Limitations of Non-Uniform Computational Models

Amit Chakrabarti

A Dissertation
Presented to the Faculty
of Princeton University
in Candidacy for the Degree
of Doctor of Philosophy

Recommended for Acceptance
By the Department of
Computer Science

November 2002

Abstract

The demonstration of a limitation on the abilities of a computational model is a central theme in complexity theory. Hardness results, which demonstrate such limitations, are crucial for complexity theory's efforts to classify problems into classes based on their inherent difficulty.

In this work, we discuss three well-studied models of computation and prove hardness results in each of them. The models studied here share the common feature of non-uniformity.

First, we study the complexity of the well-known problem of high dimensional approximate nearest neighbour searching in the cell probe model. We establish the first nontrivial lower bound on the complexity of this problem.

We then turn to the Boolean decision tree model and study what is arguably the most famous open problem in the model: Karp's Evasiveness Conjecture for graph properties. We make partial progress towards a settlement of this conjecture. We prove that the conjecture holds for all minor-closed graph properties and improve upon the best previously known results for subgraph containment and related properties.

Finally, we study the direct sum question in the simultaneous message model of communication. We show that the complexity of the equality function satisfies a direct sum theorem in this model: to solve m different equality problems one needs m times as many bits of communication as for one problem. We also prove some related results and generalisations.

Acknowledgments

It would be impossible to exaggerate how important my advisor, Bernard Chazelle, was in making my Princeton experience a positive one. I am grateful to Bernard for his support and encouragement at every step, and especially for his willingness to spend time with me on problems from outside his central research area of computational geometry.

Andrew Yao, with his enthusiasm for attacking challenging fundamental problems, has been an important source of inspiration for me, as well as an important source of problems! Two of the topics discussed in this thesis entered my research life thanks to discussions with Andrew.

The results presented herein are products of various collaborations. I am grateful to my co-authors in the works on which the thesis is based: Bernard Chazelle, Benjamin Gum, Subhash Khot, Alexey Lvov, Yaoyun Shi, Anthony Wirth, and Andrew Yao. I am no less grateful to the numerous researchers in and around Princeton who, through informal discussions, have influenced my outlook on computer science. Of these, I would like to specially mention Sanjeev Arora who has influenced this thesis in many indirect, subtle ways.

I am greatly indebted to Ajit A. Diwan, of I.I.T. Bombay, one of the very best teachers I have ever known, both for introducing me to the challenges and the joy of theoretical computer science, and for being a person to look up to, as I embark upon an academic career myself.

Many fellow graduate students at Princeton have enriched my experience here in non-academic ways. I would like to thank them, and especially Shoibal Chakravarthy, Yuqun Chen, Akihiro Nakao, Kedar Swadi, and Anastasios Viglas, for their support.

Finally, I would like to thank those whom I owe the most: my parents Sthiti and Aloknath Chakrabarti, for their constant love, encouragement, and confidence in me.

Amit Chakrabarti

Princeton, New Jersey

September 2002

I gratefully acknowledge NSF grant CCR-99-8817, NSF grant CCR-96-23768, ARO Grant DAAH04-96-1-0181, and DIMACS (summer research fellowship, 1998) for supporting my research work while I was at Princeton University.

Contents

List of Figures

Chapter 1

Introduction

From the very early years of computer science it has been clear that computational problems vary greatly in difficulty: some problems require less resources for a solution (i.e., can be solved more efficiently) than others. The branch of computer science known as complexity theory seeks to organise the universe of computational problems into classes based on how efficiently they can be solved. Of course, such an organisation into classes is most useful when one can prove that the classes are in fact distinct. Coming up with such proofs has proven to be extremely challenging and a great many class separation conjectures remain open to this day, the most famous example being the $\mathcal{P} \stackrel{?}{\neq} \mathcal{NP}$ conjecture.

To distinguish between complexity classes one needs to demonstrate that one problem is harder to solve than another, and to this end one must at the very least prove that one of the two is "hard" to solve. This is where hardness results, also called lower bound results, come into play.

Unfortunately, our success at proving lower bounds on the time required to solve explicit problems in a general model of computation, such as the Turing machine, has been very limited. A more precise statement is that we cannot currently demonstrate a super-linear Turing machine lower bound for any explicit problem in $\mathcal{NP}$ (the same is true if the computational model is changed to Boolean circuits). Such bounds are important not just in the light of our observations above, but also because lower bounds answer significant questions about the real world, e.g., "Is it possible to sort in linear time?" Furthermore, there are situations where a lower bound is a *positive* result. Some examples are:

- The proof that an algorithm is optimal
- The proof that a cryptosystem is secure
- Derandomisation via pseudorandom generators[1]

For further philosophical perspective on the role of hardness results in computer science, the survey paper of Allender [2] is strongly recommended.

The aforementioned lack of success notwithstanding, complexity theory does have several impressive hardness results to offer. Typically these results are obtained by working in more specialised models of computation than the Turing machine. In this thesis we present lower bound results in three such specialised models: the cell probe model, the Boolean decision tree model, and the simultaneous message communication model. Like most models in which lower bound research has had success, these models are *non-uniform*, meaning that the resulting algorithm for n-bit inputs in one of these models may have nothing in common with the algorithm for $(n + 1)$-bit inputs. Among the better-

[1]This novel application of lower bounds is based on results which show that if certain functions are hard to compute using small circuits, then there exist certain pseudorandom generators whose outputs "look random" to all algorithms in a certain class. This immediately implies that certain fast randomised algorithms can be converted into fast deterministic ones, by simply feeding them with pseudorandom bits instead of random bits, and then iterating over all random seeds for the pseudorandom generator used.

known models of computation, the Boolean circuit model has this feature; in contrast, in the Turing machine model, a single algorithm is required to work for all input sizes.

Before proceeding to introduce our results, we remark that all the lower bounds shown here are *unconditional*; in particular, they do not depend on any unproven complexity theoretic assumptions such as $\mathcal{P} \neq \mathcal{NP}$.

The Cell Probe Model

The cell probe model, first defined by Yao [64], was designed to capture algorithms for so-called "data structure problems," or "query problems." In such problems, one is first given some data to be preprocessed into a data structure to be stored in random access memory, and one must then answer queries about the data.

In this model, we consider the problem of Nearest Neighbour Searching, a basic problem in computational geometry with applications in a wide range of areas. In a typical application, the ambient space has high dimension and this seems to render the problem intractable. To overcome this "curse of dimensionality," it is common to relax the problem to Approximate Nearest Neighbour Searching (ANNS), in which one seeks only approximate answers. We consider high dimensional ANNS in its canonical setting: points on the d-dimensional Hamming cube, with the Hamming metric. We prove a lower bound of $\Omega(\log\log d/\log\log\log d)$ on the worst-case deterministic query time, with any polynomial-sized data structure. This result was first published by Chakrabarti, Chazelle, Gum, and Lvov [20]; here, we give a new presentation of our proof that is more explicit and detailed than the previously published version.

Our proof is purely combinatorial. At a rather high level, it can be described as relying upon the fact that the d-cube (for large d) is a large enough and rich enough space to allow the construction of certain "low-discrepancy" combinatorial structures within it.

The Boolean Decision Tree Model

Boolean decision trees evaluate a Boolean function of several Boolean variables by adaptively querying an oracle for the values of the variables until enough information has been obtained to compute the function. This model charges only for oracle queries. Thus, complexity in this model has to do with how much of the input needs to be revealed in order to compute the function, and not how much time it takes to perform the computation. An important and very general problem studied in this model is that of deciding a graph property. In this case, the Boolean inputs are entries of the graph's adjacency matrix; thus, a decision tree queries the potential edges of the graph.

A long standing conjecture [57] is that a decision tree for a *monotone* graph property must query every single edge; this is succinctly stated as "monotone properties are evasive." We prove that the conjecture is true (at least for large enough graphs) for properties which are closed under taking graph minors. We also improve the previous best lower bound on the decision tree complexity of subgraph containment properties: i.e., the property of deciding whether or not the input graph contains some fixed graph as a subgraph. These results were first published by Chakrabarti, Khot, and Shi [21].

Our proof uses the topological ideas of Kahn, Saks, and Sturtevant [43] but in a more sophisticated manner than in their original work. We add, on top of their ideas, certain algebraic and simple number-theoretic constructions.

The Simultaneous Message Communication Model

Communication complexity, in its canonical form, measures the amount of communication required between two parties, conventionally called Alice and Bob, to jointly compute a function on inputs split between them. The simultaneous message model, first defined by Yao [63], is a variant in which Alice and Bob do not communicate amongst themselves, but instead, send (randomised) messages to a third party, a referee, who then computes the output of the function.

We consider a "direct sum" question in this computational model. In general, a direct sum question asks whether m copies of a problem require m times as much of a resource to solve as one copy. When the answer happens to be affirmative, we have a direct sum theorem, a lower bound result. Karchmer, Raz, and Wigderson [45] show that a certain direct sum theorem in communication complexity would, if proven true, separate the complexity classes $\mathcal{NC}^1$ and $\mathcal{NC}^2$ (and thus, $\mathcal{NC}^1$ and $\mathcal{P}$). This is part of the motivation for studying direct sum questions at all.

We prove a direct sum theorem for computation of the equality function in the simultaneous message model. Equality of n-bit strings was known to have complexity $\Theta(\sqrt{n})$ in this model; we show that computing m such equality functions simultaneously has complexity $\Omega(m\sqrt{n})$. We also prove some related stronger results. It is worth noting that such a direct sum theorem for the equality function was known to be *false* in certain other communication models.

For our proof we invent an information-theoretic measure of complexity of simultaneous message protocols; we call our measure "informational complexity." Our lower bounds, and the informational complexity technique, were previously announced in the paper by Chakrabarti, Shi, Wirth, and Yao [22].

We suspect that our approach may prove to be much more important than just a tool in our particular results. Very recent research seems to confirm this.

Notes on Notation

All of the special notation introduced in this work is explained at the point it is first used. However, in the interest of providing a common reference point, we would like to note that

- all logarithms are to the base 2, unless otherwise specified,
- for an integer k, we denote the set $\{1, 2, \ldots, k\}$ by $[k]$,
- for a set S and integer k, $\binom{S}{k}$ denotes the set of all k-element subsets of S,
- $V(G)$ denotes the vertex set and $E(G)$ the edge set of graph G,
- for a random variable $\mathbf{x}$, $H(\mathbf{x})$ denotes its Shannon entropy,
- the notations $O(.)$, $\Omega(.)$, $o(.)$, and $\omega(.)$, used to describe asymptotic behaviour of functions, have their usual meanings (for precise definitions, we refer the reader to [29]), and
- the notation $\mathrm{poly}(f(n))$ is a synonym for $f(n)^{O(1)}$.

Chapter 2

The Cell Probe Model and Approximate Nearest Neighbour Searching

2.1 The Problem and the Model

2.1.1 The Problem

Nearest Neighbour Searching (henceforth, NNS) is a ubiquitous problem which is encountered in areas as diverse as molecular biology [66, 59], information retrieval [31, 60], and pattern recognition [30, 33], and understandably it has been a focus of attention lately [5, 6, 7, 13, 19, 23, 26, 27, 28, 47, 50, 67]. Typically, in these applications, the objects of interest are represented as points in Euclidean space by abstracting their features. The role of NNS is to find, in a *database* of objects, the most similar one to a given *query* object. In the applications considered, the dimension of the ambient space is usually high (anywhere from tens to thousands), and "classical computational geometry" lines of attack — such as Voronoi diagrams and other space decomposition techniques —

fail since they lead to algorithms that involve storage and/or running times exponential in the dimension. This failure is usually termed the "curse of dimensionality."

Recent research on the problem, due to Kushilevitz, Ostrovsky, and Rabani [49], Indyk and Motwani [42], and Har-Peled [40], has shown that the curse of dimensionality can be overcome by not insisting on the absolute nearest neighbour and allowing approximation. Also, for the purposes of applications, since the abstraction of objects into points in Euclidean space already involves heuristics and approximations, one might as well solve the easier Approximate Nearest Neighbour Searching (henceforth, ANNS) problem. Accordingly, we shall be concerned with the complexity of ANNS in high dimensional spaces.

We shall define ANNS in a general setting. Let $(\mathcal{M}, \mathrm{dist})$ be a metric space. Given a *database* $S \subseteq \mathcal{M}$ and a real number $\delta \geq 1$, a δ*-approximate nearest neighbour* (δ-ANN) of a *query* point $x \in \mathcal{M}$ with respect to S is any $z \in S$ such that $\mathrm{dist}(x, z) \leq \delta{\cdot}\mathrm{dist}(x, z')$, for any $z' \in S$. The parameter δ is called the *approximation factor*[1] of the problem. Given some δ, the problem is to preprocess S so as to be able to find a δ-ANN of any query point efficiently.

In this chapter, we shall be concerned with the complexity of ANNS when the underlying metric space is the d-dimensional *Hamming Cube*, i.e., we will have $\mathcal{M} = \{0, 1\}^d$ and dist will be the Hamming (or ℓ_1) distance. Our main result is a lower bound on the complexity of ANNS in this setting. Clearly, since the Hamming Cube is a subspace of $\mathbb{R}^d$ under the ℓ_1 metric, our results also apply to that case. Moreover, since Hamming and Euclidean distances are essentially equivalent on $\{0, 1\}^d$, the results carry over to Euclidean space as well.

[1]Most of the literature about ANN is concerned with algorithms that achieve approximation factors close to 1 and sometimes they use the term "ε-ANN" (for small positive ε) to mean what we would call a $(1 + \varepsilon)$-ANN.

2.1.2 The Model

ANNS is an example of what may be called "data structure problems." Such problems consist of a space X of "queries," a space Y of "databases," a space Z of answers, and a relation $R \subseteq X \times Y \times Z$ where $(x, y, z) \in R$ means that z is a valid answer to query x on database y. Given a database $y \in Y$, we are required to preprocess it so that, given any $x \in X$, we can efficiently find a $z \in Z$ such that $(x, y, z) \in R$ (we may assume w.l.o.g. that such a z always exists). For instance, in the case of ANNS, $X = Z = \mathcal{M}$, Y is the collection of all subsets of $\mathcal{M}$ (or perhaps a subcollection, e.g., subsets of a certain size), and $(x, y, z) \in R$ iff z is a δ-ANN of x with respect to y. Notice that in general there is no requirement that Y be a collection of subsets of X, or indeed, that Y be a "database" at all, in the traditional sense. For instance, Dodis [32] considers the problem of preprocessing a graph so as to answer various queries about its induced subgraphs.

The standard framework for analysing the complexity of data structure problems is Yao's *cell probe model* [64]. The model assumes that the preprocessing phase constructs a data structure (from the database y) which is represented as a table consisting of s cells each of which holds b bits. This preprocessing phase is modelled by a function $\pi : Y \to \{0, 1\}^{bs}$ which maps databases to tables. The query phase is modelled by a t-tuple of *probe functions* $(f_1, \dots, f_t)$ where the range of each f_i is $[s] := \{1, 2, \dots, s\}$. The query algorithm works as follows. Given a query x, it evaluates $k_1 = f_1(x)$ and probes the table $\pi(y)$, reading the contents of the k_1^{th} cell; let us denote these contents by $z_1 = \pi(y)[k_1]$. Next, it evaluates $k_2 = f_2(x, z_1)$ and probes the table again, obtaining $z_2 = \pi(y)[k_2]$. Proceeding in this fashion (note that $f_i : X \times \{0, 1\}^{b(i-1)} \to [s]$), the query algorithm probes the table t times and is now armed with the t values $z_1, \dots, z_t \in \{0, 1\}^b$. It then computes an answer $z = f(x, z_1, \dots, z_t) \in Z$, with the correctness requirement being that $(x, y, z) \in R$.

Such an algorithm is called a *t-probe* algorithm, or is said to require *time* t. Notice that there is no charge for preprocessing time or for the time required to compute the functions $\{f_i\}$ and f; also, it is assumed that a single cell of the table can be read in constant time. These assumptions might be unrealistically generous, but they only strengthen any lower bound results. Also, the model is so general that any lower bound in it can be trusted to apply to virtually any conceivable sequential algorithm.

The model, as defined above, only allows for *deterministic* query algorithms and not randomised ones. Furthermore, we only consider *static* data structure problems and not dynamic ones. Both extensions are easy to define and we shall not pursue them here. We refer the interested reader to Miltersen's survey [53] for these and other variants of the model.

2.1.3 The Result

Consider a cell probe algorithm for δ-ANN in the Hamming Cube $\{0,1\}^d$ for databases consisting of n points. Since we have no way to enforce an efficient preprocessing algorithm, we only require that the data structure be of polynomial size, i.e., that it have $\mathrm{poly}(n,d)$ cells. We shall allow each cell to hold $\mathrm{poly}(d)$ bits; this ensures that a point can be read in constant time. Under these conditions we prove:

Theorem 2.1.1 *Given an integer d and a real constant $\varepsilon > 0$, let δ satisfy $\delta \leq 2^{\lfloor (\log d)^{1-\varepsilon} \rfloor}$. If there is a t-probe algorithm for δ-ANN with $s = \mathrm{poly}(n,d)$ and $b = \mathrm{poly}(d)$ that works for all n, then*

$$t \quad \geq \quad \frac{\varepsilon \log \log d}{2 \log \log \log d} \,.$$

Let us compare this lower bound with the best known upper bounds for the problem. We can solve *exact* NNS with a single cell probe if we allow $s = 2^d$. Moreover, recent algorithmic results ([49]; see also [42, 40]), when adapted to our model, show that for *constant* $\delta > 1$ there is a polynomial sized table with $O(d^2)$-bit entries and a *randomised* algorithm that enables us to answer δ-ANN queries using $O(\log\log d)$ probes to the table. Although there seems to be only a small gap between this upper bound and our lower bound, the two bounds are in fact incomparable because of the randomisation; recall that our model only allows *deterministic* query algorithms. An important open theoretical question regarding ANNS is to extend our lower bound to allow randomisation.

Theorem 2.1.1 was first proved in joint work with Chazelle, Gum, and Lvov [20].

2.2 Combinatorial Roots of Cell Probe Lower Bounds

The proof of our lower bound for ANNS involves a complicated combinatorial construction. We would like to point out that nontrivial combinatorics is not a unique feature of our proof; indeed, we would like to suggest that *every* nontrivial cell probe lower bound has, at its core, a combinatorial theorem. To support this claim, we shall briefly review three other well-known results in the area and point out their combinatorial roots. This digression is not essential for an understanding of our proof but we believe that it gives the proof context.

A Cell Probe Lower Bound for Set Membership

We begin by reviewing the first such result, from Yao's seminal paper [64] where the cell probe model was introduced. This paper studied the set membership problem, which is the problem of storing an n-element subset of the universe $[m]$ so as to quickly answer queries of the form "does this number belong to the stored subset?" Formally, in the notation of §2.1.2:

$$X = [m], \;\; Y = \binom{[m]}{n}, \;\; Z = \{0, 1\}$$

$$\forall x \in X, y \in Y: \quad \begin{cases} (x, y, 1) \in R \;\text{ if } x \in y, \\ (x, y, 0) \in R \;\text{ otherwise.} \end{cases}$$

It is clear that the simple scheme of storing y in sorted order and using binary search to answer queries gives us a $\lceil \log(n+1) \rceil$-probe algorithm. Yao shows that no matter how the elements of y are stored, any cell probe algorithm requires $\lceil \log(n+1) \rceil$ probes. Note, however, that there is an important restriction that the table in the cell probe algorithm may only store elements of y. Yao generalises this slightly to allow, additionally, the storage of elements ("pointers," say) from a fixed set; the contents of these pointer cells in the table may depend on n but not on y. Yao also points out that such restrictions are necessary, and this point is driven home thoroughly by the $O(1)$-probe "perfect hashing" schemes of Fredman et al.[38].

Returning to the lower bound itself, the proof begins by observing that the $\lceil \log(n+1) \rceil$ probe requirement is easy to show if the preprocessing algorithm always stores the elements of y according to any one fixed permutation (such as sorted order) and $m \geq 2n-1$. The key step is to show that if m is large enough, there is enough room in the universe to force any preprocessing scheme to do just this on some $(2n-1)$-sized subset of the universe. In other words, if we classify databases y according to the permutation which the preprocessing algorithm uses to store them, then there is a set $y' \subseteq [m]$ of size $2n-1$ such that all databases $y \subseteq y'$ belong to the same class.

We are now down to a purely combinatorial statement, and indeed, this last statement above is a direct consequence of a famous combinatorial result: Ramsey's Theorem. Thus, at the heart of this particular cell probe lower bound lies Ramsey's Theorem.

A Bit Probe Lower Bound for Set Membership

Not every cell probe lower bound can be boiled down this cleanly to a combinatorial theorem this famous. However, there is at least one other very convincing example: the study of the same problem (set membership) by Buhrman et al. [18]. Among other things, they consider this problem in the cell probe model under the special conditions $b = 1$ (cells hold one bit each) and $t = 1$ (the query algorithm probes just one bit) and are interested in the smallest possible value of s, the number of cells (which, under these conditions, equals the number of bits) used to store the preprocessed database. It is clear that a deterministic query scheme forces $s = m$; Buhrman et al. show the startling result that if the query scheme can choose the one location that it probes *randomly*, and err with probability less than ε, then $s = O(\text{poly}(n/\varepsilon) \log m)$ is achievable. The precise upper bound on s is $O((n/\varepsilon)^2 \log m)$ if the algorithm is required to be always correct when the query does belong to the stored set (one-sided error) and improves to $O((n/\varepsilon^2) \log m)$ if the algorithm is allowed to err on all queries (two-sided error).

In the same paper, the authors prove essentially matching lower bounds for s in both these models. Let us consider the one-sided error model first. Recall that we use the notation $\pi(y)[k]$ to denote the contents of the k^{th} cell (i.e., the k^{th} bit) of the table $\pi(y)$. Associated with each query $x \in [m]$ is the set $S_x \subseteq [s]$ of table locations that the scheme probes with positive probability, on input x. A simple encoding trick ensures that we can assume that the query algorithm works as follows: on input x, choose a random $k \in S_x$ (not necessarily uniformly) and output $\pi(y)[k]$.

Since the algorithm must say "yes" when the query *is* in the database, it follows that $\pi(y)[k] = 1$ for all $k \in S_x$ with $x \in y$. The key step is to show that no S_x can be contained in the union of $\lceil n/\varepsilon \rceil$ other sets $S_{x'}$ or, technically speaking, that the set system $\{S_x\}_{x\in[m]}$ is $\lceil n/\varepsilon \rceil$-cover-free. The proof of this is simple: if S_0 (say) is contained in the union of $\lceil n/\varepsilon \rceil$ S_w's with $w \neq 0$, an averaging argument shows that some n of these other S_w's must cover a portion of S_0 that the query scheme uses with probability $\geq \varepsilon$, on input 0. Therefore, if we let the database y consist of the n corresponding w's, and supply query 0, the probability of error would be $\geq \varepsilon$: a contradiction.

We are now down to a purely combinatorial question: what is the smallest s such that the universe $[s]$ can accommodate an r-cover-free set system with m sets? This question is answered by the theorem of Dyachkov and Rykov [34, 36, 39]: their theorem says that $s \geq r^2 \log m/(2 \log r + O(1))$. Plugging in $r = \lceil n/\varepsilon \rceil$ gives us our desired lower bound: $s = \Omega\left(\frac{n^2 \log m}{\varepsilon^2 \log(n/\varepsilon)}\right)$.

A lower bound for s in the two-sided error model can also be proven by boiling the issue down to this same combinatorial theorem. Interestingly, the proof also involves *upper* bounds on the universe size required to accommodate an r-cover-free family. We shall not discuss the details here; we refer the interested reader to the paper of Buhrman et al. [18].

A Cell Probe Lower Bound for Exact NNS

We return to the problem of NNS on the d-dimensional Hamming Cube (but only the exact version, not ANNS) for our final example. For the purpose of proving a lower bound, it clearly suffices to focus on the decision version of the problem where a parameter λ is fixed beforehand and we are given a database of points to preprocess so as to quickly answer queries of the form "is there a database point within distance λ of this query point?"

We can formalise this as follows:

$$X = \{0,1\}^d, \;\; Y = \binom{\{0,1\}^d}{n}, \;\; Z = \{0,1\}$$

$$\forall x \in X, y \in Y : \quad \begin{cases} (x,y,0) \in R \;\text{ if } \exists\, x' \in y \text{ such that } \mathrm{dist}(x,x') \le \lambda, \\ (x,y,1) \in R \;\text{ otherwise.} \end{cases}$$

It is convenient to require, as we do above, that the algorithm output "0" to indicate that there *is* a nearby database point and "1" to indicate that there is not. In our standard notation, Barkol and Rabani [11] prove that either $s = 2^{\Omega(d/t)}$ or $b = n^{\Omega(1)}/t$; translated into a query time lower bound, this says that with $s = \mathrm{poly}(n,d)$ and $b = \mathrm{poly}(d)$ we must have $t = \Omega(d/\log n)$. The result holds even when the query algorithm is allowed to be randomised and err with probability less than a fixed constant. Explaining how this precise quantitative result is obtained would take us too far afield, but we shall provide some intuition for how such a lower bound boils down to combinatorial facts about $\{0,1\}^d$ and the Hamming distance.

We start by using a general result due to Miltersen et al. [54] which provides a recipe for proving lower bounds for Boolean data structure problems, i.e., problems where $Z = \{0,1\}$ and the relation R is a function $f : X \times Y \to \{0,1\}$. Suppose we regard f as a two-party communication problem where Alice gets x and Bob gets y and they must jointly compute $f(x,y)$ exchanging as few bits as possible. Reducing the communication problem to the data structure problem in the obvious way, we see that the communication problem can be solved by a protocol in which Alice sends $t\lceil \log s \rceil$ bits and Bob sends tb bits.

For finite sets M and N, the quantity $|M \cap N|/|N|$ is called the *density* of M in N. Consider f as a matrix with rows indexed by X and columns by Y. Miltersen et al. prove that for constants $\alpha, \beta > 0$, if the set $\{(x, y) : f(x, y) = 1\}$ has density at least α in $X \times Y$ and f has a randomised protocol in which Alice sends a bits and Bob sends b bits, then f has a submatrix M of dimensions at least $|X|/2^{O(a)} \times |Y|/2^{O(a+b)}$ such that the density of 0-entries in M is at most β. This result has come to be known as the *richness lemma*.

To recap, a good cell probe algorithm for R leads to a good communication protocol for the associated function f. Suppose the "matrix" of f has a significant fraction of 1-entries. Then, by the richness lemma, there is a large submatrix of f which does not have too large a fraction of 0-entries. This reduces the problem of proving a cell probe lower bound to the problem of proving two combinatorial results about the matrix of the problem:

i. It has many 1-entries.

ii. Any large submatrix of it has many 0-entries.

Here, "many" simply means constant density, and "large" is quantified by parameters that depend on the precise lower bound we wish to prove.

In general, statements of this kind can be quite unwieldy. However, particular problems may have enough structure to allow us to prove stronger combinatorial results which imply the above statements. Miltersen et al. give several examples of this in [54] and Barkol and Rabani show that the approach works for NNS too (or rather, for its decision version with parameter λ, as described earlier).

For any d, if n is chosen in the range $2^{o(d)} \cap d^{\omega(1)}$, the parameter λ can be set so that the volume of (i.e., number of points in) a radius-λ ball in the Hamming Cube is at most $2^{d+1}/n$. This means that for any query $x \in X$, the probability that a random point in X is at distance $> \lambda$ from x is at least $1 - 2/n$. Therefore, a random n-point database y has all its points at least λ away from x with constant probability. In other words, for any fixed x and random y, $f(x, y) = 1$ with constant probability. This proves the first of the statements listed above.

Proving the second statement is complicated, but let us highlight the core combinatorial result involved. For constant $\varepsilon > 0$, call points $x_1, x_2 \in X$ ε-far if $\operatorname{dist}(x_1, x_2) > \left(\frac{1}{2} - \sqrt{\varepsilon}\right) d$, and call a set $S \subseteq X$ ε-apart if all pairs of points in S are ε-far. The key fact is that there exists a $\delta = \delta(\varepsilon)$ such that for any ε-apart set S, $|S| \geq n^{\delta}$, the probability that a random point in X is within distance λ of S is at least $1/(2n^{1-\delta})$. This in turn follows from a careful estimation of volume of the intersection of two radius-λ balls whose centres are ε-far. Armed with this fact one can derive the second statement listed above, with a "large" submatrix being defined as one with dimensions $n^{\delta}2^{d-\Omega(d)} \times 2^{nd-\Omega(n^{\delta})}$, although considerable extra effort is required. This particular quantification of "large," together with the richness lemma, leads to an $\Omega(d/\log n)$ query time lower bound in the cell probe model.

2.3 Proof of the Lower Bound for ANNS

Having sampled some of the diverse combinatorial ideas involved in the proofs of other cell probe lower bounds, we now turn to the proof of the result stated in §2.1.3 (Theorem 2.1.1). Recall the notation introduced in §2.1.2. A formal definition of the δ-ANNS problem with database size n is the following.

$$X = \{0,1\}^d, \;\; Y = \binom{\{0,1\}^d}{n}, \;\; Z = \{0,1\}^d$$
$$\forall x, z \in X, y \in Y: \quad (x, y, z) \in R \Longleftrightarrow \forall z' \in y \;\; (\operatorname{dist}(x, z) \leq \delta \cdot \operatorname{dist}(x, z'))\,.$$

We set the relevant parameters as follows:

$$
\begin{aligned}
d &= \text{a large enough power of } 2, && (2.1)\\
s &= (dn)^{\lambda}, && (2.2)\\
b &= d^{\mu}, && (2.3)\\
\beta &= 16 \cdot 2^{\lfloor (\log d)^{1-\varepsilon} \rfloor}, && (2.4)\\
t &= \left\lceil \frac{\varepsilon \log\log d}{2 \log\log\log d} \right\rceil, && (2.5)
\end{aligned}
$$

where λ, μ and ε are arbitrary positive constants. We shall assume that, contrary to the claim of Theorem 2.1.1, there is a $(t-1)$-probe algorithm $\mathcal{A}$ for $\frac{\beta}{16}$-ANNS on X which uses an s-cell table with b-bit cells and works for any database size n. Let us outline our plan for deriving a contradiction.

2.3.1 The Plan of the Proof

Recall that with its r^{th} probe $\mathcal{A}$ examines the cell with index $k_r = f_r(x, z_1, \ldots, z_{r-1})$ and obtains the value $z_r = \pi(y)[k_r]$. We call the sequence

$$((k_1, z_1), (k_2, z_2), \ldots, (k_{t-1}, z_{t-1}))$$

the *trace* of $\mathcal{A}$ on input (x, y). Notice that we can assume w.l.o.g. that the last cell probed by $\mathcal{A}$ contains the output that $\mathcal{A}$ will give (any algorithm can be made to satisfy this property by adding 1 to its query time). Under this assumption the trace of $\mathcal{A}$ on input (x, y) clearly determines $\mathcal{A}$'s output. Therefore, the following claim will prove Theorem 2.1.1:

Claim 2.3.1 *For a suitable choice of n, there exist queries $x_1, x_2 \in X$ and a database $y \in Y$ such that the traces of $\mathcal{A}$ on inputs (x_1, y) and (x_2, y) are identical, yet there is no $z \in X$ which is a $\frac{\beta}{16}$-ANN of both x_1 and x_2 with respect to y.*

We come up with x_1, x_2 and y as follows. We start with two sets $Q_1 \subseteq X$ and $D_1 \subseteq Y$ with certain desirable properties. Based on the preprocessing function π and the probe function f_1, we identify sets $Q_2 \subseteq Q_1$ and $D_2 \subseteq D_1$, as well as values $k_1 \in [s]$ and $z_1 \in \{0,1\}^b$, such that

$$\forall (x, y) \in Q_2 \times D_2, \ \ f_1(x) = k_1 \text{ and } \pi(y)[k_1] = z_1 \,.$$

Next, we identify sets $Q_3 \subseteq Q_2$ and $D_3 \subseteq D_2$, as well as values $k_2 \in [s]$ and $z_2 \in \{0,1\}^b$, such that $\forall (x, y) \in Q_3 \times D_3$, $f_2(x, z_1) = k_2$ and $\pi(y)[k_2] = z_2$. Proceeding in this fashion, we produce the following four sequences:

$$Q_1 \supseteq Q_2 \supseteq \cdots \supseteq Q_t \,;$$

$$D_1 \supseteq D_2 \supseteq \cdots \supseteq D_t \,;$$

$$k_1, k_2, \ldots, k_{t-1} \,;$$

$$z_1, z_2, \ldots, z_{t-1} \,.$$

We require that these sequences satisfy the following invariant.

- TRACE INVARIANT: For $r \in [t]$, the trace of $\mathcal{A}$ on any input in $Q_r \times D_r$ has prefix $((k_1, z_1), \ldots, (k_{r-1}, z_{r-1}))$.

Eventually, we end up with a set Q_t of queries and a set D_t of databases, such that all inputs from $Q_t \times D_t$ produce identical traces for $\mathcal{A}$. We then show that there exist $x_1, x_2 \in Q_t$ and $y \in D_t$ which satisfy the conditions of Claim 2.3.1.

It remains to indicate how the choices of Q_r, D_r, k_r and z_r ($r \in [t]$) are made and how this proves the existence of the above x_1, x_2 and y. This is where the combinatorics and the geometry of the Hamming Cube enter the picture.

2.3.2 Combinatorial Preliminaries

The main combinatorial object we construct is a hierarchy $\mathcal{H}$ of balls in $X = \{0,1\}^d$. Formally, $\mathcal{H}$ is a rooted tree whose vertices are balls with the property that if a ball (vertex) v is a child of another ball (vertex) w, then $v \subseteq w$. Any two balls which are siblings will be required to be "far apart" in a sense we shall make precise later. Our proof will rely crucially on three quantities being large enough: the depth of $\mathcal{H}$, the degree of a non-leaf vertex in $\mathcal{H}$, and the minimum distance between balls which are sibling vertices in $\mathcal{H}$.

For each $r \in [t]$, along with our choices of Q_r and D_r, we choose a special ball v_r from $\mathcal{H}$. We shall ensure that $Q_r \subseteq v_r$. We shall also ensure that for any database $y \in D_r$, the portion of y that lies outside v_r is irrelevant to ANNS queries from the set Q_r. Since all the action at this stage is happening inside the ball v_r, we call v_r an *arena*. Like the sequences $\{Q_r\}$ and $\{D_r\}$, the arenas form a nested sequence:

$$v_1 \supseteq v_2 \supseteq \cdots \supseteq v_t \,.$$

We shall now describe the construction of $\mathcal{H}$ precisely. We begin with a geometric fact about the hypercube.

Definition 2.3.2 *A family of balls is said to be γ-separated if the distance between any two points belonging to distinct balls in the family is more than γ times the distance* *between any two points belonging to any one ball in the family. Here γ is any positive real quantity.*

Lemma 2.3.3 *There is an absolute constant k_0 such that for any ball $B \subseteq \{0,1\}^d$ of radius k, $k_0 \leq k \leq d$, and any $\gamma \geq 16$ there exists a $\frac{\gamma}{16}$-separated family of balls within B, such that the size of the family is at least $2^{k/13}$ and the radius of each ball in the family is k/γ.*

Proof: We use an argument similar to the proof of Shannon's Theorem. Let V_r be the volume of (i.e. the number of points in) a ball in $\{0,1\}^d$ of radius r, centred at a point in $\{0,1\}^d$. (Notice that this number does not depend on the centre). Clearly

$$V_r = \sum_{i=0}^{\lfloor r \rfloor} \binom{d}{i}.$$

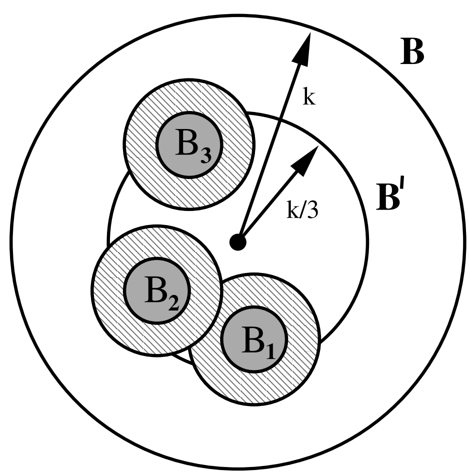

Figure 2.1: Picking the separated family of balls $B_1, B_2, \ldots$. The marked points are indicated by hatching; the picked balls by solid fill.

Consider the ball B', concentric with B and of radius $k/3$, and call its points initially *unmarked*. We proceed to mark the points of B' as follows: while there is an unmarked point left in B', pick one and mark all the points at distance at most $k/4$ from that point (see Figure 2.1). The number N of points we pick in B' satisfies

$$N \;\geq\; \frac{V_{k/3}}{V_{k/4}}\,.$$

We can bound N from below:

$$N \;\geq\; \frac{V_{k/3}}{V_{k/4}} \;=\; \frac{\sum_{i=0}^{\lfloor k/3 \rfloor} \binom{d}{i}}{\sum_{i=0}^{\lfloor k/4 \rfloor} \binom{d}{i}} \;\geq\; \frac{\binom{d}{\lfloor k/3 \rfloor}}{\sum_{i=0}^{\lfloor k/4 \rfloor} \binom{d}{i}}\,.$$

Note that in each term of the sum in the denominator i is at most $d/4$. For such i,

$$\frac{\binom{d}{i}}{\binom{d}{i-1}} \;=\; \frac{d-i+1}{i} \;\geq\; 3,$$

so

$$\sum_{i=0}^{\lfloor k/4 \rfloor} \binom{d}{i} \;\leq\; \frac{3}{2}\binom{d}{\lfloor k/4 \rfloor}\,.$$

This gives

$$\begin{aligned}
N \;&\geq\; \frac{2}{3} \cdot \frac{\binom{d}{\lfloor k/3 \rfloor}}{\binom{d}{\lfloor k/4 \rfloor}} \\
&=\; \frac{2}{3} \cdot \prod_{i=\lfloor k/4 \rfloor}^{\lfloor k/3 \rfloor} \frac{d-i+1}{i} \\
&\geq\; \frac{2}{3} \cdot \big(\underbrace{2 \times 2 \times \cdots \times 2}_{k/12-2 \text{ factors}}\big) \\
&\geq\; 2^{k/12-3},
\end{aligned}$$

and for large enough k, this implies $N \geq 2^{k/13}$.

Now pick balls of radius k/γ centred at the N picked points; their centres are in B' and their common radius is at most $k/16$, so these balls lie within B. Moreover, it is easy to see that they form a $\frac{\gamma}{16}$-separated family. To see why, suppose on the contrary that two points p and q in balls centred at distinct points p_0 and q_0 lie within $k/8$ of each other. Then,

$$\begin{aligned}
\operatorname{dist}(p_0, q_0) &\leq \operatorname{dist}(p_0, p) + \operatorname{dist}(p, q) + \operatorname{dist}(q, q_0) \\
&\leq k/\gamma + k/8 + k/\gamma \\
&\leq k/4\,,
\end{aligned}$$

since $\gamma \geq 16$. But this is a contradiction since, by construction, $\operatorname{dist}(p_0, q_0) > k/4$. □

Corollary 2.3.4 *For k divisible by β, there exists a $\frac{\beta}{16}$-separated family of radius-(k/β) balls within B, of size $2^{k/\beta}$.* □

We are ready to define the tree $\mathcal{H}$. Let us set

$$h \;=\; \lambda(\mu + 5)t\,, \tag{2.6}$$

where λ and μ are the constants from (2.2) and (2.3). The root of $\mathcal{H}$ is the radius-d ball centred at $(0, \ldots, 0)$, i.e., the entire Hamming Cube X. Choose any $2^{d/\beta}$ radius-(d/β) balls within this root ball such that they form a $\frac{\beta}{16}$-separated family; Corollary 2.3.4 guarantees that this can be done. We make these sub-balls the children of the root in $\mathcal{H}$. Next, we apply Corollary 2.3.4 to each of these children and thus create the depth-2 vertices (balls) of $\mathcal{H}$, and so on. In general, a vertex of depth k (root being of depth 0) is a ball of radius d/β^k and has $2^{d/\beta^{k+1}}$ children. We continue this construction until the leaves of $\mathcal{H}$ are of depth h^{t-1}.

We have to ensure that the balls shrink slowly enough to allow us to construct a tree of this depth. Showing that $d/\beta^{h^{t-1}} = \omega(1)$ will suffice, for we can then continue to apply Corollary 2.3.4 right up to depth h^{t-1}. In fact, some routine algebra with the expressions for h, t, and β shows the stronger statement

$$\frac{d}{\beta^{h^{t-1}}} \quad > \quad \sqrt{d}\,. \tag{2.7}$$

Remark: For the above inequality to hold, with $\beta = \Omega(1)$ and $h = \Theta(t)$, we must have $t = O(\log\log d/\log\log\log d)$. Replacing $\sqrt{d}$ with $\omega(1)$ does not help. This is what limits the strength of Theorem 2.1.1.

Fact 2.3.5 *Any two disjoint balls of $\mathcal{H}$ form a $\frac{\beta}{16}$-separated family.* $\square$

The hierarchy $\mathcal{H}$ is used to build other *derived* hierarchies. Let v be a non-leaf vertex of $\mathcal{H}$ whose depth (root being of depth 0) is divisible by h^{t-r} and not divisible by h^{t-r+1}, where $r \in [t-1]$. We define the $\mathcal{H}_r(v)$, the r^{th} *derived hierarchy at* v, to be the following tree. The root of $\mathcal{H}_r(v)$ is v. The descendants (in $\mathcal{H}$) of v at distance h^{t-r-1} from v are made the children of v in $\mathcal{H}_r(v)$. For each of these children, *its* descendants (in $\mathcal{H}$) h^{t-r-1} levels below are made its children in $\mathcal{H}_r(v)$, and so on. We continue this construction until the leaves of $\mathcal{H}_r(v)$ are of depth h, in $\mathcal{H}_r(v)$. Stated another way, if v is of depth kh^{t-r} in $\mathcal{H}$ (where k is not divisible by h), then the vertices of $\mathcal{H}_r(v)$ are precisely the vertices at depths

$$kh^{t-r},\ (kh+1)h^{t-r-1},\ (kh+2)h^{t-r-1},\ \ldots,\ (k+1)h^{t-r}$$

in $\mathcal{H}$ that belong to the subtree of $\mathcal{H}$ rooted at v.

Also, for any vertex v of $\mathcal{H}$ whose depth is not divisible by h, we define $\mathcal{H}_t(v)$ to be the depth-2 tree formed by v and its children in $\mathcal{H}$. Notice that:

i. For any non-leaf vertex v of $\mathcal{H}$ there is precisely one r for which $\mathcal{H}_r(v)$ is defined. We shall say that v is of *type* r.

ii. The root of $\mathcal{H}$ is the only vertex of type 1.

iii. For $r < t$, whenever $\mathcal{H}_r(v)$ is defined, it is a tree of depth exactly h.

iv. For $r < t$, if v is of type r and v' is an internal vertex of $\mathcal{H}_r(v)$, then v' is of type $(r+1)$. Moreover, the leaves of $\mathcal{H}_{r+1}(v')$ are the children of v' in $\mathcal{H}_r(v)$.

In the sequel, we shall be interested especially in the trees $\mathcal{H}_r(v_r)$ where $\{v_i\}$ is the sequence of arenas mentioned earlier; needless to say, we shall ensure that v_r is of type r. We shall need the following degree lower bound for these derived hierarchies:

Lemma 2.3.6 *Any internal vertex of any $\mathcal{H}_r(v)$ has more than $2^{\sqrt{d}}$ children.*

Proof: Observe that any $\mathcal{H}_t(v)$ is a subtree of some $\mathcal{H}_{t-1}(v')$; therefore it suffices to consider the case $r \in [t-1]$. Pick any such r and consider any internal vertex w of any $\mathcal{H}_r(v)$. It has as many children in $\mathcal{H}_r(v)$ as it has level-h^{t-r-1} descendants in $\mathcal{H}$. This latter number is

$$\prod_{i=1}^{h^{t-r-1}} 2^{d/\beta^{k+i}},$$

where k is the depth of v in $\mathcal{H}$. This expression is a decreasing function of both r and k. Since $r \leq t-1$ and $k \leq h^{t-1} - 1$, the expression is at least $2^{d/\beta^{h^{t-1}}}$ which, by (2.7), is greater than $2^{\sqrt{d}}$. $\square$

The Structure of Q_r and D_r

We now explain the constructions of Q_1 and D_1. Recall that these two sets do not depend on the algorithm $\mathcal{A}$ at all. The set Q_1 is very simple to describe: it consists of the centres of the balls which are leaves of $\mathcal{H}$.[2]

On the other hand, D_1 is constructed in a somewhat involved recursive fashion. To describe it, we need to introduce the notion of an r*-fragment*, for $r \in [t]$. Let v be a ball such that $\mathcal{H}_r(v)$ is defined. An r-fragment within $\mathcal{H}_r(v)$ is defined as follows:

- For $r < t$, choose any family F of *disjoint* balls (vertices) from $\mathcal{H}_r(v)$ such that F contains exactly $d^{\mu+3}$ depth-k balls, for each $k \in [h-1]$; here μ is the constant from (2.3). For each $v' \in F$, choose an $(r+1)$-fragment within $\mathcal{H}_{r+1}(v')$. The union of these $(r+1)$-fragments is defined to be an r-fragment within $\mathcal{H}_r(v)$; we say that this particular r-fragment *picks* the balls in F.

- For $r = t$, choose any $d^{\mu+3}$ balls from among the leaves of the depth-2 tree $\mathcal{H}_t(v)$. The set of centres of these balls is defined to be a t-fragment within $\mathcal{H}_t(v)$; we say that this particular t-fragment *picks* the above balls.

The first arena, v_1, is defined to be the root of $\mathcal{H}$. The set D_1 is defined to be the set of all possible 1-fragments within $\mathcal{H}_1(v_1)$.

It is easy to compute the size of a 1-fragment; doing so gives us the expression for n, the database size for which we want the algorithm $\mathcal{A}$ to work:

$$n \;=\; (h-1)^{t-1} d^{(\mu+3)t}\,. \tag{2.8}$$

[2] The reader may be concerned about whether the "centre" of a Hamming ball is well defined. It is easy to show that, except for the radius-d ball, any Hamming ball in $\{0,1\}^d$ does have a unique centre.

As mentioned earlier, as far as the ANNS problem is concerned, all the action for inputs in $Q_r \times D_r$ happens inside the arena v_r. We now formalise this statement.

Definition 2.3.7 *Suppose vertex v of $\mathcal{H}$ is of type q, with $2 \le q \le t-1$, and suppose $y \in D_1$ is a database which includes a $(q-1)$-fragment which picks v. Then the q-fragment $y \cap v$ is called the* restriction *of y to v. If $D \subseteq D_1$ is a set of databases, then the set of all q-fragments which are restrictions to v of at least one database from D is called the* restriction *of D to v.*

We denote by $\widetilde{D}_r$ the restriction of D_r to the arena v_r; thus, the elements of $\widetilde{D}_r$ are r-fragments within $\mathcal{H}_r(v_r)$. We also define $\widetilde{D}_1 = D_1$. Notice that $\widetilde{D}_1$ consists of *all* possible 1-fragments within $\mathcal{H}_1(v_1)$, but for general r, $\widetilde{D}_r$ may not contain all possible r-fragments within $\mathcal{H}_r(v_r)$.

Lemma 2.3.8 *Suppose $\tilde{y} \in \widetilde{D}_t$ is a t-fragment and suppose $y \in D_t$ is a database whose restriction to v_t is $\tilde{y}$. Then, for any query $x \in Q_t$, any $\frac{\beta}{16}$-ANN of x with respect to y must belong to $\tilde{y}$. Therefore, the set of all $\frac{\beta}{16}$-ANNs of x with respect to y is equal to the set of all $\frac{\beta}{16}$-ANNs of x with respect to $\tilde{y}$.*

Proof: Suppose, to the contrary, that $z \in \tilde{y} \setminus y$ is a $\frac{\beta}{16}$-ANN of x with respect to y. Choose an arbitrary point $\tilde{z} \in \tilde{y}$. Then x and $\tilde{z}$ lie inside the arena v_t but z does not.

Let r be the largest index in $[t]$ such that z and $\tilde{z}$ both lie inside v_r. Then the r-fragment $y \cap v_r$ must pick distinct — and therefore disjoint — balls v, $\tilde{v}$ from $\mathcal{H}_r(v_r)$ such that $z \in v$ and $\tilde{z} \in \tilde{v}$. The point $\tilde{z}$ lies in both the balls v_t and $\tilde{v}$; therefore, by construction of $\mathcal{H}$, $v_t \subseteq \tilde{v}$. Thus $\tilde{v} \supseteq v_t \supseteq Q_t \ni x$. By Fact 2.3.5, the pair $\{v, \tilde{v}\}$ is a $\frac{\beta}{16}$-separated family, whence

$$\mathrm{dist}(x, z) \quad > \quad \frac{\beta}{16} \cdot \mathrm{dist}(x, \tilde{z}) \, ,$$

contrary to the definition of a $\frac{\beta}{16}$-ANN. $\square$

The above lemma permits us to essentially forget about D_r and just work with $\widetilde{D}_r$. Indeed, we can reduce Claim 2.3.1 to the following claim.

Claim 2.3.9 *There exist queries $x_1, x_2 \in Q_t$ and a t-fragment $\tilde{y} \in \widetilde{D}_t$ so that there is no $z \in \tilde{y}$ which is a $\frac{\beta}{16}$-ANN of both x_1 and x_2 with respect to $\tilde{y}$.*

For $r > 1$, the sets Q_r and D_r will no longer have as nice a structure as Q_1 and D_1, because we are limited by the trace invariant from §2.3.1. However, we still have enough room for manoeuvre to enforce the following two additional invariants for all r:

- QUERY INVARIANT: The set Q_r intersects at least a $1/d$ fraction of the balls which are the leaves of $\mathcal{H}_r(v_r)$.
- DATABASE INVARIANT: The set $\widetilde{D}_r$ contains at least a $2^{-2d^{\mu}}$ fraction of all possible r-fragments within $\mathcal{H}_r(v_r)$.

We shall construct the sequences $\{Q_r\}$ and $\{D_r\}$ according to a strategy which ensures that these invariants are satisfied. We shall then show that if the invariants are satisfied at $r = t$ then Claim 2.3.9 is true. In the next section we prove three lemmas that form the backbone of our strategy.

2.3.3 Tools for Maintaining the Invariants

Recall that Q_r is constructed based upon the preprocessing function π and the first $r-1$ probe functions $f_1, \ldots, f_{r-1}$ of the algorithm $\mathcal{A}$. We must then construct Q_{r+1} so that f_r is constant on Q_{r+1}. Recall that the range of f_r is $[s]$. Consider the partition $Q_r = Q_r^{(1)} \cup \cdots \cup Q_r^{(s)}$ where

$$Q_r^{(i)} = \{x \in Q_r : f_r(x, z_1, \ldots, z_{r-1}) = i\}. \tag{2.9}$$

Definition 2.3.10 *A ball of $\mathcal{H}_r(v_r)$ is said to be i-*dense *if it is neither the root nor a leaf and at least a $1/d$ fraction of its children intersect $Q_r^{(i)}$. A ball is said to be* dense *if it is i-dense for some $i \in [s]$.*

Lemma 2.3.11 *The dense balls of $\mathcal{H}_r(v_r)$ cover at least a $1/(2d)$ fraction of its leaves.*

Proof: Let us fix an i and consider $Q_r^{(i)}$. Any ball of $\mathcal{H}_r(v_r)$ that intersects $Q_r^{(i)}$ is said to be *coloured*. Any ball which is covered by an i-dense ball (note that a ball covers itself) is said to be *marked*. For $k \in [h]$, let α_k be the fraction of leaves of $\mathcal{H}_r(v_r)$ whose depth-k ancestor in $\mathcal{H}_r(v_r)$ is coloured and unmarked (note that any vertex is its own ancestor, and recall that the root has depth 0). For $k > 1$, an unmarked coloured depth-k ball is the child of a coloured depth-($k-1$) ball that is *not* i-dense. If a ball is not i-dense then, by definition, at most a $1/d$ fraction of its children are coloured; it follows that $\alpha_k < \alpha_{k-1}/d$. Therefore

$$\alpha_h \;<\; \frac{\alpha_{h-1}}{d} \;<\; \frac{\alpha_{h-2}}{d^2} \;<\; \cdots \;<\; \frac{\alpha_1}{d^{h-1}} \;\leq\; \frac{1}{d^{h-1}}\,.$$

We have shown that at most a $1/d^{h-1}$ fraction of leaves intersect $Q_r^{(i)}$ but do not have an i-dense ancestor. Repeating this argument for all $i \in [s]$ shows that at most an s/d^{h-1} fraction of leaves intersect Q_r but do not have a dense ancestor. We now upper bound this fraction using the formulae for s, t, h and n (see equations (2.2), (2.5), (2.6), and (2.8)):

$$\begin{aligned}
\frac{s}{d^{h-1}} &< (dn)^{\lambda} d^{-h} \\
&\leq \left(d \cdot h^{t-1} \cdot d^{(\mu+3)t}\right)^{\lambda} \cdot d^{-h} \\
&\leq \left(d \cdot d^{t-1} \cdot d^{(\mu+3)t}\right)^{\lambda} \cdot d^{-h} \\
&= d^{\lambda(\mu+4)t-h} \\
&= d^{-\lambda t}
\end{aligned}$$

$$\leq \frac{1}{2d}.$$

Using the query invariant, we see that at least a

$$\frac{1}{d} - \frac{s}{d^{h-1}} > \frac{1}{2d}$$

fraction of leaves *do* have a dense ancestor, as required. □

Lemma 2.3.12 *There exists an $\ell \in [h-1]$ such that at least a $2^{-2d^{\mu}-1}$ fraction of the r-fragments within $\mathcal{H}_r(v_r)$ satisfy the following properties: (i) they belong to $\widetilde{D}_r$, and (ii) each of them picks at least $3d^{\mu}$ dense depth-ℓ vertices (balls) of $\mathcal{H}_r(v_r)$.*

We shall prove this lemma using the probabilistic method, for which we need the following tail estimate for the hypergeometric distribution. It can be proven directly from Theorems 1 and 4 in the paper of Hoeffding [41].

Fact 2.3.13 *Let m, N, p be integers with $m < N$ and $p < N$. Let S be a random m-element subset of $[N]$, all m-element subsets being equally likely. Then, for any real $t > 0$, we have* $\Pr\left[\frac{|S \cap [p]|}{|S|} \leq \frac{p}{N} - t\right] \leq e^{-2mt^2}$. □

Proof of Lemma 2.3.12: An α fraction of the balls at any one level of $\mathcal{H}_r(v_r)$ covers exactly an α fraction of the leaves, due to the symmetry in the construction of the hierarchies. Therefore, by Lemma 2.3.11 and the pigeonhole principle, there exists an $\ell \in [h-1]$ such that at least a $1/(2dh)$ fraction of the depth-ℓ balls are dense.

Let φ be a random r-fragment chosen uniformly from amongst all r-fragments within $\mathcal{H}_r(v_r)$. We claim that φ fails to satisfy property (ii) with probability at most $2^{-2d^{\mu}-1}$. The database invariant would then imply that φ satisfies both properties (i) and (ii) with probability at least $2^{-2d^{\mu}-1}$, as required.

Let $\mathbf{W}$ denote the (random) set of depth-ℓ balls that are not covered by any ball of depth less than ℓ picked by $\boldsymbol{\varphi}$; it is clearly enough to prove our claim for each fixed value of $\mathbf{W}$. Once $\mathbf{W}$ is fixed, the distribution of the depth-ℓ balls picked by $\boldsymbol{\varphi}$ is uniform over all $d^{\mu+3}$-sized subsets of $\mathbf{W}$. At each level $\ell' < \ell$, $\boldsymbol{\varphi}$ picks exactly $d^{\mu+3}$ balls; by Lemma 2.3.6, this represents at most a $d^{\mu+3}2^{-\sqrt{d}}$ fraction of all the depth-ℓ' balls, and therefore, these balls cover at most a $d^{\mu+3}2^{-\sqrt{d}}$ fraction of the depth-ℓ balls. Summing over all $\ell' < \ell$ we see that at most a $(\ell-1)d^{\mu+3}2^{-\sqrt{d}} \leq hd^{\mu+3}2^{-\sqrt{d}}$ fraction of the depth-ℓ balls do not belong to $\mathbf{W}$. Thus, the fraction of dense balls in $\mathbf{W}$ is at least

$$\frac{\frac{1}{2dh} - \frac{hd^{\mu+3}}{2^{\sqrt{d}}}}{1 - \frac{hd^{\mu+3}}{2^{\sqrt{d}}}} \quad > \quad \frac{1}{3dh}\,.$$

Suppose $\boldsymbol{\varphi}$ fails to satisfy property (ii). Then, among the depth-ℓ balls picked by $\boldsymbol{\varphi}$, at most $3d^{\mu}$ are dense; this represents at most a $3/d^3$ fraction. By Fact 2.3.13, the probability of this event is at most

$$e^{-2d^{\mu+3}\left(\frac{1}{3dh}-\frac{3}{d^3}\right)^2} \quad \leq \quad 2^{-2d^{\mu}-1}\,,$$

where the inequality follows from the fact that $h = d^{o(1)}$. This proves our claim. $\square$

Lemma 2.3.14 *There exists a dense ball v in $\mathcal{H}_r(v_r)$ such that at least half the $(r+1)$-fragments within $\mathcal{H}_{r+1}(v)$ are restrictions to v of databases from D_r.*

Proof: Let $\ell \in [h-1]$ be the integer whose existence is guaranteed by Lemma 2.3.12, and let $\alpha = 2^{-2d^{\mu}-1}$. For an r-fragment φ within $\mathcal{H}_r(v_r)$, let $S(\varphi)$ denote the set of dense depth-ℓ balls picked by φ. Lemma 2.3.12 says that at least an α fraction of all r-fragments φ satisfy $\varphi \in \widetilde{D}_r$ and $|S(\varphi)| \geq 3d^{\mu}$. A simple averaging argument shows that there exists a set W of dense depth-ℓ balls, with $|W| \geq 3d^{\mu}$, such that among the r-fragments φ with $S(\varphi) = W$, at least an α fraction lie in $\widetilde{D}_r$.

Let Φ be the set of r-fragments with $S(\varphi) = W$. The above fact can be restated as $|\Phi \cap \widetilde{D}_r| \geq \alpha{\cdot}|\Phi|$. Let Φ_{in} denote the (elementwise) restriction of Φ to the balls in W; let Φ_{out} denote the restriction to the remainder of the arena v_r. To be precise:

$$\begin{aligned} \Phi_{\text{in}} &= \left\{ \varphi \cap \left(\bigcup_{w \in W} w \right) \,:\, \varphi \in \Phi \right\}; \\ \Phi_{\text{out}} &= \left\{ \varphi \cap \left(v_r \setminus \bigcup_{w \in W} w \right) \,:\, \varphi \in \Phi \right\}. \end{aligned}$$

Recall the "product construction" of the fragments, which ensures that what happens inside one ball picked by a fragment is completely independent of what happens inside any other ball disjoint from it. As a result, we have $|\Phi| = |\Phi_{\text{in}}| \cdot |\Phi_{\text{out}}|$. Combining this with our earlier inequality gives

$$|\Phi \cap \widetilde{D}_r| \;\;\geq\;\; \alpha \cdot |\Phi_{\text{in}}| \cdot |\Phi_{\text{out}}| \,. \tag{2.10}$$

For each $w \in W$ let Φ_w denote the set of all $(r{+}1)$-fragments within $\mathcal{H}_{r+1}(w)$ and let Ψ_w denote the subset of this set consisting of those $(r{+}1)$-fragments which are restrictions to w of some database in D_r. Suppose $W = \{w_1, w_2, \ldots, w_{|W|}\}$. Because of the "product construction" of fragments, the map

$$\underset{\Phi \cap \widetilde{D}_r}{\overset{\varphi}{\Cap}} \quad \longmapsto \quad \big(\varphi \cap w_1,\; \varphi \cap w_2,\; \ldots,\; \varphi \cap w_{|W|},\; \varphi \cap (v_r \setminus \cup_{w \in W} w)\big)$$

is easily seen to be an injective map from $\Phi \cap \widetilde{D}_r$ to $\Psi_{w_1} \times \cdots \times \Psi_{w_{|W|}} \times \Phi_{\text{out}}$.

Therefore,

$$|\Phi \cap \widetilde{D}_r| \quad \leq \quad |\Psi_{w_1}| \times \cdots \times |\Psi_{w_{|W|}}| \cdot |\Phi_{\text{out}}| \,. \tag{2.11}$$

A third fact that follows from the "product construction" is that

$$|\Phi_{\text{in}}| \quad = \quad |\Phi_{w_1}| \times \cdots \times |\Phi_{w_{|W|}}| \,. \tag{2.12}$$

Combining (2.10), (2.11) and (2.12) gives

$$\prod_{w \in W} \frac{|\Psi_w|}{|\Phi_w|} \quad \geq \quad \alpha \,.$$

Therefore there exists a $v \in W$ such that

$$\begin{aligned} \frac{|\Psi_v|}{|\Phi_v|} \quad & \geq \quad \alpha^{1/|W|} \\ & \geq \quad \left(2^{-2d^\mu - 1}\right)^{1/(3d^\mu)} \\ & \geq \quad \tfrac{1}{2} \,, \end{aligned}$$

which completes the proof. $\square$

2.3.4 A Strategy for Maintaining the Invariants

The lemmas from §2.3.3 provide us with the tools we need to build a strategy for maintaining our three invariants. This in turn enables us to establish Claim 2.3.9, which, as we have argued before, completes the proof of our main theorem.

Lemma 2.3.15 *The sequences* $\{Q_r\}_{r=1}^t$ *and* $\{D_r\}_{r=1}^t$ *can be so chosen as to make the query, database and trace invariants hold for all* $r \in [t]$.

Proof: The case $r = 1$ is trivial. By construction, Q_1 intersects *every* ball which is a leaf of $\mathcal{H}_1(v_1)$, so the query invariant holds. Also, by construction, $\widetilde{D}_1 = D_1 = \{\text{all 1-fragments within } \mathcal{H}_1(v_1)\}$, so the database invariant holds. The trace invariant holds vacuously.

Now suppose the three invariants hold for some $r \in [t-1]$. Let v be the dense ball whose existence is guaranteed by Lemma 2.3.14. We let v be our $(r+1)^{\text{th}}$ arena; i.e., we set $v_{r+1} = v$. Since v is dense, it must be i-dense for some $i \in [s]$; we set $Q_{r+1} = Q_r^{(i)}$, where $Q_r^{(i)}$ is as defined in (2.9), and $k_r = f_r(x)$ for any $x \in Q_r^{(i)}$. Looking at Definition 2.3.10, we see that Q_{r+1} intersects at least a fraction $1/d$ of the children of v in $\mathcal{H}_r(v_r)$. These children become the leaves of $\mathcal{H}_{r+1}(v)$, and so the query invariant holds at $r + 1$.

We must also construct $D_{r+1} \subseteq D_r$ so that the table entry $\pi(y)[k_r]$ is constant for $y \in D_{r+1}$. Consider the partition $D_r = \bigcup_{z \in \{0,1\}^b} D_r^{(z)}$ where

$$D_r^{(z)} \;=\; \{y \in D_r \,:\, \pi(y)[k_r] = z\}\,.$$

Since at least half the $(r+1)$-fragments within $\mathcal{H}_{r+1}(v)$ are restrictions to v of databases from D_r, by the pigeonhole principle, there exists a $z \in \{0,1\}^b$ such that at least a

$$\frac{1}{2} \cdot \frac{1}{2^b} \;=\; 2^{-d^\mu - 1} \;\geq\; 2^{-2d^\mu}$$

fraction of these fragments are restrictions to v of databases from $D_r^{(z)}$. We set $D_{r+1} = D_r^{(z)}$ and $z_r = z$, ensuring that the database invariant holds at $r + 1$.

Finally, the r^{th} 2-tuple in the trace produced by an input in $Q_{r+1} \times D_{r+1}$ is always (k_r, z_r). Therefore the trace invariant holds at $r+1$. $\square$

Lemma 2.3.16 *There exists a t-fragment in $\widetilde{D}_t$ which picks two distinct balls of $\mathcal{H}_t(v_t)$ each of which intersects Q_t.*

Proof: We use the probabilistic method. Let $\boldsymbol{\varphi}$ be a random t-fragment chosen uniformly from amongst all t-fragments within $\mathcal{H}_t(v_t)$. Recall that $\mathcal{H}_t(v_t)$ is a depth-2 tree. The set of balls picked by $\boldsymbol{\varphi}$ is just a random $d^{\mu+3}$-element subset of the set of leaves of $\mathcal{H}_t(v_t)$. By the query invariant, at least a fraction $1/d$ of these leaves intersect Q_t.

Let $\mathcal{E}$ be the event that $\boldsymbol{\varphi}$ picks at most one Q_t-intersecting leaf; this represents at most a $1/d^{\mu+3} \leq 1/d^2$ fraction of the leaves picked by $\boldsymbol{\varphi}$. By Fact 2.3.13

$$\begin{aligned}
\Pr[\mathcal{E}] &\leq e^{-2d^{\mu+3}\left(\frac{1}{d}-\frac{1}{d^2}\right)^2} \\
&\leq e^{-2d^{\mu}} \\
&< 2^{-2d^{\mu}} \\
&\leq \Pr[\boldsymbol{\varphi} \in \widetilde{D}_t]\,,
\end{aligned}$$

where the last inequality follows from the database invariant. Therefore there exists a t-fragment for which $\mathcal{E}$ does not occur. $\square$

Proof of Theorem 2.1.1: Let φ be the t-fragment and w_1, w_2 be the two balls whose existences are guaranteed by Lemma 2.3.16. Let $y \in D_t$ be a database whose restriction to v_t is φ. Choose queries $x_1 \in w_1 \cap Q_t$ and $x_2 \in w_2 \cap Q_t$ arbitrarily. Since the leaves of $\mathcal{H}_t(v_t)$ form a $\frac{\beta}{16}$-separated family, and since φ is contained in the union of these leaves, any $\frac{\beta}{16}$-ANN of x_i with respect to φ must lie in w_i ($i = 1, 2$). Thus, no $z \in \varphi$ can be a $\frac{\beta}{16}$-ANN of both x_1 and x_2, which establishes Claim 2.3.9, and therefore proves Theorem 2.1.1. $\square$

2.4 Concluding Remarks

2.4.1 Notes on the Proof

Our proof of Theorem 2.1.1 uses, at several points, ideas from work on other problems in complexity theory. The basic plan of the proof — the collection of derived hierarchies, the recursive and "product" construction of databases and fragments, and the crucial idea of "zooming in" from arena to arena in order to identify a hard input for the algorithm — can be traced back to a seminal paper of Ajtai [1] in which he proved a nontrivial cell probe lower bound for the predecessor searching problem. We execute the plan differently from Ajtai, though, and there is one important difference that is worth mentioning. Ajtai, whose goal was to show a nonconstant lower bound, used an argument that repeatedly generated new problem instances *in new universes* and showed that one could iterate a constant number of times and still generate a nontrivial problem instance. In our argument we start with one master set of problem instances, $Q_1 \times D_1$, and construct successively smaller subsets; all our problem instances are in the *same* universe $\{0,1\}^d$. Staying in a fixed universe seems essential for proving an ANNS lower bound. Furthermore, we have carefully optimised the parameters in our combinatorial constructions so as to get the asymptotically best lower bound we can using our proof plan. These optimisations are important elsewhere too, for it turns out that this proof plan is adaptable to other closely related data structure problems. In fact, the plan is good enough to prove an asymptotically optimal lower bound for predecessor searching; such a proof can be found in Beame and Fich [12]. In related work, Chazelle [24] used this same plan to prove a cell probe lower bound for the geometric problem of point separation.

Lemma 2.3.14, although not stated using the language of probability distributions, is really about amplifying probability by projecting a product distribution on its "highest density co-ordinate." A random r-fragment within the arena v_r has only a small probability ($2^{-2d^{\mu}}$) of being "good" (i.e., of being a restriction of a database from D_r); so, before shrinking D_r to D_{r+1}, we zoom in to an arena within which fragments have a high probability ($\frac{1}{2}$) of being good. This kind of probability amplification has been used elsewhere in complexity theory as well, a famous example being the work of Karchmer and Wigderson [46] on lower bounds for monotone circuit depth.

It is important to mention that our presentation of the proof of Theorem 2.1.1 is quite different from that in Chakrabarti et al. [20]. We have done away with the abstractions of communication complexity and probabilistically defined databases (both found in [20]) and instead kept the presentation direct and focussed on detail. We hope that this approach will help readers take a closer look at the technical details of the proof.

2.4.2 Open Problems

We have shown that the problem of Approximate Nearest Neighbour Searching, even with a very weak approximation requirement, has a nontrivial lower bound in a model as strong and general as the cell probe model. The most significant limitation of our bound is that it applies only to deterministic query algorithms. Thus, although our bound almost matches the best upper bound currently known for the problem, we cannot really compare the two bounds since the latter uses randomisation. Barkol and Rabani [11] give a lower bound for NNS that does apply to randomised query schemes and is higher than the aforementioned upper bound. However, their bound does not apply when approximation is allowed. Furthermore, there is no nontrivial upper bound known for exact NNS that one can compare their bound to.

While our work nicely complements that of Barkol and Rabani (and the earlier work of Borodin et al. [17]), the present state of affairs strongly suggests that we seek a lower bound for randomised query algorithms for ANNS. This is among the most important open theoretical questions regarding Nearest Neighbour Searching.

The richness lemma of Miltersen et al. [54], which we encountered in §2.2, is a useful tool for proving randomised cell probe lower bounds. However, the technique was designed to handle Boolean problems. It seems unlikely that the technique can handle problems like ANNS, where the answer space has $\omega(1)$ size (and is, in fact, as large as the query space). Of course, NNS has this same problem, but Barkol and Rabani get around this by proving their lower bound for a decision version of the problem instead. Unfortunately, their idea will not help prove a nontrivial lower bound for randomised ANNS, because the decision version of this problem *does* have an $O(1)$-probe algorithm.

A different technique, also developed in [54], is the so-called *round elimination* technique. It can be used to derive a randomised cell probe lower bound for the predecessor problem, which makes it look promising for our purposes. However, adapting the predecessor lower bound to ANNS still seems very challenging; the fact that an ANNS query can have many different correct answers seems to be a major obstacle.

It is open whether the decision version of ANNS has a nonconstant *deterministic* query time lower bound. Notice that if such a lower bound were proven, randomisation would be shown to be essential in the algorithms of [49, 42, 40], which would be a remarkable result in its own right.

Chapter 3

Decision Trees and The Evasiveness of Monotone Graph Properties

3.1 Introduction

3.1.1 The Problem and the Model

Suppose we have an input graph G and are required to decide whether or not it has a certain property P. For instance, P could be the property of being a 2-connected graph, or the property of containing a 5-cycle. We would like to reach our decision by looking at as little of our input as possible. More precisely, we start out knowing only n, the number of vertices of G. We have access to an oracle which answers queries of the form "Is (x, y) an edge of G?" for vertices $x, y \in [n]$. We seek a strategy that specifies a sequence of such queries to the oracle, where each query may depend upon the outcomes of the previous ones, terminating when sufficient information about G has been obtained to decide whether or not G has property P.

This problem can be generalised by observing that the input object does not need to be a graph. Suppose that the objects we are interested in can be represented by m-bit strings; P can then be a property of m-bit strings, that is to say, a Boolean function $P : \{0,1\}^m \to \{0,1\}$. The input string $x = (x_1, \ldots, x_m)$, $x_i \in \{0,1\}$, is now given by an oracle which answers queries of the form "What is the value of x_i?" for $i \in [m]$. We seek a strategy that sequentially and adaptively queries the oracle, terminating when the value of $P(x)$ is determined.

Such a query strategy is called a *decision tree algorithm* since it can be represented by a rooted binary tree whose non-leaf vertices are labelled by the Boolean input variables (i.e., x_i), and leaves and edges are labelled by Boolean values (i.e., 0 or 1). The execution of the algorithm would involve tracing a path from the root of the tree to a leaf, asking the queries indicated by the vertex labels and following the edges with labels matching the oracle's responses (see Figure 3.1). Upon reaching a leaf, the algorithm would output the Boolean value at that leaf. The *cost* of such a decision tree algorithm is the worst case number of queries that it makes, i.e., the length of the longest root-to-leaf path. The *decision tree complexity* of P is the minimum cost of any decision tree algorithm for P. Notice that the complexity of any property $P : \{0,1\}^m \to \{0,1\}$ is at most m, because with m queries to the oracle we could determine the input completely and thus, since computation time is free in this model, we would know whether or not the input has property P.

In this chapter we shall primarily be concerned with the decision tree complexity of *graph properties*. An n-vertex input graph G can be written as a $\frac{1}{2}n(n-1)$-bit string, by ordering the vertex pairs $\{(x,y) \in [n] \times [n] : x < y\}$ in an arbitrary, but fixed, way and letting the bits 1 and 0 indicate presence and absence of an edge, respectively. Notice

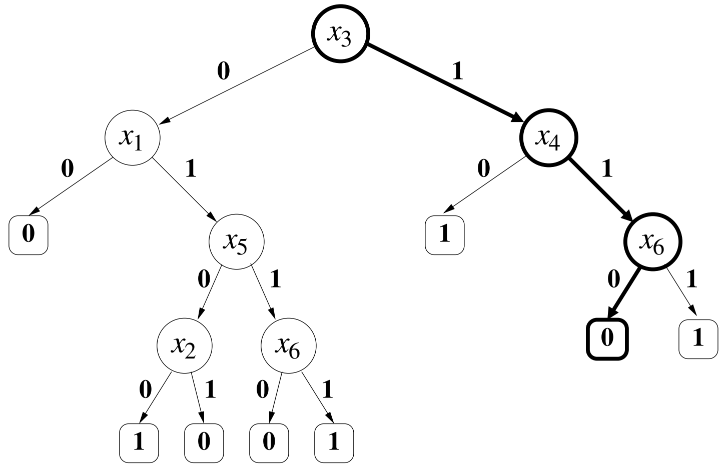

Figure 3.1: A Boolean Decision Tree. On input $x = (1, 0, 1, 1, 1, 0)$ the algorithm queries x_3, x_4, and x_6, and outputs "0". The cost of this tree is 4.

that the bit string representation of a graph depends on the labelling of its vertices. As a result, with such a representation, one can define "properties" of the type:

- G contains the edge $(2, 6)$
- The odd numbered vertices in G form an independent set

Such properties, which depend on the precise vertex labelling used, are not graph theoretically interesting. We shall disallow them by requiring that our properties be *invariant under vertex relabelling*, i.e., isomorphism-invariant. Henceforth, the term "graph property" will mean a property invariant under vertex relabelling.

Remark: Throughout this chapter "input graphs" will be assumed to be vertex-labelled. Other ("usual") graphs we encounter will be unlabelled, unless otherwise specified. Also, the term "edge," when applied to an input graph, will simply mean a labelled vertex pair.

3.1.2 Karp's Evasiveness Conjecture

In the early 1970s a number of researchers studying specific graph properties were able to show that certain properties, such as acyclicity and connectedness, had complexity $\frac{1}{2}n(n-1)$, for n-vertex graphs. In other words, any decision tree algorithm for one of these properties would, in the worst case, have to query *every* edge of the graph for presence or absence. Such properties have come to be called *evasive*.[1] Of course, the concept can be defined more generally for Boolean properties of bit strings.

Definition 3.1.1 *A property $P : \{0,1\}^m \to \{0,1\}$ is said to be evasive if it has decision tree complexity m.*

How large is the collection of evasive graph properties? Might all nontrivial graph properties (the "always true" and "always false" properties being the trivial ones) be evasive? The answer turns out to be negative; in fact Best et al. [14] define a class of graphs called *scorpions* and show that the property of being a scorpion has decision tree complexity $O(n)$. However, this property lacks the feature of *monotonicity*: both addition and deletion of edges can change a scorpion to a non-scorpion.

Definition 3.1.2 *A property $P : \{0,1\}^m \to \{0,1\}$ is said to be* monotone *if for all $x, y \in \{0,1\}^m$ such that $\forall i\ (x_i \leq y_i)$, we have $P(x) \leq P(y)$. In other words, changing input variables from 0 to 1 never falsifies the property.*

Thus, for graph properties, monotonicity means that starting with a graph which has the property, the addition of edges does not destroy the property. Connectedness, non-planarity, non-k-colourability and the property of containing a perfect matching are all examples of nontrivial monotone properties (for sufficiently large n). In fact, for most

[1] Some authors call such properties "elusive" instead of evasive.

natural graph properties P, it is the case that either P itself is monotone, or its negation $\neg P$ is. Since the decision tree complexities of P and $\neg P$ are always equal, monotone graph properties form a particularly important subclass of the class of all graph properties.

Remarkably, to date there is no known nontrivial example of a non-evasive monotone graph property. In other words, the following bold conjecture, which Rosenberg [57] attributes to Karp, is still open.

- KARP CONJECTURE: For every n, every nontrivial monotone n-vertex graph property is evasive.

While we do not settle this conjecture in this chapter, we do provide some interesting results which make partial progress towards a proof that the conjecture is true.

3.2 Towards a Settlement of the Karp Conjecture

3.2.1 Prior Work

During the 1970s, several researchers devised adversarial strategies which proved the evasiveness, for all n, of certain specific graph properties (see e.g. [14], [51] and [16, Ch.8]). These strategies worked for the properties of acyclicity, connectedness, 2-connectedness, planarity and simple variants on these. The most sophisticated of these adversarial strategies was one used by Bollobás [15] to prove the evasiveness of the property of containing a k-clique, for any k, $2 \leq k \leq n$. In general, it seemed that the adversarial strategy needed to be adjusted to the property under consideration, and it was unclear how these approaches could handle a statement as general as the Karp Conjecture.

A weaker version of the Karp Conjecture had been proposed earlier by Aanderaa and Rosenberg [57]; they conjectured that all nontrivial monotone graph properties have decision tree complexity $\Omega(n^2)$.[2] In an important breakthrough, Rivest and Vuillemin [56] proved that all such properties have complexity at least $n^2/16$, thereby settling this weaker conjecture. Importantly, their proof technique was based on ideas quite different from the aforementioned adversarial strategies.

The next big advance was the work of Kahn, Saks, and Sturtevant [43] who discovered a connection between evasiveness and topology. In a sense, their topological approach was a highly evolved version of Rivest and Vuillemin's techniques. They used the topological approach to prove that the Karp Conjecture holds whenever n is a prime power. As a by-product they also improved the complexity lower bound for nontrivial monotone graph properties to $n^2/4 + o(n^2)$; this remains the best result as of this writing. A decade later Triesch [61, 62] used the topological approach, together with complicated algebraic constructions, to prove the evasiveness of some special classes of properties: specifically, these papers established evasiveness of graph properties that are always false when (i) the graph contains either a 3-cycle or a 4-cycle, and when (ii) the graph is not bipartite, respectively. Meahwhile, Yao [65] had used the topological method to prove a related result: he showed that nontrivial monotone *bipartite* graph properties (i.e., the input graph is guaranteed to be bipartite and the algorithm starts out knowing the bipartition) are evasive.

3.2.2 Our Results

In this chapter we prove a few new evasiveness results. For certain classes of monotone n-vertex graph properties, we show evasiveness for all large enough n. The properties we consider are (i) properties closed under the operation of taking graph minors, and

[2]Since Karp's conjecture is a strengthening of this one, some authors refer to it as the Aanderaa-Karp-Rosenberg (or AKR) conjecture.

(ii) properties having to do with subgraph containment (also known as subgraph isomorphism).

Definition 3.2.1 *A graph H is called a* minor *of the graph G if H can be obtained by starting with G and performing a sequence of edge deletions, edge contractions, and isolated vertex removals. A family $\mathcal{M}$ of graphs is said to be* minor-closed *if, for any $G \in \mathcal{M}$ and any minor H of G, we have $H \in \mathcal{M}$. Such a family $\mathcal{M}$ is said to be* nontrivial *if it is infinite and does not equal the family of all graphs.*

Let $\mathcal{M}$ be a minor-closed family and let $R_n^{\mathcal{M}}$ denote the n-vertex graph property of belonging to $\mathcal{M}$ (we shall call such properties "minor-closed properties"). We shall assume that $\mathcal{M}$ is nontrivial, otherwise $R_n^{\mathcal{M}}$ would be a trivial property for all large n; with the assumption it is easy to show that $R_n^{\mathcal{M}}$ is nontrivial for all large n. Note that $R_n^{\mathcal{M}}$ is not monotone, but its negation, $\neg R_n^{\mathcal{M}}$, is. We prove the following result about such properties.

Theorem 3.2.2 *The property $R_n^{\mathcal{M}}$ is evasive for all large enough n.*

The most well-known minor-closed property is planarity. There are lots of others natural graph properties which are minor-closed, examples being embeddability on an arbitrary surface, outerplanarity, linkless embeddability in $\mathbb{R}^3$, the property of being a series-parallel graph, the property of bounded treewidth, etc. Thus, the above result is a major generalisation of the evasiveness of planarity due to Best et al. [14].[3]

Let H be any fixed graph. Let Q_n^H denote the property of containing H as a subgraph (not necessarily as an induced subgraph). As mentioned before, Bollobás [15] proved that Q_n^H is evasive for all n in the special case when H is a complete graph. This raises

[3]However, our result is not the strongest possible generalisation to "minor-closed" properties, since planarity has been proven evasive whenever n is large enough for it to be a nontrivial property (i.e. $n \geq 5$).

the natural question: what can we say about general H? Here we prove the following results about Q_n^H.

Theorem 3.2.3 *If H is bipartite, then Q_n^H is evasive for large enough n.*

Theorem 3.2.4 *For any H there exists an integer r_0 with the following property. Suppose $n = \sum_{i=1}^{r} q^{\alpha_i}$ where q is a prime power, $q \geq |V(H)|$, each $\alpha_i \geq 1$ and $r \equiv 1 \pmod{r_0}$. Then Q_n^H is evasive.*

Here, as in the rest of this chapter, for any graph G, $V(G)$ denotes its vertex set and $E(G)$ its edge set.

From the theorem of Kahn, Saks, and Sturtevant [43] we already knew that Q_n^H would be evasive for prime powers n; is Theorem 3.2.4 much of an improvement? The following two corollaries clarify that our theorem *is* indeed a significant contribution. Neither corollary is derivable from the results of [43], because there are arbitrarily large gaps in the sequence of prime powers.

Corollary 3.2.5 *For any graph H there exist infinitely many primes p such that for all sufficiently large n divisible by p, the property Q_n^H is evasive. In particular, for any H there exists an infinite arithmetic progression $\mathcal{A} = \mathcal{A}(H)$ such that Q_n^H is evasive for $n \in \mathcal{A}$.*

Corollary 3.2.6 *For any graph H there exists a constant $c = c(H)$ such that for all sufficiently large n, the decision tree complexity of Q_n^H is at least $\frac{1}{2}n^2 - cn$.*

Finally, we can generalise Theorem 3.2.4 from subgraph containment to any monotone Boolean combination of finitely many subgraph containment properties. Thus, for example, if H_1, H_2 and H_3 are fixed graphs, then the property of containing as subgraph either H_1 or both of H_2 and H_3 is still evasive for those n which satisfy the conditions of Theorem 3.2.4. We formally state this below; the proof is a straightforward modification of our proof of Theorem 3.2.4, so we shall not give the details here.

Theorem 3.2.7 *Let $f : \{0,1\}^k \to \{0,1\}$ be a nontrivial monotone Boolean function and let $H_1, \ldots, H_k$ be arbitrary fixed graphs. Define the composite property $Q_n = f(Q_n^{H_1}, \ldots, Q_n^{H_k})$. Then there exists an integer r_0 with the following property. Suppose $n = \sum_{i=1}^{r} q^{\alpha_i}$ where q is a prime power, $q \geq \max_{1 \leq i \leq k} |V(H_i)|$, each $\alpha_i \geq 1$ and $r \equiv 1 \pmod{r_0}$. Then Q_n is evasive.* $\square$

The proofs of Theorems 3.2.2–3.2.4 all use the topological approach of Kahn, Saks, and Sturtevant. In the next section we review the topological approach and after that we present the proofs of our results. Our results in this chapter were obtained in joint work with Khot and Shi [21]. We remark that our proofs closely follow those in Chakrabarti, Khot, and Shi [21], although our presentation differs slightly from that paper.

3.3 The Topological Approach

Suppose $P : \{0,1\}^m \to \{0,1\}$ is a nontrivial monotone property. Consider the collection of subsets $S \subseteq [m]$ with the following property: setting the variables indexed by S to 1 and those indexed by $[m] \setminus S$ to 0 yields an m-tuple which *does not satisfy* P (in other words, the characteristic vector of S does not satisfy P). Since P is monotone, this collection of sets is downward closed under set inclusion. Such a downward closed collection of sets is called an *abstract complex*, the sets in this collection are called the *faces*, and the elements of $[m]$ the *vertices* of the complex.

Definition 3.3.1 *If P is monotone, then the abstract complex associated with P, denoted $\Delta(P)$, is defined as follows:*

$$\Delta(P) = \{S \subseteq [m] \,:\, \text{If } x_i = 1 \Longleftrightarrow i \in S, \text{ then } (x_1, \ldots, x_m) \text{ does not satisfy } P\}.$$

Associated with an abstract complex Δ is a topologically important number called its *Euler characteristic* which is denoted $\chi(\Delta)$ and is defined as follows:

$$\chi(\Delta) \quad = \quad \sum_{\emptyset \neq F \in \Delta} (-1)^{|F|-1} \,. \tag{3.1}$$

For our purposes this purely combinatorial view of abstract complexes and the their Euler characteristics will suffice. We note, in passing, that an abstract complex can be associated in a natural way with a topological space (such as a simplicial complex) and that the Euler characteristic of the abstract complex is a fundamental topological invariant of the space.

Kahn, Saks, and Sturtevant [43] showed that non-evasiveness of P has topological consequences for $\Delta(P)$. The following theorem is implicit in their work:

Theorem 3.3.2 (Kahn, Saks, and Sturtevant [43]) *If the monotone property P is not evasive, then $\chi(\Delta(P)) = 1$.* $\square$

For our result, we shall need to use a stronger theorem from [43]. Let Δ be an abstract complex defined on $[m]$ and let Γ be a finite group which acts on the set $[m]$, preserving the faces of Δ. The action partitions $[m]$ into orbits, say $A_1, \ldots, A_k$. We use the action of Γ to define another abstract complex Δ_Γ on $[k]$ as follows:

$$\Delta_\Gamma \quad = \quad \Big\{S \subseteq [k] \,:\, \bigcup_{i \in S} A_i \in \Delta\Big\} \,. \tag{3.2}$$

Sometimes, as in the case of our work, $\Delta(P)$ is too complicated to handle, but one can find a group Γ whose action greatly "simplifies" the complex to $(\Delta(P))_\Gamma$ (note that the group action must be an action on the *edges*, i.e., the vertex pairs, of the input graph). The next theorem, the most important tool in [43], says that if Γ has certain rather restrictive properties, then non-evasiveness of P has a topological consequence for this new complex.

Theorem 3.3.3 (Kahn, Saks, and Sturtevant [43]) *Suppose Γ has a normal subgroup Γ_1 which is such that $|\Gamma_1|$ is a prime power and the quotient group Γ/Γ_1 is cyclic. Then, if P is not evasive, $\chi((\Delta(P))_\Gamma) = 1$.* $\square$

An application of this result leads to the following theorem, the main result of [43].

Theorem 3.3.4 (Kahn et al. [43]) *Let P_n be a nontrivial monotone property of n-vertex graphs. If n is a prime power, then P_n is evasive.* $\square$

In order to derive Theorem 3.3.4 from Theorem 3.3.3, Kahn, Saks, and Sturtevant construct a group Γ which acts on the vertices of the input graph and thus, indirectly, on the edges. They then show that their group acts *transitively* on the edges, meaning that any edge can be mapped, by some group element, to any other edge. Thus, the action of Γ on the edges produces one single orbit consisting of all the edges, whence $(\Delta(P_n))_\Gamma$ is a one-vertex complex. Since P_n is monotone and nontrivial, P_n must hold for the complete graph K_n; therefore, $(\Delta(P_n))_\Gamma$ must be a trivial complex $\{\emptyset\}$ which has Euler characteristic 0, contradicting Theorem 3.3.3.

The number theoretic constraint on n is a consequence of the fact that this action depends crucially on being able to view the vertices of the graph as elements of a finite field. Our approach to proving evasiveness for more general n will be to devise a more sophisticated group action.

3.4 Proofs of Our Evasiveness Results

We shall now prove the results stated in §3.2.2. We first design a group action on labelled graphs which will let us easily prove Theorems 3.2.2 and 3.2.3 from Theorem 3.3.3. The proof of Theorem 3.2.4 will require considerable extra effort, so we shall tackle it last.

3.4.1 Our Group Action

Suppose we are considering n-vertex graph properties, where

$$n = \sum_{i=1}^{r} q^{\alpha_i}, \tag{3.3}$$

where q is a large enough prime power, r is arbitrary and the α_i are arbitrary positive integers. Note that this condition is satisfied for all large enough n, because for any q_0 there exists an n_0 such that all integers $n > n_0$ are divisible by a prime power larger than q_0.

For integer p, let $\mathbb{F}_p$ denote the finite field with p elements (if it exists), and let $\mathbb{F}_p^*$ denote the multiplicative group of $\mathbb{F}_p$ as well as the set $\mathbb{F}_p \setminus \{0\}$. We partition the vertex set $[n]$ into r *clusters* $V_1, \ldots, V_r$, with $|V_i| = q^{\alpha_i}$ and identify vertices in V_i with elements of $\mathbb{F}_{q^{\alpha_i}}$. We then define a permutation group Γ on the vertices as follows:

$$\Gamma = \left\{ \langle a, b_1, b_2, \ldots, b_r \rangle \; : \; a \in \mathbb{F}_q^*, \; b_i \in \mathbb{F}_{q^{\alpha_i}} \right\},$$

where $\langle a, b_1, b_2, \ldots, b_r \rangle$ denotes a permutation which sends $x \in V_i = \mathbb{F}_{q^{\alpha_i}}$ to $ax + b_i \in V_i$. Let $\Gamma_1 = \{\langle 1, b_1, \ldots, b_r \rangle \; : \; b_i \in \mathbb{F}_{q^{\alpha_i}}\}$. It is easy to check that Γ_1 is a normal subgroup of Γ, $|\Gamma_1| = q^{\alpha_1 + \cdots + \alpha_r}$, a prime power, and $\Gamma/\Gamma_1 \cong \mathbb{F}_q^*$, which is always a cyclic group. Thus Γ satisfies the hypotheses of Theorem 3.3.3.

Now let P_n be one of the following properties:

i. The property Q_n^H for some fixed graph H

ii. The property $\neg R_n^{\mathcal{M}}$ for some nontrivial minor-closed family $\mathcal{M}$

Let $\Delta = \Delta(P_n)$ be the abstract complex associated with P_n. The action of Γ induces a group action on the edges (i.e., labelled vertex pairs) of the input graph and thus partitions the edge set into orbits. Let $\mathcal{A}$ denote the set of these orbits. We define the complex Δ_Γ as in (3.2):

$$\Delta_\Gamma \;=\; \Big\{\mathcal{D} \subseteq \mathcal{A} \,:\, \bigcup_{A\in\mathcal{D}} A \,\in \Delta\Big\}\,. \tag{3.4}$$

Notice that an orbit $A \in \mathcal{A}$ is just a set of edges and can be thought of as the labelled n-vertex graph whose edge set is A (or, to put it more vividly, the n-vertex graph obtained by "turning on" exactly those edges of the input graph which belong to A). Likewise, the union of several orbits can also be thought of as an n-vertex graph. We shall frequently use this correspondence between (sets of) orbits and graphs throughout the proof.

Our intention is to show that the Euler characteristic $\chi(\Delta_\Gamma) \neq 1$. To this end, let us investigate what the faces of Δ_Γ look like. Call an edge an *intracluster edge* if both its end points lie in the same V_i for some i; call the edge an *intercluster edge* otherwise.

Lemma 3.4.1 *An orbit containing an intracluster edge is not contained in any face of* Δ_Γ.

Proof: Let $A \in \mathcal{A}$ be the orbit of the intracluster edge (u, v), $u, v \in V_i$. Then $A = \{(au + b, av + b) : \, b \in \mathbb{F}_{q^{\alpha_i}}, \, a \in \mathbb{F}_q^*\}$. Set $w = v - u$. Then $(0, w) \in A$. Consider the set of vertices $X = \{wz : \; z \in \mathbb{F}_q\} \subseteq V_i$. For $0 \neq x \in X$ we clearly have $(0, x) \in A$. Thus, for any pair of distinct vertices $x_1, x_2 \in X$, we have $(0, x_2 - x_1) \in A$, whence

$(x_1, x_2) \in A$. So A contains all edges among vertices in X; in particular, A contains K_q, the complete graph on $q = |X|$ vertices.

Suppose $P_n = Q_n^H$. Since q, in (3.3), is large enough, we can assume that $q \geq |V(H)|$. For such q, A, considered as a graph, clearly satisfies P_n. From Definition 3.3.1 we see that no superset of A can be a face of Δ. By (3.4), no face of Δ_Γ can contain A.

Suppose $P_n = \neg R_n^{\mathcal{M}}$. Since $\mathcal{M}$ is nontrivial, there exists an integer h such that $K_h \notin \mathcal{M}$. Therefore, for $q \geq h$, we have $A \notin \mathcal{M}$ whence A satisfies $\neg R_n^{\mathcal{M}}$. As before, we conclude that no face of Δ_Γ can contain A. $\square$

Our group action is clearly vertex transitive within each cluster. Therefore, if $u \in V_i, v \in V_j, i < j$, then the orbit of the intercluster edge (u, v) is the set E_{ij} of all edges between V_i and V_j. Let $\mathcal{E} = \{E_{ij} : i < j\} \subseteq \mathcal{A}$. The preceding lemma reduces (3.4) to:

$$\Delta_\Gamma = \Big\{\mathcal{D} \subseteq \mathcal{E} : \bigcup_{E_{ij} \in \mathcal{D}} E_{ij} \in \Delta\Big\}. \tag{3.5}$$

3.4.2 Minor-closed Properties and Bipartite Subgraph Containment

We are all set to prove our first two evasiveness results.

Proof of Theorem 3.2.2: Suppose $P_n = \neg R_n^{\mathcal{M}}$ for a nontrivial minor-closed family $\mathcal{M}$. Then $K_h \notin \mathcal{M}$ for some integer h. Since K_h is a minor of the complete bipartite graph $K_{h,h}$, we have $K_{h,h} \notin \mathcal{M}$. Suppose $q \geq h$ where q is as in (3.3). Then any orbit E_{ij} contains $K_{h,h}$ as a subgraph and therefore satisfies $\neg R_n^{\mathcal{M}}$. Thus, none of the orbits E_{ij} can be contained in a face of Δ and this reduces (3.5) to the very simple $\Delta_\Gamma = \{\emptyset\}$.

Thus, $\chi(\Delta_\Gamma) = 0 \neq 1$ whence, by Theorem 3.3.3, P_n is evasive. $\square$

Proof of Theorem 3.2.3: Suppose $P_n = Q_n^H$ for a fixed bipartite graph H. For $q \geq |V(H)|$, any orbit E_{ij} contains H as a subgraph, and therefore, satisfies P_n. Arguing exactly as in the previous proof, we see that P_n is evasive. $\square$

3.4.3 General Subgraph Containment

Unfortunately, the situation is not so simple for the property Q_n^H where H is a non-bipartite graph. The complex Δ_Γ now *does* contain some nonempty faces and we will have to carefully analyse its Euler characteristic. For this, we shall need to consider an auxiliary class of graph properties and prove a technical lemma about the abstract complexes associated with such properties.

Consider the following operation on a graph G. Let the vertices of G be coloured, using *all* the colours in some set C, so that no two adjacent vertices get the same colour. Let G' be a graph with vertex set C where two distinct vertices $c_1, c_2 \in C$ are adjacent iff the colouring assigns colours c_1 and c_2 to the end-points of some edge in G. We shall call G' a *compression of graph G induced by colouring C*. If there exists a C which induces a compression G' of G, we shall write $G' \lhd G$. It is easy to check that "$\lhd$" is a partial order on graphs.

Definition 3.4.2 *A family $\mathcal{F}$ of graphs is said to be* closed under compression *if, for graphs G, H such that $G \in \mathcal{F}$ and $H \lhd G$, we have $H \in \mathcal{F}$.*

Let $\mathcal{F}$ be a nonempty finite family of nontrivial graphs (a *trivial graph* being one with no edges) that is closed under compression. The property $P_n^{\mathcal{F}}$ that an input graph G on n vertices contains some member of $\mathcal{F}$ as a subgraph is clearly nontrivial, for n large enough, and monotone. Let $\chi_n = \chi(\Delta(P_n^{\mathcal{F}}))$ be the Euler characteristic of its associated complex. We shall show that $\chi_n \neq 1$ infinitely often. Let us set

$$T = 2^{2^t}, \text{ where } t \text{ is the smallest integer such that } T \geq \min_{F \in \mathcal{F}} |V(F)|. \tag{3.6}$$

We shall prove:

Lemma 3.4.3 (Technical Lemma) *If* $n \equiv 1 \pmod{T-1}$*, then* $\chi_n \equiv 0 \pmod 2$.

Since we only care about $\chi_n \bmod 2$, we can use the fact that addition and subtraction are equivalent mod 2 in (3.1) to get[4]

$$\chi_n \equiv \#\{G : G \text{ is nontrivial and does not satisfy } P_n^{\mathcal{F}}\} \pmod 2. \tag{3.7}$$

Consider n-vertex input graphs with vertices labelled 0 through $n-1$. For $n > T$, let us define a group action on such graphs as follows. For $a, b \in \{0, 1, 2, \ldots, T-1\}$ and a odd, let permutation $\phi_{a,b}$ be defined by mapping vertex i to vertex $(ai + b) \bmod T$ for $i \in \{0, 1, \ldots, T-1\}$. The other $n - T$ vertices are left fixed. It is routine to check that the set of all these permutations forms a group Φ under composition, thereby defining a group action on the labelled vertices. This action induces an action on graphs in the obvious manner, thereby partitioning the set of all labelled n-vertex graphs into orbits. Since the order $|\Phi|$ of the group is $\frac{1}{2}T^2$, a power of 2, each orbit has size a power of 2. Therefore, (3.7) can be modified to

$$\chi_n \equiv \#\left\{G : \begin{array}{c} G \text{ is nontrivial, invariant under} \\ \Phi\text{, and does not satisfy } P_n^{\mathcal{F}} \end{array}\right\} \pmod 2. \tag{3.8}$$

The action of Φ on the vertices also induces an action on edges (or rather, on vertex pairs), not to be confused with the action on labelled graphs mentioned above. Therefore

[4]Note that we are counting not graphs, but labelled graphs.

the set of edges amongst vertices $0, 1, \ldots, T-1$ is partitioned into orbits. Since any odd integer is invertible mod T, we get 2^t orbits $E_0, E_1, \ldots, E_{2^t-1}$, where

$$E_i \;=\; \big\{(x, y) \,:\, 0 \le x < y < T,\;\; y - x = 2^i k \text{ for some odd number } k\big\}\,. \quad (3.9)$$

Let G be a Φ-invariant graph. From now on, let us refer to the vertices $0, 1, \ldots, T-1$ as *left vertices* and the rest as *right vertices*. Let G_{left} and G_{right} denote the subgraphs of G induced by the left and right vertices, respectively. By invariance of G, the set of right vertices adjacent to any left vertex is the same for each left vertex; let $\mathcal{R}(G)$ denote this set. Also, the set of edges $E(G_{\text{left}})$ is the union of a certain number of the orbits E_i; let $\operatorname{orb}(G)$ denote this number. We shall show that whether or not G has the property $P_n^{\mathcal{F}}$ is completely determined once $G_{\text{right}}, \mathcal{R}(G)$ and this number $\operatorname{orb}(G)$ are fixed; the specific G_{left} does not matter.

For a graph G, let $\operatorname{chr}(G)$ denote its chromatic number and $\operatorname{clq}(G)$ denote the size of its largest clique.[5]

Lemma 3.4.4 *For any invariant G, we have* $\operatorname{chr}(G_{\text{left}}) = \operatorname{clq}(G_{\text{left}}) = 2^{\operatorname{orb}(G)}$.

Proof: Let $I \subseteq \{0, 1, \ldots, 2^t - 1\}$ be such that $E(G_{\text{left}}) = \bigcup_{i \in I} E_i$; then we have $|I| = \operatorname{orb}(G)$. Consider two vertices x, y of G_{left}. If their binary representations agree on the bit positions indexed by I, then $x - y = \sum_{i \in I'} \pm 2^i$ for some set I' disjoint from I. By (3.9), this implies $(x, y) \notin E(G_{\text{left}})$. Therefore, the vertices of G_{left} can be partitioned into $2^{|I|}$ independent sets; thus $\operatorname{chr}(G_{\text{left}}) \le 2^{\operatorname{orb}(G)}$. On the other hand, if x, y are such that the bits in positions outside I are all zero, then $x - y = \sum_{i \in I''} \pm 2^i$ for some $I'' \subseteq I$,

[5] We avoid the customary symbol "χ" for chromatic number since we have used it already to mean Euler characteristic.

which, by (3.9) implies that $(x, y) \in E(G_{\text{left}})$. Therefore, G_{left} has a clique of size $2^{|I|} = 2^{\text{orb}(G)}$. The lemma follows. $\square$

Lemma 3.4.5 *Let G_1, G_2 be two invariant n-vertex labelled graphs such that $G_{1,\text{right}} = G_{2,\text{right}}$, $\mathcal{R}(G_1) = \mathcal{R}(G_2)$ and $\text{orb}(G_1) = \text{orb}(G_2)$. Then G_1 has property $P_n^{\mathcal{F}}$ if and only if G_2 does.*

Proof: Suppose G_1 has property $P_n^{\mathcal{F}}$; we shall show that G_2 does too. Suppose G_1 contains $F \in \mathcal{F}$ as a subgraph. We fix a particular occurrence of F within G_1 so that we can talk about $F_{\text{left}}, F_{\text{right}}$ and $\mathcal{R}(F) := \mathcal{R}(G_1) \cap V(F)$. Let $h = \text{chr}(F_{\text{left}})$; using Lemma 3.4.4 and the hypotheses, we obtain

$$h \leq \text{chr}(G_{1,\text{left}}) = \text{clq}(G_{1,\text{left}}) = 2^{\text{orb}(G_1)} = 2^{\text{orb}(G_2)} = \text{clq}(G_{2,\text{left}}) .$$

Therefore $G_{2,\text{left}}$ contains K_h as a subgraph. Fix a particular occurrence of K_h and, starting with the graph F_{right}, connect each of the h left vertices in this occurrence to each vertex in $\mathcal{R}(F)$. Let F' be the resulting graph. Since $\mathcal{R}(F) \subseteq \mathcal{R}(G_1) = \mathcal{R}(G_2)$ and since F_{right} is a subgraph of $G_{1,\text{right}} = G_{2,\text{right}}$, it follows that F' is a subgraph of G_2.

Consider the following colouring of the graph F: we use h colours for its left vertices and colour each right vertex with a distinct colour, never using any of these h colours. Let $F'' \lhd F$ be the compression of F induced by this colouring. It is not hard to see that F'' is isomorphic to a subgraph of F', and therefore, to a subgraph of G_2. Since $\mathcal{F}$ is closed under compression, $F'' \in \mathcal{F}$. Therefore G_2 has property $P_n^{\mathcal{F}}$. $\square$

Let k be a fixed integer with $0 \leq k \leq 2^t$. Recall that the group action on the edges creates 2^t orbits. Consider the family of all n-vertex invariant graphs G with $\mathcal{R}(G)$ and G_{right} fixed, and $\text{orb}(G) = k$. By Lemma 3.4.5, either all graphs in this family have property $P_n^{\mathcal{F}}$ or none of them does. The size of this family is $\binom{2^t}{k}$, which is even if $k \neq 0$ and $k \neq 2^t$. If $k = 2^t$, G_{left} is a complete graph, and so G contains a clique of size T, whence, by (3.6), we see that G has property $P_n^{\mathcal{F}}$. Therefore, by (3.8),

$$\chi_n \quad \equiv \quad \#\left\{G \; : \; \begin{array}{c} \operatorname{orb}(G) = 0 \text{ and } G \text{ is nontrivial,} \\ \Phi\text{-invariant, and does not satisfy } P_n^{\mathcal{F}} \end{array}\right\} \pmod 2\,. \qquad (3.10)$$

Lemma 3.4.6 *For $n \geq T = 2^{2^t}$, we have $\chi_n \equiv \chi_{n-T+1} \pmod 2$.*

Proof: Suppose we take a Φ-invariant G with $\operatorname{orb}(G) = 0$ and collapse all its left vertices into one vertex which we connect to every vertex in $\mathcal{R}(G)$ and to no others, thereby yielding a graph $\widetilde{G}$. This gives a bijection from labelled n-vertex invariant graphs G with $\operatorname{orb}(G) = 0$ to labelled $(n-T+1)$-vertex graphs.

It is clear that if $\widetilde{G}$ has property $P_{n-T+1}^{\mathcal{F}}$, then G has property $P_n^{\mathcal{F}}$. Now suppose G has property $P_n^{\mathcal{F}}$ and let $F \in \mathcal{F}$ be a subgraph of G. Since $\operatorname{orb}(G) = 0$, the vertices in F_{left} form an independent set; thus we may colour them all with one colour and then colour each remaining vertex of F with a distinct colour different from the one just used. This colouring produces a compression $\widetilde{F} \lhd F$ which is clearly a subgraph of $\widetilde{G}$. Since $\mathcal{F}$ is closed under compression, we have $\widetilde{F} \in \mathcal{F}$ and so $\widetilde{G}$ has property $P_{n-T+1}^{\mathcal{F}}$.

Using (3.10) and this bijection, we get

$$\chi_n \quad \equiv \quad \left\{\widetilde{G} \; : \; \widetilde{G} \text{ is nontrivial and does not satisfy } P_{n-T+1}^{\mathcal{F}}\right\} \pmod 2\,,$$

whence, by (3.7) $\chi_n \equiv \chi_{n-T+1} \pmod 2$. □

Proof of Lemma 3.4.3 (Technical Lemma): Set $n = T = 2^{2^t}$. The only way for an n-vertex graph G to have $\operatorname{orb}(G) = 0$ is for it to have no edges, i.e., to be the trivial graph. Using (3.10), this implies $\chi_T \equiv 0 \pmod 2$. Lemma 3.4.6 completes the proof. □

With this technical lemma in hand, we are finally in a position to tackle general subgraph containment properties.

Consider equation (3.5) from §3.4.1 and let $\mathcal{D}$ be any subset of $\mathcal{E}$. Then $G_{\mathcal{D}} := \bigcup_{A \in \mathcal{D}} A$ is a graph on n vertices with no intracluster edges and such that if $i \neq j$, the edges between V_i and V_j are either all present or all absent. Define a graph $\widetilde{G}_{\mathcal{D}}$ on r vertices $v_1, \ldots, v_r$ such that (v_i, v_j) is an edge iff all edges between V_i, V_j are present in $G_{\mathcal{D}}$.

Let $\mathcal{I}(H)$ denote the family of all graphs $\widetilde{H}$ such that $\widetilde{H} \lhd H$; in poset terminology, this is the principal order ideal generated by H under the partial order "$\lhd$." Clearly, $\mathcal{I}(H)$ is closed under compression. The following lemma connects the preceding discussion with that of §3.4.1.

Lemma 3.4.7 *The graph H is a subgraph of $G_{\mathcal{D}}$ if and only if there is a graph $\widetilde{H} \in \mathcal{I}(H)$ such that $\widetilde{H}$ is a subgraph of $\widetilde{G}_{\mathcal{D}}$. In other words, $G_{\mathcal{D}}$ satisfies Q_n^H iff $\widetilde{G}_{\mathcal{D}}$ satisfies $P_r^{\mathcal{I}(H)}$.*

Proof: Suppose H is a subgraph of $G_{\mathcal{D}}$. Consider the following colouring of $G_{\mathcal{D}}$: all vertices in a cluster are coloured the same and no two clusters use the same colour. This is a valid colouring since each cluster of vertices is an independent set. This colouring induces a colouring of H which in turn induces a compression $\widetilde{H} \lhd H$. Clearly this $\widetilde{H}$ is a subgraph of $\widetilde{G}_{\mathcal{D}}$.

Now suppose $\widetilde{H} \lhd H$ is a subgraph of $\widetilde{G}_{\mathcal{D}}$. Recall that q, from equation (3.3), is given to be large enough; suppose it is larger than $|V(H)|$. Consider the graph H_1 with vertices in $\cup_{i=1}^r V_i$ formed by taking all edges in E_{ij} whenever v_i and v_j are adjacent in $\widetilde{H}$. Since each $|V_i| \geq q \geq |V(H)|$, a straightforward argument shows that H is a subgraph of H_1, and therefore, of $G_{\mathcal{D}}$. $\square$

We now prove the remaining results from §3.2.2.

Proof of Theorem 3.2.4: Suppose Q_n^H is not evasive. From Theorem 3.3.3, we have $\chi(\Delta_\Gamma) = 1$. If $r = 1$, there is only one cluster, so by Lemma 3.4.1 we have $\Delta_\Gamma = \{\emptyset\}$, whence $\chi(\Delta_\Gamma) = 0$, a contradiction. Therefore $r > 1$. Equation (3.5) and Lemma 3.4.7 imply that there is a one-to-one correspondence between faces of Δ_Γ and nontrivial r-vertex graphs not satisfying property $P_r^{\mathcal{I}(H)}$. Hence the abstract complex Δ_Γ is the same as the abstract complex $\Delta\big(P_r^{\mathcal{I}(H)}\big)$. It follows from the definition of compression that $\mathcal{I}(H)$ contains the complete graph $K_{\mathrm{chr}(H)}$ and contains no smaller graph. Therefore, (3.6) yields $t = \lceil \log\log \mathrm{chr}(H) \rceil$. Setting $r_0 = 2^{2^t} - 1$ and applying the Technical Lemma, we have $\chi\big(\Delta\big(P_r^{\mathcal{I}(H)}\big)\big) \neq 1$ and so $\chi(\Delta_\Gamma) \neq 1$, a contradiction. □

Proof of Corollary 3.2.5: Choose an integer t such that $T = 2^{2^t}$ is at least $|V(H)|$. By Dirichlet's theorem there exist infinitely many primes p such that $p \equiv 2 \pmod{T-1}$. Fix one such $p \geq T$ and pick any $n \geq p^2(T-1)$ divisible by p. Now $p-1$ is relatively prime to $T-1$; therefore there is an integer x such that $x(p-1) \equiv n/p - 1 \pmod{T-1}$ and $0 \leq x < T-1$. From the lower bound on n we have $n/p - px > 0$. Therefore we can write

$$n \;=\; \sum_{i=1}^{x} p^2 + \sum_{i=1}^{n/p-px} p \,,$$

which is an expression of n as a sum of powers of p. The number of summands in this expression is $x + n/p - px \equiv 1 \pmod{T-1}$. Since $p \geq T \geq |V(H)|$, we can apply Theorem 3.2.4 to conclude that Q_n^H is evasive. □

Proof of Corollary 3.2.6: Let $\mathcal{A}$ be the arithmetic progression guaranteed by Corollary 3.2.5 and let c be the difference between two consecutive integers in $\mathcal{A}$. Let $f(n)$ denote the decision tree complexity of Q_n^H. If $n \in \mathcal{A}$ then $f(n) = \frac{1}{2}n^2 - \frac{1}{2}n$ and we are done. If not, there exists $n' \in \mathcal{A}$ with $n - c + 1 \leq n' < n$. Given an n'-vertex input graph

G', we can pretend that we have an n-vertex input graph G that induces the subgraph G' on vertices 1 through n' and contains no additional edges (i.e., we can simulate an oracle for such a graph G). We can now use a decision tree algorithm for Q_n^H to decide $Q_{n'}^H$ and conclude that $f(n') \leq f(n)$. This shows that $f(n) \geq \frac{1}{2}n'(n'-1) \geq \frac{1}{2}(n-c+1)(n-c) \geq \frac{1}{2}n^2 - cn$, as required. $\square$

3.5 Concluding Remarks

The major open question in the area of decision tree complexity of graph properties is to settle the Karp Conjecture. The pioneering work of Kahn, Saks, and Sturtevant [43] has given us a possible direction to follow in attempting to settle this conjecture. Our work takes steps in this direction by extending their topological approach to prove stronger results for a fairly general class of graph properties.

It is worth taking a high-level view of how our work extends that of [43]. Let us say that an action of group Γ on the vertex set of the input graph *absorbs* the property P if $(\Delta(P))_\Gamma = \{\emptyset\}$. Kahn, Saks, and Sturtevant's proof of Theorem 3.3.4 can be succinctly put thus: they construct, for any prime power n, a group action that absorbs *every* non-trivial monotone n-vertex graph property. A summary of *our* proofs for Theorems 3.2.2 and 3.2.3 would be that we construct a group action that absorbs minor-closed properties and bipartite subgraph containment for n-vertex graphs with $n \in N_0$. Here $N_0 \subseteq \mathbb{Z}$ is a special set of integers that depends on the exact property under consideration. Of course, what makes our result significant is that N_0 turns out to be much larger than the set of prime powers; in particular N_0 contains all large enough integers.

Despite the progress made by our methods here, there are significant hurdles to overcome before we can settle the Karp Conjecture. For now, our methods, and more generally, methods based on the topological approach are limited by the rather strong algebraic conditions that Theorem 3.3.3 imposes on the group Γ. These conditions make it very hard to produce results which hold for many values of n and for large classes of graph properties at the same time. Constructions such as ours seem to indicate that for many "natural" properties it may be much easier to prove the weaker statement that there is an n_0 such that the property is evasive for $n > n_0$.

The other difficulty with extending our approach has to do with the nature of the group action we construct. Our action has the effect of "collapsing" large vertex clusters in the input graph for certain specific graph properties. This collapse occurs because the properties under consideration are satisfied once the graph contains a large clique; the rest of the graph, outside of this clique, is immaterial. This is a feature of subgraph containment and minor-closed properties but we run into problems when trying to handle more "global" monotone properties such as bounded diameter.

The two hurdles we have mentioned suggest that one needs to look at more sophisticated group actions to use in Theorem 3.3.3.

A different, and more concrete, open problem suggested by our work here is to study the sequence χ_n as defined in §3.4.3. We conjecture that $\chi_n \neq 1$ for large enough n. If proved true, this conjecture would remove all number theoretic restrictions in Theorem 3.2.4.

Chapter 4

Informational Complexity and a Direct Sum Theorem

4.1 Introduction

Given m copies of the same problem, does it take m times the amount of resources to solve these m problems? This meta-question is called the *direct sum question*. It is a fundamental question that has been studied in many computational models. In this chapter, we consider this question in the simultaneous message (SM) model of communication introduced by Yao [63], and prove direct sum theorems for certain computations in the SM model.

Our proofs are based on a new complexity measure that we introduce: we call this measure *informational complexity*. In this chapter we define informational complexity in a way that connects it to SM complexity, but it will be clear that the notion is relevant to a more general class of computational models. Since the publication of the results presented here [22], other authors have successfully used informational complexity ideas in other computational settings [58, 10], showing that our notion is of independent interest.

4.1.1 The Problem and the Model

We consider the problem of computing a function $f(x, y)$ whose inputs are split between two parties: Alice, who holds x, and Bob, who holds y. The two-party communication model was introduced by Yao [63] to study the number of bits Alice and Bob need to communicate to each other to compute f. In the same paper Yao also introduced the simultaneous message (SM) model — which is the model we focus on, here — in which Alice and Bob do not talk to each other, but send messages to a third party, a referee, who then computes f based on these messages. We shall give formal definitions later (see §4.2.2).

Kushilevitz and Nisan's book [48] provides a comprehensive treatment of communication complexity, including the SM model. The survey paper by Babai [8] provides an overview of recent research.

Considerable effort has been spent in trying to understand the SM complexity of the (Boolean) equality function, EQ_n, defined as $\mathsf{EQ}_n(x, y) = 1$ iff $x = y$, for $x, y \in \{0, 1\}^n$. It is easy to show that deterministic protocols for EQ_n require n bits of communication, both in the two-party and the SM model. The problem becomes considerably more interesting if we allow randomised protocols (that err with low probability) with Alice and Bob flipping private coins to decide on their messages to the referee. From here on, we shall assume that our SM model is randomised, with private coins for Alice and Bob.

Various SM protocols for EQ_n that communicate only $O(\sqrt{n})$ bits were discovered by Ambainis [4], Newman and Szegedy [55], and Babai and Kimmel [9], amongst others. The last two papers also prove that this is tight, and therefore, that the SM complexity of

EQ_n is $\Theta(\sqrt{n})$. The paper [9] actually proves the following general lower bound on SM complexity, from which the bound for EQ_n follows.

Theorem 4.1.1 (Babai and Kimmel [9]) *For any Boolean function f, its SM complexity $C_{\mathrm{SM}}(f)$ and its deterministic two-party communication complexity $C_{\mathrm{2P}}(f)$ are related as follows: $C_{\mathrm{SM}}(f) = \Omega\big(\sqrt{C_{\mathrm{2P}}(f)}\big)$.*

Let us now ask the following direct sum question. Suppose Alice has m strings $x_1, \ldots, x_m \in \{0,1\}^n$ and Bob has m strings $y_1, \ldots, y_m \in \{0,1\}^n$. For each $i \in [m]$, let the bit $b_i \in \{0,1\}$ indicate equality between x_i and y_i; i.e., $b_i = \mathsf{EQ}_n(x_i, y_i)$. What is the SM complexity of computing *all* m bits $b_1, \ldots, b_m$? More generally, if $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ is an arbitrary Boolean function and $f^m : \{0,1\}^{mn} \times \{0,1\}^{mn} \to \{0,1\}^m$ is defined as the function obtained by concatenating the outputs of m copies of f with separate inputs for each copy, can we relate the complexity of f^m to that of f?

4.1.2 Our Results

Let us define $C(f) = C_{\mathrm{SM}}(f)$ for the remainder of this chapter. We answer the above question by proving the following direct sum theorem.

Theorem 4.1.2 $C(\mathsf{EQ}_n^m) = \Omega(m\sqrt{n})$.

A possible criticism of the above result is that we are proving a lower bound for a problem which requires outputting m bits. What if, instead of having to output all the bits $b_1, \ldots, b_m$, we only had to compute the majority (say) of these bits? Our next theorem defends us against this criticism. Let us define the Boolean functions

- $\mathsf{OREQ}_n^m(x_1, \ldots, x_m, y_1, \ldots, y_m) = \bigvee_{i=1}^m b_i\,,$
- $\mathsf{XOREQ}_n^m(x_1, \ldots, x_m, y_1, \ldots, y_m) = \bigoplus_{i=1}^m b_i\,,$

- $\mathsf{MAJEQ}_n^m(x_1, \ldots, x_m, y_1, \ldots, y_m) = 1$ iff $|\{i : b_i = 1\}| \geq \frac{1}{2}m$, and

- $\mathsf{THREQ}_n^{m,k}(x_1, \ldots, x_m, y_1, \ldots, y_m) = 1$ iff $|\{i : b_i = 1\}| \geq k$.

Theorem 4.1.3 *Let* $0 < \lambda < \frac{1}{2}$ *be a constant and suppose* $m \leq \lambda 2^n$. *Then*

i. $C(\mathsf{OREQ}_n^m) = \Omega(m\sqrt{n})$,

ii. $C(\mathsf{XOREQ}_n^m) = \Omega(m\sqrt{n})$,

iii. $C(\mathsf{MAJEQ}_n^m) = \Omega(m\sqrt{n})$, *and*

iv. $C(\mathsf{THREQ}_n^{m,k}) = \Omega((m - k + 1)\sqrt{n})$.

Note that each of the Boolean functions listed above has deterministic two-party communication complexity at most mn, wherefore an application of Theorem 4.1.1 yields at best an $\Omega(\sqrt{mn})$ lower bound for the SM complexity.

All of the above results were obtained in joint work with Shi, Wirth, and Yao [22], and our proofs here closely follow those in the original paper, although the presentation here is slightly different.

4.1.3 Our Techniques

We obtain the above results using an information theoretic approach. For any communication problem f we define a quantity called its *informational complexity*, denoted $IC(f)$, which describes the minimum requirement on the capacity of the "communication channels" between Alice/Bob and the referee for a uniformly distributed input (for precise definitions see §4.2.2). This quantity never exceeds the SM complexity $C(f)$.

Lemma 4.1.4 *Every function* $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ *satisfies* $IC(f) \leq C(f)$.

Moreover, informational complexity is a "well-behaved" measure with a neat direct sum property.

Theorem 4.1.5 (Additivity Theorem for Informational Complexity) *For any Boolean function f and integer m, $IC(f^m) \geq m \cdot IC(f)$.*

It would be very nice if we could prove a relation of the type $IC(f) = \Omega(C(f))$. Then, by Lemma 4.1.4 and the Additivity Theorem, we would have a direct sum theorem for the SM complexity of any function f. However, things are not so simple and we can only prove a much weaker inequality in the direction opposite to that of Lemma 4.1.4.

Definition 4.1.6 *For $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ and $X \subseteq \{0,1\}^n$, let $f|_X$ be the restriction of f to $X \times X$. We define*

$$\widetilde{C}(f) \quad = \quad \min_{X:\,|X| \geq (2/3)2^n} C(f|_X)\,.$$

We call f robust *if $\widetilde{C}(f) = \Omega(C(f))$.*

From the downward self-reducibility of the equality function, it follows that if $|X| \geq \frac{2}{3} 2^n$ then $C(\mathsf{EQ}_n) \leq C(\mathsf{EQ}_n|_X) + 1 \leq 2C(\mathsf{EQ}_n|_X)$. Thus, EQ_n is robust.

Our Main Lemma, which complements Lemma 4.1.4, says:

Lemma 4.1.7 (Main Lemma) *Every function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ satisfies $IC(f) = \Omega(\widetilde{C}(f)) - O(\log n)$. In particular, if f is robust, $IC(f) = \Omega(C(f)) - O(\log n)$.*

It should be clear that the above three results together imply Theorem 4.1.2. Of the three, the Main Lemma turns out to be the most technically involved, and the bulk of this chapter is devoted to its proof.

4.1.4 Relation to Earlier Work

Let us digress a little to compare our work to a few other closely related results. Feder, Kushilevitz, Naor, and Nisan [37] studied the direct sum question in the deterministic two-party communication model. They proved that there exists a partial function f such that $C_{\mathrm{2P}}(f) = \Theta(\log n)$, whereas for computing m copies of f we have $C_{\mathrm{2P}}(f^m) = O(m + \log m \cdot \log n)$; thus the average cost per copy is $O(1)$ for large m. This shows that a direct sum theorem does *not* hold for this function in this model. Incidentally, they also proved the lower bound $C_{\mathrm{2P}}(f^m) \geq m(\sqrt{C_{\mathrm{2P}}(f)/2} - \log n - O(1))$. In the same paper Feder et al. also considered the ε-error randomised two-party communication model and showed that for the equality problem, $C_{\mathrm{RAND}}(\mathsf{EQ}_n^m) = O(m + \log n)$, in contrast to the single copy case where $C_{\mathrm{RAND}}(\mathsf{EQ}_n) = \Theta(\log n)$.

In contrast to the above "anti-direct-sum" theorems, Feder et al. [37] showed that in the one-round deterministic model, for any f, essentially $C_{\mathrm{1R}}(f^m) = mC_{\mathrm{1R}}(f)$. This is a fairly strong direct sum theorem since it is not restricted to a particular function f. Karchmer, Kushilevitz, and Nisan [44] proved the same result in the two-round deterministic model. and in certain non-deterministic communication models.

No discussion on the direct sum question would be complete without citing a remarkable result due to Karchmer, Raz, and Wigderson [45]. They studied communication complexity for *relations*, a concept invented by Karchmer and Wigderson [46], where Alice, who holds a string x and Bob, who holds y, want to agree on a string z such that a relation $R(x, y, z)$ is true. They showed that the direct sum problem in this model is closely connected to the Boolean circuit depth problem, and that the proof of a certain direct sum theorem would lead to important results such as $\mathcal{NC}^1 \neq \mathcal{NC}^2$. Some results for this model were obtained in Karchmer, Raz, and Wigderson [45], Karchmer, Kushilevitz, and Nisan [44], and Edmonds, Impagliazzo, Rudich, and Sgall [35]. However, the crucial direct sum questions for this model remain wide open. In this thesis we will not have occasion to study communication complexity for relations.

4.2 Preliminaries for the Proof

Let us now return to the proof of our results. This section introduces some basic conventions and formally defines the concepts that we have introduced informally before. In §4.3 we prove Lemmas 4.1.4 and 4.1.7, thereby connecting informational complexity with SM complexity. In §4.4 we prove the Additivity Theorem and our main results: Theorems 4.1.2 and 4.1.3.

4.2.1 Conventions

We use a few special conventions throughout our proof and we call the reader's attention to them. Random variables in this chapter are always distributed over finite universes, and are denoted by boldface characters. For random variables $\mathbf{x}$ and $\mathbf{y}$, $H(\mathbf{x})$ denotes the Shannon entropy of $\mathbf{x}$ (or, more precisely, of the underlying probability distribution), $H(\mathbf{x} \mid \mathbf{y})$ denotes the entropy of $\mathbf{x}$ conditioned on $\mathbf{y}$ and $I(\mathbf{x}, \mathbf{y})$ denotes the mutual information between $\mathbf{x}$ and $\mathbf{y}$.

Let us quickly review these concepts. Suppose $\mathbf{x}$ is distributed over universe X. For each $x \in X$, let $p_x = \Pr[\mathbf{x} = x]$. Then $H(\mathbf{x})$ is defined as follows:

$$H(\mathbf{x}) \quad = \quad \sum_{\substack{x \in X \\ p_x \neq 0}} p_x \log \frac{1}{p_x} \,.$$

If $\mathbf{y}$ is another random variable, distributed over universe Y and possibly correlated with $\mathbf{x}$, the conditional entropy $H(\mathbf{x} \mid \mathbf{y})$ is defined as follows:

$$H(\mathbf{x} \mid \mathbf{y}) \quad = \quad \sum_{y \in Y} \Pr[\mathbf{y} = y] \cdot H(\mathbf{x} \mid \mathbf{y} = y) \,.$$

Let $\mathbf{x}$ and $\mathbf{y}$ be as above. We shall need the following well-known facts about entropy.

- $0 \le H(\mathbf{x}) \le \log|X|$.
- $H(\mathbf{x}) + H(\mathbf{y} \mid \mathbf{x}) = H(\mathbf{y}) + H(\mathbf{x} \mid \mathbf{y})$.

Finally, the mutual information $I(\mathbf{x}, \mathbf{y})$ is defined thus:

$$I(\mathbf{x}, \mathbf{y}) \quad = \quad H(\mathbf{x}) - H(\mathbf{x} \mid \mathbf{y}) \quad = \quad H(\mathbf{y}) - H(\mathbf{y} \mid \mathbf{x})\,.$$

We call the reader's attention to the following **unusual notation** that we shall use here: matrices are denoted by uppercase letters and their entries by the corresponding lowercase letters with indices in parentheses. Thus, if A is a matrix, its $(i, j)^{\mathrm{th}}$ entry is written as $a(i, j)$.

4.2.2 Definitions

Consider a Boolean function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$. A simultaneous message (SM) protocol for f with error bound ε is a quintuple (k, ℓ, P, Q, R) with k, ℓ integers and P, Q, R matrices of shape $2^n \times k$, $2^n \times \ell$ and $k \times \ell$ respectively. We assume that the rows of P and Q are indexed by elements of $\{0,1\}^n$. The matrices P and Q are required to be stochastic (i.e. their rows are required to sum to 1) and R is required to have entries in $\{0,1\}$. Finally, the product $F^* = PRQ^{\mathsf{T}}$ is required to satisfy

$$|f^*(x, y) - f(x, y)| \quad < \quad \varepsilon\,, \qquad \text{for all } x, y \in \{0,1\}^n\,. \tag{4.1}$$

The protocol works as follows: suppose Alice receives $x \in \{0,1\}^n$ and Bob receives $y \in \{0,1\}^n$. Alice interprets row x of matrix P as a probability distribution over the set

$[k]$ and picks a "message" from $[k]$ according to this distribution. Bob similarly encodes y into a message from $[\ell]$. They both send their messages to a referee, thereby communicating $\lceil \log k \rceil$ and $\lceil \log \ell \rceil$ bits respectively. The referee uses the messages to index into matrix R and outputs the corresponding entry of R. [1] Inequality (4.1) says that the probability that the referee correctly outputs $f(x, y)$ exceeds $1 - \varepsilon$.

We denote the set of all such ε-error protocols for f by $\Pi_\varepsilon(f)$. For a protocol $\mathcal{P} = (k, \ell, P, Q, R) \in \Pi_\varepsilon(f)$ we define $\mathrm{cost}_A(\mathcal{P}) = \lceil \log k \rceil$, $\mathrm{cost}_B(\mathcal{P}) = \lceil \log \ell \rceil$ and $\mathrm{cost}(\mathcal{P}) = \mathrm{cost}_A(\mathcal{P}) + \mathrm{cost}_B(\mathcal{P})$. The *simultaneous message communication complexity* (or SM complexity) of f is defined to be

$$C(f) \quad = \quad \min_{\mathcal{P} \in \Pi_{1/3}(f)} \mathrm{cost}(\mathcal{P})\,.$$

For protocol $\mathcal{P}$, let $\mathbf{x}, \mathbf{y}$ be random variables uniformly distributed over $\{0, 1\}^n$. Let random variables $\mathbf{u}$ and $\mathbf{v}$ denote the messages that Alice and Bob (respectively) generate upon receiving inputs $\mathbf{x}$ and $\mathbf{y}$. Clearly the distributions of $\mathbf{u}$ and $\mathbf{v}$ depend on the matrices P and Q. The *informational cost*, $\mathrm{icost}(\mathcal{P})$, of the protocol is defined to be $\mathrm{icost}_A(\mathcal{P}) + \mathrm{icost}_B(\mathcal{P})$ where $\mathrm{icost}_A(\mathcal{P}) = I(\mathbf{x}, \mathbf{u})$ and $\mathrm{icost}_B(\mathcal{P}) = I(\mathbf{y}, \mathbf{v})$. The *informational complexity* of f is defined to be

$$IC(f) \quad = \quad \min_{\mathcal{P} \in \Pi_{1/3}(f)} \mathrm{icost}(\mathcal{P})\,.$$

We remark that the choice of the constant $1/3$ in the above definitions is arbitrary; the error can be reduced from any constant to any smaller constant by trivially "repeating" the protocol and increasing the cost or informational cost only by a constant factor.

[1] One can make a more general definition where R has entries in $[0, 1]$ and the referee makes a probabilistic decision whether to output 0 or 1. However it is well known [55] that any such protocol can be converted to a protocol with deterministic referee by at most doubling the error.

For studying the direct sum question, we also need to define protocols for functions with multiple bits of output. For a function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$, we require Alice and Bob to behave as before, but let the referee use m 0-1 matrices $R_1, \ldots, R_m$, one for each bit. We call this an ε-error protocol if, for each i, the protocol given by R_i is an ε-error protocol for the i^{th} bit of f.[2] The concepts cost, icost, C and IC are defined exactly as before.

From here on, we shall consistently use the symbols $\mathbf{x}, \mathbf{y}, \mathbf{u}$, and $\mathbf{v}$ with the same meanings as in the above discussion.

4.3 Connecting Informational and SM Complexities

We now prove the two lemmas which essentially "sandwich" $IC(f)$ between $C(f)$ and $\Omega(\widetilde{C}(f))$. The upper bound follows almost trivially from the definition of informational complexity.

Proof of Lemma 4.1.4: Consider any protocol $\mathcal{P} = (k, \ell, P, Q, R) \in \Pi_{1/3}(f)$ and let $\mathbf{x}, \mathbf{y}, \mathbf{u}, \mathbf{v}$ have their usual meanings, as described above. Then

$$\begin{aligned}
\text{icost}(\mathcal{P}) &= H(\mathbf{u}) - H(\mathbf{u} \,|\, \mathbf{x}) + H(\mathbf{v}) - H(\mathbf{v} \,|\, \mathbf{y}) \\
&\leq H(\mathbf{u}) + H(\mathbf{v}) \\
&\leq \log k + \log \ell \\
&\leq \lceil \log k \rceil + \lceil \log \ell \rceil \\
&= \text{cost}(\mathcal{P}) .
\end{aligned}$$

[2] Another plausible definition would have required the referee to correctly output the *entire* m-bit result with probability more than $1 - \varepsilon$. Our weaker definition only makes our lower bound results stronger.

Since this holds for all $\mathcal{P}$, we conclude that $IC(f) \leq C(f)$. $\square$

Now let $\mathcal{P} = (k, \ell, P, Q, R) \in \Pi_\varepsilon(f)$ be an ε-error SM protocol for f, with ε a sufficiently small constant. We prove the Main Lemma (Lemma 4.1.7) in two major stages. In the first stage, we transform $\mathcal{P}$ into a new protocol that satisfies a property we call *uniformity*, while increasing the informational cost essentially by a constant factor. In the second stage, we convert this uniform protocol into a new protocol that has cost comparable to the informational cost of the uniform protocol. Unfortunately, this second stage also "damages" the protocol slightly, so that the function computed by the final protocol is not quite the same as that computed by $\mathcal{P}$. This is the reason why we can only use $\widetilde{C}(f)$, and not $C(f)$, to lower bound $IC(f)$.

For each $x \in \{0,1\}^n$ and $u \in [k]$ let us define the following sets, which depend on the protocol $\mathcal{P}$.

$$\begin{aligned} U_x &= \{u' \in [k] : p(x, u') > 0\} \\ X_u &= \{x' \in \{0,1\}^n : p(x', u) > 0\} \end{aligned}$$

A message $u \in [k]$ is said to be *smooth* if $p(x_1, u) = p(x_2, u)$, for all $x_1, x_2 \in X_u$.

Definition 4.3.1 *$\mathcal{P}$ is said to be (δ, s)-*uniform for Alice*, for an integer s and a real $\delta \in [0,1]$, if there is a subset $Z \subseteq [k]$ of her messages such that*

i. $\Pr[\mathbf{u} \in Z] < \delta$.

ii. *Each message $u \in [k] \setminus Z$ is smooth.*

iii. *Each message $u \in [k] \setminus Z$ satisfies $s \leq |X_u| < 2s$.*

The messages in Z are called irregular *messages and the rest* regular. *The quantity δ is called the* irregularity *of the protocol. Uniformity for Bob is defined similarly.*

For brevity, we shall call a protocol which is (δ, s)-uniform for Alice and (δ, s')-uniform for Bob a (δ, s, s')-uniform protocol.

Lemma 4.3.2 (Stage One Lemma) *Given a protocol $\mathcal{P} \in \Pi_\varepsilon(f)$ and a constant $0 < \delta < 1$, there is a protocol $\mathcal{P}' \in \Pi_\varepsilon(f)$ and integers s, s', such that $\mathcal{P}'$ is (δ, s, s')-uniform and* $\mathrm{icost}(\mathcal{P}') = O(\mathrm{icost}(\mathcal{P})) + O(\log n)$.

Lemma 4.3.3 (Stage Two Lemma) *For sufficiently small constants ε and δ, suppose $\mathcal{P}' \in \Pi_\varepsilon(f)$ is (δ, s, s')-uniform for some s and s'. Then there is an $X \subseteq \{0,1\}^n$ and a protocol $\mathcal{P}'' \in \Pi_{1/3}(f|_X)$ such that $|X| \geq \frac{2}{3} \cdot 2^n$ and* $\mathrm{cost}(\mathcal{P}'') \leq \mathrm{icost}(\mathcal{P}') + O(\log n)$.

Proof of Lemma 4.1.7 (Main Lemma): Suppose, for the moment, that the above two lemmas have been proven. Applying the Stage One Lemma to a protocol $\mathcal{P} \in \Pi_\varepsilon(f)$ and then applying the Stage Two Lemma to the resulting protocol $\mathcal{P}'$ gives a protocol $\mathcal{P}''$ with

$$\mathrm{cost}(\mathcal{P}'') = O(\mathrm{icost}(\mathcal{P})) + O(\log n),$$

and $\mathcal{P}'' \in \Pi_{1/3}(f|_X)$ for an X of size at least $\frac{2}{3} \cdot 2^n$. Clearly

$$\mathrm{cost}(\mathcal{P}'') \geq C(f|_X) \geq \widetilde{C}(f).$$

Therefore $\mathrm{icost}(\mathcal{P}) = \Omega(\widetilde{C}(f)) - O(\log n)$. $\square$

4.3.1 Stage One: Converting to a Uniform Protocol

Define the *degree* of a message u to be $|X_u|$. We convert the general protocol $\mathcal{P}$ into a uniform protocol $\mathcal{P}'$ by applying some transformations to Alice's side of the protocol and then applying these exact same transformations to Bob's side. Let us concentrate on Alice's side. We apply the transformations in two steps; the first step smoothens the messages and the second equalises the degrees up to a factor of 2. Each step consists of a "split" operation which divides each of Alice's messages into several new messages.

Let us formally define this split operation. Suppose we wanted to split one of Alice's messages, u, into K_u new messages $u^{(1)}, \ldots, u^{(K_u)}$. We replace the existing protocol with a new one in which Alice sends one of the $u^{(i)}$ at random (according to some distribution) whenever she would have sent u in the old protocol. The referee, for his part treats each of the messages $u^{(i)}$ just like u in the old protocol. Thus, if the old protocol was the quintuple (k, ℓ, P, Q, R), the new one is the quintuple $(k-1+K_u, \ell, P', Q, R')$ where P' is obtained by replacing the u^{th} column of P with K_u new column vectors which add up to this u^{th} column, and R' is obtained by replicating the u^{th} row of R K_u times. Finally, we can apply this operation to every message u and doing so gives us the most general split operation. We call the operation a *K-fold split* if $\max_u K_u \leq K$.

It is easy to see that applying a split operation to a protocol does not change its behaviour (i.e., the function computed by it) in any way.

It is also clear that a K-fold split increases the cost, on Alice's side, of the protocol by at most $1 + \log K$. The next lemma shows that such a bound applies for the informational cost as well.

Lemma 4.3.4 *Suppose protocol $\mathcal{P}'$ is obtained from protocol $\mathcal{P}$ by a K-fold split. Then* $\operatorname{icost}_A(\mathcal{P}') \leq \operatorname{icost}_A(\mathcal{P}) + \log K$.

Proof: Let $\mathbf{u}, \mathbf{u}'$ denote the random messages generated by Alice in protocols $\mathcal{P}, \mathcal{P}'$ respectively, when given a uniform random input $\mathbf{x}$. Suppose a particular message u gets split into K_u messages, $\{u^{(1)}, u^{(2)}, \ldots, u^{(K_u)}\}$, $K_u \leq K$, with associated new probabilities $p'\big(x, u^{(i)}\big)$. Let $p(u) = \Pr[\mathbf{u} = u]$ and $p'\big(u^{(i)}\big) = \Pr[\mathbf{u}' = u^{(i)}]$. Because of the way the probabilities are redistributed, $p(u) = \sum_{i \in [K_u]} p'\big(u^{(i)}\big)$.

Now, by definition of entropy, $H(\mathbf{u}) = \sum_{u \in [k]} -p(u) \log p(u)$. Moreover,

$$
\begin{aligned}
H(\mathbf{u}') &= \sum_{u \in [k]} \sum_{i \in [K_u]} -p'\big(u^{(i)}\big) \log p'\big(u^{(i)}\big) \\
&= \sum_u K_u \sum_i -\frac{1}{K_u} p'\big(u^{(i)}\big) \log p'\big(u^{(i)}\big) \\
&\leq \sum_u K_u \cdot \left(-\frac{p(u)}{K_u} \log \frac{p(u)}{K_u} \right) \qquad (4.2) \\
&= \sum_u -p(u) \left(\log p(u) - \log K_u \right) \\
&\leq H(\mathbf{u}) + \log K \, ,
\end{aligned}
$$

where (4.2) is obtained by applying Jensen's inequality to the inner summations.

It is also clear that, for any fixed x, $H(\mathbf{u}' \,|\, x) \geq H(\mathbf{u} \,|\, x)$. Therefore $H(\mathbf{u}' \,|\, \mathbf{x}) \geq H(\mathbf{u} \,|\, \mathbf{x})$. Thus,

$$
\begin{aligned}
I(\mathbf{x}, \mathbf{u}') &= H(\mathbf{u}') - H(\mathbf{u}' \,|\, \mathbf{x}) \\
&\leq H(\mathbf{u}) + \log K - H(\mathbf{u} \,|\, \mathbf{x}) \\
&= I(\mathbf{x}, \mathbf{u}) + \log K \, .
\end{aligned}
$$

Noting that $I(\mathbf{x}, \mathbf{u}) = \mathrm{icost}_A(\mathcal{P})$ and $I(\mathbf{x}, \mathbf{u}') = \mathrm{icost}_A(\mathcal{P}')$ completes the proof. $\square$

Smoothening

We begin by applying a split operation to $\mathcal{P}$ that "smoothens" Alice's messages. Fix a message $u \in [k]$ and let $\pi_u = \max_{x \in X_u} p(x,u)$. Let j be the integer that satisfies

$$\frac{1}{2^j} \;\leq\; \pi_u \;<\; \frac{1}{2^{j-1}}\,. \tag{4.3}$$

We shall replace message u by $n+c+2$ new messages, $\{u^{(0)}, u^{(1)}, \ldots, u^{(n+c)}, u_{\text{irr}}\}$, for some constant c, common to all the u's. The message u_{irr} will eventually end up an irregular message.

Let P' denote the matrix for Alice's new strategy. We "distribute" the probability $p(x,u)$ amongst the new probabilities given by P' as follows. Consider the binary representation of $p(x,u)$:

$$p(x,u) \;=\; \sum_{i=0}^{\infty} \frac{a_{j+i}}{2^{j+i}}\,,$$

where $a_i \in \{0,1\}$ (it does not matter that the a_i's are not uniquely determined). We set

$$p'\big(x, u^{(i)}\big) \;=\; \frac{a_{j+i}}{2^{j+i}} \tag{4.4}$$

for $0 \leq i \leq n+c$, and

$$p'(x, u_{\text{irr}}) \;=\; \sum_{i=n+c+1}^{\infty} \frac{a_{j+i}}{2^{j+i}}\,. \tag{4.5}$$

The end result, once we perform this operation for each $u \in [k]$, is an $(n+c+2)$-fold split. By Lemma 4.3.4, this causes $\text{icost}_A(\mathcal{P})$ to increase by at most $\log(n+c+2) = O(\log n)$.

Let $Z_1 = \{u_{\text{irr}} : u \in [k]\}$. From (4.4) we see that all the newly created messages not in Z_1 are smooth. Let $\mathbf{u}$ and $\mathbf{u}'$ denote the random messages that Alice sends in the original and modified protocols, respectively, upon receiving a uniformly distributed random input $\mathbf{x}$. We argue that $\mathbf{u}'$ is unlikely to land in Z_1. Consider an arbitrary $u \in [k]$. By (4.5),

$$p'(x, u_{\text{irr}}) \;\leq\; \sum_{i=n+c+1}^{\infty} \frac{1}{2^{j+i}} \;=\; \frac{1}{2^{j+n+c}} \;\leq\; \frac{\pi_u}{2^{n+c}}\,,$$

where the last inequality follows from (4.3). From the definition of π_u, we have

$$\pi_u \;\leq\; \sum_{x \in X_u} p(x, u) \;=\; 2^n \Pr[\mathbf{u} = u]\,.$$

Therefore $p'(x, u_{\text{irr}}) \leq \Pr[\mathbf{u} = u]/2^c$. Now

$$\Pr[\mathbf{u}' \in Z_1] \;=\; \frac{1}{2^n} \sum_{x \in \{0,1\}^n} \sum_{u \in [k]} p'(x, u_{\text{irr}}) \;\leq\; \sum_{u \in [k]} \frac{\Pr[\mathbf{u} = u]}{2^c} \;=\; \frac{1}{2^c}\,. \tag{4.6}$$

Equalising degrees of messages

We now have a protocol in which all messages, except those in Z_1, are smooth. Their degrees, however, could vary considerably. We apply another split operation (working only with the messages not in Z_1) whose effect is to "partition" the inputs for each message so that the degrees become more balanced.

For every message $u \notin Z_1$, let $I(u) = H(\mathbf{x}) - H(\mathbf{x} \mid u)$; since u is smooth,

$$I(u) \;=\; n - \log |X_u|\,. \tag{4.7}$$

Let $Z_2 = \{u : u \notin Z_1 \text{ and } I(u) > bI(\mathbf{x}, \mathbf{u})\}$, for some constant b. The messages in Z_2 will not be split and will end up as additional irregular messages. Since $I(\mathbf{x}, \mathbf{u}) =$

$\sum_{u\in U} \Pr[\mathbf{u}=u]\, I(u)$, Markov's inequality gives

$$\Pr[\mathbf{u} \in Z_2] \;\; \leq \;\; 1/b\,. \tag{4.8}$$

Now let Z, the final set of irregular messages, be defined as $Z = Z_1 \cup Z_2$. Combining (4.6) and (4.8), and setting b and c to large enough constants, we can ensure that

$$\Pr[\mathbf{u} \in Z] \;\; \leq \;\; \frac{1}{2^c} + \frac{1}{b} \;\; < \;\; \delta\,,$$

which bounds the irregularity of the final protocol.

Let $s = 2^{n - bI(\mathbf{x},\mathbf{u})}$. By (4.7), $|X_u| \geq s$ for all $u \notin Z_2$. We split every $u \notin Z_2$ into $K_u = \lfloor |X_u|/s \rfloor$ new messages, $\{u^{(1)}, \ldots, u^{(K_u)}\}$. The probabilities are distributed as follows: partition the inputs in X_u into K_u parts arbitrarily; if x is in the i^{th} part then assign $p'\big(x, u^{(i)}\big) = p(x,u)$, otherwise assign $p'\big(x, u^{(i)}\big) = 0$. This partitioning can be arranged so that the degrees of the new messages (i.e., the sizes of the parts) are all in the range $[s, 2s)$. Clearly the new messages are all smooth and so we have a protocol that is (δ, s)-uniform for Alice.

When a message u is split, it is divided into

$$K_u \;\; = \;\; \lfloor |X_u|/s \rfloor \;\; \leq \;\; 2^n/s \;\; = \;\; 2^{bI(\mathbf{x},\mathbf{u})}$$

new messages. Therefore the entire split operation is $2^{bI(\mathbf{x},\mathbf{u})}$-fold. By Lemma 4.3.4, the informational cost on Alice's side increases by at most $bI(\mathbf{x},\mathbf{u}) = O(\mathrm{icost}_A(\mathcal{P})) + O(\log n)$, where $\mathcal{P}$ was the original protocol, before the smoothening step.

Proof of Lemma 4.3.2 (Stage One Lemma): Starting with the given protocol, apply the two splitting operations described above to Alice's side. By the preceding discussion, the resulting protocol $\mathcal{P}_1$ is (δ, s)-uniform for Alice and $\mathrm{icost}_A(\mathcal{P}) = O(\mathrm{icost}_A(\mathcal{P})) + O(\log n)$. Notice that the transformation does not affect Bob's side of the protocol. Repeat the procedure for Bob to obtain a (δ, s, s')-uniform protocol $\mathcal{P}'$ with $\mathrm{icost}_B(\mathcal{P}') = O(\mathrm{icost}_B(\mathcal{P}_1)) + O(\log n) = O(\mathrm{icost}_B(\mathcal{P})) + O(\log n)$ and $\mathrm{icost}_A(\mathcal{P}') = \mathrm{icost}_A(\mathcal{P}_1) = O(\mathrm{icost}_A(\mathcal{P})) + O(\log n)$. $\square$

4.3.2 Stage Two: Working with a Uniform Protocol

Now suppose $\mathcal{P}'$ is a (δ, s, s')-uniform protocol in $\Pi_\varepsilon(f)$, where ε and δ are sufficiently small constants. Let Z denote the set of irregular messages of Alice. Once again, let $\mathbf{x}, \mathbf{y}, \mathbf{u}, \mathbf{v}$ have their usual meanings. By the definition of uniformity, we clearly have

$$n - \log s - 1 \quad < \quad I(\mathbf{x}, \mathbf{u}) \quad \leq \quad n - \log s\,. \tag{4.9}$$

The idea of the proof is to replace Alice's strategy by a new one so that her message set changes in size from k to about $O(2^n/s)$ while ensuring that the error of the protocol increases from ε to at most $O(\sqrt{\varepsilon})$ which can be made less than any constant. Alice's new message set will be an appropriate random sample of $[k]$, possibly with some elements of $[k]$ appearing multiple times (these multiple occurrences are, of course, considered distinct messages in the new protocol). With the right choice of parameters the new protocol will have cost on Alice's side comparable to $\mathrm{icost}_A(\mathcal{P}')$.

Fix inputs x, y for Alice and Bob and let P_x, Q_y be the corresponding rows of matrices P, Q. Let $V_x = U_x \setminus Z$ be the set of regular messages that Alice generates with positive probability, on input x. Since $\mathcal{P}'$ is a valid ε-error protocol, we have $|P_x R Q_y^{\mathsf{T}} - f(x,y)| < \varepsilon$. Consider the vector $R_y^* = RQ_y^{\mathsf{T}}$; its entries are to be thought of as being indexed by

Alice's messages. Notice that the u^{th} entry of R_y^* is simply the probability that the referee outputs "1" given that Alice sent message u.

Call a message $u \in V_x$ *defective* for (x, y) if the u^{th} entry of R_y^* differs from $f(x, y)$ by more than $\sqrt{\varepsilon}$. Messages not in V_x are defined to be not defective. Let $\tilde{\mathbf{u}}_x$ be a random variable distributed according to P_x. In other words, $\tilde{\mathbf{u}}_x$ is a random message that Alice generates according to her strategy on input x. From (4.1) we know that $|P_x R_y^* - f(x, y)| < \varepsilon$. A simple averaging argument then shows that

$$\Pr[\tilde{\mathbf{u}}_x \text{ is defective}] \quad < \quad \sqrt{\varepsilon}\,. \tag{4.10}$$

Definition 4.3.5 *An input x for Alice is said to be* active *if* $\Pr[\tilde{\mathbf{u}}_x \in Z] < \sqrt{\delta}$. *Active inputs for Bob are defined similarly.*

From the definition of uniformity and an averaging argument it follows that at most a $\sqrt{\delta}$ fraction of Alice's inputs are inactive. The same is true of Bob's inputs as well. If we choose δ small enough, there exists an $X \subseteq \{0, 1\}^n$ such that $|X| \geq \frac{2}{3} \cdot 2^n$ and inputs in $X \times X$ are active for both Alice and Bob.

Let $T = (u_1, u_2, \ldots)$ be a finite sequence of messages and let $|T|$ denote its length.

Definition 4.3.6 *We say that T is* good *for (x, y) if $T \cap V_x \neq \emptyset$ and*

$$|\{i : \, u_i \textit{ is defective}\}| \quad \leq \quad 6\sqrt{\varepsilon} \cdot |T \cap V_x|\,,$$

where $T \cap V_x$ is the subsequence of T consisting of only those elements that are in V_x.

Note that only those elements of T which belong to V_x affect whether or not T is good for (x, y).

We would like to prove that there exists a sequence T of appropriate length that is good for all active input pairs (x, y). We use the probabilistic method: we argue that a T chosen at random in a suitable manner is good for any fixed active pair (x, y) with probability greater than $1 - 2^{-2n}$. Since there are at most 2^{2n} different pairs (x, y), the result follows.

Lemma 4.3.7 *Suppose x is active and $u \in V_x$. Then*

$$\tfrac{1}{3}\cdot\Pr[\tilde{\mathbf{u}}_x = u] \;\;<\;\; \Pr[\mathbf{u} = u \mid \mathbf{u} \in U_x] \;\;<\;\; 3\cdot\Pr[\tilde{\mathbf{u}}_x = u]\,.$$

Proof: Recall that the protocol is (δ, s)-uniform for Alice. Therefore,

$$\begin{aligned}
\Pr[\mathbf{u} = u \mid \mathbf{u} \in U_x] &= \frac{\frac{1}{2^n}|X_u|p(x,u)}{\sum_{u' \in V_x} \frac{1}{2^n}|X_{u'}|p(x,u')} \\
&< \frac{2sp(x,u)}{\sum_{u' \in V_x} sp(x,u')} \\
&= \frac{2p(x,u)}{1 - \Pr[\tilde{\mathbf{u}}_x \in Z]} \\
&< \frac{2p(x,u)}{1 - \sqrt{\delta}} \\
&< 3\cdot\Pr[\tilde{\mathbf{u}}_x = u]\,.
\end{aligned}$$

The other inequality follows similarly. □

Define a random sequence $\mathbf{T}(\tau) = (\mathbf{u}_1, \ldots, \mathbf{u}_\tau)$ where the $\mathbf{u}_i$ are independent random variables distributed identically to $\mathbf{u}$. The length τ is a parameter.

Lemma 4.3.8 *Fix an index set $I \subseteq [\tau]$ and let $\mathbf{T} = \mathbf{T}(\tau)$. Let $\mathbf{T}_I$ denote the subsequence of $\mathbf{T}$ consisting of the elements indexed by I. Then*

$$\Pr\left[\mathbf{T} \textit{ is good} \mid \mathbf{T} \cap V_x = \mathbf{T}_I\right] \;\;\geq\;\; 1 - e^{-\frac{3}{4}\sqrt{\varepsilon}|I|}$$

Proof: To begin with, notice that the lemma holds vacuously if $I = \emptyset$.

Conditioned on $\mathbf{T} \cap V_x = \mathbf{T}_I$, it is the case that $\mathbf{T}$ is good iff $\mathbf{T}_I$ is. Now under this condition each element of $\mathbf{T}_I$ is distributed identically to $(\mathbf{u} \mid \mathbf{u} \in V_x)$. By Lemma 4.3.7 we see that the expected number of defective elements in $\mathbf{T}_I$ is at most $3|I|$ times the probability that $\tilde{\mathbf{u}}_x$ is defective, which, by (4.10), is at most $3\sqrt{\varepsilon}|I|$. By standard Chernoff bounds [3], the probability that $\mathbf{T}_I$ has at least $6\sqrt{\varepsilon}|I|$ defective elements is at most $e^{-\frac{3}{4}\sqrt{\varepsilon}|I|}$ and the result follows. □

Lemma 4.3.9 *Set $t = 2^{n+4}n/(s\sqrt{\varepsilon})$. Then there exists a sequence T with $|T| = t$ that is good for all (x, y) with x active.*

Proof: Consider the random sequence $\mathbf{T}(t)$ and fix a particular (x, y) such that x is active. By the definition of uniformity, each $\mathbf{u}_i$ lands in the set V_x with probability $\sum_{u \in V_x} \frac{1}{2^n}|X_u|p(x, u) \geq (s/2^n) \cdot \Pr[\tilde{\mathbf{u}}_x \in V_x] \geq (s/2^n)(1 - \sqrt{\delta}) \geq s/2^{n+1}$, for small enough δ. Therefore $\mathrm{E}[|\mathbf{T}(t) \cap V_x|] \geq ts/2^{n+1} = 8n/\sqrt{\varepsilon}$. Using a standard Chernoff bound [3],

$$\Pr\left[|\mathbf{T}(t) \cap V_x| \leq 4n/\sqrt{\varepsilon}\right] \quad \leq \quad e^{-n/\sqrt{\varepsilon}}. \tag{4.11}$$

Let $I \subseteq [t]$ be an index set with $|I| \geq 4n/\sqrt{\varepsilon}$. By Lemma 4.3.8,

$$\Pr\left[\mathbf{T}(t) \text{ is good} \mid \mathbf{T}(t) \cap V_x = \mathbf{T}(t)_I\right] \quad \geq \quad 1 - e^{-3n}.$$

Therefore,

$$\Pr\left[\mathbf{T}(t) \text{ is good}\right] \quad \geq \quad \Pr\left[\mathbf{T}(t) \text{ is good} \;\wedge\; |\mathbf{T}(t) \cap V_x| \geq 4n/\sqrt{\varepsilon}\right]$$

$$
\begin{aligned}
&\geq \sum_{\substack{I\subseteq[t] \\ |I|\geq 4n/\sqrt{\varepsilon}}} \left(1-e^{-3n}\right)\cdot \Pr\left[\mathbf{T}(t)\cap V_x = \mathbf{T}(t)_I\right] \\
&= (1-e^{-3n})\cdot \Pr\left[|\mathbf{T}(t)\cap V_x| \geq 4n/\sqrt{\varepsilon}\right] \\
&> 1-2^{-2n},
\end{aligned}
$$

where the last inequality follows from (4.11) and the fact that ε is small enough.

Since there are at most 2^{2n} pairs (x, y) with x active it follows that $\mathbf{T}(t)$ is good for all such (x, y) with positive probability. This proves the lemma. $\square$

Proof of Lemma 4.3.3 (Stage Two Lemma): Let X be a subset of $\{0,1\}^n$ of size at least $\frac{2}{3}\cdot 2^n$ such that all inputs $(x, y) \in X \times X$ are active for both Alice and Bob. Let T be a sequence of length $t = 2^{n+4}n/(s\sqrt{\varepsilon})$ that is good for all $(x, y) \in X \times X$; we know that such a sequence exists from Lemma 4.3.9.

We change Alice's strategy as follows: on input x, Alice chooses one of the elements in the subsequence $T \cap V_x$ uniformly at random and sends it to the referee. Let $\mathcal{P}_1$ be this new protocol.

To analyse the correctness of $\mathcal{P}_1$, consider an arbitrary input $(x_0, y_0) \in X \times X$. Since T is good for (x_0, y_0), the message chosen by Alice is defective for (x_0, y_0) with probability at most $6\sqrt{\varepsilon}$. By definition of defectiveness, if Alice's message is not defective, the referee's output is wrong with probability at most $\sqrt{\varepsilon}$. Therefore, overall, the referee errs with probability at most $7\sqrt{\varepsilon}$ on this input.

Applying (4.9) we have $\mathrm{cost}_A(\mathcal{P}_1) = \lceil \log t \rceil \leq I(\mathbf{x}, \mathbf{u}) + 6 + \log n - \log\sqrt{\varepsilon} = \mathrm{icost}_A(\mathcal{P}') + O(\log n)$.

Note that this transformation has not affected Bob's part of the protocol. Therefore, we can do another such transformation for Bob, this time not affecting Alice, and obtain a protocol $\mathcal{P}''$ with $\mathrm{cost}_A(\mathcal{P}'') = \mathrm{cost}_A(\mathcal{P}_1)$ and $\mathrm{cost}_B(\mathcal{P}'') = \mathrm{icost}_B(\mathcal{P}_1) + O(\log n) =$

$\text{icost}_B(\mathcal{P}') + O(\log n)$. The error, for inputs in $X \times X$, is now at most $7\sqrt{7\sqrt{\varepsilon}}$ which can be made at most $1/3$ by choosing ε small enough. Thus, $\mathcal{P}'' \in \Pi_{1/3}(f|_X)$ and we are done. $\square$

4.4 Proving SM Complexity Lower Bounds

In this section we prove the Additivity Theorem for informational complexity, which, together with Lemmas 4.1.4 and 4.1.7, implies Theorem 4.1.2. To prove Theorem 4.1.3 we have to replace the Additivity Theorem by a new theorem which lower bounds the informational complexity of problems like OREQ_n^m. Unlike the Additivity Theorem, this latter theorem makes use of a special property of the equality function, namely, that if x and y are drawn from $\{0,1\}^n$ uniformly at random, then $\mathsf{EQ}_n(x,y) = 0$ with very high probability.

4.4.1 A Direct Sum Theorem for SM Complexity

Proof of Theorem 4.1.5 (Additivity Theorem): We use the symbols x_i, y_i to denote fixed strings in $\{0,1\}^n$ and the symbols $\mathbf{x}_i, \mathbf{y}_i$ to denote random strings distributed uniformly in $\{0,1\}^n$.

Let $\mathcal{P} \in \Pi_{1/3}(f^m)$. For $1 \le i \le m$ and arbitrary strings $a = x_1x_2\ldots x_{i-1} \in \{0,1\}^{(i-1)\cdot n}$ and $b = y_1y_2\ldots y_{i-1} \in \{0,1\}^{(i-1)\cdot n}$, define the protocol $\mathcal{P}_{a,b}^{(i)}$ for f as follows: on receiving inputs x and y, Alice and Bob simulate $\mathcal{P}$ on the random inputs $x_1 \ldots x_{i-1} x\, \mathbf{x}_{i+1} \ldots \mathbf{x}_m$ and $y_1 \ldots y_{i-1} y\, \mathbf{y}_{i+1} \ldots \mathbf{y}_m$ respectively, and the referee uses the matrix for the i^{th} bit of output to determine f. Clearly, the referee will correctly output $f(x,y)$ with probability at least $2/3$, whence $\mathcal{P}_{a,b}^{(i)} \in \Pi_{1/3}(f)$.

Consider the probability distribution of Alice's message when she follows protocol $\mathcal{P}$ with the random input $z_1 \ldots z_j \, \mathbf{z}_{j+1} \ldots \mathbf{z}_m$. Let $H_j(z_1, \ldots, z_j)$ denote the entropy of this distribution and let h_j denote the expectation $\mathrm{E}[H_j(\mathbf{z}_1, \ldots, \mathbf{z}_j)]$. It is easy to see that

$$\mathrm{icost}_A(\mathcal{P}) \;=\; h_0 - h_m\,. \tag{4.12}$$

Further,

$$\mathrm{icost}_A\big(\mathcal{P}^{(i)}_{a,b}\big) \;=\; H_{i-1}(x_1, \ldots, x_{i-1}) - \mathrm{E}_{\mathbf{x}}[H_i(x_1, \ldots, x_{i-1}, \mathbf{x})]\,,$$

whence

$$\mathrm{E}_{\mathbf{a}}\left[\mathrm{icost}_A\big(\mathcal{P}^{(i)}_{\mathbf{a},b}\big)\right] \;=\; h_{i-1} - h_i\,, \tag{4.13}$$

where the expectation is over all $\mathbf{a}$ consisting of $i-1$ blocks of n-bit strings. Since this statement is true for all b, we can average over all $2^{(i-1)\cdot n}$ possible values of b in (4.13) and combine the resulting equation with (4.12) to obtain

$$\mathrm{icost}_A(\mathcal{P}) \;=\; \sum_{i=1}^{m} \mathrm{E}_{\mathbf{a},\mathbf{b}}\left[\mathrm{icost}_A\big(\mathcal{P}^{(i)}_{\mathbf{a},\mathbf{b}}\big)\right].$$

A similar equation can be derived for Bob's part of the protocols, whence

$$\begin{aligned} \mathrm{icost}(\mathcal{P}) \;&=\; \sum_{i=1}^{m} \mathrm{E}_{\mathbf{a},\mathbf{b}}\left[\mathrm{icost}\big(\mathcal{P}^{(i)}_{\mathbf{a},\mathbf{b}}\big)\right] \\ &\geq\; \sum_{i=1}^{m} IC(f)\,. \end{aligned}$$

Thus, $IC(f^m) \geq m \cdot IC(f)$. □

Proof of of Theorem 4.1.2: Recall that EQ_n is robust and that $C(\mathsf{EQ}_n) = \Omega(\sqrt{n})$. By the Main Lemma, $IC(\mathsf{EQ}_n) = \Omega(\sqrt{n}) - O(\log n) = \Omega(\sqrt{n})$. By the Additivity Theorem, we now have $IC(\mathsf{EQ}_n^m) = \Omega(m\sqrt{n})$. Therefore, by Lemma 4.1.4, $C(\mathsf{EQ}_n^m) = \Omega(m\sqrt{n})$ as desired. $\square$

4.4.2 Lower Bounds for Boolean Compositions of Equality Problems

Lemma 4.4.1 *For any constant λ with $0 < \lambda < \frac{1}{2}$ and integers m, n with $m \leq \lambda 2^n$, $IC(\mathsf{OREQ}_n^m) \geq \Omega(m\sqrt{n})$.*

Proof: We start with a protocol in $\mathcal{P}_0 \in \Pi_{1/3}(\mathsf{OREQ}_n^m)$ and "repeat" it $O(1)$ times to obtain a protocol $\mathcal{P} \in \Pi_\varepsilon(\mathsf{OREQ}_n^m)$ for some constant $\varepsilon < \frac{1}{2} - \lambda$. For $1 \leq i \leq m$ and an arbitrary string $a = w_1 w_2 \ldots w_{i-1} \in \{0,1\}^{(i-1)\cdot n}$, define the protocol $\mathcal{Q}_a^{(i)}$ for EQ_n as follows: on receiving inputs x and y, Alice and Bob simulate $\mathcal{P}$ on the random inputs $w_1 \ldots w_{i-1} x\, \mathbf{x}_{i+1} \ldots \mathbf{x}_m$ and $\overline{w}_1 \ldots \overline{w}_{i-1} y\, \mathbf{y}_{i+1} \ldots \mathbf{y}_m$ respectively, and the referee uses the same matrix as before. Here $\overline{w}_i$ denotes the bit-wise complement of w_i. Since

$$\Pr_{\mathbf{u},\mathbf{v} \in \{0,1\}^n} [\mathsf{EQ}_n(\mathbf{u}, \mathbf{v}) = 1] \;=\; 2^{-n}\,,$$

the referee fails to output $\mathsf{EQ}_n(x, y)$ correctly with probability at most $\varepsilon + (m-i) \cdot 2^{-n} \leq \varepsilon + m/2^n \leq \varepsilon + \lambda$, which is a constant less than $\frac{1}{2}$. Therefore we can get a protocol $\mathcal{P}_a^{(i)} \in \Pi_{1/3}(\mathsf{EQ}_n)$ by "repeating" $\mathcal{Q}_a^{(i)}$ an appropriate constant number of times, say c times.

Arguing exactly as in the proof of Theorem 4.1.5, we have

$$\begin{aligned}
\mathrm{icost}(\mathcal{P}) &= \sum_{i=1}^{m} \mathrm{E}_{\mathbf{a}}\left[\mathrm{icost}\big(\mathcal{Q}_{\mathbf{a}}^{(i)}\big)\right] \\
&= \sum_{i=1}^{m} \mathrm{E}_{\mathbf{a}}\left[\frac{1}{c}\cdot \mathrm{icost}\big(\mathcal{P}_{\mathbf{a}}^{(i)}\big)\right] \\
&\geq \frac{1}{c}\sum_{i=1}^{m} IC(\mathsf{EQ}_n) \\
&= \Omega(m\sqrt{n}),
\end{aligned}$$

and so $\mathrm{icost}(\mathcal{P}_0) = \Omega(m\sqrt{n})$. The result follows. □

Proof of Theorem 4.1.3: Combining Lemmas 4.4.1, 4.1.4, the Main Lemma and the Additivity Theorem, we easily derive $C(\mathsf{OREQ}_n^m) = \Omega(m\sqrt{n})$. The arguments for XOREQ_n^m, MAJEQ_n^m and $\mathsf{THREQ}_n^{m,k}$ are very similar — one just needs to modify Lemma 4.4.1 appropriately — and we shall omit the details. □

4.5 Concluding Remarks

We have introduced the powerful notion of informational complexity and used it to prove new lower bounds in the simultaneous message communication model. Information theoretic methods are a standard approach to problems about communication and it is natural to expect their application to be fruitful in communication *complexity*. Since several other areas of complexity theory have close ties with communication complexity [46, 52], we believe that our techniques have considerable potential.

We remark that the informational complexity approach also works for the one round two-party communication model, where Alice sends a single message to Bob who then computes the desired function. A straightforward analogue of the concept of informational complexity can be defined and used in analogues of Lemma 4.1.7 and Theorem 4.1.5. An observation of this type has been used by Saks and Sun [58] (implicitly), and by Bar-Yossef et al. [10] in their studies of the complexity of data stream computations. The latter paper also generalises our notion of informational complexity by allowing random inputs to be distributed non-uniformly on their domains, and by allowing conditioning on the entropy measures that define the informational cost. With these generalisations, Bar-Yossef et al. are then able to attack certain (multi-party, multi-round) communication complexity problems which easily lead to results about data stream computations.

An obvious open question raised by our work is whether we can avoid "damaging" the protocol while translating low informational cost into low communication cost. If so, we could then replace $\widetilde{C}(f)$ by $C(f)$ in the Main Lemma (Lemma 4.1.7) and we would have a neat direct sum theorem for all functions with $C(f) = \omega(\log n)$. Should this goal be too ambitious, one can then ask whether a non-damaging transformation is possible for a limited class of functions f, or, failing that, whether one can demonstrate robustness for functions significantly different from equality.

Chapter 5

Conclusions

Hardness results play a central role in complexity theory, and thus, in our understanding of computers and computing. Proving hardness results presents one of the greatest challenges in theoretical computer science. This thesis has described some of our work in this area and the new hardness results that we have obtained for some well-studied problems.

5.1 Summary of Contributions: Results

- We established the first non-trivial lower bound on the complexity of high dimensional approximate nearest neighbour searching. Our bound holds in the very general cell probe model, and works even for very weak approximation, but has the limitation that it only applies to deterministic algorithms.

- We made progress towards a settlement of Karp's evasiveness conjecture by showing that the conjecture holds for all minor-closed graph properties. We also obtained results for some other classes of graph properties, including subgraph containment.

- We proved the first non-trivial direct sum theorem in the simultaneous message model of communication complexity. Our result also gives the first lower bound in this model that is stronger than what can be proven using the Babai-Kimmel technique [9].

5.2 Summary of Contributions: Techniques

- We devised a framework for proving lower bounds on the cell probe complexity of deterministic query schemes via adversarial arguments, by adapting the pioneering ideas of Ajtai [1]. Our framework is general enough to admit results for a broader class of data structure problems [25, 12, 24].

- Using new algebraic and number-theoretic constructions, we extended the topological approach of Kahn, Saks, and Sturtevant [43] in order to prove the evasiveness of fairly general classes of graph properties.

- We introduced the notion of informational complexity and showed that it is a useful tool for proving lower bounds in the area of communication complexity. Further work by other researchers [58, 10] strongly suggests that our notion has quite a lot of potential waiting to be tapped.

Bibliography

[1] M. Ajtai. A lower bound for finding predecessors in Yao's cell probe model. *Combinatorica*, 8:235–247, 1988.

[2] E. Allender. Some pointed questions concerning asymptotic lower bounds, and news from the isomorphism front. In G. Paun, G. Rozenberg, and A. Salomaa, editors, *Current Trends in Theoretical Computer Science*, pages 25–41. World Scientific, 2001.

[3] N. Alon and J. H. Spencer. *The Probabilistic Method*. Wiley-Interscience, New York, 2000.

[4] A. Ambainis. Communication complexity in a 3-computer model. *Algorithmica*, 16(3):298–301, 1996.

[5] S. Arya and D. M. Mount. Approximate nearest neighbor searching. In *Proc. 4th Annu. ACM-SIAM Symp. Disc. Alg.*, pages 271–280, 1993.

[6] S. Arya, D. M. Mount, and O. Narayan. Accounting for boundary effects in nearest-neighbor searching. *Disc. Comput. Geom.*, 16(2):155–176, 1996.

[7] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Wu. An optimal algorithm for approximate nearest neighbor searching. *J. ACM*, 45(6):891–923, 1998. Preliminary version in *Proc. 5th Annu. ACM-SIAM Symp. Disc. Alg.*, pages 573–582, 1994.

[8] L. Babai. Communication complexity. Technical Report TR-2001-09, University of Chicago, 2001.

[9] L. Babai and P. G. Kimmel. Randomized simultaneous messages: Solution of a problem of Yao in communication complexity. In *Proc. 12th Annu. IEEE Conf. Comput. Complexity*, pages 239–246, 1997.

[10] Z. Bar-Yossef, T. S. Jayram, R. Kumar, and D. Sivakumar. An information statistics approach to data stream and communication complexity. In *Proc. 43rd Annu. IEEE Symp. Found. Comp. Sci.*, 2002. To appear.

[11] O. Barkol and Y. Rabani. Tighter lower bounds for nearest neighbor search and related problems in the cell probe model. In *Proc. 32nd Annu. ACM Symp. Theory Comput.*, pages 388–396, 2000.

[12] P. Beame and F. Fich. Optimal bounds for the predecessor problem. In *Proc. 31st Annu. ACM Symp. Theory Comput.*, pages 295–304, 1999.

[13] M. Bern. Approximate closest-point queries in high dimensions. *Inform. Process. Lett.*, 45(2):95–99, 1993.

[14] M. R. Best, P. van Emde Boas, and H. W. Lenstra, Jr. A sharpened version of the Aanderaa-Rosenberg conjecture. Technical Report Report ZW 30/74, Mathematisch Centrum, Amsterdam, The Netherlands, 1974.

[15] B. Bollobás. Complete subgraphs are elusive. *J. Combin. Theory Ser. B*, 21(1):1–7, 1976.

[16] B. Bollobás. *Extremal Graph Theory*. Academic Press, London, New York, 1978.

[17] A. Borodin, R. Ostrovsky, and Y. Rabani. Lower bounds for high dimensional nearest neighbor search and related problems. In *Proc. 31st Annu. ACM Symp. Theory Comput.*, pages 312–321, 1999.

[18] H. Buhrman, P. B. Miltersen, J. Radhakrishnan, and S. Venkatesh. Are bitvectors optimal? In *Proc. 32nd Annu. ACM Symp. Theory Comput.*, pages 449–458, 2000.

[19] F. Cazals. Effective nearest neighbours searching on the hyper-cube, with applications to molecular clustering. In *Proc. 14th Annu. ACM Symp. Comput. Geom.*, pages 222–230, 1998.

[20] A. Chakrabarti, B. Chazelle, B. Gum, and A. Lvov. A lower bound on the complexity of approximate nearest-neighbor searching on the hamming cube. In *Proc. 31st Annu. ACM Symp. Theory Comput.*, pages 305–311, 1999.

[21] A. Chakrabarti, S. Khot, and Y. Shi. Evasiveness of subgraph containment and related properties. *SIAM J. Comput.*, 31(3):866–875, 2002.

[22] A. Chakrabarti, Y. Shi, A. Wirth, and A. C. Yao. Informational complexity and the direct sum problem for simultaneous message complexity. In *Proc. 42nd Annu. IEEE Symp. Found. Comp. Sci.*, pages 270–278, 2001.

[23] T. Chan. Approximate nearest neighbor queries revisited. *Disc. Comput. Geom.*, 20(3):359–373, 1998. Preliminary version in *Proc. 13th Annu. ACM Symp. Comput. Geom.*, pages 352–358, 1997.

[24] B. Chazelle. Geometric searching over the rationals. In *Proc. 7th Annu. European Symp. Alg.*, pages 354–365, 1999.

[25] B. Chazelle. *The Discrepancy Method: Randomness and Complexity*. Cambridge University Press, Cambridge, 2000.

[26] K. L. Clarkson. A probabilistic algorithm for the post office problem. In *Proc. 17th Annu. ACM Symp. Theory Comput.*, pages 175–184, 1985.

[27] K. L. Clarkson. A randomized algorithm for closest-point queries. *SIAM J. Comput.*, 17(4):830–847, 1988.

[28] K. L. Clarkson. An algorithm for approximate closest-point queries. In *Proc. 10th Annu. ACM Symp. Comput. Geom.*, pages 160–164, 1994.

[29] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms, Second Edition*. MIT Press, Cambridge, 2001.

[30] T. M. Cover and P. E. Hart. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory*, 13(1):21–27, 1967.

[31] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *J. Amer. Soc. Inf. Sci.*, 41(6):391–407, 1990.

[32] Y. Dodis. *Space-Time Tradeoffs for Graph Properties*. PhD thesis, MIT, 1998.

[33] R. O. Duda and P. E. Hart. *Pattern classification and scene analysis*. John Wiley, New York, NY, 1973.

[34] A. G. Dyachkov and V. V. Rykov. Bounds on the length of disjunctive codes. *Problemy Peredaci Informacii*, 18(3):7–13, 1982.

[35] J. Edmonds, R. Impagliazzo, S. Rudich, and J. Sgall. Communication complexity towards lower bounds on circuit depth. *Comput. Complexity*, 10(3):210–246, 2001. Preliminary version in *Proc. 32nd Annu. IEEE Symp. Found. Comp. Sci.*, pages 249–257, 1991.

[36] P. Erdös, P. Frankl, and Z. Füredi. Families of finite sets in which no set is covered by the union of r others. *Israel J. Math.*, 51:79–89, 1985.

[37] T. Feder, E. Kushilevitz, M. Naor, and N. Nisan. Amortized communication complexity. *SIAM J. Comput.*, 24(4):736–750, 1995. Preliminary version in *Proc. 32nd Annu. IEEE Symp. Found. Comp. Sci.*, pages 239–248, 1991.

[38] M. L. Fredman, J. Komlós, and E. Szemerédi. Storing a sparse table with $O(1)$ worst case access time. *J. ACM*, 31(3):538–544, 1984. Preliminary version in *Proc. 23rd Annu. IEEE Symp. Found. Comp. Sci.*, pages 165–169.

[39] Z. Füredi. On r-cover-free families. *J. Combin. Theory Ser. A*, 73(1):172–173, 1996.

[40] S. Har-Peled. A replacement for Voronoi diagrams of near linear size. In *Proc. 42nd Annu. IEEE Symp. Found. Comput. Sci.*, pages 94–103, 2001.

[41] W. Hoeffding. Probability inequalities for sums of bounded random variables. *J. Amer. Stat. Assoc.*, 58(301):13–30, 1963.

[42] P. Indyk and R. Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proc. 30th ACM Symp. Theory Comput.*, pages 604–613, 1998.

[43] J. Kahn, M. Saks, and D. Sturtevant. A topological approach to evasiveness. *Combinatorica*, 4:297–306, 1984. Preliminary version in *Proc. 24th Annu. IEEE Symp. Found. Comp. Sci.*, pages 31–39, 1983.

[44] M. Karchmer, E. Kushilevitz, and N. Nisan. Fractional covers and communication complexity. *SIAM J. Disc. Math.*, 8(1):76–92, 1995. Preliminary version in *Structures in Complexity Theory*, pages 262–274, 1992.

[45] M. Karchmer, R. Raz, and A. Wigderson. Super-logarithmic depth lower bounds via the direct sum in communication complexity. *Comput. Complexity*, 5(3/4):191–204, 1995. Preliminary version in *Structures in Complexity Theory*, pages 299–304, 1991.

[46] M. Karchmer and A. Wigderson. Monotone circuits for connectivity require super-logarithmic depth. *SIAM J. Disc. Math.*, 3(2):255–265, 1990.

[47] J. M. Kleinberg. Two algorithms for nearest neighbor search in high dimensions. In *Proc. 29th Annu. ACM Symp. Theory Comput.*, pages 599–608, 1997.

[48] E. Kushilevitz and N. Nisan. *Communication Complexity*. Cambridge University Press, Cambridge, 1997.

[49] E. Kushilevitz, R. Ostrovsky, and Y. Rabani. Efficient search for approximate nearest neighbor in high-dimensional spaces. *SIAM J. Comput.*, 30(2):457–474, 2000. Preliminary version in *Proc. 30th Annu. ACM Symp. Theory Comput.*, pages 614–623, 1998.

[50] N. Linial, E. London, and Y. Rabinovich. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15(2):215–245, 1995. Preliminary version in *Proc. 35th Annu. IEEE Symp. Found. Comput. Sci.*, pages 577–591, 1994.

[51] E. C. Milner and D. J. A. Welsh. On the computational complexity of graph theoretical properties. In C. S. J. A. Nash-Williams and J. Sheehan, editors, *Proc. 5th British Combin. Conf.*, pages 471–487, Aberdeen, Scotland, 1976.

[52] P. B. Miltersen. Lower bounds for union-split-find related problems on random access machines. In *Proc. 26th Annu. ACM Symp. Theory Comput.*, pages 625–634, 1994.

[53] P. B. Miltersen. Cell probe complexity — a survey. *J. Disc. Alg.*, to appear.

[54] P. B. Miltersen, N. Nisan, S. Safra, and A. Wigderson. On data structures and asymmetric communication complexity. *J. Comput. Syst. Sci.*, 57(1):37–49, 1998. Preliminary version in *Proc. 27th Annu. ACM Symp. Theory Comput.*, pages 103–111, 1995.

[55] I. Newman and M. Szegedy. Public vs. private coin flips in one round communication games. In *Proc. 28th Annu. ACM Symp. Theory Comput.*, pages 561–570, 1996.

[56] R. L. Rivest and J. Vuillemin. On recognizing graph properties from adjacency matrices. *Theoretical Comput. Sci.*, 3(3):371–384, 1976.

[57] A. L. Rosenberg. On the time required to recognize properties of graphs: a problem. *SIGACT News*, 5:15–16, 1973.

[58] M. Saks and X. Sun. Space lower bounds for distance approximation in the data stream model. In *Proc. 34th Annu. ACM Symp. Theory Comput.*, pages 360–369, 2002.

[59] A. A. Salamov and V. V. Solovyev. Prediction of protein secondary structure by combining nearest-neighbor algorithms and multiple sequence alignments. *J. Mol. Biol.*, 247(1):11–15, 1995.

[60] C. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, New York, NY, 1983.

[61] E. Triesch. Some results on elusive graph properties. *SIAM J. Comput.*, 23(2):247–254, 1994.

[62] E. Triesch. On the recognition complexity of some graph properties. *Combinatorica*, 16(2):259–268, 1996.

[63] A. C. Yao. Some complexity questions related to distributive computing. In *Proc. 11th Annu. ACM Symp. Theory Comput.*, pages 209–213, 1979.

[64] A. C. Yao. Should tables be sorted? *J. ACM*, 28(3):615–628, 1981.

[65] A. C. Yao. Monotone bipartite graph properties are evasive. *SIAM J. Comput.*, 17(3):517–520, 1988.

[66] T. M. Yi and E. S. Lander. Protein secondary structure prediction using nearest-neighbor methods. *J. Mol. Biol.*, 232(4):1117–1129, 1993.

[67] P. N. Yianilos. Data structures and algorithms for nearest neighbor search in general metric spaces. In *Proc. 4th Annu. ACM-SIAM Symp. Disc. Alg.*, pages 311–321, 1993.