using R_3 as a one-time-pad (see [66]) and writes the result to the blackboard. Then the second player decrypts the transcript using the shared one-time-pad R_3 to obtain the input X_2, the only input that is not seen by him, and computes the output $f(X_1, \ldots, X_k)$. Clearly, this protocol computes $f(X_1, \ldots, X_k)$, but the mutual information of the transcript and the inputs is zero. This problem can be solved by conditioning on R, but treating the random inputs

of a randomized protocol different than the inputs for the computed function introduces new problems. It complicates the use of direct sum arguments like Theorem 3.3.30 that depend on the fact that the inputs and the random inputs of a randomized protocol are "interchangeable" such that inputs of the function can be interpreted as random inputs of a subfunction.

3.4.2 The Pointer Jumping Function $\mathrm{PJ}_{k,n}$

In the following we will consider the one-way NOF information complexity of a pointer jumping function that is defined as follows:

Definition 3.4.1 (Pointer jumping function)**.** Let $f_1, \ldots, f_k$ be functions with the domain and range $\{1, \ldots, n\}$. Then the k-party pointer jumping function $\mathrm{PJ}_{k,n}$ is defined as follows:

$$\mathrm{PJ}_{k,n}(f_1, \ldots, f_k) = (f_k \circ f_{k-1} \circ \cdots \circ f_1)(1) \ .$$

Note that the value of $\mathrm{PJ}_{k,n}(f_1, \ldots, f_k)$ does not depend on $f_1(2), \ldots, f_1(n)$. This part of the input is redundant, it is only present for the sake of a more uniform notation. In addition to the notation from Definition 3.2.7, the following definition will be used throughout this section.

Definition 3.4.2 (Notation)**.** Let $f = (f_1, \ldots, f_k)$ be an input for $\mathrm{PJ}_{k,n}$. Then

- let $\widetilde{f_i} = f_i \circ f_{i-1} \circ \cdots \circ f_1$ denote the composition of first i functions from f, and
- let $f_{-(i,j)} = (f_1, \ldots, f_{i-1}, f_{j+1}, \ldots, f_k)$ denote the vector of all f_ℓ where $\ell < i$ or $\ell > j$.

An alternative definition of the function $\mathrm{PJ}_{k,n}$ as a graph problem explains the name "pointer jumping": We have a layered digraph with $k+1$ layers and n nodes in each layer. All nodes in the first k layers have exactly one outgoing edge, the nodes in the last layer have no outgoing edges. Let $v_{i,j}$ denote the jth node in the ith layer. The graph contains the edges from $v_{i,j}$ to $v_{i+1,f_i(j)}$ for all $i \in \{1, \ldots, k\}$ and $j \in \{1, \ldots, n\}$. Then f_i corresponds to the edges from layer i to layer $i+1$ and the value of the function $\mathrm{PJ}_{k,n}$ is the number of the unique node that is reached by chasing the edges from node $v_{1,1}$ to the last layer.

Several variants of pointer jumping functions have been defined in the literature. Section 3.4.4 gives references to some results about pointer jumping functions in various communication models. A common variation is, for example, a binary variant of $\mathrm{PJ}_{k,n}$ where the range of the last function f_k is replaced by $\{0, 1\}$. In this case the resulting function for $k = 2$ players is the index function IND_n that was introduced in Section 3.3.1 and the binary variant for $k \geq 3$ can be seen as a multi-party version of the index function.

3.4.3 Conservative, Myopic, and Collapsing One-Way Protocols for $\mathrm{PJ}_{k,n}$

Because of the difficulties that are described in Section 3.4.1, we will only be able to prove a lower bound on the information cost of a restricted subset of deterministic one-way NOF protocols for $\mathrm{PJ}_{k,n}$: We will consider protocols where each player "can see only one layer ahead", hence the message of the ith player only depends on the messages of the preceding players and the inputs $f_1, f_2, \ldots, f_{i-1}$ and f_{i+1}. We call protocols that respect our restriction *myopic protocols*. First, we will restate the restriction in information theoretical terms.

Definition 3.4.3 (Myopic protocol)**.** Let the random variables $X = (X_1, \ldots, X_k)$ be the input of a deterministic k-party one-way NOF protocol P and let $T(X) = (T_1, \ldots, T_k)$ be the transcript of P for the input X where T_i denotes the part of the transcript that was written by the ith player. Then P is called myopic, if for all $i \in \{1, \ldots, k\}$

$$\mathrm{I}(T_{1,i-2} : X_i | X_{-i}) = 0 \; .$$

Consider the function $\mathrm{PJ}_{k,n}$ for random inputs $F = (F_1, \ldots, F_k)$. If the mutual information of $T(F)$ and F is small then $T(F)$ can only convey limited information about each function F_i. The definition of myopic protocols is motivated by the intuition that the ith player can put the limited information in his message T_i to the best use if he can compute a good prediction of $\widetilde{F}_i(1)$ using the information at his disposal, namely the inputs F_{-i} and the messages $T_{1,i-1}$. Clearly, if he can compute $\widetilde{F}_i(1)$ exactly then he can immediately compute the output $\widetilde{F}_k(1)$ as a function of $\widetilde{F}_i(1)$ and F_{-i}. But even less than perfect information about $\widetilde{F}_i(1)$ is useful because it helps to avoid "wasting information" on random variables that do not have an effect on the value of $\mathrm{PJ}_{k,n}(F)$. For example, if the ith player knows that $\widetilde{F}_i(1) \neq j$ then he should not waste the limited information in his message T_i on the useless random variable $F_{i+1}(j)$. Therefore it seems to be useful to give as much information as possible to the next player in a one-way protocol since this helps the next player to put the limited information in his message to the best use and, intuitively, the amount of wasted information should be minimized by this strategy. Unfortunately, today it is known that this intuition is flawed. New results by Chakrabarti [24] and Brody and Chakrabarti [19] show that myopic protocols are suboptimal (see also the discussion of related work in Sect. 3.4.4).

Before we prove some basic properties of myopic protocols, we will describe classes of restricted one-way NOF protocols that have been considered by other researchers and we will discuss the significance of lower bounds on the communication complexity of functions for restricted classes of protocols: Damm, Jukna, and Sgall [32] investigated the communication cost of *conservative* one-way NOF protocols. In a conservative protocol the message of the ith player may only depend on the messages $T_{1,i-1}$ of the preceding players, the inputs $f_{i+1}, \ldots, f_k$ and the value of $\widetilde{f}_{i-1}(1)$. Hence the ith player knows the node that is reached by the first $i-1$ pointers, but he does not know the exact path by which the node is reached. The access to the pointers "ahead" of the ith player is not restricted. The situation is inverted for *collapsing* one-way NOF protocols which were introduced by Brody and Chakrabarti [19]. Here the message of the ith player may, in addition to the messages $T_{1,i-1}$ of the preceding players, depend on the inputs $f_1, \ldots, f_{i-1}$ and on the function $f_k \circ f_{k-1} \circ \cdots \circ f_{i+1}$. Here the ith player's access to the preceding pointers is not restricted. But for the pointers that lie ahead he only knows which node in the final layer is reached from each node of layer $i+1$, but not the exact path by which the node in the final layer is reached.

Restricted classes of one-way NOF protocols can be considerably weaker than unrestricted one-way NOF protocols. The recent results by Chakrabarti [24] and by Brody and Chakrabarti [19] show that conservative, myopic, and collapsing protocols for $\mathrm{PJ}_{k,n}$ are suboptimal. Nevertheless lower bounds in these models are useful. The lower bounds can guide the design of efficient protocols for $\mathrm{PJ}_{k,n}$ since they rule out the efficiency of certain classes of protocols. For example, in an optimal protocol for $\mathrm{PJ}_{k,n}$ the message of the ith player *must* depend on the inputs $f_{i+1}, \ldots, f_k$ for at least one $i \in \{1, \ldots, k-1\}$. But lower bounds for restricted classes of protocols can also contribute to our understanding of the information complexity for unrestricted protocols: They rule out the applicability of some proof strate-

gies, for example the wrong intuition about myopic protocols that is described above, and thereby guide our search for properties of the function $\mathrm{PJ}_{k,n}$ that can be used for the proof of lower bounds for unrestricted protocols. The exploration of restricted models has proved useful before, for example for branching programs (see [75] for details).

Now we will prove some properties of myopic protocols that will be useful in the following.

Proposition 3.4.4. *Let P be a deterministic k-party one-way NOF protocol for $\mathrm{PJ}_{k,n}$ and let $T(F) = (T_1, \ldots, T_k)$ denote the transcript of P for uniformly distributed random inputs $F = (F_1, \ldots, F_k)$. If P is myopic then*

(i) *$F_{i+1}(1), \ldots, F_{i+1}(n)$ are conditionally independent given $(F_{-(i,i+1)}, T_{1,i-1})$,*

(ii) *F_i and F_{i+1} are conditionally independent given $(F_{-(i,i+1)}, T_{1,i-1})$, and*

(iii) *$\mathrm{H}(\widetilde{F}_{i+1}(1)|F_{-(i+1)}, T_{1,i-1}) = \log(n)$.*

Proof. First observe that F_{i+1} is independent of $(F_{-(i+1)}, T_{1,i-1})$ for all $i \in \{1, \ldots, k-1\}$ since, by the definition of myopic protocols and the fact that F_{i+1} and $F_{-(i+1)}$ are independent

$$\begin{aligned} 0 &= \mathrm{I}(T_{1,i-1} : F_{i+1}|F_{-(i+1)}) && (3.109)\\ &= \mathrm{H}(F_{i+1}|F_{-(i+1)}) - \mathrm{H}(F_{i+1}|T_{1,i-1}, F_{-(i+1)}) && (3.110)\\ &= \mathrm{H}(F_{i+1}) - \mathrm{H}(F_{i+1}|T_{1,i-1}, F_{-(i+1)})\ . && (3.111)\end{aligned}$$

This implies that $\mathrm{H}(F_{i+1}) = \mathrm{H}(F_{i+1}|T_{1,i-1}, F_{-(i+1)})$ and F_{i+1} and $(F_{-(i+1)}, T_{1,i-1})$ are independent by Proposition 2.2.11. Then claim (i) of the proposition follows immediately from the independence of F_{i+1} and $(F_{-(i+1)}, T_{1,i-1})$ since the independent random variables $F_{i+1}(1), \ldots, F_{i+1}(n)$ remain independent if we condition on a random variable that is independent of these variables. For the proof of claim (ii), by Proposition 2.2.13, it suffices to show that $\mathrm{H}(F_{i+1}|T_{1,i-1}, F_{-(i,i+1)}) = \mathrm{H}(F_{i+1}|T_{1,i-1}, F_{-(i+1)})$. This also follows immediately from the independence of F_{i+1} and $(F_{-(i+1)}, T_{1,i-1})$ since it implies the independence of F_{i+1} and $(F_{-(i,i+1)}, T_{1,i-1})$ and we have

$$\mathrm{H}(F_{i+1}|T_{1,i-1}, F_{-(i+1)}) = \mathrm{H}(F_{i+1}) = \mathrm{H}(F_{i+1}|T_{1,i-1}, F_{-(i,i+1)})\ . \tag{3.112}$$

Claim (iii) follows from the independence of F_{i+1} and $(F_{-(i+1)}, T_{1,i-1})$ and the fact that $\widetilde{F}_i$ is a function of $F_{-(i+1)}$: By the definition of $\widetilde{F}_{i+1}$ and by Proposition 2.2.12, we have

$$\begin{aligned} \mathrm{H}(\widetilde{F}_{i+1}(1)|F_{-(i+1)}, T_{1,i-1}) &= \mathrm{H}(F_{i+1}(\widetilde{F}_i(1))|F_{-(i+1)}, T_{1,i-1}, \widetilde{F}_i(1)) && (3.113)\\ &= \sum_{p=1}^{n} \Pr\{\widetilde{F}_i(1)=p\} \cdot \mathrm{H}(F_{i+1}(p)|F_{-(i+1)}, T_{1,i-1}, \widetilde{F}_i(1)=p) && (3.114)\\ &= \sum_{p=1}^{n} \Pr\{\widetilde{F}_i(1)=p\} \cdot \mathrm{H}(F_{i+1}(p)) && (3.115)\\ &= \log(n)\ . && (3.116)\end{aligned}$$

In the second to the last line we used that $F_{i+1}(p)$ is independent of $(F_{-(i+1)}, T_{1,i-1}, \widetilde{F}_i(1))$ and in the last line we used that $\mathrm{H}(F_{i+1}(p)) = \log(n)$ for all $i \in \{1, \ldots, k\}$ and all $p \in \{1, \ldots, n\}$. □

3.4.4 The Information Cost of Myopic Protocols for $\mathrm{PJ}_{k,n}$

Bar-Yossef, Jayram, Kumar, and Sivakumar [12] were the first to apply the information complexity approach of Chakrabarti, Shi, Wirth, and Yao [26] to the NOF model. They translated a combinatorial proof method of Babai, Gál, Kimmel, and Lokam [6] for simultaneous message protocols into the domain of information theory. In the simultaneous message model the message of each player depends only on the inputs seen by the player, but not on the messages of the other players, hence the players do not interact directly. A natural next step are protocols where the interaction of the players is limited, for example one-way protocols. In the following we will prove a lower bound on the information cost of myopic one-way NOF protocols for $\mathrm{PJ}_{k,n}$, a first step into this direction.

Theorem 3.4.5. *Let P be a deterministic myopic k-party one-way NOF protocol that computes $\mathrm{PJ}_{k,n}$ with distributional error ϵ for uniformly distributed inputs $F = (F_1, \ldots, F_k)$ such that $\mathrm{cost}(P) < n/2$ and let $T(F) = (T_1, \ldots, T_k)$ denote the transcript of P for the input F. Then*

$$\mathrm{icost}(P; F) \geq \log(n) \cdot \left(2^{-(1+1/k)} n^{(1-\epsilon)/k} - 1\right) .$$

Theorem 3.4.5 was the first attempt to extend the information complexity approach to the one-way NOF model, albeit only for protocols with additional artificial information theoretical restrictions. Note that the lower bound for myopic protocols has been improved significantly by Chakrabarti [24] since the publication [43] of Theorem 3.4.5. More details on this can be found in the following section on related work.

Related Work

Several variants of pointer jumping problems have been defined in the literature, common to all of these variants is the basic problem of "chasing" edges in a given graph. The variants of the problem differ with respect to the topology of the underlying graph, the allocation of the edges to the players in communication protocols, and the result of the function. We refer the reader to the papers that are cited below for details. All pointer jumping problems are designed to capture the hardness of inherently sequential communication problems for protocols that obtain the inputs in the wrong order. The index function IND_n (see Sect. 3.3.1) belongs to this class of problems. In the following results the value n roughly measures the size of the input. The exact meaning of n depends on the concrete variant of pointer jumping.

The communication complexity of pointer jumping problems in Yao's two-player model of communication has bees studied thoroughly, for example by Nisan and Wigderson [62] and Ponzio, Radhakrishnan, and Venkatesh [63].

The proof method of Babai *et al.* [6] yields an $\Omega(n^{1/k})$ lower bound on the communication complexity of certain pointer jumping problems for simultaneous message protocols in the k-party NOF model [64].

The first result on the communication complexity of pointer jumping problems in the one-way multi-party NOF model is an unpublished result due to Wigderson who proved an $\Omega(n^{1/2})$ lower bound (the result was described by Babai, Hayes, and Kimmel [7]). For $k > 3$ players progress on the one-way communication complexity of pointer jumping in the NOF model has been slow, therefore restricted one-way models that allow simpler proofs of strong lower bounds have been considered by several researchers. Damm, Jukna, and Sgall proved an $\Omega(n/k^2)$ lower bound for up to $O(n^{1/3-\epsilon})$ players in the conservative model (see Sect. 3.4.3).

Gronemeier [43] proved an $\Omega(n^{(1-\epsilon)/k})$ lower bound for ϵ-error protocols in the myopic model (Theorem 3.4.5). Chakrabarti [24] significantly improved the bound for myopic protocols to $\Omega(n/k)$ and extended the result to protocols that mix conservative and myopic behavior of the players. An $n - O(\log n)$ lower bound for collapsing protocols was obtained by Brody and Chakrabarti [19]. An optimal $\Omega(n^{1/(k-1)})$ lower bound on the communication complexity of pointer jumping on n-ary trees of depth k for unrestricted one-way protocols in the k-party NOF model was proved recently by Viola and Wigderson [71].

The best currently known upper bound on the communication complexity of pointer jumping in the one-way NOF model is due to Brody and Chakrabarti [19]. They describe a protocol that has communication cost $O\left(n((k\log\log n)/\log(n))^{(k-2)/(k-1)}\right)$. For $k=3$ players this yields an upper bound of $O\left(n\sqrt{\log\log n/\log n}\right)$ ruling out the possibility of a linear lower bound for a constant number of players. Surprisingly, all players except for the first player in their protocol are collapsing. Taken together with the lower bound for collapsing protocols, this shows that the communication complexity of $\mathrm{PJ}_{k,n}$ for restricted protocols is very sensitive to minimal changes of the restriction.

Outline of the Proof

We will show that the message T_i of the ith player can not reveal much information about $\widetilde{F}_{i+1}(1)$ if the conditional entropy of $\widetilde{F}_i(1)$ given $(F_{-i}, T_{1,i-1})$ is large and the conditional mutual information of T_i and F_{i+1} given $(F_{-(i+1)}, T_{1,i-1})$ is small. This property is mainly due to the fact that for myopic protocols the variables $F_{i+1}(1),\ldots,F_{i+1}(n)$ are still independent after the first $i-1$ players have written their part of the transcript. Then, intuitively, the ith player needs to partition the mutual information of T_i and F_{i+1} among the variables $F_{i+1}(1),\ldots,F_{i+1}(n)$ whereas only one of the variables, namely $F_{i+1}(\widetilde{F}_i(1))$, contains information about $\widetilde{F}_{i+1}(1)$. Since the ith player cannot predict $\widetilde{F}_i(1)$ reliably, he needs to "waste a lot of information" to reveal at least a little information about $F_{i+1}(\widetilde{F}_i(1))$. The technical details of this argument are contained in Lemma 3.4.6. This property is used in Lemma 3.4.7 to prove a lower bound on the conditional entropy of $\widetilde{F}_k(1)$ given $(F_{-k}, T_{1,k-1})$ by induction. Finally, Theorem 3.4.5 is proved by applying Fano's inequality to our lower bound on the conditional entropy of $\widetilde{F}_k(1)$ given $(F_{-k}, T_{1,k-1})$.

Proof of the Lower Bound

The following lemma can be seen as a variant of Lemma 3.3.3 for the index function with weakened premises. Here we only use a lower bound on the entropy of the index Y instead of an upper bound on the value of $p_{\max}$ from Lemma 3.3.3. An upper bound on $p_{\max}$ is equivalent to a lower bound on the so-called *min-entropy* of the index Y. The min-entropy $\mathrm{H}_\infty(Y)$ of a random variable Y is defined by $\mathrm{H}_\infty(Y) = \min\{-\log(\Pr\{Y=y\}) : y \in \mathrm{supp}(Y)\}$. The min-entropy $\mathrm{H}_\infty(Y)$ is a lower bound on the Shannon entropy $\mathrm{H}(Y)$ according to Definition 2.2.1, but a random variable with a large Shannon entropy can have a small min-entropy. Hence, provided the entropy of the index Y is sufficiently large, the following lemma can still prove strong bounds in cases where Lemma 3.3.3 fails due to a small min-entropy of Y. Additionally, replacing the lower bound on the min-entropy of the index Y in Lemma 3.3.3 by the entropy will enable us to apply the lemma inductively for pointer jumping, the multi-party variant of the index function. Note that, contrary to the discussion above and the usage in Lemma 3.3.3,

in the following lemma the index is a function $P(Y)$ of the random variable Y. The reason for this slight change will become clear later on.

Lemma 3.4.6. *Let $X = (X_1, \ldots, X_n)$ be a random variable such that the random variables $X_1, \ldots, X_n$ are independent and $\mathrm{H}(X_p) \leq \log n$ for all $p \in \{1, \ldots, n\}$. Additionally, let Y and T be random variables such that X and T are jointly independent of Y and let $P\colon \mathrm{range}(Y) \longrightarrow \{1, \ldots, n\}$ be a function. If $\lceil \mathrm{I}(T : X|Y)/\log n \rceil \leq C < n/2$ then*

$$\mathrm{I}(T : X_{P(Y)}|Y) \leq \frac{\log(n-C) + 1 - \mathrm{H}(P(Y))}{\log(n-C) - \log(C)} \log n \ .$$

Proof. Clearly, $\mathrm{I}(T : X|Y) = \mathrm{I}(T : X)$ since X and T are jointly independent of Y. Then, by the superadditivity of mutual information for independent random variables (Prop. 2.2.27),

$$\mathrm{I}(T : X_1, \ldots, X_n|Y) = \mathrm{I}(T : X_1, \ldots, X_n) \geq \sum_{p=1}^{n} \mathrm{I}(T : X_p) \ . \tag{3.117}$$

Furthermore $\mathrm{I}(T : X_{P(Y)}|Y) = \mathrm{I}(T : X_{P(Y)}|Y, P(Y))$ by Proposition 2.2.12 since $P(Y)$ is a function of Y and $\mathrm{I}(T : X_{P(Y)}|Y, P(Y) = p) = \mathrm{I}(T : X_p)$ since X and T are jointly independent of Y. Therefore

$$\begin{aligned}
\mathrm{I}(T : X_{P(Y)}|Y) &= \mathrm{I}(T : X_{P(Y)}|Y, P(Y)) && (3.118)\\
&= \sum_{p=1}^{n} \Pr\{P(Y)=p\} \cdot \mathrm{I}(T : X_{P(Y)}|Y, P(Y) = p) && (3.119)\\
&= \sum_{p=1}^{n} \Pr\{P(Y)=p\} \cdot \mathrm{I}(T : X_p) \ . && (3.120)
\end{aligned}$$

Now assume w.l.o.g. that $\Pr\{P(Y)=1\} \geq \Pr\{P(Y)=2\} \geq \cdots \geq \Pr\{P(Y)=n\}$. Then the sum (3.120) is maximized if $\mathrm{I}(T : X_p)$ is as large as possible for small values of p. Since $\mathrm{I}(T : X_p) \leq \mathrm{H}(X_p) \leq \log n$ and $\sum_{p=1}^{n} \mathrm{I}(T : X_p) \leq \mathrm{I}(T : X_1, \ldots, X_n) \leq C \log n$, we obtain an upper bound on the value the sum (3.120) by assuming that $\mathrm{I}(T : X_p) = \log n$ if $p \leq C$ and $\mathrm{I}(T : X_p) = 0$ if $p > C$. Let Z be a random variable such that $Z = 1$ if $P(Y) \leq C$ and $Z = 0$ if $P(Y) > C$. Then, by using again that X and T are jointly independent of Y and that Z is a function of Y, the upper bound on the sum (3.120) can be expressed as follows:

$$\begin{aligned}
\mathrm{I}(T : X_{P(Y)}|Y) &= \mathrm{I}(T : X_{P(Y)}|Y, Z) && (3.121)\\
&= \Pr\{Z=1\}\, \mathrm{I}(T : X_{P(Y)}|Y, Z=1) + \Pr\{Z=0\}\, \mathrm{I}(T : X_{P(Y)}|Y, Z=0) && (3.122)\\
&\leq \Pr\{Z=1\} \cdot \log n \ . && (3.123)
\end{aligned}$$

We have $\mathrm{H}(P(Y)) = \mathrm{H}(P(Y), Z) = \mathrm{H}(Z) + \mathrm{H}(P(Y)|Z)$ since Z is also a function of $P(Y)$ and, by using that $\mathrm{H}(Z) \leq 1$, we obtain

$$\mathrm{H}(P(Y)|Z) = \mathrm{H}(P(Y)) - \mathrm{H}(Z) \geq \mathrm{H}(P(Y)) - 1 \ . \tag{3.124}$$

On the other hand, by Proposition 2.2.4 and the fact that $P(Y) \in \{1, \ldots, C\}$ given that $Z = 1$ whereas $P(Y) \in \{C+1, \ldots, n\}$ under the condition that $Z = 0$, we obtain

$$\begin{aligned}
\mathrm{H}(P(Y)|Z) &= \Pr\{Z=1\} \cdot \mathrm{H}(P(Y)|Z=1) + (1 - \Pr\{Z=1\}) \cdot \mathrm{H}(P(Y)|Z=0) && (3.125)\\
&\leq \Pr\{Z=1\} \cdot \log(C) + (1 - \Pr\{Z=1\}) \cdot \log(n-C) \ . && (3.126)
\end{aligned}$$

By combining the two inequalities for $\mathrm{H}(P(Y)|Z)$ we get

$$\Pr\{Z=1\}\,(\log(n-C)-\log(C)) \leq \log(n-C)+1-\mathrm{H}(P(Y)) \tag{3.127}$$

and since the premise $C < n/2$ implies that $\log(n-C)-\log(C) > 0$ we finally obtain

$$\Pr\{Z=1\} \leq \frac{\log(n-C)+1-\mathrm{H}(P(Y))}{\log(n-C)-\log(C)}\ . \tag{3.128}$$

By substituting this into our estimate of $\mathrm{I}(T : X_{P(Y)}|Y)$, we obtain the claim of the lemma:

$$\mathrm{I}(T : X_{P(Y)}|Y) \leq \Pr\{Z=1\}\cdot\log n \tag{3.129}$$

$$\leq \frac{\log(n-C)+1-\mathrm{H}(P(Y))}{\log(n-C)-\log(C)}\log n\ . \tag{3.130}$$

□

Consider the situation of the ith player in a myopic protocol for $\mathrm{PJ}_{k,n}$: The ith player would like to provide as much information as possible about $F_{i+1}(\widetilde{F}_i(1))$ to player $i+1$. But if the entropy of $\widetilde{F}_i(1)$ is large and the amount of information that can be provided by the ith player is limited then, by Lemma 3.4.6, the entropy of $\widetilde{F}_{i+1}(1)$ will also be large and player $i+1$ will be in a similar position as player i. This argument can be extended inductively to several players. The following lemma contains the technical details.

Lemma 3.4.7. *Let P be a deterministic k-party one-way NOF protocol for $\mathrm{PJ}_{k,n}$ and let $T(F) = (T_1, \ldots, T_k)$ denote the transcript of P for uniformly distributed random inputs $F = (F_1, \ldots, F_k)$. Suppose that P is myopic, that $\lceil \mathrm{cost}(P)/\log n\rceil < n/2$, and that $\mathrm{I}(T_i : F_{i+1}|F_{-(i+1)}, T_{1,i-1})/\log(n) \leq U$ for all $i < k$. Then for all $i \leq k$ we have*

$$\mathrm{H}(\widetilde{F}_i(1)|F_{-i}, T_{1,i-1}) \geq \log n - i - i\log(U+1)\ .$$

Proof. For brevity, let $A_i = \mathrm{H}(\widetilde{F}_i(1)|F_{-i}, T_{1,i-1})$. We will first show a recurrence relation for A_i: Clearly, we have $A_1 = \log n$ since the empty transcript at the beginning of the protocol does not contain any information about F_1. By the definition of mutual information, we have

$$\mathrm{I}(T_i : \widetilde{F}_{i+1}(1)|F_{-(i+1)}, T_{1,i-1}) = \mathrm{H}(\widetilde{F}_{i+1}(1)|F_{-(i+1)}, T_{1,i-1}) - \mathrm{H}(\widetilde{F}_{i+1}(1)|F_{-(i+1)}, T_{1,i}) \tag{3.131}$$

$$= \mathrm{H}(\widetilde{F}_{i+1}(1)|F_{-(i+1)}, T_{1,i-1}) - A_{i+1}\ . \tag{3.132}$$

By claim (iii) of Proposition 3.4.4, we have $\mathrm{H}(\widetilde{F}_{i+1}(1)|F_{-(i+1)}, T_{1,i-1}) = \log(n)$, therefore we obtain

$$A_{i+1} = \mathrm{H}(\widetilde{F}_{i+1}(1)|F_{-(i+1)}, T_{1,i-1}) - \mathrm{I}(T_i : \widetilde{F}_{i+1}(1)|F_{-(i+1)}, T_{1,i-1}) \tag{3.133}$$

$$= \log(n) - \mathrm{I}(T_i : \widetilde{F}_{i+1}(1)|F_{-(i+1)}, T_{1,i-1})\ . \tag{3.134}$$

Let $E_i(f,t)$ denote the event that $F_{-(i,i+1)} = f$ and $T_{1,i-1} = t$, let

$$B_i(f,t) = \mathrm{I}(T_i : \widetilde{F}_{i+1}(1)|F_{-(i+1)}, T_{1,i-1}, E_i(f,t)) \tag{3.135}$$

and let

$$C_i(f,t) = \lceil \mathrm{I}(T_i : F_{i+1}|F_{-(i+1)}, T_{1,i-1}, E_i(f,t))/\log(n)\rceil\ . \tag{3.136}$$

Then, by our previous observations, we have

$$A_{i+1} = \log(n) - \sum_{f,t} \Pr\{E_i(f,t)\} \cdot B_i(f,t) \; . \tag{3.137}$$

Assume that the event $E_i(f,t)$ happened. Then the random variables $T_{1,i-1}$ and $F_{-(i,i+1)}$ are fixed to constants. Hence $\widetilde{F}_i(1)$ only depends on F_i and the message T_i of player i only depends on F_{i+1} since player i does not see F_i. By claim (ii) of Proposition 3.4.4, F_i and F_{i+1} are conditionally independent given $E_i(f,t)$, therefore F_{i+1} and T_i are jointly independent of F_i and $\widetilde{F}_i(1)$ under this condition. Finally, we have $\widetilde{F}_{i+1}(1) = F_{i+1}(\widetilde{F}_i(1))$ and, by claim (i) of Proposition 3.4.4, the random variables $F_{i+1}(1), \ldots, F_{i+1}(n)$ are independent given $E_i(f,t)$. Therefore we can apply Lemma 3.4.6 for the parameters $X_p = F_{i+1}(p)$, $Y = F_i$, $P(Y) = \widetilde{F}_i(1)$, $T = T_i$, and $C = C_i(f,t)$ to obtain an upper bound on $B_i(f,t)$. Note that $\mathrm{H}(F_{i+1}(p)) \leq \log(n)$ for all $p \in \{1, \ldots, n\}$ and that $\lceil C_i(f,m) \rceil \leq \mathrm{cost}(P)/\log(n) < n/2$ if n is sufficiently large. This is due to the fact that even under the condition $E_i(f,t)$ the entropy of the transcript is a lower bound on the communication cost of P by the same argument as in Theorem 3.2.10. Hence all requirements for the application of Lemma 3.4.6 are met and we obtain

$$B_i(f,t) \leq \frac{\log(n - C_i(f,t)) + 1 - \mathrm{H}(\widetilde{F}_i(1)|E_i(f,t))}{\log(n - C_i(f,t)) - \log(C_i(f,t))} \log n \tag{3.138}$$

and therefore

$$\log(n) - B_i(f,t) \geq \left(1 - \frac{\log(n - C_i(f,t)) + 1 - \mathrm{H}(\widetilde{F}_i(1)|E_i(f,t))}{\log(n - C_i(f,t)) - \log(C_i(f,t))}\right) \log(n) \tag{3.139}$$

$$= \frac{\mathrm{H}(\widetilde{F}_i(1)|E_i(f,t)) - \log(C_i(f,t)) - 1}{\log(n - C_i(f,t)) - \log(C_i(f,t))} \log(n) \tag{3.140}$$

$$\geq \mathrm{H}(\widetilde{F}_i(1)|E_i(f,t)) - \log(C_i(f,t)) - 1 \; . \tag{3.141}$$

For the last line we used that $0 < \log(n - C_i(f,t)) - \log(C_i(f,t)) \leq \log(n)$. Then, by plugging this result into (3.137), we obtain

$$A_{i+1} = \sum_{f,t} \Pr\{E_i(f,t)\} \cdot (\log(n) - B_i(f,t)) \tag{3.142}$$

$$\geq \sum_{f,t} \Pr\{E_i(f,t)\} \cdot \Big(\mathrm{H}(\widetilde{F}_i(1)|E_i(f,t)) - \log(C_i(f,t)) - 1\Big) \tag{3.143}$$

$$= \mathrm{H}(\widetilde{F}_i(1)|F_{-(i,i+1)}, T_{1,i-1}) - 1 - \sum_{f,t} \Pr\{E_i(f,t)\} \cdot \log(C_i(f,t)) \; . \tag{3.144}$$

Since conditioning reduces entropy (Prop. 2.2.11), we have

$$\mathrm{H}(\widetilde{F}_i(1)|F_{-(i,i+1)}, T_{1,i-1}) \geq \mathrm{H}(\widetilde{F}_i(1)|F_{-i}, T_{1,i-1}) = A_i \tag{3.145}$$

and, by Jensen's inequality (Theorem A.2.1), we have

$$\sum_{f,t} \Pr\{E_i(f,t)\} \cdot \log(C_i(f,t)) \leq \log\left(\sum_{f,t} \Pr\{E_i(f,t)\} \cdot C_i(f,t)\right) \tag{3.146}$$

$$\leq \log\left(\mathrm{I}(T_i : F_{i+1}|F_{-(i+1)}, T_{1,i-1})/\log(n) + 1\right) \tag{3.147}$$

$$\leq \log\left(U + 1\right) \; . \tag{3.148}$$

We used the fact that $C_i(f,t) \leq \mathrm{I}(T_i : F_{i+1}|F_{-(i+1)}, T_{1,i-1}, E_i(f,t))/\log(n) + 1$ in the second inequality. Finally, we obtain the recurrence relation

$$A_{i+1} \geq A_i - 1 - \log(U+1) \tag{3.149}$$

and the claim of the lemma follows immediately from this recurrence relation and the base case $A_1 = \log(n)$ which was mentioned earlier in the proof. □

Using the technical preparations in the preceding Lemmas, we can prove Theorem 3.4.5.

Proof of Theorem 3.4.5. The kth player of the protocol P uses F_{-k} and $T_{1,k-1}$ to predict $\widetilde{F}_k(1)$ with an error probability of at most ϵ. Then, by Fano's inequality (Thm. 2.2.29), we have

$$\mathrm{H}(\widetilde{F}_k(1)|F_{-k}, T_{1,k-1}) \leq h_2(\epsilon) + \epsilon\log(n-1) \leq 1 + \epsilon\log(n)\ . \tag{3.150}$$

Let $U = \mathrm{icost}(P;F)/\log(n) = \max\{\mathrm{I}(T_i : F_{i+1}|F_{-(i+1)}, T_{1,i-1})/\log(n)\ :\ 1 \leq i < k\}$. Then for all i such that $1 \leq i < k$ we have $\mathrm{I}(T_i : F_{i+1}|F_{-(i+1)}, T_{1,i-1})/\log(n) \leq U$ and, by Lemma 3.4.7, we obtain

$$\mathrm{H}(\widetilde{F}_k(1)|F_{-k}, T_{1,k-1}) \geq \log(n) - k - k\log(U+1)\ . \tag{3.151}$$

Combining (3.150) and (3.151) and solving for U yields

$$U \geq 2^{-(1+1/k)} n^{(1-\epsilon)/k} - 1\ . \tag{3.152}$$

The claim of the theorem follows immediately from this, concluding the proof. □

Chapter 4

Algorithms

In this section we will describe data stream algorithms for basic problems under extreme space restrictions. First we will consider the problem of counting events approximately using only $O(\log \log n)$ bits of memory for n events. Unlike the previously known algorithms for this problem, our algorithm provides good approximations of the actual number of events after every event with an adjustable error probability, even for an infinite number of events. Counting the number of data stream elements is a basic operation of many data stream algorithms, hence this algorithm is a useful building block for data stream algorithms. Then we will use this building block to design an algorithm that samples an element approximately uniformly at random from a data stream using only $O(\log \log n)$ bits of memory in addition to the sample. Sampling uniformly at random from a data stream is also a basic operation that is needed in many randomized data stream algorithms. Finally, we will apply our algorithms for the basic problems of counting and sampling to the computation of frequency moments for very long data streams. But before we proceed to the algorithms, we will give a detailed introduction to data stream algorithms.

4.1 Data Stream Algorithms

Data stream algorithms address a recent new trend in data processing and algorithm design: The rate at which we produce data is growing at an accelerating rate. Networks process data packets at a rate of several gigabits per second, therefore it is hardly possible to store all packets or even some small information about each packet. Nevertheless such data contains valuable information that can be used for network management. Large webservers produce huge amounts of log data that contains valuable information, e.g. for marketing purposes. But this information can only be obtained by processing the data which becomes more difficult and costly as the amount of data grows. Large online services like *facebook*[1], *flickr*[2], or *gmail*[3] have to manage petabytes of userdata. The recent research in sensor networks foreshadows an even larger growth of the steady stream of information about our world that is constantly available to us. But all of this information is useless if we are not able to process this huge amount of data at the same rate at which it is produced.

[1] `http://www.facebook.com/`
[2] `http://www.flickr.com/`
[3] `http://www.gmail.com/`

A proposed solution to these challenges are data stream algorithms. A data stream algorithm reads its input as a sequence of data elements that are processed one-by-one in an order that is given by an adversary in a single pass over the input or a small constant number of such passes. Data stream algorithms are considered to be efficient if each data element can be processed efficiently and if the space complexity of the algorithm is sublinear, preferably polylogarithmic, in the length of the input. Given these restrictions, a data stream algorithm cannot simply read the entire input and access the input randomly during the computation of the output. It must rather process each data element online as it arrives and cannot store every data element for later reference.

Provably, many important problems cannot be solved exactly by deterministic algorithms within the sublinear space constraints of data stream algorithms, therefore we often have to resort to randomized algorithms that approximate the solution of a problem. The following notion of approximation has turned out to be adequate for data stream algorithms.

Definition 4.1.1 ((ϵ, δ)-approximation)**.** Let $U = \{u_1, \ldots, u_m\}$ be a set and let $f\colon U^* \longrightarrow \mathbb{R}$ be a function. A randomized data stream algorithm A computes an (ϵ, δ)-approximation of f if the output $A(x)$ of A satisfies

$$\Pr\{|A(x) - f(x)| \leq \epsilon f(x)\} \geq 1 - \delta$$

for every data stream $x = (x_1, \ldots, x_n) \in U^*$ where the probability is taken with respect to the random decisions of the randomized algorithm.

This definition can be generalized to functions $f : U^* \longrightarrow R$ for arbitrary sets R by introducing a function that measures the quality of an approximate solution for a given data stream.

Clearly, the model of data stream algorithms matches the scenario in which a steady stream of data elements has to be processed at a rate such that a permanent storage of the data is impossible, for example network devices and to same extent log files. But this model also applies to general computations on large data sets. Even if we are able to store a large data set then a random access of many items in the data set can be too costly. Modern hard drives can access data sequentially at high rates, but random access is slow since the positioning of the read head (seek) is a mechanical process which cannot be sped up significantly. The gap between sequential and random access is even increasing as the capacity of hard drives grows since the speed of sequential reading grows with the storage density whereas the time for random access is dominated by the seek time which essentially remained constant in the recent past. This precludes the application of conventional algorithms for large data sets on external storage devices. Hence, even if a random access to the input is possible in principle, a data stream algorithm may be the only feasible solution in practice.

A comprehensive introduction to data stream algorithms can be found, for example, in the surveys by Muthukrishnan [61] and by Babcock, Babu, Datar, Motwani, and Widom [9].

4.2 Approximate Counting

The first data stream algorithm that we consider is for a very basic problem: Counting the number of elements in a data stream. Clearly, this problem can be solved exactly using $O(\log n)$ bits of memory for data streams of length n. It is also easy to show that every algorithm that computes the length of a data stream exactly must use $\Omega(\log n)$ bits of memory. Since data stream algorithms with polylogarithmic space complexity are usually considered

efficient, one might wonder how data stream algorithms for counting the number of stream elements can be improved. Nevertheless, in 1978 when memory was scarce and expensive, Robert Morris faced the problem that he had to maintain many counters for a large number of events that had to be stored in small registers, 8-bit registers in his case. To solve his problem, he devised an algorithm that computes (ϵ, δ)-approximations of the counters using space $O(\log \log n)$ for n events [59]. This algorithm is a very early example of a data stream algorithm and already demonstrates basic ideas that are commonly used in the design of data stream algorithms. The random process that is implemented by this algorithm is interesting on its own, it was analyzed in detail by Flajolet [38] and by Hofri and Kechris [49]. But Morris' algorithm has one shortcoming: It computes a single (ϵ, δ)-approximation of the number of events after all events happened. It was not designed to provide a good approximation of the number of events after each event while the events are counted. This severely limits the utility of Morris' algorithm as a building block for other algorithms, since many algorithms that use counters access these counters frequently. An example of this is given in Section 4.3. In this section we will presents an algorithm that counts events approximately using space $O(\log \log n)$ after n events such that with a high probability the approximate count is always a good approximation of the number of events that were counted so far.

4.2.1 Morris' Algorithm

We will briefly sketch Morris' algorithm [59] for approximately counting n events using only $O(\log \log n)$ space. Morris' algorithm approximately maintains the logarithm of the number of events that have happened so far with respect to a fixed base $b > 1$. For n events this can be done using $O(\log \log n)$ bits of memory. To this end the algorithm stores a number r, the *register value*, that is initialized with $r = 0$. The register value r represents the approximate count $v(r) = \frac{b^r - 1}{b-1}$. Clearly, there does not exist a register value r for every $n \in \mathbb{N}$ such that $v(r) = n$. Therefore, in general, it is impossible to increment $v(r)$ by one to count a single event. The main idea of Morris' algorithm is to increment the register with the probability $p(r) = (v(r+1) - v(r))^{-1} = b^{-r}$. Then $p(r)v(r+1) + (1 - p(r))v(r) = v(r) + 1$ and the *expected value* of $v(r)$ after this randomized increment is $v(r)+1$. By the fact that $v(0) = 0$ and by the linearity of expectation, the expected approximate count after n randomized increments is n. The probability that the actual approximate count deviates too far from the average can be bounded easily by Chebyshev's inequality. For details of this analysis we refer the reader to [59] and [38, 46].

4.2.2 Counting an Infinite Number of Events Approximately

We will improve Morris' approximate counting algorithm in two ways. The first modification ensures that we can realize a Bernoulli trial with a success probability of approximately $1/C_t$ for the approximate count C_t after t events within the space bounds of the algorithm using only independent unbiased random bits. This idea was introduced by Gronemeier and Sauerhoff in [46]. The second modification ensures that the improved algorithm gives a good approximation of the actual count of events after *every* event with a sufficiently large probability. This even holds for an infinite number of events. Morris algorithm, in contrast, only guarantees a good approximation with a sufficiently high probability for a single query of the approximate number of events. This problem was also addressed in [46], but it was not solved entirely: Gronemeier and Sauerhoff proposed an algorithm that yields a good approximation

of the number of events after every event with a large probability for up to n events using only space $O(\log \log n)$. Unfortunately, the parameters of the algorithm depend on n, hence the upper bound n on the number of events has to be fixed in advance and the approximation guarantees of their algorithm only hold for up to n events. The properties of the improved approximate counting algorithm are summarized in the following theorem.

Theorem 4.2.1. *Let $0 < \epsilon, \delta < 1$. Then the following is guaranteed to hold for the improved approximate counting algorithm with a probability of at least $1 - \delta$: Let C_t denote the approximate count after t events. Then for every $t \in \mathbb{N}$ after the tth event*

(i) *the approximate count C_t satisfies $|C_t - t| \leq \epsilon t$,*

(ii) *the algorithm uses $O\left(\log \frac{1}{\epsilon} + \log \frac{1}{\delta} + \log \log t\right)$ bits of memory,*

(iii) *a Bernoulli trial with the success probability $\frac{1}{C'_t}$ such that $|C'_t - t| < 2\epsilon t$ can be realized within the space bounds of the algorithm using independent unbiased random bits.*

Our improved algorithm has two parameters $c, d \in \mathbb{N}$ that depend on the error parameters ϵ and δ of Theorem 4.2.1. The choice of these parameters is described later on. The algorithm maintains a register value r that represents the actual approximate count of events. We interpret the register value r like Gronemeier and Sauerhoff [46]: The algorithm has two phases. In the first phase $2^d - 1$ events are counted exactly using the register r. In the second phase the register value r approximately represents the logarithm of the number of events with respect to the base $b = 1 + \beta$ where $\beta = (2^d - 1)^{-1}$. The register value r represents the value $v(r) = (1 + \beta)^r / \beta$. Note that the count after $2^d - 1$ events at the start of the second phase is correct since $v(0) = 1/\beta = 2^d - 1$. We will see in the following section that our slightly modified representation of the approximate count compared to Morris' algorithm will enable us to realize a Bernoulli trial with a success probability of approximately $1/C_t$ within the space bounds of the algorithm using only independent unbiased random bits. In addition to the register value, the algorithm stores the phase of the algorithm using a single bit of memory and it maintains a counter s that is described below. Algorithm 1 summarizes the initialization of all variables:

Algorithm 1 Initialization of the improved approximate counting algorithm

Require: r, i, phase are global variables.

```
1: procedure INIT
2:     phase ← 1                ▷ Start with Phase 1
3:     r ← 0                    ▷ Register
4:     s ← 0                    ▷ Counter
5: end procedure
```

The approximate count can be queried at any time. Since the binary representation of the number that is represented by r does not respect the space bounds of Theorem 4.2.1, we use the function representation(phase, r) to obtain a succinct representation of $v(r)$. For our purposes it is sufficient to represent $v(r)$ by *phase* and the register value r, therefore we simply assume that representation$(\mathit{phase}, r) = (\mathit{phase}, r)$ and query the approximate count as it is shown in Algorithm 2.

The second improvement on Morris' algorithm is due to changes in the randomized increment of the approximate counter. It is easy to see that r has to be incremented with

Algorithm 2 Get approximate count

Require: $r, phase$ are global variables.

```
1: function GetCount
2:     return representation(phase, r)
3: end function
```

the probability $p(r) = (1+\beta)^{-r}$ in the second phase of the algorithm to obtain an expected increment of 1 of the approximate count $v(r)$. We omit the simple analysis from [46] since our analysis of the improved algorithm does not use this fact immediately. For our analysis a slightly different interpretation of Morris' algorithm will be useful: For each value r of the register a random experiment E_r is performed and the register value is incremented if the experiment terminates. In this random experiment for each event that is counted by the algorithm a Bernoulli trial with the success probability $(1+\beta)^{-r}$ is performed and the experiment terminates if the Bernoulli trial is successful. Clearly, this is just a different description of Morris' algorithm. For the purpose of probability amplification we will perform several independent copies of the experiment E_r in parallel and increment the register value if at least a $(1-e^{-1})$-fraction of the experiments terminated.

Definition 4.2.2. For each event that is counted by Morris' algorithm, conceptually, a single step of the experiment E_r is simulated. In a single step of the experiment E_r a Bernoulli trial with the success probability $(1+\beta)^{-r}$ is performed. The experiment E_r is said to terminate after s steps if the first successful Bernoulli trial happens after s steps. Consider $c(r+2)$ independent copies of the experiment E_r and let $E_{r,j}$ denote the jth copy of the experiment E_r.

Note that a single copy $E_{r,j}$ of the experiment E_r does not have any state except for the fact that it has terminated or that it is still running: Every Bernoulli trial in a running experiment is independent of the previous random decisions in the experiment and, given two independent copies $E_{r,j}$ and $E_{r,j'}$ that have not terminated, both copies terminate with the same probability after the next event. Since the running experiments are indistinguishable, for the simulation of $c(r+2)$ independent copies of the experiment E_r it is sufficient to store the total number of experiments and the number of experiments that are still running, therefore we can simulate $c(r+2)$ copies of E_r using space $O(\log c(r+2))$. We will prove below that the success probability of the improved approximate counting algorithm is increased significantly by this change. Algorithm 3 explains the implementation of this idea in detail. Note that at this point it is not obvious that Algorithm 3 can be implemented within the constraints of Theorem 4.2.1 since we have not yet described how the Bernoulli trials can be realized using unbiased random bits. This problem will be addressed in the analysis of the algorithm. Additionally, we have to deal with the fact that Euler's number e is irrational. To this end, we remark that our analysis in the following sections remains valid if we approximate $e^{-1}c(r+2)$ up to a small constant absolute error and that this approximation can be computed within the space bounds of the algorithm using the exponential series for e^{-1}.

The Bernoulli trial with a success probability of approximately $1/C_t$ for the approximate count C_t according to claim (iii) of Theorem 4.2.1 is detailed in Algorithm 4. It is realized using a Bernoulli trial with the parameter $(1+\beta)^{-r}$ and Bernoulli trials with the constant parameters β' and $p_1, \ldots, p_{2^d-2}$ that are described in the analysis of the algorithm. We will explain in the analysis how this algorithm can be implemented using only unbiased random bits.

Algorithm 3 Increment approximate count

Require: $r, s, phase$ are global variables, c, d are fixed parameters, $\beta = (2^d - 1)^{-1}$.

```
1: function INCREMENT
2:     if phase = 1 then
3:         r ← r + 1
4:         if r = 2^d − 1 then
5:             phase ← 2
6:             r ← 0
7:             s ← c(r + 2)
8:         end if
9:     else
10:        Simulate s Bernoulli trials with success probability (1 + β)^{−r} and
11:        subtract the number of successes from s.
12:        if s ≤ e^{−1} c(r + 2) then
13:            r ← r + 1
14:            s ← c(r + 2)
15:        end if
16:    end if
17: end function
```

Algorithm 4 Realize Bernoulli trial with success probability $1/C'_t$ (see Thm. 4.2.1)

Require: $r, s, phase$ are global variables, c, d are fixed parameters, $\beta = (2^d - 1)^{-1}$

Require: The constants β' and $p_1, \ldots, p_{2^d-2}$ are described in the analysis.

```
1: function BERNOULLITRIAL
2:     if phase = 1 then
3:         if r = 0 then
4:             Error!
5:         else
6:             Simulate Bernoulli trial with parameter p_r and
7:             return the result.
8:         end if
9:     else
10:        Simulate Bernoulli trials with the parameters (1 + β)^{−r} and β'.
11:        Return success if both trials were successful and failure otherwise.
12:    end if
13: end function
```

Realizing the Random Decisions with Random Bits

Our choice of the base $b = 1 + \beta$ with $\beta = (2^d - 1)^{-1}$ enables us to realize the Bernoulli trials for an increment of the counter (Alg. 3) within the space bounds of the algorithm using $d \cdot r$ independent unbiased random bits. We need to realize a Bernoulli trial with the success probability $p(r) = (1 + \beta)^{-r}$. Note that

$$p(r) = (1 + \beta)^{-r} = \left(1 + \frac{1}{2^d - 1}\right)^{-r} = \left(1 - 2^{-d}\right)^r \ . \tag{4.1}$$

Clearly, Bernoulli trials with the success probability $1 - 2^{-d}$ can be realized using d random bits. A Bernoulli trial with the parameter $p(r)$ can be realized as the simultaneous success of r independent Bernoulli trials with the success probability $1 - 2^{-d}$. Note that we need $\log r$ bits to count the number of trials. Algorithm 3 sequentially performs $c(r + 2)$ Bernoulli trials. The number of performed and successful trials can be counted using $O(\log c + \log r)$ bits of memory. Overall, the random decisions of Algorithm 3 can be realized with unbiased independent random bits using space

$$O(\log c + \log r) \ . \tag{4.2}$$

The Bernoulli trials of Algorithm 4 in the second phase of the algorithm are realized as follows: We already know how to realize a Bernoulli trial with the parameter $(1 + \beta)^{-r}$. According to Theorem 4.2.1, after t events we need to realize a Bernoulli trial with a success probability of $1/C'_t$ such that $|C'_t - t| \leq 2\epsilon t$. First recall that C_t denotes the approximate count after t events and note that

$$\frac{1}{C_t} = (1 + \beta)^{-r}\beta \ , \tag{4.3}$$

hence, if we choose the parameter $\beta' = \beta$ in Algorithm 4 then we implement a Bernoulli trial with the success probability $1/C_t$. With unbiased random bits we can only realize Bernoulli trials with a success probability β' such that β' is an approximation of β. Hence, the success probability of the Bernoulli trial that is realized by Algorithm 4 will only be an approximation $1/C'_t$ of $1/C_t$ and we need to analyze the error that is introduced by this approximation. Let $\epsilon' = \frac{\epsilon}{1+\epsilon}$. We will use an approximation β' of β such that the register value r and the approximate count C_t after t events satisfy

$$\frac{1}{(1 + \epsilon')C_t} \leq \beta'(1 + \beta)^{-r} \leq \frac{1}{(1 - \epsilon')C_t} \ . \tag{4.4}$$

Then, since $(1 + \epsilon')(1 + \epsilon) = 1 + 2\epsilon$, $(1 - \epsilon')(1 - \epsilon) \geq 1 - 2\epsilon$, and $|C_t - t| \leq \epsilon t$, we obtain a Bernoulli trial with the parameter $1/C'_t$ such that $|C'_t - t| \leq 2\epsilon t$. To this end we choose $\beta' = \frac{\lceil 2^\ell \beta \rceil}{2^\ell}$ for a suitably chosen value $\ell \in \mathbb{N}$. Clearly, a Bernoulli trial with parameter β' can be realized using $O(\ell)$ random bits and $O(\ell)$ space. By plugging $C_t = \frac{(1+\beta)^r}{\beta}$ into inequality (4.4), we obtain the requirement

$$\frac{\beta}{1 + \epsilon'} \leq \beta' \leq \frac{\beta}{1 - \epsilon'} = \beta + \frac{\epsilon'\beta}{1 - \epsilon'} \tag{4.5}$$

for β' and, by the definition of β', we have

$$\frac{\beta}{1 + \epsilon'} \leq \beta \leq \beta' \leq \frac{2^\ell \beta + 1}{2^\ell} = \beta + 2^{-\ell} \ . \tag{4.6}$$

Therefore it is sufficient to choose ℓ such that $2^{-\ell} \leq \frac{\epsilon'\beta}{1-\epsilon'}$ and a Bernoulli trial with the parameter β' can be realized with independent unbiased random bits using space

$$O(\ell) = O(\log((1-\epsilon')/(\epsilon'\beta))) = O(\log(1/\epsilon) + \log(1/\beta)) \,. \tag{4.7}$$

The Bernoulli trials with success probabilities $p_1, \ldots, p_{2^d-2}$ in the first phase of Algorithm 4 are realized analogously such that

$$\frac{1}{(1+2\epsilon)i} \leq p_i \leq \frac{1}{(1-2\epsilon)i} \tag{4.8}$$

for all $i \in \{1, \ldots, 2^d - 2\}$. This can also be done using space $O(\log(1/\epsilon) + \log(1/\beta))$. Overall, Algorithm 4 can be realized with unbiased independent random bits using space

$$O(\log c + \log r + \log(1/\epsilon) + \log(1/\beta)) \,. \tag{4.9}$$

Bounding the Probability of a Large Relative Error

Obviously, the error of our improved approximate counting algorithm is zero in the first phase of the algorithm. To prove claim (i) of Theorem 4.2.1 for the second phase of the algorithm we will analyze the number of events that happen while $r = i$ in the second phase of Algorithm 3.

Definition 4.2.3. Events that lead to an increment of the register value in the second phase of Algorithm 3 are called *increment events* in the following. Let the random variable N_i denote the number of events that happen in the second phase of Algorithm 3 while $r = i$ including the increment event that leads to the increment of the register value to $i+1$. Then $N_{<i} = 2^d - 1 + \sum_{j=0}^{i-1} N_j$ denotes the number of events that happened up to the ith increment event including the increment event.

Note that $N_{<i}$ includes the $2^d - 1 = \beta^{-1}$ events that are counted exactly in the first phase of the algorithm. We will first bound the probability that the approximate C_t is not a good approximation of t for values of t that correspond to increment events. Immediately after an increment event exactly $N_{<r}$ events happened. Assume for the moment that $N_i = (1+\beta)^i$ for all $i \in \{1, \ldots, r-1\}$. Then the output of Algorithm 2 immediately after an increment event is the exact number of events that happened so far:

$$N_{<r} = \frac{1}{\beta} + \sum_{i=0}^{r-1} N_i = \frac{1}{\beta} + \sum_{i=0}^{r-1} (1+\beta)^i = \frac{1}{\beta} + \frac{(1+\beta)^r - 1}{\beta} = \frac{(1+\beta)^r}{\beta} \,. \tag{4.10}$$

This is equal to the approximate count $v(r)$ for the register value r. We will show in the following that N_i does not deviate too far from $(1+\beta)^i$ for all $i \in \mathbb{N}$ with a large probability if the parameters of the algorithm are chosen appropriately. Then, by the same argument as above, the output of Algorithm 2 after an increment event is an ϵ-approximation of $N_{<r}$ if $|N_i - (1+\beta)^i| \leq \epsilon N_i$ for all $i < r$. This condition is implied by $|N_i - (1+\beta)^i| \leq \frac{\epsilon}{1+\epsilon}(1+\beta)^i$ since in this case

$$N_i \leq \left(1 + \frac{\epsilon}{1+\epsilon}\right)(1+\beta)^i \leq \left(1 + \frac{\epsilon}{1-\epsilon}\right)(1+\beta)^i = \frac{1}{1-\epsilon}(1+\beta)^i \tag{4.11}$$

and

$$N_i \geq \left(1 - \frac{\epsilon}{1+\epsilon}\right)(1+\beta)^i = \frac{1}{1+\epsilon}(1+\beta)^i \tag{4.12}$$

and therefore $(1+\beta)^i \geq (1-\epsilon)N_i$ and $(1+\beta)^i \leq (1+\epsilon)N_i$. Consequently, we can prove claim (i) of Theorem 4.2.1 for values of t that correspond to increment events by showing that the probability for the existence of an $i \in \mathbb{N}$ such that $|N_i - (1+\beta)^i| \geq \frac{\epsilon}{1+\epsilon}(1+\beta)^i$ is bounded from above by δ. Before we proceed with this plan, we need some necessary technical preparations. First we will need the following estimates.

Proposition 4.2.4. *Let $x \in [0,1]$. Then*

(i) $1 - e^{-x} \geq \frac{x}{2}$ *and*

(ii) $e^{-1} - e^{-(1+x)} \geq \frac{x}{5}$.

Proof. Let $\ell(x) = 1 - e^{-x}$ and $r(x) = \frac{x}{2}$ denote the left hand and right hand side of the inequality in claim (i), respectively. For the proof of claim (i) it is sufficient to verify that $\ell(0) = r(0)$, that $\ell(1) \geq r(1)$, and that $\ell(x)$ is concave. This implies that the concave function $\ell(x)$ is larger than the linear function $r(x)$ on the unit interval. Claim (ii) can be verified using the same arguments. □

The next lemma bounds the probability that many independent copies of a geometric random variable simultaneously deviate from their expectation by a large amount.

Lemma 4.2.5. *Let $0 < \epsilon < 1$ and assume that the random variables $X_1, \ldots, X_n$ are independent copies of a geometric random variable X with parameter p and define random variables $L(t) = |\{i : X_i > t\}|$ for $t \in \{1, \ldots, n\}$.*

(i) *If $t \leq (1-\epsilon)(\mathrm{E}[X] - 1)$ then $\Pr\{L(t) \leq e^{-1}n\} \leq \exp\left(-\frac{\epsilon}{50}n\right)$.*

(ii) *If $t \geq (1+\epsilon)\,\mathrm{E}[X]$ then $\Pr\{L(t) \geq e^{-1}n\} \leq \exp\left(-\frac{\epsilon}{50}n\right)$.*

Proof. We will first prove claim (i). Assume that $t \leq (1-\epsilon)(\mathrm{E}[X] - 1)$. Each variable X_i is a geometric random variable with parameter p. Hence

$$\Pr\{X_i > t\} = 1 - \sum_{k=1}^{t} p(1-p)^{k-1} \tag{4.13}$$

$$= (1-p)^t \tag{4.14}$$

$$\geq (1-p)^{(1-\epsilon)(\mathrm{E}[X]-1)} \tag{4.15}$$

$$= (1-p)^{(1-\epsilon)(1/p-1)} \tag{4.16}$$

$$\geq e^{-(1-\epsilon)} \tag{4.17}$$

for every $i \in \{1, \ldots, n\}$. In the last line we used the well-known fact that $(1-x)^{1/x-1} \geq e^{-1}$ for all $x \in (0,1]$. The events $X_i > t$ for $i \in \{1, \ldots, n\}$ are independent because the random variables X_i are independent, hence $L(t)$ is distributed binomially with the parameters n and $q \geq e^{-(1-\epsilon)}$ and $\mathrm{E}[L(t)] = qn \geq e^{-(1-\epsilon)}n$. Let $c_1 = 1 - e^{-\epsilon}$. Then $c_1 > 0$ and

$$(1-c_1)\,\mathrm{E}[L(t)] \geq e^{-1}n\,. \tag{4.18}$$

By Chernoff bounds (see Thm. A.2.2),

$$\Pr\left\{L(t) \leq e^{-1}n\right\} \leq \Pr\left\{L(t) \leq (1-c_1)\,\mathrm{E}[L(t)]\right\} \tag{4.19}$$

$$\leq \exp\left(-\frac{c_1^2}{2}\,\mathrm{E}[L(t)]\right)\,. \tag{4.20}$$

Using claim (i) of Proposition 4.2.4 we have

$$\frac{c_1^2}{2}\,\mathrm{E}[L(t)] \geq \frac{\left(1-e^{-\epsilon}\right)^2}{2} e^{-(1-\epsilon)} n \geq \frac{\epsilon^2}{8} e^{-1} n \geq \frac{\epsilon^2}{50} n\ . \tag{4.21}$$

By plugging this into the last result we obtain claim (i) of the lemma.

For the proof of claim (ii) assume that $t \geq (1+\epsilon)\,\mathrm{E}[X]$. Let $S(t) = |\{i | X_i \leq t\}|$, then $S(t) + L(t) = n$. By the fact that each X_i is a geometric random variable with parameter t we get the following for every $i \in \{1, \ldots, n\}$:

$$\begin{aligned}
\Pr\{X_i \leq t\} &= \sum_{k=1}^{t} p(1-p)^{k-1} && (4.22)\\
&= 1 - (1-p)^t && (4.23)\\
&\geq 1 - (1-p)^{(1+\epsilon)\,\mathrm{E}[X]} && (4.24)\\
&= 1 - (1-p)^{(1+\epsilon)/p} && (4.25)\\
&\geq 1 - e^{-(1+\epsilon)}\ . && (4.26)
\end{aligned}$$

In the last line we used the well-known fact that $(1-x)^{1/x} \leq e^{-1}$ for all $x \in (0,1]$. Clearly, $S(t)$ is distributed binomially with the parameters n and $q \geq 1 - e^{-(1+\epsilon)}$ and $\mathrm{E}[S(t)] = qn \geq (1 - e^{-(1+\epsilon)})n$. By the fact that $S(t) + L(t) = n$, we have

$$\Pr\{L(t) \geq e^{-1}n\} = \Pr\{S(t) \leq (1-e^{-1})n\}\ . \tag{4.27}$$

Let $c_2 = 1 - \frac{1-e^{-1}}{1-e^{-(1+\epsilon)}}$. Then $c_2 > 0$ and

$$(1-c_2)\,\mathrm{E}[S(t)] \geq (1-e^{-1})n \tag{4.28}$$

By Chernoff bounds, we have

$$\begin{aligned}
\Pr\left\{L(t) \geq e^{-1}n\right\} &= \Pr\left\{S(t) \leq (1-e^{-1})n\right\} && (4.29)\\
&\leq \Pr\{S(t) \leq (1-c_2)\,\mathrm{E}[S(t)]\} && (4.30)\\
&\leq \exp\left(-\frac{c_2^2}{2}\,\mathrm{E}[S(t)]\right)\ . && (4.31)
\end{aligned}$$

By claim (ii) of Proposition 4.2.4 we obtain

$$\frac{c_2^2}{2}\,\mathrm{E}[S(t)] \geq \frac{1}{2}\left(1 - \frac{1-e^{-1}}{1-e^{-(1+\epsilon)}}\right)^2 \left(1 - e^{-(1+\epsilon)}\right) n = \frac{\left(e^{-1} - e^{-(1+\epsilon)}\right)^2}{2\left(1-e^{-(1+\epsilon)}\right)} n \geq \frac{\epsilon^2}{50} n\ . \tag{4.32}$$

Then claim (ii) of the lemma follows from the last to result. □

Now, with our technical preparations in place, we can prove that N_i does not deviate too far from $(1+\beta)^i$ with a large probability if the parameter c of Algorithm 3 is chosen appropriately.

Lemma 4.2.6. *Let $0 < \epsilon' < 1$ and $0 < \delta' \leq \frac{1}{2}$. If the parameter c of Algorithm 3 satisfies $c \geq \frac{50 \ln(1/\delta')}{\epsilon'}$ then $\Pr\{|N_i - (1+\beta)^i| > \epsilon'(1+\beta)^i\} \leq (\delta')^{i+1}$ for all $i \in \mathbb{N}$.*

Proof. Recall from our introductory discussion of Algorithm 3 that for every register value r, conceptually, $c(r+2)$ independent copies $E_{r,j}$ of the random experiment E_r are simulated and that the register value is incremented if at least a $(1-e^{-1})$-fraction of these experiments terminated (see Def. 4.2.2). Let the random variable $X_{r,j}$ denotes the number of events after which the experiment $E_{r,j}$ terminates and let $\mathrm{E}[X_r]$ denote the expected number of events until a single copy of E_r terminates. Clearly, we have $\mathrm{E}[X_{r,j}] = \mathrm{E}[X_r] = (1+\beta)^r$ for all $j \in \{1,\dots,c(r+2)\}$ since $X_{r,j}$ is an independent geometric random variable with the parameter $(1+\beta)^{-r}$. Now suppose that $N_r < (1-\epsilon')(1+\beta)^r = (1-\epsilon')\,\mathrm{E}[X_r]$. Then at most $(1-\epsilon')\,\mathrm{E}[X_r] - 1$ events happened until the register value was incremented to $r+1$ and at least $(1-e^{-1})c(r+2)$ copies of the experiment E_r terminated after at most $(1-\epsilon')\,\mathrm{E}[X_r] - 1$ events. Clearly, if $E_{r,j}$ is one of the terminated experiments then $X_{r,j} \le (1-\epsilon')\,\mathrm{E}[X_r] - 1$. Hence, there are at most $e^{-1}c(r+2)$ variables $X_{r,j}$ such that $X_{r,j} > (1-\epsilon')\,\mathrm{E}[X_r] - 1$ and we can apply claim (i) of Lemma 4.2.5 for the parameters $n = c(r+2)$, $t = (1-\epsilon')\,\mathrm{E}[X_r] - 1$, and $\epsilon = \epsilon'$ to obtain

$$\Pr\{N_r < (1-\epsilon')(1+\beta)^r\} \le \exp\left(-\frac{\epsilon'}{50}c(r+2)\right) . \tag{4.33}$$

Next, suppose that $N_r > (1+\epsilon')(1+\beta)^r = (1+\epsilon')\,\mathrm{E}[X_r]$. Then at least $(1+\epsilon')\,\mathrm{E}[X_r]$ events happened before the register value was incremented to $r+1$ and less than $(1-e^{-1})c(r+2)$ copies of the experiment E_r terminated after $(1+\epsilon')\,\mathrm{E}[X_r]$ events. Hence, at least $e^{-1}c(r+2)$ copies of E_r did not terminate after $(1+\epsilon')\,\mathrm{E}[X_r]$ events and we can apply claim (ii) of Lemma 4.2.5 for the parameters $n = c(r+2)$, $t = (1+\epsilon')\,\mathrm{E}[X_r]$, and $\epsilon = \epsilon'$ to obtain

$$\Pr\{N_r > (1+\epsilon')(1+\beta)^r\} \le \exp\left(-\frac{\epsilon'}{50}c(r+2)\right) . \tag{4.34}$$

By the assumption that $c \ge \frac{50\ln(1/\delta')}{\epsilon'}$, we have

$$\exp\left(-\frac{\epsilon'}{50}c\right) \le \delta' . \tag{4.35}$$

Since $|N_r - (1+\beta)^r| > \epsilon'(1+\beta)^r$ if and only if $N_r < (1-\epsilon')(1+\beta)^r$ or $N_r > (1+\epsilon')(1+\beta)^r$, our last results and the union bound imply that

$$\begin{aligned}
\Pr\{|N_r - (1+\beta)^r| > \epsilon'(1+\beta)^r\} &\le 2\exp\left(-\frac{\epsilon'}{50}c(r+2)\right) && (4.36)\\
&\le 2(\delta')^{r+2} && (4.37)\\
&\le (\delta')^{r+1} . && (4.38)
\end{aligned}$$

In the last line we used that $\delta' \le \frac{1}{2}$. This completes the proof. □

By using the last lemma we can now prove claim (i) of Theorem 4.2.1 for values of t that correspond to increment events.

Lemma 4.2.7. *Let $0 < \epsilon', \delta < 1$. Suppose that the parameter c of Algorithm 3 satisfies $c \ge \frac{50\ln((\delta+1)/\delta)}{\epsilon'}$. Then, with a probability of at least δ, the approximate count C_t that is maintained by the algorithm satisfies $|C_t - t| \le \epsilon' t$ for all $t \in \mathbb{N}$ that correspond to increment events.*

Proof. Assume that the tth event is an increment event and that r is the register value after the tth event. We have already observed that $|C_t - t| \leq \epsilon' t$ if $|N_i - (1+\beta)^i| < \frac{\epsilon'}{1+\epsilon'}(1+\beta)^i$ for all $i \leq r$ and that conversely $|C_t - t| > \epsilon' t$ implies the existence of an $i \in \{0, \ldots, r\}$ such that $|N_i - (1+\beta)^i| > \frac{\epsilon'}{1+\epsilon'}(1+\beta)^i$. By Lemma 4.2.6, for every constant $0 < \delta' \leq \frac{1}{2}$

$$\Pr\left\{|N_i - (1+\beta)^i| > \frac{\epsilon'}{1+\epsilon'}(1+\beta)^i\right\} \leq (\delta')^{i+1} \tag{4.39}$$

if we choose the parameter c of Algorithm 3 such that $c \geq \frac{50\ln(1/\delta')}{\epsilon'}$. Then, by the union bound, the probability that $|C_t - t| > \epsilon' t$ for any $t \in \mathbb{N}$ is bounded by

$$\sum_{i=0}^{\infty} \Pr\left\{|N_i - (1+\beta)^i| > \frac{\epsilon'}{1+\epsilon'}(1+\beta)^i\right\} \leq \sum_{i=0}^{\infty} (\delta')^{i+1} = \frac{1}{1-\delta'} - 1\ . \tag{4.40}$$

To complete the proof it suffices to choose $\delta' = \frac{\delta}{\delta+1}$, then $\frac{1}{1-\delta'} - 1 = \delta$ and $\delta' \leq \frac{1}{2}$ as it is required for the application of Lemma 4.2.6. □

Now we will extend our proof of claim (i) in Theorem 4.2.1 to values of t that do not correspond to increment events. Suppose that the error of the approximate count after ℓ events and after u events, where $\ell < u$, is small. Then, since the approximate count of Algorithm 3 does not decrease, the error between event ℓ and u is also small if u/ℓ is not too large. This observation is quantified in the following proposition.

Proposition 4.2.8. *For a constant ϵ with $0 < \epsilon < 1$ let $\ell, u \in \mathbb{N}$ such that $1 \leq u/\ell \leq 1 + \epsilon/4$, let $t \in [\ell, u]$, and let $\widetilde{\ell}$ and $\widetilde{u}$ be $(\epsilon/4)$-approximations of ℓ and u, respectively. If $\widetilde{t}$ satisfies $\widetilde{\ell} \leq \widetilde{t} \leq \widetilde{u}$ then $\widetilde{t}$ is an ϵ-approximation of t, thus $|\widetilde{t} - t| \leq \epsilon t$.*

Proof. Since $\widetilde{\ell}$ and $\widetilde{u}$ are $(\epsilon/4)$-approximations of ℓ and u, by the bounds on $\widetilde{t}$, we have

$$(1 - \epsilon/4)\ell \leq \widetilde{t} \leq (1 + \epsilon/4)u\ . \tag{4.41}$$

Thus $t, \widetilde{t} \in [(1-\epsilon/4)\ell, (1+\epsilon/4)u]$ and therefore

$$\begin{aligned}
|t - \widetilde{t}| &\leq \left(1 + \frac{\epsilon}{4}\right)u - \left(1 - \frac{\epsilon}{4}\right)\ell && (4.42)\\
&= \ell\left(\left(1 + \frac{\epsilon}{4}\right)\frac{u}{\ell} - \left(1 - \frac{\epsilon}{4}\right)\right) && (4.43)\\
&\leq \ell\left(\left(1 + \frac{\epsilon}{4}\right)^2 - \left(1 - \frac{\epsilon}{4}\right)\right) && (4.44)\\
&= \ell\left(\frac{\epsilon}{2} + \frac{\epsilon}{4} + \frac{\epsilon^2}{16}\right) && (4.45)\\
&\leq \epsilon\ell\ . && (4.46)
\end{aligned}$$

By using that $\ell \leq t$ we finally obtain the claim of the proposition.

$$\frac{|\widetilde{t} - t|}{t} \leq \frac{|\widetilde{t} - t|}{\ell} \leq \frac{\epsilon\ell}{\ell} = \epsilon\ . \tag{4.47}$$

□

Finally, we can complete the proof of Theorem 4.2.1 (i): Recall that the parameters ϵ and δ specify the approximation error and the error probability of the improved approximate counting algorithm, respectively, and that the algorithm has the parameters c and d. We choose the Parameter d of the improved approximate counting algorithm such that $\epsilon/32 \leq \beta = \frac{1}{2^d-1} \leq \epsilon/16$. Furthermore, we define the constant $\epsilon' = \frac{\epsilon}{32+\epsilon}$. Note that $\frac{1+\epsilon'}{1-\epsilon'} = 1 + \epsilon/16$. Then we choose the parameter c as follows:

$$c = \frac{50 \ln((\delta+1)/\delta)}{\epsilon'} = \frac{50 \ln((\delta+1)/\delta)(32+\epsilon)}{\epsilon} . \tag{4.48}$$

Assume that the register value was incremented to the value i for the t_ith event. Then $|C_{t_i} - t_i| < \epsilon' t_i < (\epsilon/4) t_i$ for all $i \in \mathbb{N}$ with a probability of at least δ by Lemma 4.2.7. Now assume that this event happened. Then $C_{t_i} \geq (1-\epsilon')t_i$ and $C_{t_i} \leq (1+\epsilon')t_i$ for all i and therefore,

$$\frac{t_{i+1}}{t_i} \leq \frac{(1+\epsilon')C_{t_{i+1}}}{(1-\epsilon')C_{t_i}} = \frac{(1+\epsilon')(1+\beta)^{i+1}}{(1-\epsilon')(1+\beta)^i} \leq (1+\epsilon/16)^2 \leq 1+\epsilon/4 . \tag{4.49}$$

By Proposition 4.2.8, we get $|C_t - t| < \epsilon t$ for all i and all $t \in \mathbb{N}$ such that $t_i \leq t \leq t_{i+1}$. This completes the proof of Theorem 4.2.1, claim (i).

Space Complexity of the Algorithm

We have already observed that the random decisions of the algorithms can be realized using

$$O(\log c + \log r + \log(1/\epsilon) + \log(1/\beta)) = O(\log c + \log r + \log(1/\epsilon)) \tag{4.50}$$

bits of memory. Obviously, a single bit of storage suffices to store *phase*. The algorithm uses $O(\log r)$ bits of memory to store r and $O(\log(c(r+2))) = O(\log c + \log r)$ bits to store s. By our choice of c in the last section and the fact that $\ln(1+1/\delta) \leq 1/\delta$ we have

$$\log c = \log \frac{50 \ln((\delta+1)/\delta)(32+\epsilon)}{\epsilon} \leq \log \frac{50(32+\epsilon)}{\delta\epsilon} = O\left(\log \frac{1}{\delta} + \log \frac{1}{\epsilon}\right) . \tag{4.51}$$

If $|C_t - t| \leq \epsilon t$ then $r = O(\log_{(1+\beta)} t)$. By using that $\log(1+x) \geq x$ and that $\beta \geq \frac{\epsilon}{32}$ we get

$$O(\log r) = O\left(\frac{\log t}{\log(1+\beta)}\right) = O(\log\log t - \log\log(1+\beta)) = O(\log\log t + \log(1/\epsilon)) . \tag{4.52}$$

Overall, the space complexity of the algorithm under the condition that it does not fail is

$$O(\, \log\log t + \log(1/\epsilon) + \log(1/\delta)\,) . \tag{4.53}$$

This proves claim (ii) of Theorem 4.2.1 and completes our proof of Theorem 4.2.1.

4.3 Approximate Reservoir Sampling

In the following section we will apply our improved approximate counting algorithm in an algorithm that accesses the counter frequently: Vitter's algorithm for sampling uniformly at random from a data stream. Sampling a stream element uniformly at random is a basic

operation that is used by many data stream algorithms. Here the challenge lies in the fact that the length n of the data stream is not known in advance, hence we cannot simply choose a random number $i \in \{1, \ldots, n\}$ and store the ith stream element when we read it. Vitter [72] proposed an algorithm that chooses a stream element uniformly at random in a single pass over the data stream without knowing the length of the data stream in advance. His algorithm counts the number of data stream elements that have been read so far and uses this number in each step of the algorithm. We will replace this counter by approximate counting. While Morris' algorithm was not designed for this setting, our improved approximate counting algorithm is applicable in this situation. If we use approximate counting in Vitter's algorithm, then the resulting sample is no longer distributed uniformly. In this section we will analyze the effect of the approximation error of approximate counting on the distribution of the sample.

4.3.1 Vitter's Algorithm

We briefly describe Vitter's [72] algorithm for uniformly sampling a random element from a data stream if the length n of the data stream is unknown in advance. Vitter's algorithm counts the number of data stream elements that have been read so far and inductively maintains a current sample that is drawn uniformly at random from these data stream elements. Clearly, if only one stream element has been read then this element has to be the current sample which is stored by the algorithm. If the next stream element is the tth element, then the current sample is replaced by this element with a probability of $\frac{1}{t}$. It is easy to verify that this choice maintains the invariant that the current sample is drawn uniformly at random from the stream elements that have been processed so far. The memory requirements of this algorithm are obvious: Besides the storage space for the current sample the algorithm has to count the number of elements, to this end after t steps $O(\log t)$ bits of memory are needed.

4.3.2 Reservoir Sampling and Approximate Counting

The contribution of the length n of a data stream to the space complexity of Vitter's algorithm becomes relevant if the sample that is maintained by the algorithm can be stored using $o(\log n)$ space. In this case the space complexity is asymptotically dominated by the size of the counter for the data stream length. A brief claim by Alon, Matias, and Szegedy in [3] implies that the contribution of the data stream length n to the space complexity of Vitter's algorithm can be lowered to $O(\log \log n)$ if Morris' approximate counting algorithm is used, but this claim is not substantiated by a rigorous analysis. Here we will prove that the claimed improvement is possible if *the improved approximate counting algorithm* is used and we will analyze the impact of this change on the distribution of the sample. We doubt that the same result can be achieved using the original approximate counting algorithm by Morris since it does only guarantee a good approximation for a single query of the counter whereas the counter is queried for every stream element in Vitter's algorithm.

Definition 4.3.1 (Approximate reservoir sampling)**.** Let $0 < \epsilon, \delta < 1$. We modify Vitter's reservoir sampling algorithm such that the counter for the number of stream elements that have been processed so far is replaced by an approximate counter according to Theorem 4.2.1 with the parameters $\epsilon/2$ and δ. The current sample is replaced by the tth stream element if a Bernoulli trial with a success probability of $1/C'_t$ according to claim (iii) of Theorem 4.2.1 is successful. The resulting algorithm is called (ϵ, δ)-*approximate reservoir sampling*.

Note that the value of C'_t in the parameter $1/C'_t$ of the Bernoulli trial is an ϵ-approximation of t for every $t \in \mathbb{N}$. This property is guaranteed by the improved approximate counting algorithm, but not by Morris' original algorithm. With this stronger guarantee we can prove the claim of Alon, Matias, and Szegedy and quantify the impact on the resulting distribution of the sample.

Theorem 4.3.2. *Let $0 < \epsilon, \delta < 1$ and let X_t denote the index of the current sample that has been chosen by the (ϵ, δ)-approximate reservoir sampling algorithm after t data stream elements have been read. Then, with a probability of at least $1 - \delta$, for all $t \in \mathbb{N}$ the (ϵ, δ)-approximate reservoir sampling algorithm uses $O(\log \frac{1}{\epsilon} + \log \frac{1}{\delta} + \log\log t)$ space after t steps in addition to the space for the sample and the total variation distance of the distribution of X_t and the uniform distribution on $\{1, \ldots, t\}$ is bounded from above by ϵ.*

Proof. In the following we will work under the assumption that for all $t \in \mathbb{N}$ the improved approximate counting algorithm uses $O(\log \frac{1}{\epsilon} + \log\log \frac{1}{\delta} + \log\log t)$ space after t steps and that $|C'_t - t| \leq \epsilon t$ for all $t \in \mathbb{N}$. By Theorem 4.2.1, this holds with a probability of at least $1 - \delta$. Then the claim about the space requirements of (ϵ, δ)-approximate reservoir sampling is obvious. We will prove the claim about the total variation distance by induction on t. Let V_t denote the total variation distance of the uniform distribution on $\{1, \ldots, t\}$ and the distribution of X_t. Clearly, the claim of the theorem is true for $t = 1$ since $\Pr\{X_1 = 1\} = 1$. Now, for the induction, assume that $V_{t-1} \leq \epsilon$. First observe that the current sample is replaced in the tth step with the probability $p_t = \frac{1}{(1+\epsilon_t)t}$ where $|\epsilon_t| \leq \epsilon < 1$. This random decision is independent of the previous random decisions, hence

$$\Pr\{X_t = t\} = \frac{1}{(1+\epsilon_t)t} \,. \tag{4.54}$$

For $i \neq t$ we have $X_t = i$ if and only if $X_{t-1} = i$ and if the current sample is not replaced in the tth step of the approximate reservoir sampling algorithm. Thus for $i \in \{1, \ldots, t-1\}$

$$\Pr\{X_t = i\} = \left(1 - \frac{1}{(1+\epsilon_t)t}\right) \Pr\{X_{t-1} = i\} \,. \tag{4.55}$$

By using this and the triangle inequality we obtain

$$V_i = \frac{1}{2} \sum_{i=1}^{t} \left| \Pr\{X_t = i\} - \frac{1}{t} \right| \tag{4.56}$$

$$= \frac{1}{2} \sum_{i=1}^{t-1} \left| \left(1 - \frac{1}{(1+\epsilon_t)t}\right) \Pr\{X_{t-1} = i\} - \frac{1}{t} \right| + \frac{1}{2} \left| \Pr\{X_t = t\} - \frac{1}{t} \right| \tag{4.57}$$

$$\begin{aligned} &\leq \frac{1}{2} \left(1 - \frac{1}{(1+\epsilon_t)t}\right) \sum_{i=1}^{t-1} \left| \Pr\{X_{t-1} = i\} - \frac{1}{t-1} \right| \\ &\quad + \frac{1}{2} \sum_{i=1}^{t-1} \left| \left(1 - \frac{1}{(1+\epsilon_t)t}\right) \frac{1}{t-1} - \frac{1}{t} \right| + \frac{1}{2} \left| \Pr\{X_t = t\} - \frac{1}{t} \right| \end{aligned} \tag{4.58}$$

$$\begin{aligned} &= \left(1 - \frac{1}{(1+\epsilon_t)t}\right) V_{t-1} \\ &\quad + \frac{1}{2} \sum_{i=1}^{t-1} \left| \frac{1}{t-1} \left(1 - \frac{1}{(1+\epsilon_t)t}\right) - \frac{1}{t} \right| + \frac{1}{2} \left| \Pr\{X_t = t\} - \frac{1}{t} \right| \,. \end{aligned} \tag{4.59}$$

Now we will examine the second and third term in the last equality separately: Note that, since $|\epsilon_t| \leq \epsilon < 1$, we have that $|1 + \epsilon_t| = 1 + \epsilon_t$. Then, for the second term, we obtain

$$\frac{1}{2}\sum_{i=1}^{t-1}\left|\frac{1}{t-1}\left(1-\frac{1}{(1+\epsilon_t)t}\right)-\frac{1}{t}\right| = \frac{t-1}{2}\left|\frac{\epsilon_t}{(1+\epsilon_t)(t-1)t}\right| = \frac{1}{2(1+\epsilon_t)t}\left|\epsilon_t\right| . \tag{4.60}$$

For the third term we also get

$$\frac{1}{2}\left|\Pr\{X_t{=}t\}-\frac{1}{t}\right| = \frac{1}{2}\left|\frac{1}{(1+\epsilon_t)t}-\frac{1}{t}\right| = \frac{1}{2}\left|\frac{-\epsilon_t}{(1+\epsilon_t)t}\right| = \frac{1}{2(1+\epsilon_t)t}\left|\epsilon_t\right| . \tag{4.61}$$

If we plug this into our first result then the induction hypothesis and the assumption that $|\epsilon_t| \leq \epsilon$ yield the claimed result for $t \geq 2$:

$$V_t \leq \left(1-\frac{1}{(1+\epsilon_t)t}\right)V_{t-1} + \frac{1}{(1+\epsilon_t)t}\left|\epsilon_t\right| \tag{4.62}$$

$$\leq \left(1-\frac{1}{(1+\epsilon_t)t}\right)\epsilon + \frac{1}{(1+\epsilon_t)t}\cdot\epsilon \tag{4.63}$$

$$= \epsilon . \tag{4.64}$$

□

Note that the L^∞-distance of the distribution of X_t and the uniform distribution on $\{1,\ldots,t\}$ has been examined before by Gronemeier and Sauerhoff [46], but this result is too weak to bound the total variation distance by a constant.

4.4 Frequency Moments

The computation of frequency moments of a data stream is an important and already thoroughly studied class of problems. The kth frequency moment of a data stream is defined as follows.

Definition 4.4.1 (histogram, frequency moments)**.** Let $U = \{u_1,\ldots,u_m\}$ be a set, let $a = (a_1,\ldots,a_n)$ be a data stream such that $a_i \in U$ for all $i \in \{1,\ldots,n\}$, and let $k \in \mathbb{R}$ be a constant subject to $k \geq 0$. Then

- $f_i(a) = |\{j \in \{1,\ldots,n\} : a_j = u_i\}|$ is the *absolute frequency* of the element u_i in a,
- $f(a) = (f_1(a),\ldots,f_m(a))$ is called *the histogram* of a, and
- $F_k(a) = \sum_{i=1}^m f_i(a)^k$ is called the *kth frequency moment* of the data stream a.

The frequency moments of a data stream are a statistical measure for the degree of skew in the distribution of the stream elements. An important application of frequency moments is query optimization in relational databases since the size of the join of two relations is closely related to the second frequency moment [47]. The connection to data stream algorithms is as follows: In database applications the sequence of database operations can be considered as a data stream. For query optimization we need a small data structure that can be kept in the main memory which "summarizes" the database operations and supports the estimation of

frequency moments. This data structure must be updated efficiently for every database operation in order to reflect the actual contents of the database. From the preceding description, we see that the computation of this data structure is in fact a data stream algorithm.

In their seminal paper [3] Alon, Matias, and Szegedy described efficient data stream algorithms that compute (ϵ, δ)-approximations of F_2 and F_k for $k \geq 1$ using space $O((1/\epsilon^2)\log(1/\delta)(\log m + \log n))$ and $O((1/\epsilon^2)\log(1/\delta)km^{1-1/k}(\log m + \log n))$, respectively. The results of Alon, Matias, and Szegedy were to some extent a catalyst for the recent interest in data stream algorithms and triggered the publication of various new data stream algorithms, including improved algorithms for (ϵ, δ)-approximations of frequency moments by Coppersmith and Kumar [28], Ganguly [40], and Indyk and Woodruff [50]. The last publication presents an algorithm that achieves an essentially optimal space complexity of $\widetilde{O}(m^{1-2/k})$ and time complexity of $\widetilde{O}(1)$ per update where the $\widetilde{O}$-notation suppresses a factor of the order $\text{poly}((1/\epsilon), \log(1/\delta), \log m, \log n)$. The quite complicated algorithm by Indyk and Woodruff was finally simplified by Bhuvanagiri, Ganguly, Kesh, and Saha [17].

In this section we consider the computation of frequency moments for very long data streams. The contribution of the data stream length n to the space complexity of the known data stream algorithms becomes relevant if the length n of the data stream is not polynomially bounded in the size m of the universe. In [3] Alon, Matias and Szegedy have briefly remarked that the space complexity of their algorithms can be decreased for large n by replacing exact counting with approximate counting using Morris' algorithm, but they did not support this claim by a detailed analysis. While their claim is easily verified for their algorithm that approximates F_2, it turns out that additional ideas are required for the general algorithm that approximates F_k for $k \geq 1$. We described these ideas in the preceding sections on approximate counting and reservoir sampling. In the following we will apply approximate counting and approximate reservoir sampling to the computation of frequency moments and show that the claim in [3] holds if our improved variants of approximate counting and reservoir sampling are used. We doubt that the same result can be obtained by using Morris' original approximate counting algorithm.

4.4.1 The Algorithm of Alon, Matias, and Szegedy

In [3] Alon, Matias, and Szegedy present an algorithm for computing $F_k(a)$ for $k \in \mathbb{R}$ such that $k \geq 1$. Their algorithm, the AMS algorithm for short, is easily generalized to functions of the histogram and the length of the data stream that are of the following form:

Definition 4.4.2 (Function F_g of the histogram)**.** Let $a = (a_1, \ldots, a_n)$ be a data stream with elements from the universe $U = \{u_1, \ldots, u_m\}$. Let $g\colon \mathbb{R} \times \mathbb{R} \longrightarrow \mathbb{R}$ be a continuous function that is nondecreasing in the first coordinate and partially differentiable with respect to the first coordinate on $\mathbb{R}$ such that for every $n \in \mathbb{N}$ we have that $g(0, n) = 0$ and that $g'(x, n) = \frac{d}{dx}g(x, n)$ is nondecreasing in x. Then define

$$F_g(a) = \sum_{i=1}^{m} g(f_i(a), n) \; .$$

Clearly, frequency moments are a special case of this definition. Note however that the space complexity of the generalized AMS algorithm will depend heavily on the choice of the function g and that the algorithm is not necessarily an efficient algorithm for functions g that do not correspond to frequency moments. Here we will briefly describe and analyze the

generalized AMS algorithm and apply this analysis to the computation of frequency moments because our improvement for very long data streams and its analysis will use these results.

The AMS algorithm is based on the following unbiased estimator for $F_g(a)$:

Definition 4.4.3 (Basic estimator Y for $F_g(a)$)**.** Let $a = (a_1, \ldots, a_n)$ be a data stream with elements from $U = \{u_1, \ldots, u_m\}$ and let $r_i = |\{j : i \leq j \leq n, a_j = a_i\}|$ denote the number of elements after a_i that are identical to a_i including the element a_i itself. Let X be chosen uniformly at random from $\{1, \ldots, n\}$ and define the estimator Y for $F_g(a)$ as follows:

$$Y = n(g(r_X, n) - g(r_X - 1, n)) \; .$$

First observe that r_X can be computed by a data stream algorithm using a slight variation of Vitter's reservoir sampling algorithm: While the data stream elements are processed sequentially in Vitter's algorithm, the elements that are identical to the current sample are counted. Whenever the current sample is replaced by a new stream element, the counter is reset to the value 1. Then, in the end, the value of the counter is r_X. The value of Y is easily computed by a data stream algorithm that counts the number of stream elements, uses Vitter's algorithm to obtain r_X, and computes Y after the whole stream has been read. Clearly, this algorithm uses $O(\log n + \log m)$ bits of memory. Let c be a constant that will be defined later on. The AMS algorithm uses c independent copies $Y_1, \ldots, Y_c$ of the estimator Y and computes $Z = \frac{1}{c}\sum_{i=1}^{c} Y_i$ as the output of the algorithm. It remains to verify that Z is an unbiased estimator for $F_g(a)$ and to estimate the error probability of this estimator. Clearly, $\mathrm{E}[Z] = \mathrm{E}[Y]$ and $\mathrm{Var}[Z] = \frac{\mathrm{Var}[Y]}{c} \leq \frac{\mathrm{E}[Y^2]}{c}$. The error probability will be analyzed using Chebyshev's inequality, therefore we need to estimate $\mathrm{E}[Y^2]$. The following theorem shows that Y is an unbiased estimator for $F_g(a)$ and provides an upper bound on $\mathrm{E}[Y^2]$. Surprisingly, this upper bound is expressed in terms of $F_h(a)$ for a function h that is closely related to the function g.

Lemma 4.4.4 (Based on [3])**.** *Let $h(x, n) = n \cdot g(x, n) \cdot g'(x, n)$ where $g'(x, n) = \frac{d}{dx} g(x, n)$. The estimator Y from Definition 4.4.3 satisfies $\mathrm{E}[Y] = F_g(a)$ and $\mathrm{E}[Y^2] \leq F_h(a)$.*

Proof. In the following we will abbreviate $f_i(a)$ as f_i. First observe that $\Pr\{a_X = u_i\} = \frac{f_i}{n}$ and that $\Pr\{r_X = j | a_X = u_i\} = \frac{1}{f_i}$ for all $j \in \{1, \ldots, f_i\}$. Then, by the law of total probability,

$$\mathrm{E}[Y] = \sum_{i=1}^{m} \Pr\{a_X = u_i\} \sum_{j=1}^{f_i} \Pr\{r_X = j | a_X = u_i\} \cdot n(g(j, n) - g(j-1, n)) \tag{4.65}$$

$$= \sum_{i=1}^{m} \frac{f_i}{n} \sum_{j=1}^{f_i} \frac{1}{f_i} n(g(j, n) - g(j-1, n)) \tag{4.66}$$

$$= \sum_{i=1}^{m} \sum_{j=1}^{f_i} (g(j, n) - g(j-1, n)) \tag{4.67}$$

$$= \sum_{i=1}^{m} \sum_{j=1}^{f_i} g(f_i, n) \tag{4.68}$$

$$= F_g(a) \; . \tag{4.69}$$

For the second to the last equation observe that the inner sum is a telescope sum and that $g(0,n) = 0$.

For an estimate of $\mathrm{E}[Y^2]$, in addition to the telescope sum property, we will use that $g(x,n) - g(x-1,n) = g'(c,n)$ for some $c \in [x-1,x]$ by the mean value theorem. Then, by the fact that $g'(x,n)$ is nondecreasing in x, we have $g(x,n) - g(x-1,n) \leq g'(x,n)$. Additionally, we have $g(x,n) - g(x-1,n) \geq 0$ since g is nondecreasing, therefore we obtain

$$\begin{aligned}
\mathrm{E}[Y^2] &= \sum_{i=1}^{m} \frac{f_i}{n} \sum_{j=1}^{f_i} \frac{1}{f_i} \cdot n^2 (g(j,n) - g(j-1,n))^2 && (4.70)\\
&\leq \sum_{i=1}^{m} \sum_{j=1}^{f_i} n \cdot (g(j,n) - g(j-1,n)) \cdot g'(f_i,n) && (4.71)\\
&= \sum_{i=1}^{m} n \cdot g(f_i,n) \cdot g'(f_i,n) && (4.72)\\
&= F_h(a)\ . && (4.73)
\end{aligned}$$

□

By using the preceding lemma it is easy to analyze the space complexity of the AMS algorithm for the computation of an (ϵ,δ)-approximation of $F_g(a)$. The result is summarized in the following theorem.

Theorem 4.4.5 (Based on [3]). *Let $a = (a_1,\ldots,a_n)$ be a data stream with elements from the universe $U = \{u_1,\ldots,u_m\}$. Furthermore, let $h(x,n) = n \cdot g(x) \cdot g'(x)$ and let*

$$D = \sup\left\{ \frac{F_h(a')}{F_g(a')^2} : n' \in \mathbb{N},\ a' \text{ is a data stream of length } n' \right\}\ . \tag{4.74}$$

Let $0 < \epsilon, \delta' < 1$ be constants. The AMS algorithm with $c = \frac{D}{\delta'\epsilon^2}$ independent copies of the basic estimator Y computes an (ϵ,δ')-approximation of $F_g(a)$ using space

$$O\left(\frac{D \cdot (\log n + \log m)}{\delta'\epsilon^2} \right)\ . \tag{4.75}$$

Proof. Recall that the AMS algorithm computes the estimator Z which is the average of c independent copies of the basic estimator Y according to Definition 4.4.3. By Chebyshev's inequality, we have

$$\Pr\{|Z - \mathrm{E}[Z]| \geq \epsilon\,\mathrm{E}[Z]\} \leq \frac{\mathrm{Var}[Z]}{\epsilon^2\,\mathrm{E}[Z]^2} \leq \frac{\mathrm{E}[Y^2]}{c\,\epsilon^2\,\mathrm{E}[Y]^2}\ . \tag{4.76}$$

The output of the AMS algorithm is an (ϵ,δ')-approximation of $\mathrm{E}[Z] = F_g(a)$ if the right hand side of this inequality is bounded from above by δ'. Note that $\mathrm{E}[Y]$ and $\mathrm{E}[Y^2]$ depend on the input a of the AMS algorithm. By Lemma 4.4.4, we have

$$\frac{\mathrm{E}[Y^2]}{c\,\epsilon^2\,\mathrm{E}[Y]^2} \leq \frac{F_h(a)}{c\,\epsilon^2 F_g(a)^2}\ . \tag{4.77}$$

To obtain an (ϵ, δ')-approximation of $F_g(a)$ for every input a we have to choose c such that for all $n \in \mathbb{N}$ and all data streams of a length n

$$\frac{F_h(a)}{c\epsilon^2 F_g(a)^2} \leq \delta' \,. \tag{4.78}$$

By the definition of D, this is the case if we choose $c = \frac{D}{\delta'\epsilon^2}$. The computation of each of the c independent copies of the basic estimator Y requires $O(\log n + \log m)$ bits of memory. In total the space complexity is

$$O(c(\log n + \log m)) = O\left(\frac{D \cdot (\log n + \log m)}{\delta'\epsilon^2}\right) \,. \tag{4.79}$$

□

Now we will apply the analysis of the generalized AMS algorithm to the computation of the kth frequency moment F_k for $k \geq 1$.

Corollary 4.4.6 (Based on [3])**.** *Let $a = (a_1, \ldots, a_n)$ be a data stream with elements from the universe $U = \{u_1, \ldots, u_m\}$. The AMS algorithm computes (ϵ, δ)-approximations of $F_k(a)$ using space $O((1/\epsilon^2)\log(1/\delta)km^{1-1/k}(\log n + \log m))$.*

Proof. It is easy to verify that

$$\frac{n^k}{m^{k-1}} \leq F_k(a) \leq n^k \,. \tag{4.80}$$

It is also a well-known fact that the L^p-norm satisfies $\|x\|_i \geq \|x\|_j$ for $1 \leq i \leq j$ and, by the definition of the histogram of a data stream (Def. 4.4.1), we have $\|f(a)\|_k^k = F_k(a)$ for all $k \geq 1$. Using these observations we obtain

$$F_{2k-1}(a) \leq F_k(a)^{(2k-1)/k} = F_k(a)^{2-1/k} \leq F_k(a)^2 \left(\frac{n^k}{m^{k-1}}\right)^{-1/k} = F_k(a)^2 \frac{m^{1-1/k}}{n} \,. \tag{4.81}$$

Let $g(x, n) = x^k$ and $h(x, n) = n \cdot g(x) \cdot g'(x) = n \cdot k \cdot x^{2k-1}$. Then we have $F_g(a) = F_k(a)$ and $F_h(a) = n \cdot k \cdot F_{2k-1}(a)$ and, by our initial observations, we obtain

$$\frac{F_h(a)}{F_g(a)^2} = \frac{knF_{2k-1}(a)}{F_k(a)^2} \leq km^{1-1/k} \,. \tag{4.82}$$

Since the right hand side is independent of n, this is an upper bound on the value of D in Theorem 4.4.5. Note that it is sufficient to apply Theorem 4.4.5 for a constant error probability $\delta' < \frac{1}{2}$. Then the error probability can be reduced to any constant $\delta > 0$ by taking the median of $O(\log(1/\delta))$ independent copies of the algorithm. □

4.4.2 Frequency Moments of Very Long Data Streams

In this section we use the approximate reservoir sampling algorithm from Theorem 4.3.2 and the approximate counting algorithm from Theorem 4.2.1 to modify the AMS algorithm for the computation of the frequency moment F_k such that the contribution of the data stream length n to the space complexity of the algorithm is reduced to $O(\log \log n)$. We will state separate bounds on the space complexity of our modified algorithm for the more important *online phase* in which the data stream is read and the *offline phase* in which the output is computed. This following theorem will be proved in this section.

Theorem 4.4.7. *Let $a = (a_1, \ldots, a_n)$ be a data stream with elements from the universe $U = \{u_1, \ldots, u_m\}$. The modified AMS algorithm computes (ϵ, δ)-approximations of $F_k(a)$ using space*

$$O\left(\left(1/\epsilon^2\right)\log(1/\delta)km^{1-1/k}\left(\log(1/\epsilon) + \log k + \log m + \log\log n\right)\right) \tag{4.83}$$

in the online phase and space

$$O\left(\text{poly}(\log(1/\epsilon), \log\log(1/\delta), \log k, \log m, \log\log n)\right)$$

in the offline phase of the algorithm.

Let $a = (a_1, \ldots, a_n)$ be a data stream. Recall that the AMS algorithm uses independent copies of the basic estimator Y (see Sect. 4.4.1). The estimator Y is computed by sampling a data stream element a_X uniformly at random from a using reservoir sampling and by counting the number r_X of the following stream elements that are identical to a_X. Additionally, the length n of the data stream is counted. The basic estimator $Y = n((r_X)^k - (r_X - 1)^k)$ is computed as a function of r_X and n. We modify the data stream algorithm for the computation of the basic estimator Y as follows: We use approximate reservoir sampling to sample approximately uniformly at random a stream element $a_{\widetilde{X}}$ and use approximate counting to obtain an approximation $\widetilde{R}_{\widetilde{X}}$ of $r_{\widetilde{X}}$. Additionally, the approximate length $\widetilde{N}$ of the data stream is obtained by approximate counting. The basic estimator of the modified algorithm is $\widetilde{Y} = \widetilde{N}((\widetilde{R}_{\widetilde{X}})^k - (\widetilde{R}_{\widetilde{X}} - 1)^k)$. The output of the complete algorithm is the average $\widetilde{Z} = \frac{1}{c}\sum_{i=1}^{c}\widetilde{Y}_i$ of c copies of the basic estimator $\widetilde{Y}$ that use independent samples $\widetilde{X}_1, \ldots, \widetilde{X}_c$ and independent approximate counters for $\widetilde{R}_{\widetilde{X}_1}, \ldots, \widetilde{R}_{\widetilde{X}_c}$, but share the same approximation $\widetilde{N}$ of the data stream length n, hence $\widetilde{Y}_i = \widetilde{N}((\widetilde{R}_{\widetilde{X}_i})^k - (\widetilde{R}_{\widetilde{X}_i} - 1)^k)$. The choice of c and the approximation parameters is described in the analysis of the algorithm. Note that the computation of the estimators $\widetilde{Y}_i$ and $\widetilde{Z}$ from $\widetilde{N}$ and $\widetilde{R}_{\widetilde{X}_i}$ for $i \in \{1, \ldots, c\}$ poses a problem since the numbers that are represented by the approximate counts are $\log(n)$-bit numbers. Therefore, using only $O(\log\log n)$ bits of memory, we cannot convert $\widetilde{N}$ and $\widetilde{R}_{\widetilde{X}_i}$ to their binary representations to compute $\widetilde{Y}_i$ and finally $\widetilde{Z}$ from the approximate counts. We postpone this problem and first analyze the space complexity of the algorithm while the stream elements are processed.

Analysis of the Online Phase

In the analysis we will compare the modified AMS algorithm that uses approximate reservoir sampling and approximate counting to the original algorithm by Alon, Matias and Szegedy with the same parameters for a fixed input. The following definition summarizes the variables that will be used in this comparison.

Definition 4.4.8. Let $0 < \epsilon_1 < 1$ and $0 < \delta_1 < 1$ be fixed parameters for the AMS algorithm and let $c = \frac{1}{\delta_1 \epsilon_1^2} k m^{1-1/k}$ be the number of basic estimators that is used for these parameters in the AMS algorithm. The modified algorithm also uses c copies of its basic estimator. In the modified algorithm the length of the data stream is counted approximately using the parameters ϵ_1 and δ_1. Each basic estimator uses (ϵ_2, δ_2)-approximate reservoir sampling and approximate counting with the parameters ϵ_3 and δ_3 to obtain $\widetilde{R}_{\widetilde{X}_i}$. Let $a = (a_1, \ldots, a_n)$ be a fixed data stream with elements from the universe $U = \{u_1, \ldots, u_m\}$ and assume that we run the original and the modified AMS algorithm for the input a. Then let F_k be an abbreviation for $F_k(a)$ and for $i \in \{1, \ldots, c\}$ let

- $\widetilde{N}$ denote the approximation of the data stream length n in the modified algorithm,
- $\widetilde{X}_i$ and X_i denote the index of the data stream element that is sampled in the ith copy of the basic estimator for the modified and original algorithm, respectively, and let $\widetilde{X} = (\widetilde{X}_1, \ldots, \widetilde{X}_c)$ and $X = (X_1, \ldots, X_c)$.
- $\widetilde{R}_{\widetilde{X}_i}$ denote the approximate value of $r_{\widetilde{X}_i}$ in the modified algorithm,
- $\widetilde{Y}_i = \widetilde{N}((\widetilde{R}_{\widetilde{X}_i})^k - (\widetilde{R}_{\widetilde{X}_i} - 1)^k)$ and $Y_i = n(r_{X_i})^k - (r_{X_i} - 1)^k)$ denote the basic estimators in the modified and original algorithm, respectively, and
- let $\widetilde{Z} = \frac{1}{c}\sum_{i=1}^{c} \widetilde{Y}_i$ and $Z = \frac{1}{c}\sum_{i=1}^{c} Y_i$ denote the estimates of F_k for the modified and original algorithm, respectively.

By Corollary 4.4.6, we have

$$\Pr\{|Z - F_k| \geq \epsilon_1 F_k\} \leq \delta_1 \ . \tag{4.84}$$

Now define

$$Y_i' = n\left(r_{\widetilde{X}_i}^k - (r_{\widetilde{X}_i} - 1)^k\right) \quad \text{and} \quad Z' = \frac{1}{c}\sum_{i=1}^{v} Y_i' \ . \tag{4.85}$$

In the modified algorithm we will use approximate reservoir sampling with the parameters $\epsilon_2 = \delta_1/c$ and $\delta_2 = \delta_1/c$. Hence, for each $i \in \{1, \ldots, c\}$ with a probability of at least $(1 - \delta_2)$ the total variation distance of the distributions of $\widetilde{X}_i$ and X_i is bounded by ϵ_2. Then, by the union bound and by Proposition 2.2.38, with a probability of at least $1 - c\delta_2 = 1 - \delta_1$ the total variation distance $\mathrm{V}(X, \widetilde{X})$ of X and $\widetilde{X}$ is bounded by $c\epsilon_2 = \delta_1$. If this event happens then, by Proposition 2.2.37, we have

$$\Pr\{|Z' - F_k| \geq \epsilon_1 F_k\} \leq \delta_1 + \mathrm{V}(X, \widetilde{X}) \leq 2\delta_1 \ . \tag{4.86}$$

In the following we work under the assumption that (4.86) holds. We call this assumption *assumption 1*. Note that *assumption 1* fails to hold with a probability of at most δ_1 with respect to the random decisions of the reservoir sampling algorithm and that the random decisions of the approximate counters for $\widetilde{N}$ and $\widetilde{R}_{\widetilde{X}_i}$ are independent of this assumption.

Next, we fix a random choice $\widetilde{X} = x$ of the samples in the modified algorithm where $x = (x_1, \ldots, x_c) \in \{1, \ldots, n\}^c$. By our last assumption, with a probability of at least $1 - 2\delta$ we choose a random sample x such that under the condition that $\widetilde{X} = x$

$$|Z' - F_k| \leq \epsilon_1 F_k \ . \tag{4.87}$$

In the following we assume that (4.87) holds for the random choice $\widetilde{X} = x$ (*assumption 2*) and continue our analysis under the condition that $\widetilde{X} = x$. Note that *assumption 2* fails with a probability of at most $2\delta_1$ and that the random decisions of the approximate counters for $\widetilde{N}$ and $\widetilde{R}_{\widetilde{X}_i} = \widetilde{R}_{x_i}$ for $i \in \{1, \ldots, c\}$ are independent of *assumption 2* since each instance of the approximate counting algorithm uses independent random bits for its random decision.

The modified algorithm uses an (ϵ_1, δ_1)-approximation $\widetilde{N}$ of n in its basic estimators $\widetilde{Y}_i$, therefore $\Pr\{|\widetilde{N} - n| \geq \epsilon_1 n\} \leq \delta_1$. Let

$$Y_i'' = \widetilde{N}\left(r_{\widetilde{X}_i}^k - (r_{\widetilde{X}_i} - 1)^k\right) \quad \text{and} \quad Z'' = \frac{1}{c}\sum_{i=1}^{v} Y_i'' \,. \tag{4.88}$$

Then, with a probability of at least $1 - \delta_1$, we have

$$|Z'' - Z'| < \epsilon_1 Z' \,. \tag{4.89}$$

In the following we work under the assumption (*assumption 3*) that (4.89) holds. This assumption fails to hold with a probability of at most δ_1 and the random decisions in the computation of $\widetilde{R}_{\widetilde{X}_i}$ for $i \in \{1, \ldots, c\}$ remain independent of our current assumptions.

Before we proceed with the analysis of the online phase, we will prove a technical proposition that is useful for this analysis.

Proposition 4.4.9. *Let $\epsilon \in [0, 1]$ and $k, x \in \mathbb{N}$ such that $2k\epsilon \leq 1$ and $(1-\epsilon)x \geq 1$ and let $f(x) = x^k - (x-1)^k$. Then $f((1-\epsilon)x) \geq (1-2k\epsilon)f(x)$ and $f((1+\epsilon)x) \leq (1+2k\epsilon)f(x)$.*

Proof. By using that $(1-\epsilon)x \geq 1$, we have

$$\begin{aligned} f((1-\epsilon)x) &= (1-\epsilon)^k x^k - ((1-\epsilon)x - 1)^k && (4.90)\\ &\geq (1-\epsilon)^k x^k - ((1-\epsilon)x - (1-\epsilon))^k && (4.91)\\ &= (1-\epsilon)^k (x^k - (x-1)^k) && (4.92)\\ &= (1-\epsilon)^k f(x) \,. && (4.93) \end{aligned}$$

Now, for the proof of the first inequality, it suffices to show that $(1-\epsilon)^k \geq 1 - 2k\epsilon$. To this end let $\ell(x) = (1-x)^k$ and $r(x) = 1 - kx$. The derivatives of these functions are $\ell'(x) = -k(1-x)^{k-1}$ and $r'(x) = -k$, respectively. Now observe that $\ell(0) = r(0) = 1$, that $\ell(x)$ is convex on the unit interval since $k \geq 1$, and that $\ell'(0) = r'(0) = -k$. Hence, the linear function $r(x)$ is the asymptote of the convex function $\ell(x)$ for $x = 0$. This implies that $\ell(x) > r(x)$ for all $x \in [0, 1]$ and therefore $(1-\epsilon)^k \geq 1 - kx \geq 1 - 2k\epsilon$.

The second inequality of the lemma is shown similarly. Here we use that $(1+\epsilon)^k \leq 1 + 2k\epsilon$ for $x \in [0, 1]$ which is implied by the well known fact that $\epsilon/2 \leq \ln(1+\epsilon) \leq \epsilon$:

$$\ln\left((1+\epsilon)^k\right) = k \ln\left((1+\epsilon)\right) \leq k\epsilon \leq \ln(1+2k\epsilon) \,. \tag{4.94}$$

The inequality $(1+\epsilon)^k \leq 1 + 2k\epsilon$ follows by taking the exponential function of both sides. □

Recall that we work under the assumption that $\widetilde{X} = (x_1, \ldots, x_c)$. The modified AMS algorithm uses approximations $\widetilde{R}_{x_i}$ of r_{x_i} that are obtained by approximate counting with the parameters ϵ_3 and δ_3. Thus $\Pr\{|\widetilde{R}_{x_i} - r_{x_i}| \geq \epsilon_3 r_{x_i}\} \leq \delta_3$ for $i \in \{1, \ldots, c\}$. We

choose $\epsilon_3 = \epsilon_1/(2k)$ and $\delta_3 = \delta_1/c$. Then, by the union bound, $|\widetilde{R}_{x_i} - r_{x_i}| < \epsilon_3 r_{x_i}$ holds simultaneously for all $i \in \{1, \ldots, c\}$ with a probability of at least $1 - \delta_1$. In this case, by Proposition 4.4.9, we have $|Y_i'' - Y_i| \leq 2k\epsilon_3 Y_i = \epsilon_1 Y_i$ for all $i \in \{1, \ldots, c\}$ and therefore

$$|\widetilde{Z} - Z''| \leq \epsilon_1 Z'' \,. \tag{4.95}$$

In the following we assume that (4.95) holds (*assumption 4*). This assumption fails with a probability of at most δ_1.

If our four assumptions hold then, by the triangle inequality, the inequalities (4.87), (4.89), and (4.95), and the fact that $\epsilon_1 \leq 1$, we get the following upper bound on the approximation error of the modified AMS algorithm:

$$\begin{aligned}
|\widetilde{Z} - F_k| &\leq |\widetilde{Z} - Z''| + |Z'' - Z'| + |Z' - F_k| && (4.96)\\
&\leq \epsilon_1 Z'' + \epsilon_1 Z' + \epsilon_1 F_k && (4.97)\\
&\leq \epsilon_1(1+\epsilon_1)Z' + \epsilon_1(1+\epsilon_1)F_k + \epsilon_1 F_k && (4.98)\\
&\leq \epsilon_1(1+\epsilon_1)^2 F_k + \epsilon_1(1+\epsilon_1)F_k + \epsilon_1 F_k && (4.99)\\
&= (\epsilon_1^3 + 3\epsilon_1^2 + 3\epsilon_1)F_k && (4.100)\\
&\leq 7\epsilon_1 F_k \,. && (4.101)
\end{aligned}$$

By choosing $\epsilon_1 = \epsilon/7$ we obtain an ϵ-approximation of F_k under the condition that our four assumptions hold. By the union bound, the probability that at least one of our assumptions fails is bounded by $5\delta_1$. If we choose $\delta_1 = 1/20$ then the probability that the algorithm fails is bounded by $\frac{1}{4}$ and the failure probability can be reduced to any constant $\delta > 0$ by taking the median of $O(\log(1/\delta))$ independent copies of the algorithm.

The claim about the space complexity of the online phase in Theorem 4.4.7 is under the condition that the algorithm does not fail. This is only guaranteed if our four assumptions hold. In this case all instances of the approximate reservoir sampling algorithm respect the space bound of Theorem 4.3.2 and all instances of the approximate counting algorithm respect the space bound of Theorem 4.2.1. The improved AMS algorithm uses

$$c = \frac{1}{\delta_1 \epsilon_1^2} k m^{1-1/k} = \frac{490}{\epsilon^2} k m^{1-1/k} \tag{4.102}$$

instances of the approximate reservoir sampling algorithm and the approximate counting algorithm. The approximate reservoir sampling algorithm has the parameters $\epsilon_2 = \delta_1/c = 1/(10c)$ and $\delta_2 = \delta_1/c = 1/(10c)$. Then, by Theorem 4.3.2, each instance uses

$$\begin{aligned}
O\left(\log\frac{1}{\epsilon_2} + \log\frac{1}{\delta_2} + \log\log n\right) &= O\left(\log c + \log\log n\right) && (4.103)\\
&= O\left(\log\frac{1}{\epsilon} + \log k + \log m + \log\log n\right) && (4.104)
\end{aligned}$$

bits of memory. The parameters of approximate counting are $\epsilon_3 = \epsilon_1/(2k) = \epsilon/(14k)$ and $\delta_3 = \delta_1/c = 1/(10c)$. By Theorem 4.2.1, each instance uses

$$\begin{aligned}
O\left(\log\frac{1}{\epsilon_3} + \log\frac{1}{\delta_3} + \log\log n\right) &= O\left(\log\frac{1}{\epsilon} + \log k + \log c + \log\log n\right) && (4.105)\\
&= O\left(\log\frac{1}{\epsilon} + \log k + \log m + \log\log n\right) && (4.106)
\end{aligned}$$

bits of memory. For the purpose of probability amplification we use the median of $\log(1/\delta)$ independent copies of the algorithm. Overall, in the online phase of the algorithm

$$O\left(\left(1/\epsilon^2\right)\log(1/\delta)km^{1-1/k}\left(\log(1/\epsilon)+\log k+\log m+\log\log n\right)\right) \tag{4.107}$$

bits of memory are used under the condition that the algorithm does not fail. This proves the claim of Theorem 4.4.7 about the space complexity of the modified AMS algorithm in the online phase.

Analysis of the Offline Phase

In the offline phase we cannot simply convert $\widetilde{R}_{\widetilde{X}_i}$ and $\widetilde{N}$ to binary numbers and compute $\widetilde{Y}_i = \widetilde{N}((\widetilde{R}_{\widetilde{X}_i})^k - (\widetilde{R}_{\widetilde{X}_i}-1)^k)$ since this would increase the contribution of the data stream length n to the space complexity of the algorithm to $\Omega(\log n)$. We solve this problem by using a space efficient simulation of a uniform circuit that computes the output from $\widetilde{R}_{\widetilde{X}_i}$ and $\widetilde{N}$. An introduction to circuit complexity can be found, for example, in [11]. Here we will only need a few facts from circuit theory: A binary circuit over the standard base $\vee$,$\wedge$, and $\neg$ with fan-in at most two can be encoded as a set of tuples of the form (g, b, g_ℓ, g_r) such that g is the number of the gate that is described by the tuple, $b \in \{\vee, \wedge, \neg, \mathit{input}\}$ is the type of the gate, and g_ℓ and g_r are the numbers of the gates that are the left and right input of gate g, respectively. For a circuit with n inputs and m outputs the gates with the numbers $1, \ldots, n$ are of the type *input*. These gates represent the inputs of the circuit. The output of the circuit is computed by the gates $\{n+1, \ldots, n+m\}$. This representation of circuits is usually called the *standard encoding* of circuits. Uniform circuits are circuits whose standard encoding can be computed by deterministic Turing machines that use little space.

Definition 4.4.10. Let $S = \{S_n : n \in \mathbb{N}\}$ be a sequence of circuits with n inputs and one output. The sequence S is called $\mathrm{U_{BC}}$-uniform, or briefly uniform, if there is a deterministic Turing machine that, given the input 1^n, computes the standard encoding of the circuit S_n using space $O(\log n)$. Furthermore, let $\mathrm{U_{BC}}$-SIZE, DEPTH$(c(n), d(n))$ denote the set of all languages $L \subseteq \{0,1\}^*$ which can be decided by a $\mathrm{U_{BC}}$-uniform family $S = \{S_n : n \in \mathbb{N}\}$ of circuits of size $c(n)$ and depth $d(n)$.

The following theorem is a basic result in circuit complexity. A proof of this theorem can be found, for instance, in [10].

Theorem 4.4.11. *For $d(n) \geq \log n$ we have that*

$$\mathrm{U_{BC}}\text{-SIZE, DEPTH}\left(2^{d(n)}, d(n)\right) \subseteq \mathrm{DSPACE}(d(n)) \ .$$

This result can be easily extended to uniform families of circuits that compute families of functions $f_n \colon \{0,1\}^n \longrightarrow \{0,1\}^{m(n)}$ with $m(n) > 1$ outputs. Here each bit of the output is simply computed by a separate $\mathrm{U_{BC}}$-uniform circuit with one output. Hence, the upper bound on the space complexity in the offline phase of the improved AMS algorithm in Theorem 4.4.7 can be proved by showing that the median of $O(\log(1/\delta))$ independent copies of the estimator $\widetilde{Z}$ for F_k can be computed by a uniform circuit of depth $O\left(\mathrm{poly}(\log(1/\epsilon), \log\log(1/\delta), \log k, \log m, \log\log n)\right)$ from the register values of the associated approximate counters.

We first note that the basic operations that are needed for the computation of $\widetilde{Z}$ can be computed by uniform circuits of logarithmic depth: Addition, multiplication and integer division of two n-bit numbers can be computed by uniform circuits of depth $O(\log n)$. A uniform circuit of depth $O(\text{poly}(\log k) \cdot \log n)$ for computing the median of k numbers with n bits can be obtained by combining a uniform sorting network of depth $O(\text{poly}(\log k))$ with uniform circuits for the comparison of two n-bit integers of depth $O(\log n)$. Details on circuits for basic functions can be found in the monographs [73, 74] by Wegener.

Then we observe that the values of $\widetilde{N}$ and $\widetilde{R}_{\widetilde{X}_i}$ for $i \in \{1, \ldots, c\}$ have binary representations of length $O(\log n)$ if the algorithm does not fail since in this case all approximate counts have a bounded relative error. Let r and r_i denote the register values that corresponds to the approximate counts $\widetilde{N}$ and $\widetilde{R}_{\widetilde{X}_i}$, respectively. We will first describe a circuit that computes $\widetilde{R}_{\widetilde{X}_i}$ from r_i. Recall that the improved approximate counting algorithm uses the parameter d and the value $\beta = (2^d - 1)^{-1}$ such that

$$\widetilde{R}_{\widetilde{X}_i} = (1 + \beta)^{r_i}/\beta = \frac{2^{dr_i}}{(2^d - 1)^{r_i - 1}} \, . \tag{4.108}$$

By the approximation error of $\widetilde{R}_{\widetilde{X}_i}$ we have that $r_i = O(\log n)$, and $d = O(1/\epsilon)$ by the choice of d. Hence, the values 2^{dr_i} and $(2^d - 1)^{r_i}$ can be computed by using a binary tree of multiplications (iterated squaring) of depth $O(\log r_i) = O(\log(1/\epsilon) + \log\log n)$. Further on, 2^{dr_i} and $(2^d - 1)^{r_i - 1}$ have binary representations of length $O((1/\epsilon) \log n)$ and all intermediate values in the tree of multipliers can be multiplied by circuits of depth $O(\log(1/\epsilon) + \log\log n)$. Then the resulting circuit has the depth $O((\log(1/\epsilon) + \log\log n)^2)$. Note that this circuit is uniform if we use uniform multipliers since the tree-structure is very simple and can be computed using space $O(\log(1/\epsilon) + \log\log n)$. A uniform circuit for the computation of $\widetilde{R}_{\widetilde{X}_i}$ is obtained from the circuits for 2^{dr_i} and $(2^d - 1)^{r_i}$ by using an integer division. Note that an integer division is sufficient at this point since an absolute error of at most 1 increases the relative error of $\widetilde{R}_{\widetilde{X}_i}$ by a vanishingly small amount if $\widetilde{R}_{\widetilde{X}_i}$ grows to infinity. This increase of the approximation error can be compensated by adjusting the parameters of the AMS algorithm in Corollary 4.4.6 accordingly. A uniform circuit for the computation of $\widetilde{N}$ from the register value r is constructed analogously. The value of $\widetilde{Y}_i$ can be computed from $\widetilde{R}_{\widetilde{X}_i}$ by a uniform circuit of depth $O(\log(1/\epsilon) + \log k + \log\log n)$ that is constructed using the same ideas. The value $\widetilde{Y}_i$ and all intermediate values in its computation have binary representations of length $O(\text{poly}(\log n) + \log k)$. The estimator $\widetilde{Z}$ is the average of the c independent basic estimators $\widetilde{Y}_i$ for $i \in \{1, \ldots, c\}$ where

$$c = \frac{490}{\epsilon^2} k m^{1 - 1/k} \tag{4.109}$$

is the number of basic estimators that was chosen in the analysis of the online phase. The sum $\sum_{i=1}^{c} \widetilde{Y}_i$ can be computed using a binary tree of additions of depth

$$\log(c) = O(\log m + \log k + \log(1/\epsilon)) \, . \tag{4.110}$$

Each intermediate result in this tree has a binary representation of length

$$O(\log(1/\epsilon) + \log k + \log m + \text{poly}(\log n)) \, . \tag{4.111}$$

The depth of a circuit for a single addition in the tree is logarithmic in this length, hence the depth of the whole circuit for the sum, including the contribution of the circuits for each addition, is bounded by

$$O(\text{poly}(\log(1/\epsilon), \log k, \log m, \log\log n)) \; . \tag{4.112}$$

The division by c in the computation of $\widetilde{Z} = \frac{1}{c}\sum_{i=1}^{c} \widetilde{Y}_i$ can be realized by a simple truncation if we round the parameter c up to the nearest power of two. Like before, the vanishingly small relative error that is introduced by the truncation can be compensated by adjusting the parameters of the AMS algorithm accordingly.

The output of the improved AMS algorithm is the median of $\log(1/\delta)$ independent copies of the estimator $\widetilde{Z}$. By using a uniform sorting network of depth $O(\text{poly}(\log\log(1/\delta)))$ to sort the estimators which have binary representations of length $O(\log(1/\epsilon) + \log k + \log m + \text{poly}(\log n))$ we finally obtain a uniform circuit of depth

$$O\left(\text{poly}(\log(1/\epsilon), \log\log(1/\delta), \log k, \log m, \log\log n)\right) \tag{4.113}$$

that computes the output of the algorithm. The uniformity of the circuit follows from the fact that it combines uniform subcircuits in a very simple way: The uniform subcircuits replace nodes in very simple graphs, mostly trees, that can be computed using logarithmic space. The upper bound on the space requirements of the improved AMS algorithm in the offline phase follows from the space-efficient simulation of this circuit according to Theorem 4.4.11.

4.4.3 The Space Complexity of Data Stream Algorithms for F_k

Now we will use communication complexity to prove a lower bound on the space complexity of any randomized data stream algorithm that computes (ϵ, δ)-approximations of the kth frequency moment in a constant number of passes over the input. The following reduction was introduced by Alon, Matias, and Szegedy [3].

Theorem 4.4.12. *Let $U = \{u_1, \ldots, u_m\}$ be a set, let $k, r \in \mathbb{N}$, $0 < \epsilon < 1$, and $0 < \delta < \frac{1}{2}$ be constants, and let $t = ((1 + 3\epsilon/(1-\epsilon))m)^{1/k}$. Every data stream algorithm that computes (ϵ, δ)-approximations of $F_k(a)$ for data streams a with elements from U while making at most r passes over the data stream uses space $\Omega\left(\frac{1}{rt} \, \mathrm{R}_{\delta}^{\mathrm{NIH}}\left(\mathrm{DISJ}_{t,m}^{\mathrm{unique}}\right)\right)$.*

Proof. Suppose that the r-pass data stream algorithm A, given the data stream a with elements from U as the input, computes an (ϵ, δ)-approximation $\widetilde{F}_k(a)$ of $F_k(a)$ using s bits of memory. We will use A to construct a randomized ϵ-error t-party NIH protocol P for $\mathrm{DISJ}_{t,m}^{\mathrm{unique}}$ such that $\mathrm{cost}(P) \leq rts$. Then the claim of the theorem follows immediately from the existence of this protocol since $rts \geq \mathrm{cost}(P) \geq \mathrm{R}_{\delta}^{\mathrm{NIH}}\left(\mathrm{DISJ}_{t,m}^{\mathrm{unique}}\right)$.

Let $(x_1, \ldots, x_t) \in (\{0,1\}^m)^t$ be an input for $\mathrm{DISJ}_{t,m}^{\mathrm{unique}}$. Every player uses his input x_i to compute a partial data stream a_i that contains all elements $u_j \in U$ subject to $x_{i,j} = 1$. Each element from U is contained at most once in a_i, the order of the elements in a_i is irrelevant. Then the players simulate the data stream algorithm A on the input $a = (a_1, \ldots, a_t)$ as follows: The first player simulates A on the input a_1. Then he appends the contents of the memory that is used by algorithm A to the transcript and the second player continues the simulation on a_2 using the memory contents from the transcript, and so on. If $r > 1$ then

the last player appends the contents of the memory to the transcript and the first player continues to simulate the next pass over the input in the same manner until all r passes have been simulated. In the end the last player computes the output $\widetilde{F}_k(a)$ of A. The output of the protocol P is 1 if $\widetilde{F}_k(a) > (1+\epsilon)m$ and 0 otherwise. The last player appends the output to the transcript such that all players know the output at the end of the protocol.

We will first analyze the cost of P, then we will show that P is a δ-error protocol for $\mathrm{DISJ}^{\mathrm{unique}}_{t,m}$: For each round, except for the last round in which the last player writes a single bit, each player appends the contents of the memory that is used by algorithm A to the transcript. Hence less than rts bits are written to the blackboard. Now we show that P computes the function $\mathrm{DISJ}^{\mathrm{unique}}_{t,m}$ if we assume that $|\widetilde{F}_k(a) - F_k(a)| \leq \epsilon F_k(a)$. By the properties of A, this assumption fails to holds with a probability of δ, thus P has the desired error probability. First suppose that $\mathrm{DISJ}^{\mathrm{unique}}_{t,m}(x_1,\ldots,x_t) = 0$. Then, by the unique intersection promise, for each $j \in \{1,\ldots,m\}$ there is at most one $i \in \{1,\ldots,t\}$ such that $x_{i,j} = 1$. Thus each element from U appears at most once in a and therefore we have $F_k(a) \leq m$. Then, by our assumption, we get $\widetilde{F}_k(a) \leq (1+\epsilon)m$ and the output of P is 0. Now suppose that $\mathrm{DISJ}^{\mathrm{unique}}_{t,m}(x_1,\ldots,x_t) = 1$. In this case there is at least one $j \in \{1,\ldots,m\}$ such that $x_{i,j} = 1$ for all $i \in \{1,\ldots,t\}$. Thus at least one element from U appears t times in a and therefore $F_k(a) \geq t^k$. Then, by our assumption, we have $\widetilde{F}_k(a) \geq (1-\epsilon)t^k$ and, by using the definition of t, we obtain

$$\widetilde{F}_k(a) \geq (1-\epsilon)t^k = (1-\epsilon)(1+3\epsilon/(1-\epsilon))m = (1+2\epsilon)m > (1+\epsilon)m\,. \tag{4.114}$$

Hence the output of P is 1, concluding the proof. □

By using the reduction of Alon, Matias, and Szegedy in conjunction with our lower bound on the information complexity of the disjointness function with the unique intersection promise we obtain the following result.

Corollary 4.4.13. *Let $U = \{u_1,\ldots,u_m\}$ be a set and let $0 < \epsilon < 1$ and $0 < \delta < \frac{1}{2}$ be constants. Every data stream algorithm that computes (ϵ,δ)-approximations of $F_k(a)$ for data streams a with elements from U while making a constant number of passes over the data stream uses space $\Omega\left(m^{1-2/t}\right)$.*

Proof. Suppose that A is a r-pass data stream algorithm that computes (ϵ,δ)-approximations of $F_k(a)$. Let s be a function such that, given the data stream $a = (a_1,\ldots,a_n)$ as the input, A uses $s(m,n)$ bits of memory. By using standard probability amplification techniques, we obtain a randomized data stream algorithm A' that computes (ϵ,δ')-approximations of $F_k(a)$ for a constant δ' such that $\delta' < (3/10)\left(1-\sqrt{(1/2)\log(4/3)}\right)$ using space $O(s(m,n))$. Let $t = ((1+3\epsilon/(1-\epsilon))m)^{1/k}$. By the fact that information complexity is a lower bound on communication complexity (Thm. 3.2.5) and our lower bound on the information complexity of $\mathrm{DISJ}^{\mathrm{unique}}_{t,m}$ (Cor. 3.3.35) we have

$$\mathrm{R}^{\mathrm{NIH}}_{\delta'}\left(\mathrm{DISJ}^{\mathrm{unique}}_{t,m}\right) = \Omega(m/t)\,. \tag{4.115}$$

By applying Theorem 4.4.12 to A' we then obtain

$$s(m,n) = \Omega\left(\frac{m}{rt^2}\right) = \Omega\left(m^{1-2/k}\right)\,. \tag{4.116}$$

This concludes the proof. □

The last Corollary is only a marginal improvement on the previously known $\Omega\left(m^{1-2/k}/\log m\right)$ lower bound for a constant number of passes by Chakrabarti, Khot, and Sun [25]. Actually, upper bounds on the space complexity of algorithms that compute (ϵ, δ)-approximations of F_k are usually stated using the $\widetilde{O}$-Notation that suppresses factors of the order $\text{poly}(\log m)$ (see Sect. 4.4). Nevertheless, our lower bound shows that not even small asymptotic improvements over the previously known $\Omega\left(m^{1-2/k}\right)$ lower bound for single pass data stream algorithms can be gained by using any constant number of passes over the data stream. This possibility was not excluded by the lower bound for a constant number of passes by Chakrabarti, Khot, and Sun.

Chapter 5

Conclusions and Outlook

In the first part of this thesis we have obtained an optimal lower bound on the number in the hand multi-party information complexity of the AND function and the disjointness function with the unique intersection promise. Previous results on the information complexity of the AND function mainly used the Hellinger distance of probability distributions as the main tool of the proof. Our proof adds the Kullback-Leibler distance to the mathematical toolbox of information complexity. The close connection of the Kullback-Leibler distance and mutual information and the interesting analytical properties of the Kullback-Leibler distance enabled us to improve on the known lower bounds for the AND function and the disjointness function. We have also observed that known results in communication complexity can be generalized and simplified if the combinatorial proof is replaced by an information theoretical proof. In summary, information complexity offers a powerful and intuitively accessible approach to communication complexity in the number in the hand model. The application of this technique to new problems offers plentiful opportunities for future research.

In the number on the forehead model we were less successful. We only obtained lower bounds on the information cost of an artificially restricted subset of one-way protocols. Extending the information complexity approach to the number on the forehead model seems to be a worthwhile research goal. The information complexity of pointer jumping functions for unrestricted one-way protocols looks like a good candidate for this plan. In general, proving strong lower bounds on the communication complexity of functions in the number on the forehead model is a major open problem of communication complexity. Apparently, in many applications of communication complexity the limits of the number in the hand model have been reached. For example, the currently best time-space tradeoff results for binary branching programs by Beame, Saks, Sun, and Vee [15] are based on sophisticated refinements of two-player communication complexity. Despite these successes, strong lower bounds on the size of unrestricted binary branching programs with n input variables of depth $\Omega(n \log n)$ still seem to be out of reach for today's proof methods. First applications of number on the forehead multi-party communication complexity to branching programs by Babai, Nisan, and Szegedy [8] and Beame and Vee [16] indicate that stronger results may be obtained by using multi-party communication complexity. Additionally, new results in multi-party communication complexity could potentially solve some long-standing open problems in circuit complexity. Currently, progress for these problems is mainly hindered by the embarrassing lack of strong proof methods for lower bounds on the multi-party communication complexity of functions in the number on the forehead model.

An additional research perspective is the application of information statistics to combinatorial problems outside of communication complexity. We briefly sketched a general information statistics approach for the proof of lower bounds on the size of sets. We strongly believe that the application of information statistics to combinatorial problems has not yet been fully explored.

In the second part of this thesis we designed data stream algorithms for approximate counting and random sampling that have a doubly logarithmic space complexity. These algorithms show that nontrivial and interesting data stream algorithms can be designed even under extreme space restrictions that do not even allow to store the length of the data stream as a binary number. Our improvements on known algorithms for these problems significantly improve the utility of the algorithms as a building block for larger algorithms. The utility of our algorithms as a building block for other algorithms shows in the application to the computation of frequency moments for very long data streams, although our improvements for this problem are most likely more of theoretical interest than of practical use.

Algorithm design is driven by the needs of applications. With the seemingly infinite stream of new applications that emerge steadily, pointing out research opportunities in the field of algorithm design is almost redundant. Our results on approximate counting demonstrate that even "ancient results" like Morris' approximate counting algorithm [59] from 1978 are often not fully understood and that they offer a potential for improvements and new applications. Therefore we point out that it can be a worthwhile effort to revisit old results once in a while and to reconsider known algorithms in the light of current algorithmic topics.

Bibliography

[1] F. M. Ablayev. Lower bounds for one-way probabilistic communication complexity and their application to space complexity. *Theor. Comput. Sci.*, 157(2):139–159, 1996.

[2] S. M. Ali and S. D. Silvey. A general class of coefficients of divergence of one distribution from another. *Journal of the Royal Statistical Society, Series B*, 28(1):131–140, 1966.

[3] N. Alon, Y. Matias, and M. Szegedy. The space complexity of approximating the frequency moments. *J. Comput. Syst. Sci.*, 58(1):137–147, 1999.

[4] N. Alon and J. H. Spencer. *The Probabilistic Method*. Wiley-Interscience, second edition, 2000.

[5] S. Arora and B. Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009.

[6] L. Babai, A. Gál, P. G. Kimmel, and S. V. Lokam. Communication complexity of simultaneous messages. *SIAM J. Comput.*, 33(1):137–166, 2004.

[7] L. Babai, T. P. Hayes, and P. G. Kimmel. The cost of the missing bit: Communication complexity with help. *Combinatorica*, 21(4):455–488, 2001.

[8] L. Babai, N. Nisan, and M. Szegedy. Multiparty protocols, pseudorandom generators for logspace, and time-space trade-offs. *J. Comput. Syst. Sci.*, 45(2):204–232, 1992.

[9] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom. Models and issues in data stream systems. In *Proc. of 21st PODS*, pages 1–16, 2002.

[10] J. L. Balcázar, J. Díaz, and J. Gabarró. *Structural Complexity II*. Springer, 1990.

[11] J. L. Balcázar, J. Díaz, and J. Gabarró. *Structural Complexity I*. Springer, second edition, 1994.

[12] Z. Bar-Yossef, T. S. Jayram, R. Kumar, and D. Sivakumar. Information theory methods in communication complexity. In *Proc. of 17th CCC*, pages 93–102, 2002.

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

[14] P. Beame, T. Pitassi, N. Segerlind, and A. Wigderson. A strong direct product theorem for corruption and the multiparty communication complexity of disjointness. *Comput. Complex.*, 15(4):391–432, 2006.

[15] P. Beame, M. Saks, X. Sun, and E. Vee. Time-space trade-off lower bounds for randomized computation of decision problems. *JACM*, 50(2):154–195, 2003.

[16] P. Beame and E. Vee. Time-space tradeoffs, multiparty communication complexity, and nearest-neighbor problems. In *Proc. of 34th STOC*, pages 688–697, 2002.

[17] L. Bhuvanagiri, S. Ganguly, D. Kesh, and C. Saha. Simpler algorithm for estimating frequency moments of data streams. In *Proc. of 17th SODA*, pages 708–713, 2006.

[18] B. Bollig, M. Sauerhoff, and I. Wegener. On the nonapproximability of Boolean functions by OBDDs and read-k-times branching programs. *Information and Computation*, 178:263–278, 2002.

[19] J. Brody and A. Chakrabarti. Sublinear communication protocols for multi-party pointer jumping and a related lower bound. In *Proc. of 25th STACS*, pages 145–156, 2008.

[20] R. E. Bryant. Symbolic manipulation of Boolean functions using a graphical representation. In *Proc. of 22nd DAC*, pages 688–694, 1985.

[21] R. E. Bryant. Graph-based algorithms for Boolean function manipulation. *IEEE Transactions on Computers*, C-35(8):677–691, August 1986.

[22] R. E. Bryant. On the complexity of VLSI implementations and graph representations of Boolean functions with application to integer multiplication. *IEEE Transactions on Computers*, 40(2):205–213, 1991.

[23] L. Le Cam and G. L. Yang. *Asymptotics in Statistics*. Springer, second edition, 2000.

[24] A. Chakrabarti. Lower bounds for multi-player pointer jumping. In *Proc. of 22th CCC*, pages 33–45, 2007.

[25] A. Chakrabarti, S. Khot, and X. Sun. Near-optimal lower bounds on the multi-party communication complexity of set disjointness. In *Proc. of 18th CCC*, pages 107–117, 2003.

[26] A. Chakrabarti, Y. Shi, A. Wirth, and A. C. Yao. Informational complexity and the direct sum problem for simultaneous message complexity. In *Proc. of 42nd FOCS*, pages 270–278, 2001.

[27] A. K. Chandra, M. L. Furst, and R. J. Lipton. Multi-party protocols. In *Proc. of 15th STOC*, pages 94–99, 1983.

[28] D. Coppersmith and R. Kumar. An improved data stream algorithm for frequency moments. In *Proc. of 15th SODA*, pages 151–156, 2004.

[29] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley-Interscience, 1991.

[30] I. Csiszár. Information-type measures of difference of probability distributions and indirect observation. *Studia Sci. Math. Hungar.*, 2:229–318, 1967.

[31] I. Csiszár and P. Shields. Information theory and statistics: A tutorial. *Foundations and Trends in Communications and Information Theory*, 1(4):417–528, 2004.

[32] C. Damm, S. Jukna, and J. Sgall. Some bounds on multiparty communication complexity of pointer jumping. *Comput. Complex.*, 7(2):109–127, 1998.

[33] J. Dean and S. Ghemawat. MapReduce: Simplified data processing on large clusters. *CACM*, 51(1):107–113, 2008.

[34] D. Dolev and T. Feder. Multiparty communication complexity. In *Proc. of 30th FOCS*, pages 428–433, 1989.

[35] R. M. Fano. *Class notes for Transmission of Information, MIT Course 6.574*, 1952.

[36] R. M. Fano. *Transmission of Information*. The M.I.T. Press, 1961.

[37] J. Feldman, S. Muthukrishnan, A. Sidiropoulos, C. Stein, and Z. Svitkina. On the complexity of processing massive, unordered, distributed data. *CoRR*, abs/cs/0611108, 2006.

[38] P. Flajolet. Approximate counting: A detailed analysis. In *BIT*, pages 113–134, 1985.

[39] A. Gal and P. Gopalan. Lower bounds on streaming algorithms for approximating the length of the longest increasing subsequence. In *Proc. of 48th FOCS*, pages 294–304, 2007.

[40] S. Ganguly. Estimating frequency moments of update streams using random linear combinations. In *Proc. of 8th RANDOM*, pages 369–380, 2004.

[41] A. L. Gibbs and F. E. Su. On choosing and bounding probability metrics. *International Statistical Review*, 70:419, 2002.

[42] S. W. Golomb. Run-length encodings. *IEEE Trans. Info. Theory*, 12(3):399–401, 1966.

[43] A. Gronemeier. NOF-multiparty information complexity bounds for pointer jumping. In *MFCS*, volume 4162 of *LNCS*, pages 459–470, 2006.

[44] A. Gronemeier. Approximating Boolean functions by OBDDs. *Discrete Applied Mathematics*, 155(2):194–209, 2007.

[45] A. Gronemeier. Asymptotically optimal lower bounds on the NIH-multi-party information complexity of the AND-function and disjointness. In *Proc. of 26th STACS*, pages 505–516, 2009.

[46] A. Gronemeier and M. Sauerhoff. Applying approximate counting for computing the frequency moments of long data streams. *Theor. Comp. Sys.*, 44(3):332–348, 2009.

[47] P. Haas, J. F. Naughton, S. Seshadri, and L. Stokes. Sampling-based estimation of the number of distinct values of an attribute. In *Proc. of 21st VLDB*, pages 311–322, 1995.

[48] T. Hagerup and C. Rüb. A guided tour of chernoff bounds. *Inf. Process. Lett.*, 33(6):305–308, 1990.

[49] M. Hofri and N. Kechris. Probabilistic counting of a large number of events. Technical Report TR #UH-CS-92-23, University of Houston, 1992.

[50] P. Indyk and D. Woodruff. Optimal approximations of the frequency moments. In *Proc. of 37th STOC*, pages 202–208, 2005.

[51] T. S. Jayram. Hellinger strikes back: A note on the multi-party information complexity of AND. In *Proc. of 12th APPROX and 13th RANDOM*, volume 5687 of *LNCS*, pages 562–573, 2009.

[52] S. Jukna. *Extremal Combinatorics*. Springer, 2001.

[53] S. Kakutani. On the equivalence of infinite product measures. *Ann. Math.*, 49:214–224, 1948.

[54] S. Kullback. A lower bound for discrimination information in terms of variation. *IEEE Trans. Inform. Theory*, 4:126–127, 1967.

[55] S. Kullback and R. A. Leibler. On information and sufficiency. *Ann. Math. Stat.*, 22:79–86, 1951.

[56] E. Kushilevitz and N. Nisan. *Communication Complexity*. Cambridge University Press, 1997.

[57] F. Liese and I. Vajda. On divergences and informations in statistics and information theory. *IEEE Transactions on Information Theory*, 52(10):4394–4412, 2006.

[58] M. Mitzenmacher and E. Upfal. *Probability and Computing*. Cambridge University Press, 2005.

[59] R. Morris. Counting large numbers of events in small registers. *CACM*, 21(10):840–842, 1978.

[60] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[61] S. Muthukrishnan. *Data Streams: Algorithms and Applications*. now Publishers Inc., 2005.

[62] N. Nisan and A. Wigderson. Rounds in communication complexity revisited. *SIAM J. Comput.*, 22(1):211–219, 1993.

[63] S. Ponzio, J. Radhakrishnan, and S. Venkatesh. The communication complexity of pointer chasing. *J. Comput. Syst. Sci.*, 62(2):323–355, 2001.

[64] P. Pudlák, V. Rödl, and J. Sgall. Boolean circuits, tensor ranks, and communication complexity. *SIAM J. Comput.*, 26(3):605–633, 1997.

[65] R. Raz. The BNS-Chung criterion for multi-party communication complexity. *Comput. Complex.*, 9(2):113–122, 2000.

[66] B. Schneier. *Applied Cryptography*. John Wiley & Sons, 1996.

[67] R. Sedgewick and P. Flajolet. *An Introduction to the Analysis of Algorithms*. Addison-Wesley, 1996.

[68] A. Sgarro. Informational divergence and the dissimilarity of probability distributions. *Calcolo*, 18(3):293–302, 1981.

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

[70] C. E. Shannon and W. Weaver. *The Mathematical Theory of Communication*. University of Illinois Press, 1949.

[71] E. Viola and A. Wigderson. One-way multi-party communication lower bound for pointer jumping with applications. In *FOCS*, pages 427–437, 2007.

[72] J. S. Vitter. Random sampling with a reservoir. *ACM Trans. Math. Softw.*, 11(1):37–57, 1985.

[73] I. Wegener. *The Complexity of Boolean Functions*. Wiley-Teubner, 1987.

[74] I. Wegener. *Effiziente Algorithmen für grundlegende Funktionen*. Teubner, 1989.

[75] I. Wegener. *Branching Programs and Binary Decision Diagrams: Theory and Applications*. SIAM, Philadelphia, PA, 2000.

[76] A. C. Yao. Some complexity questions related to distributive computing (preliminary report). In *Proc. of 11th STOC*, pages 209–213, 1979.

Appendix A

Some Mathematical Facts

A.1 Conditional Independence

We will frequently use the concept of conditional independence. The events $A_1, \ldots, A_n$ are conditionally independent given an event B if the events $A_1, \ldots, A_n$ are independent with respect to the conditional distribution given that the event B happened.

Definition A.1.1 (Conditionally independent events)**.** Let $A_1, \ldots, A_n \in \Omega$ and $B \in \Omega$ be events in a probability space with the sample space Ω. The events $A_1, \ldots, A_n$ are *conditionally independent given* B if for all subsets $S \subseteq \{1, \ldots, n\}$

$$\Pr\left\{ \bigcap_{i \in S} A_i \,\middle|\, B \right\} = \prod_{i \in S} \Pr\{A_i | B\} \; .$$

Conditional independence can be generalized to random variables quite naturally.

Definition A.1.2 (Conditionally independent random variables)**.** Let $X_i \in \mathcal{X}_i$ for $i \in \{1, \ldots, n\}$ and $Y \in \mathcal{Y}$ be finite random variables. The random variables $X_1, \ldots, X_n$ are *conditionally independent given* Y if for all $S \subseteq \{1, \ldots, n\}$, $x_i \in \mathcal{X}_i$, and $y \in \mathcal{Y}$

$$\Pr\left\{ \bigwedge_{i \in S} X_i = x_i \,\middle|\, Y = y \right\} = \prod_{i \in S} \Pr\{X_i = x_i | Y = y\} \; .$$

The following proposition follows immediately from the definition of conditional independence.

Proposition A.1.3. *Let $X_1 \in \mathcal{X}_1, \ldots, X_n \in \mathcal{X}_n$ and $Y \in \mathcal{Y}$ be finite random variables such that the variables $X_1, \ldots, X_n$ are conditionally independent given Y. Furthermore, let $x_1 \in \mathcal{X}_1, \ldots, x_n \in \mathcal{X}_n$ and $y \in \mathcal{Y}$ be constants and let X_{-i} and x_{-i} denote the vectors $(X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_n)$ and $(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$, respectively. Then*

$$\Pr\{X_i = x_i | X_{-i} = x_{-i} \wedge Y = y\} = \Pr\{X_i = x_i | Y = y\} \; .$$

Proof. By the fact that $\sum_{x_i \in \mathcal{X}_i} \Pr\{X_i = x_i | Y = y\} = 1$ and by the definition of conditional

independence, we have

$$\Pr\{X_{-i} = x_{-i} | Y = y\} = \sum_{x_i \in \mathcal{X}_i} \Pr\{X_i = x_i \wedge X_{-i} = x_{-i} | Y = y\} \tag{A.1}$$

$$= \sum_{x_i \in \mathcal{X}_i} \prod_{j=1}^{n} \Pr\{X_j = x_j | Y = y\} \tag{A.2}$$

$$= \prod_{j \in \{1,\ldots,n\} - \{i\}} \Pr\{X_j = x_j | Y = y\} \; . \tag{A.3}$$

Then, by plugging this into the definition of conditional probabilities, we immediately obtain the claim of the proposition:

$$\Pr\{X_i = x_i | X_{-i} = x_{-i} \wedge Y = y\} = \frac{\Pr\{X_i = x_i \wedge X_{-i} = x_{-i} | Y = y\}}{\Pr\{X_{-i} = x_{-i} | Y = y\}} \tag{A.4}$$

$$= \frac{\prod_{j=1}^{n} \Pr\{X_j = x_j | Y = y\}}{\prod_{j \in \{1,\ldots,n\} - \{i\}} \Pr\{X_j = x_j | Y = y\}} \tag{A.5}$$

$$= \Pr\{X_i = x_i | Y = y\} \; . \tag{A.6}$$

□

A.2 Useful Inequalities

Theorem A.2.1 (Jensen's inequality)**.** *Suppose that f is a convex function and that X is a random variable. Then*

$$\mathrm{E}[f(X)] \leq f(\mathrm{E}[X]) \; .$$

If f is strictly convex, then $\mathrm{E}[f(X)] = f(\mathrm{E}[X])$ implies that $X = \mathrm{E}[X]$ with probability 1.

A proof of Jensen's inequality can be found, for instance, in [29].

Theorem A.2.2 (Chernoff bound)**.** *Let $X_1, \ldots, X_n$ be independent Poisson trials such that $\Pr\{X_i = 1\} = p_i$ for $i \in \{1, \ldots, n\}$ and let $X = \sum_{i=1}^{n} X_i$. Then, for all $0 < \delta < 1$*

$$\Pr\{X \leq (1 - \delta)\,\mathrm{E}[X]\} \leq \exp\left(-\frac{\delta^2}{2}\,\mathrm{E}[X]\right) \; .$$

There are many flavors of Chernoff bounds. An overview of Chernoff bounds is given, for example, in [58], [60], and [48]. A proof of this specific Chernoff bound can be found in [58].

Appendix B

Reference

B.1 List of Important Symbols and Notation

Symbol	Explanation	Reference	Page
$\|S\|$	Size of the set S	Sect. 2.1.1	p. 5
$\|x\|$	Absolute value of number x	Sect. 2.1.4	p. 6
$[a, b]$	Closed interval	Sect. 2.1.1	p. 5
(a, b)	Open interval	Sect. 2.1.1	p. 5
$(a, b]$	Half-open interval	Sect. 2.1.1	p. 5
$[a, b)$	Half-open interval	Sect. 2.1.1	p. 5
$X \sim Y$	Identically distributed random variables	Sect. 2.1.2	p. 5
$X \sim \mu$	X distributed w.r.t. probability mass function μ	Sect. 2.1.2	p. 5
$(X\|E)$	Conditional distribution of X given event E	Sect. 2.1.2	p. 5
AND_k	k-party AND-function	Def. 3.3.5	p. 44
$\mathrm{AND}_k^{\mathrm{unique}}$	k-party AND-function with uniqueness promise	Def. 3.3.5	p. 44
$\mathrm{C}^A(f)$	Deterministic communication complexity	Def. 3.1.2	p. 29
$\mathrm{C}^{A,\text{one-way}}(f)$	Det. one-way communication complexity	Def. 3.1.8	p. 31
$\mathrm{C}^{\mathrm{A}\to\mathrm{B}}(f)$	Det. two-player one-way communication complexity	Def. 3.1.8	p. 33
$\mathrm{cost}(P)$	Cost of communication protocol P	Def. 3.1.1	p. 28
$\mathrm{D}_{\mu,\epsilon}^A(f)$	Distributional communication complexity	Def. 3.1.5	p. 30
$\mathrm{D}_{\mu,\epsilon}^{A,\text{one-way}}(f)$	Dist. one-way communication complexity	Def. 3.1.8	p. 31
$\mathrm{D}_{\mu,\epsilon}^{\mathrm{A}\to\mathrm{B}}(f)$	Dist. one-way communication complexity	Def. 3.1.8	p. 33
$\mathrm{D}_f(p, q)$	f-divergence of probability mass function p and q	Def. 2.2.30	p. 20
$\mathrm{D}(p, q)$	Kullback-Leibler distance of p and q	Def. 2.2.33	p. 21
$\mathrm{DIC}_\epsilon^A(f; X\|D)$	Deterministic information complexity	Def. 3.2.2	p. 37
$\mathrm{DISJ}_{k,n}$	k-party disjointness function	Def. 3.3.32	p. 62
$\mathrm{DISJ}_{k,n}^{\mathrm{unique}}$	disjointness function with unique intersection promise	Def. 3.3.33	p. 62

Continued on next page...

Symbol	Explanation	Reference	Page
$\mathrm{E}[X]$	Expectation of the random variable X	Sect. 2.1.2	p. 5
$F_k(a)$	kth frequency moment of data stream a	Def. 4.4.1	p. 90
$f_i(a)$	absolute frequency of ith element in data stream a	Def. 4.4.1	p. 90
$\mathrm{H}(X)$	Entropy of random variable X	Def. 2.2.1	p. 8
$\mathrm{H}(X,Y)$	Joint entropy of random variables X and Y	Def. 2.2.8	p. 10
$\mathrm{H}(X\|Y)$	Conditional entropy of X given Y	Def. 2.2.9	p. 11
$\mathrm{h}(p,q)$	Hellinger distance of p and q	Def. 2.2.39	p. 24
$h_2(p)$	Binary entropy function	Def. 2.2.3	p. 9
$\mathrm{I}(X:Y)$	Mutual information of X and Y	Def. 2.2.18	p. 15
$\mathrm{I}(X:Y\|Z)$	Conditional mutual information of X and Y given Z	Def. 2.2.23	p. 16
$\mathrm{IC}_\epsilon^A(f;X\|D)$	Randomized information complexity	Def. 3.2.2	p. 37
$\mathrm{IC}_\epsilon^{\mathrm{A}\to\mathrm{B}}(f;X\|D)$	Randomized two-player one-way information complexity	Def. 3.2.3	p. 37
$\mathrm{icost}(P;X)$	Information cost of protocol P w.r.t. X	Def. 3.2.1	p. 36
$\mathrm{PJ}_{k,n}$	k-party pointer jumping function	Def. 3.4.1	p. 65
$\Pr\{A\}$	Probability of event A	Sect. 2.1.2	p. 5
$\Pr\{A\|B\}$	Conditional probability of A given B	Sect. 2.1.2	p. 5
$\mathrm{R}_\epsilon^A(f)$	Randomized communication complexity	Def. 3.1.4	p. 30
$\mathrm{R}_\epsilon^{A,\text{one-way}}(f)$	Rand. one-way communication complexity	Def. 3.1.8	p. 31
$\mathrm{R}_\epsilon^{\mathrm{A}\to\mathrm{B}}(f)$	Rand. two-player one-way communication complexity	Def. 3.1.8	p. 33
$\mathrm{range}(X)$	Range of the random variable X	Sect. 2.1.2	p. 5
$\mathrm{supp}(X)$	Support set of the random variable X	Sect. 2.1.2	p. 5
$\mathrm{V}(p,q)$	Total variation distance of p and q	Def. 2.2.36	p. 23
$\mathrm{Var}[X]$	Variance of the random variable X	Sect. 2.1.2	p. 5