Limits on Efficient Computation in the Physical World

by

Scott Joel Aaronson

Bachelor of Science (Cornell University) 2000

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Computer Science

in the

GRADUATE DIVISION
of the
UNIVERSITY of CALIFORNIA, BERKELEY

Committee in charge:

Professor Umesh Vazirani, Chair
Professor Luca Trevisan
Professor K. Birgitta Whaley

Fall 2004

The dissertation of Scott Joel Aaronson is approved:

Chair Date

Date

Date

University of California, Berkeley

Fall 2004

Limits on Efficient Computation in the Physical World

Abstract

Limits on Efficient Computation in the Physical World

by

Scott Joel Aaronson
Doctor of Philosophy in Computer Science

University of California, Berkeley

Professor Umesh Vazirani, Chair

More than a speculative technology, quantum computing seems to challenge our most basic intuitions about how the physical world should behave. In this thesis I show that, while some intuitions from classical computer science must be jettisoned in the light of modern physics, many others emerge nearly unscathed; and I use powerful tools from computational complexity theory to help determine which are which.

In the first part of the thesis, I attack the common belief that quantum computing resembles classical exponential parallelism, by showing that quantum computers would face serious limitations on a wider range of problems than was previously known. In particular, any quantum algorithm that solves the *collision problem*—that of deciding whether a sequence of n integers is one-to-one or two-to-one—must query the sequence $\Omega\left(n^{1/5}\right)$ times. This resolves a question that was open for years; previously no lower bound better than constant was known. A corollary is that there is no "black-box" quantum algorithm to break cryptographic hash functions or solve the Graph Isomorphism problem in polynomial time. I also show that relative to an oracle, quantum computers could not solve NP-complete problems in polynomial time, even with the help of nonuniform "quantum advice states"; and that any quantum algorithm needs $\Omega\left(2^{n/4}/n\right)$ queries to find a local minimum of a black-box function on the n-dimensional hypercube. Surprisingly, the latter result also leads to new *classical* lower bounds for the local search problem. Finally, I give new lower bounds on quantum one-way communication complexity, and on the quantum query complexity of total Boolean functions and recursive Fourier sampling.

The second part of the thesis studies the relationship of the quantum computing model to physical reality. I first examine the arguments of Leonid Levin, Stephen Wolfram, and others who believe quantum computing to be fundamentally impossible. I find their arguments unconvincing without a "Sure/Shor separator"—a criterion that separates the already-verified quantum states from those that appear in Shor's factoring algorithm. I argue that such a separator should be based on a complexity classification of quantum states, and go on to create such a classification. Next I ask what happens to the quantum computing model if we take into account that the speed of light is finite—and in particular, whether Grover's algorithm still yields a quadratic speedup for searching a database. Refuting a claim by Benioff, I show that the surprising answer is yes. Finally, I analyze hypothetical models of computation that go even beyond quantum computing. I show that

many such models would be as powerful as the complexity class PP, and use this fact to give a simple, quantum computing based proof that PP is closed under intersection. On the other hand, I also present one model—wherein we could sample the entire history of a hidden variable—that appears to be more powerful than standard quantum computing, but only *slightly* so.

Professor Umesh Vazirani
Dissertation Committee Chair

Contents

List of Figures **vii**

List of Tables **viii**

1 "Aren't You Worried That Quantum Computing Won't Pan Out?" **1**

2 Overview **6**
2.1 Limitations of Quantum Computers . . . 7
2.1.1 The Collision Problem . . . 8
2.1.2 Local Search . . . 9
2.1.3 Quantum Certificate Complexity . . . 10
2.1.4 The Need to Uncompute . . . 11
2.1.5 Limitations of Quantum Advice . . . 11
2.2 Models and Reality . . . 13
2.2.1 Skepticism of Quantum Computing . . . 13
2.2.2 Complexity Theory of Quantum States . . . 13
2.2.3 Quantum Search of Spatial Regions . . . 14
2.2.4 Quantum Computing and Postselection . . . 15
2.2.5 The Power of History . . . 16

3 Complexity Theory Cheat Sheet **18**
3.1 The Complexity Zoo Junior . . . 19
3.2 Notation . . . 20
3.3 Oracles . . . 21

4 Quantum Computing Cheat Sheet **23**
4.1 Quantum Computers: N Qubits . . . 24
4.2 Further Concepts . . . 27

I Limitations of Quantum Computers **29**

5 Introduction **30**
5.1 The Quantum Black-Box Model . . . 31
5.2 Oracle Separations . . . 32

6 The Collision Problem **34**
6.1 Motivation . . . 36
6.1.1 Oracle Hardness Results . . . 36
6.1.2 Information Erasure . . . 36
6.2 Preliminaries . . . 37
6.3 Reduction to Bivariate Polynomial . . . 38
6.4 Lower Bound . . . 41
6.5 Set Comparison . . . 43
6.6 Open Problems . . . 46

7 Local Search **47**
7.1 Motivation . . . 49
7.2 Preliminaries . . . 51
7.3 Relational Adversary Method . . . 52
7.4 Snakes . . . 57
7.5 Specific Graphs . . . 60
7.5.1 Boolean Hypercube . . . 60
7.5.2 Constant-Dimensional Grid Graph . . . 64

8 Quantum Certificate Complexity **67**
8.1 Summary of Results . . . 68
8.2 Related Work . . . 70
8.3 Characterization of Quantum Certificate Complexity . . . 70
8.4 Quantum Lower Bound for Total Functions . . . 72
8.5 Asymptotic Gaps . . . 74
8.5.1 Local Separations . . . 76
8.5.2 Symmetric Partial Functions . . . 77
8.6 Open Problems . . . 78

9 The Need to Uncompute **79**
9.1 Preliminaries . . . 81
9.2 Quantum Lower Bound . . . 82
9.3 Open Problems . . . 87

10 Limitations of Quantum Advice **88**
10.1 Preliminaries . . . 91
10.1.1 Quantum Advice . . . 92
10.1.2 The Almost As Good As New Lemma . . . 93
10.2 Simulating Quantum Messages . . . 93
10.2.1 Simulating Quantum Advice . . . 96
10.3 A Direct Product Theorem for Quantum Search . . . 99
10.4 The Trace Distance Method . . . 103
10.4.1 Applications . . . 106
10.5 Open Problems . . . 110

11 Summary of Part I **112**

II Models and Reality 114

12 Skepticism of Quantum Computing **116**
12.1 Bell Inequalities and Long-Range Threads 119

13 Complexity Theory of Quantum States **126**
13.1 Sure/Shor Separators 127
13.2 Classifying Quantum States 130
13.3 Basic Results 135
13.4 Relations Among Quantum State Classes 138
13.5 Lower Bounds 141
13.5.1 Subgroup States 142
13.5.2 Shor States 146
13.5.3 Tree Size and Persistence of Entanglement 148
13.6 Manifestly Orthogonal Tree Size 149
13.7 Computing With Tree States 154
13.8 The Experimental Situation 157
13.9 Conclusion and Open Problems 160

14 Quantum Search of Spatial Regions **162**
14.1 Summary of Results 162
14.2 Related Work 164
14.3 The Physics of Databases 165
14.4 The Model 167
14.4.1 Locality Criteria 168
14.5 General Bounds 169
14.6 Search on Grids 173
14.6.1 Amplitude Amplification 174
14.6.2 Dimension At Least 3 175
14.6.3 Dimension 2 180
14.6.4 Multiple Marked Items 181
14.6.5 Unknown Number of Marked Items 184
14.7 Search on Irregular Graphs 185
14.7.1 Bits Scattered on a Graph 189
14.8 Application to Disjointness 190
14.9 Open Problems 191

15 Quantum Computing and Postselection **192**
15.1 The Class PostBQP 193
15.2 Fantasy Quantum Mechanics 196
15.3 Open Problems 198

16 The Power of History **199**
16.1 The Complexity of Sampling Histories 200
16.2 Outline of Chapter 201
16.3 Hidden-Variable Theories 203
16.3.1 Comparison with Previous Work 205
16.3.2 Objections 206
16.4 Axioms for Hidden-Variable Theories 206
16.4.1 Comparing Theories 207
16.5 Impossibility Results 208
16.6 Specific Theories 211
16.6.1 Flow Theory 211
16.6.2 Schrödinger Theory 215
16.7 The Computational Model 218
16.7.1 Basic Results 219
16.8 The Juggle Subroutine 220
16.9 Simulating SZK 221
16.10 Search in $N^{1/3}$ Queries 224
16.11 Conclusions and Open Problems 226

17 Summary of Part II **228**

Bibliography **229**

List of Figures

1.1 Conway's Game of Life 2

3.1 Known relations among 14 complexity classes 21

4.1 Quantum states of one qubit 25

7.1 A snake of vertices flicks its tail 58
7.2 The coordinate loop in 3 dimensions 64

13.1 Sure/Shor separators 128
13.2 Tree representing a quantum state 129
13.3 Known relations among quantum state classes 131

14.1 Quantum robot searching a 2D grid 163
14.2 The 'starfish' graph 171
14.3 Disjointness in $O(\sqrt{n})$ communication 191

15.1 Simulating PP using postselection 195

16.1 Flow network corresponding to a unitary matrix 211

List of Tables

8.1 Query complexity and certificate complexity measures 68

10.1 Expressions for $p_{x,ijkl}$ 109

12.1 Four objections to quantum computing 116

14.1 Summary of bounds for spatial search 163
14.2 Divide-and-conquer versus quantum walks 165

16.1 Four hidden-variable theories and the axioms they satisfy 208

Acknowledgements

My adviser, Umesh Vazirani, once said that he admires the quantum adiabatic algorithm because, like a great squash player, it achieves its goal while moving as little as it can get away with. Throughout my four years at Berkeley, I saw Umesh inculcate by example his "adiabatic" philosophy of life: a philosophy about which papers are worth reading, which deadlines worth meeting, and which research problems worth a fight to the finish. Above all, the concept of "beyond hope" does not exist in this philosophy, except possibly in regard to computational problems. My debt to Umesh for his expert scientific guidance, wise professional counsel, and generous support is obvious and beyond my ability to embellish. My hope is that I graduate from Berkeley a more adiabatic person than when I came.

Admittedly, if the push to finish this thesis could be called adiabatic, then the spectral gap was exponentially small. As I struggled to make the deadline, I relied on the help of David Molnar, who generously agreed to file the thesis in Berkeley while I remained in Princeton; and my committee—consisting of Umesh, Luca Trevisan, and Birgitta Whaley—which met procrastination with flexibility.

Silly as it sounds, a principal reason I came to Berkeley was to breathe the same air that led Andris Ambainis to write his epochal paper "Quantum lower bounds by quantum arguments." Whether or not the air in 587 Soda did me any good, Part I of the thesis is essentially a 150-page tribute to Andris—a colleague whose unique combination of genius and humility fills everyone who knows him with awe.

The direction of my research owes a great deal as well to Ronald de Wolf, who periodically emerges from his hermit cave to challenge non-rigorous statements, eat dubbel zout, or lament American ignorance. While I can see eye-to-eye with Ronald about (say) the $\mathrm{D}(f)$ versus $\mathrm{bs}(f)^2$ problem, I still feel that Andrei Tarkovsky's *Solaris* would benefit immensely from a car chase.

For better or worse, my conception of what a thesis should be was influenced by Dave Bacon, quantum computing's elder clown, who entitled the first chapter of his own 451-page behemoth "Philosonomicon." I'm also indebted to Chris Fuchs and his *samizdat*, for the idea that a document about quantum mechanics more than 400 pages long can be worth reading most of the way through.

I began working on the best-known result in this thesis, the quantum lower bound for the collision problem, during an unforgettable summer at Caltech. Leonard Schulman and Ashwin Nayak listened patiently to one farfetched idea after another, while John Preskill's weekly group meetings helped to ensure that the mysteries of quantum mechanics, which inspired me to tackle the problem in the first place, were never far from my mind. Besides Leonard, Ashwin, and John, I'm grateful to Ann Harvey for putting up with the growing mess in my office. For the record, I never once slept in the office; the bedsheet was strictly for doing math on the floor.

I created the infamous Complexity Zoo web site during a summer at CWI in Amsterdam, a visit enlivened by the presence of Harry Buhrman, Hein Röhrig, Volker Nannen, Hartmut Klauck, and Troy Lee. That summer I also had memorable conversations with David Deutsch and Stephen Wolfram. Chapters 7, 13, and 16 partly came into being during a semester at the Hebrew University in Jerusalem, a city where "Aaron's sons" were already obsessing about cubits three thousand years ago. I thank Avi Wigderson, Dorit

Aharonov, Michael Ben-Or, Amnon Ta-Shma, and Michael Mallin for making that semester a fruitful and enjoyable one. I also thank Avi for pointing me to the then-unpublished results of Ran Raz on which Chapter 13 is based, and Ran for sharing those results.

A significant chunk of the thesis was written or revised over two summers at the Perimeter Institute for Theoretical Physics in Waterloo. I thank Daniel Gottesman, Lee Smolin, and Ray Laflamme for welcoming a physics doofus to their institute, someone who thinks the string theory versus loop quantum gravity debate should be resolved by looping over all possible strings. From Marie Ericsson, Rob Spekkens, and Anthony Valentini I learned that theoretical physicists have a better social life than theoretical computer scientists, while from Dan Christensen I learned that complexity and quantum gravity had better wait before going steady.

Several ideas were hatched or incubated during the yearly QIP conferences; workshops in Toronto, Banff, and Leiden; and visits to MIT, Los Alamos, and IBM Almaden. I'm grateful to Howard Barnum, Andrew Childs, Elham Kashefi, Barbara Terhal, John Watrous, and many others for productive exchanges on those occasions.

Back in Berkeley, people who enriched my grad-school experience include Neha Dave, Julia Kempe, Simone Severini, Lawrence Ip, Allison Coates, David Molnar, Kris Hildrum, Miriam Walker, and Shelly Rosenfeld. Alex Fabrikant and Boriska Toth are forgiven for the cruel caricature that they attached to my dissertation talk announcement, provided they don't try anything like that ever again. The results on one-way communication in Chapter 10 benefited greatly from conversations with Oded Regev and Iordanis Kerenidis, while Andrej Bogdanov kindly supplied the explicit erasure code for Chapter 13. I wrote Chapter 7 to answer a question of Christos Papadimitriou.

I did take some actual ... *courses* at Berkeley, and I'm grateful to John Kubiatowicz, Stuart Russell, Guido Bacciagaluppi, Richard Karp, and Satish Rao for not failing me in theirs. Ironically, the course that most directly influenced this thesis was Tom Farber's magnificent short fiction workshop. A story I wrote for that workshop dealt with the problem of transtemporal identity, which got me thinking about hidden-variable interpretations of quantum mechanics, which led eventually to the collision lower bound. No one seems to believe me, but it's true.

The students who took my "Physics, Philosophy, Pizza" course remain one of my greatest inspirations. Though they were mainly undergraduates with liberal arts backgrounds, they took nothing I said about special relativity or Gödel's Theorem on faith. If I have any confidence today in my teaching abilities; if I think it possible for students to show up to class, and to participate eagerly, without the usual carrot-and-stick of grades and exams; or if I find certain questions, such as how a superposition over exponentially many 'could-have-beens' can collapse to an 'is,' too vertiginous to be pondered only by nerds like me, then those pizza-eating students are the reason.

Now comes the part devoted to the mist-enshrouded pre-Berkeley years. My initiation into the wild world of quantum computing research took place over three summer internships at Bell Labs: the first with Eric Grosse, the second with Lov Grover, and the third with Rob Pike. I thank all three of them for encouraging me to pursue my interests, even if the payoff was remote and, in Eric's case, not even related to why I was hired. Needless to say, I take no responsibility for the subsequent crash of Lucent's stock.

As an undergraduate at Cornell, I was younger than my classmates, invisible to many of the researchers I admired, and profoundly unsure of whether I belonged there or had any future in science. What made the difference was the unwavering support of one professor, Bart Selman. Busy as he was, Bart listened to my harebrained ideas about genetic algorithms for SAT or quantum chess-playing, invited me to give talks, guided me to the right graduate programs, and generally treated me like a future colleague. As a result, his conviction that I could succeed at research gradually became my conviction too. Outside of research, Christine Chung, Fion Luo, and my Telluride roommate Jason Stockmann helped to warm the Ithaca winters, Lydia Fakundiny taught me what an essay is, and Jerry Abrams provided a much-needed boost.

Turning the clock back further, my earliest research foray was a paper on hypertext organization, written when I was fifteen and spending the year at Clarkson University's unique Clarkson School program. Christopher Lynch generously agreed to advise the project, and offered invaluable help as I clumsily learned how to write a C program, prove a problem NP-hard, and conduct a user experiment (one skill I've never needed again!). I was elated to be trading ideas with a wise and experienced researcher, only months after I'd escaped from the prison-house of high school. Later, the same week the rejection letters were arriving from colleges, I learned that my first paper had been accepted to SIGIR, the main information retrieval conference. I was filled with boundless gratitude toward the entire scientific community—for struggling, against the warp of human nature, to judge ideas rather than the personal backgrounds of their authors. Eight years later, my gratitude and amazement are undiminished.

Above all, I thank Alex Halderman for a friendship that's spanned twelve years and thousands of miles, remaining as strong today as it was amidst the *Intellectualis minimi* of Newtown Junior High School; my brother David for believing in me, and for making me prouder than he realizes by doing all the things I didn't; and my parents for twenty-three years of harping, kvelling, chicken noodle soup, and never doubting for a Planck time that I'd live up to my potential—even when I couldn't, and can't, share their certainty.

Chapter 1

"Aren't You Worried That Quantum Computing Won't Pan Out?"

For a century now, physicists have been telling us strange things: about twins who age at different rates, particles that look different when rotated 360°, a force that is transmitted by gravitons but is also the curvature of spacetime, a negative-energy electron sea that pervades empty space, and strangest of all, "probability waves" that produce fringes on a screen when you don't look and don't when you do. Yet ever since I learned to program, I suspected that such things were all "implementation details" in the source code of Nature, their study only marginally relevant to forming an accurate picture of reality. Physicists, I thought, would eventually realize that the state of the universe can be represented by a finite string of bits. These bits would be the "pixels" of space, creating the illusion of continuity on a large scale much as a computer screen does. As time passed, the bits would be updated according to simple rules. The specific form of these rules was of no great consequence—since according to the Extended Church-Turing Thesis, any sufficiently complicated rules could simulate any other rules with reasonable efficiency.[1] So apart from practical considerations, why worry about Maxwell's equations, or Lorentz invariance, or even mass and energy, if the most fundamental aspects of our universe already occur in Conway's Game of Life (see Figure 1.1)?

Then I heard about Shor's algorithm [219] for factoring integers in polynomial time on a quantum computer. Then as now, many people saw quantum computing as at best a speculative diversion from the "real work" of computer science. Why devote one's research career to a type of computer that might never see application within one's lifetime, that faces daunting practical obstacles such as decoherence, and whose most publicized success to date has been the confirmation that, with high probability, $15 = 3 \times 5$ [234]? Ironically, I might have agreed with this view, had I not taken the Extended Church-Turing Thesis so seriously as a claim about reality. For Shor's algorithm forces us to accept that, under

[1] Here "extended" refers to the efficiency requirement, which was not mentioned in the original Church-Turing Thesis. Also, I am simply using the standard terminology, sidestepping the issue of whether Church and Turing themselves intended to make a claim about physical reality.

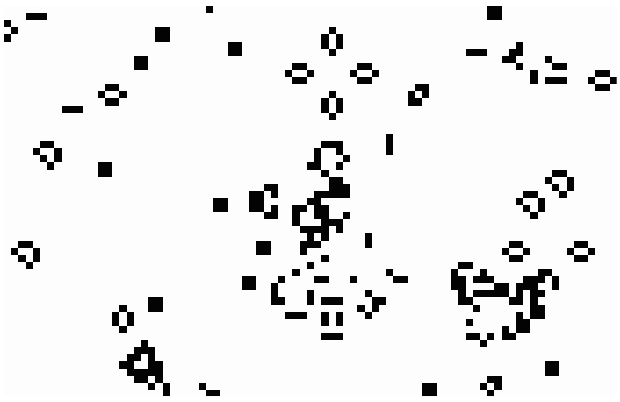

Figure 1.1: In Conway's Game of Life, each cell of a 2D square grid becomes 'dead' or 'alive' based on how many of its eight neighbors were alive in the previous time step. A simple rule applied iteratively leads to complex, unpredictable behavior. In what ways is our physical world similar to Conway's, and in what ways is it different?

widely-believed assumptions, that Thesis conflicts with the experimentally-tested rules of quantum mechanics as we currently understand them. Either the Extended Church-Turing Thesis is false, or quantum mechanics must be modified, or the factoring problem is solvable in classical polynomial time. All three possibilities seem like wild, crackpot speculations—but at least one of them is true!

The above conundrum is what underlies my interest in quantum computing, far more than any possible application. Part of the reason is that I am neither greedy, nefarious, nor number-theoretically curious enough ever to have hungered for the factors of a 600-digit integer. I *do* think that quantum computers would have benign uses, the most important one being the simulation of quantum physics and chemistry.[2] Also, as transistors approach the atomic scale, ideas from quantum computing are likely to become pertinent even for *classical* computer design. But none of this quickens my pulse.

For me, quantum computing matters because it combines two of the great mysteries bequeathed to us by the twentieth century: the nature of quantum mechanics, and the ultimate limits of computation. It would be astonishing if such an elemental connection between these mysteries shed no new light on either of them. And indeed, there is already a growing list of examples [9, 22, 151]—we will see several of them in this thesis—in which ideas from quantum computing have led to new results about *classical* computation. This should not be surprising: after all, many celebrated results in computer science involve only *deterministic* computation, yet it is hard to imagine how anyone could have proved them had computer scientists not long ago "taken randomness aboard."[3] Likewise, taking quantum mechanics aboard could lead to a new, more general perspective from which to revisit the central questions of computational complexity theory.

The other direction, though, is the one that intrigues me even more. In my view,

[2]Followed closely by Recursive Fourier Sampling, parity in $n/2$ queries, and efficiently deciding whether a graph is a scorpion.

[3]A few examples are primality testing in P [17], undirected connectivity in L [202], and inapproximability of 3-SAT unless $\mathsf{P} = \mathsf{NP}$ [224].

quantum computing has brought us *slightly* closer to the elusive Beast that devours Bohmians for breakfast, Copenhagenists for lunch, and a linear combination of many-worlders and consistent historians for dinner—the Beast that tramples popularizers, brushes off arXiv preprints like fleas, and snorts at the word "decoherence"—the Beast so fearsome that physicists since Bohr and Heisenberg have tried to argue it away, as if semantics could banish its unitary jaws and complex-valued tusks. But no, the Beast is there whenever you aren't paying attention, following all possible paths in superposition. Look, and suddenly the Beast is gone. But what does it even mean to look? If you're governed by the same physical laws as everything else, then why don't *you* evolve in superposition too, perhaps until someone else looks at you and thereby 'collapses' you? But then who collapses whom first? Or if you never collapse, then what determines what *you*-you, rather than the superposition of you's, experience? Such is the riddle of the Beast,[4] and it has filled many with terror and awe.

The contribution of quantum computing, I think, has been to show that the real nature of the Beast lies in its *exponentiality*. It is not just two, three, or a thousand states held in ghostly superposition that quantum mechanics is talking about, but an astronomical multitude, and these states could in principle reveal their presence to us by factoring a five-thousand-digit number. Much more than even Schrödinger's cat or the Bell inequalities, this particular discovery ups the ante—forcing us either to swallow the full quantum brew, or to stop saying that we believe in it. Of course, this is part of the reason why Richard Feynman [108] and David Deutsch [90] introduced quantum computing in the first place, and why Deutsch, in his defense of the many-worlds interpretation, issues a famous challenge to skeptics [92, p. 217]: if parallel universes are not physically real, then *explain how Shor's algorithm works*.

Unlike Deutsch, here I will not use quantum computing to defend the many-worlds interpretation, or any of its competitors for that matter. Roughly speaking, I agree with *every* interpretation of quantum mechanics to the extent that it acknowledges the Beast's existence, and disagree to the extent that it claims to have caged the Beast. I would adopt the same attitude in computer science, if instead of freely admitting (for example) that P versus NP is an open problem, researchers had split into "equalist," "unequalist," and "undecidabilist" schools of interpretation, with others arguing that the whole problem is meaningless and should therefore be abandoned.

Instead, in this thesis I will show how adopting a computer science perspective can lead us to *ask better questions*—nontrivial but answerable questions, which put old mysteries in a new light even when they fall short of solving them. Let me give an example. One of the most contentious questions about quantum mechanics is whether the individual components of a wavefunction should be thought of as "really there" or as "mere potentialities." When we don our computer scientist goggles, this question morphs into a different one: *what resources are needed to make a particular component of the wavefunction manifest?* Arguably the two questions are related, since something "real" ought to take less work to manifest than something "potential." For example, this thesis gradually became

[4] Philosophers call the riddle of the Beast the "measurement problem," which sounds less like something that should cause insomnia and delirious raving in all who have understood it. Basically, the problem is to reconcile a picture of the world in which "everything happens simultaneously" with the fact that you (or at least I!) have a sequence of definite experiences.

more real as less of it remained to be written.

Concretely, suppose our wavefunction has 2^n components, all with equal amplitude. Suppose also that we have a procedure to recognize a particular component x (i.e., a function f such that $f(x) = 1$ and $f(y) = 0$ for all $y \neq x$). Then how often must we apply this procedure before we make x manifest; that is, observable with probability close to 1? Bennett, Bernstein, Brassard, and Vazirani [51] showed that $\sim 2^{n/2}$ applications are necessary, even if f can be applied to all 2^n components in superposition. Later Grover [139] showed that $\sim 2^{n/2}$ applications are also sufficient. So if we imagine a spectrum with "really there" (1 application) on one end, and "mere potentiality" ($\sim 2^n$ applications) on the other, then we have landed somewhere in between: closer to the "real" end on an absolute scale, but closer to the "potential" end on the polynomial versus exponential scale that is more natural for computer science.

Of course, we should be wary of drawing grand conclusions from a single data point. So in this thesis, I will imagine a hypothetical resident of Conway's Game of Life, who arrives in our physical universe on a computational complexity safari—wanting to know exactly which intuitions to keep and which to discard regarding the limits of efficient computation. Many popular science writers would tell our visitor to throw *all* classical intuitions out the window, while quantum computing skeptics would urge retaining them all. These positions are actually two sides of the same coin, since the belief that a quantum computer would necessitate the first is what generally leads to the second. I will show, however, that neither position is justified. Based on what we know today, there really is a Beast, but it usually conceals its exponential underbelly.

I'll provide only one example from the thesis here; the rest are summarized in Chapter 2. Suppose we are given a procedure that computes a two-to-one function f, and want to find distinct inputs x and y such that $f(x) = f(y)$. In this case, by simply preparing a uniform superposition over all inputs to f, applying the procedure, and then measuring its result, we can produce a state of the form $(|x\rangle + |y\rangle)/\sqrt{2}$, for some x and y such that $f(x) = f(y)$. The only problem is that if we measure this state, then we see either x or y, but not both. The task, in other words, is no longer to find a needle in a haystack, but just to find two needles in an otherwise empty barn! Nevertheless, the *collision lower bound* in Chapter 6 will show that, if there are 2^n inputs to f, then any quantum algorithm for this problem must apply the procedure for f at least $\sim 2^{n/5}$ times. Omitting technical details, this lower bound can be interpreted in at least seven ways:

(1) Quantum computers need exponential time even to compute certain *global* properties of a function, not just local properties such as whether there is an x with $f(x) = 1$.

(2) Simon's algorithm [220], and the period-finding core of Shor's algorithm [219], cannot be generalized to functions with no periodicity or other special structure.

(3) Any "brute-force" quantum algorithm needs exponential time, not just for NP-complete problems, but for many structured problems such as Graph Isomorphism, approximating the shortest vector in a lattice, and finding collisions in cryptographic hash functions.

(4) It is unlikely that all problems having "statistical zero-knowledge proofs" can be efficiently solved on a quantum computer.

(5) Within the setting of a collision algorithm, the components $|x\rangle$ and $|y\rangle$ in the state $(|x\rangle + |y\rangle)/\sqrt{2}$ should be thought of as more "potentially" than "actually" there, it being impossible to extract information about both of them in a reasonable amount of time.

(6) The ability to map $|x\rangle$ to $|f(x)\rangle$, "uncomputing" x in the process, can be exponentially more powerful than the ability to map $|x\rangle$ to $|x\rangle|f(x)\rangle$.

(7) In hidden-variable interpretations of quantum mechanics, the ability to sample the entire history of a hidden variable would yield even more power than standard quantum computing.

Interpretations (5), (6), and (7) are examples of what I mean by putting old mysteries in a new light. We are not brought face-to-face with the Beast, but at least we have fresh footprints and droppings.

Well then. *Am* I worried that quantum computing won't pan out? My usual answer is that I'd be *thrilled* to know it will *never* pan out, since this would entail the discovery of a lifetime, that quantum mechanics is false. But this is not what the questioner has in mind. What if quantum mechanics holds up, but building a useful quantum computer turns out to be so difficult and expensive that the world ends before anyone succeeds? The questioner is usually a classical theoretical computer scientist, someone who is not known to worry excessively that the world will end before $\log \log n$ exceeds 10. Still, it would be nice to see nontrivial quantum computers in my lifetime, and while I'm cautiously optimistic, I'll admit to being *slightly* worried that I won't. But when faced with the evidence that one was born into a universe profoundly unlike Conway's—indeed, that one is living one's life on the back of a mysterious, exponential Beast comprising everything that ever could have happened—what is one to do? "Move right along... nothing to see here..."

Chapter 2

Overview

> "Let a computer smear—with the right kind of quantum randomness—and you create, in effect, a 'parallel' machine with an astronomical number of processors ... All you have to do is be sure that when you collapse the system, you choose the version that happened to find the needle in the mathematical haystack."
>
> —From *Quarantine* [103], a 1992 science-fiction novel by Greg Egan

Many of the deepest discoveries of science are *limitations*: for example, no superluminal signalling, no perpetual-motion machines, and no complete axiomatization for arithmetic. This thesis is broadly concerned with limitations on what can efficiently be computed in the physical world. The word "quantum" is absent from the title, in order to emphasize that the focus on quantum computing is not an arbitrary choice, but rather an inevitable result of taking our current physical theories seriously. The technical contributions of the thesis are divided into two parts, according to whether they accept the quantum computing model as given and study its fundamental limitations; or question, defend, or go beyond that model in some way. Before launching into a detailed overview of the contributions, let me make some preliminary remarks.

Since the early twentieth century, two communities—physicists[1] and computer scientists—have been asking some of the deepest questions ever asked in almost total intellectual isolation from each other. The great joy of quantum computing research is that it brings these communities together. The trouble was initially that, although each community would nod politely during the other's talks, eventually it would come out that the physicists thought NP stood for "Non Polynomial," and the computer scientists had no idea what a Hamiltonian was. Thankfully, the situation has improved a lot—but my hope is that it improves further still, to the point where computer scientists have internalized the problems faced by physics and vice versa. For this reason, I have worked hard to make the thesis as accessible as possible to both communities. Thus, Chapter 3 provides a "complexity theory cheat sheet" that defines NP, P/poly, AM, and other computational complexity classes that appear in the thesis; and that explains oracles and other important

[1] As in Saul Steinberg's famous *New Yorker* world map, in which 9^{th} Avenue and the Hudson River take up more space than Japan and China, from my perspective chemists, engineers, and even mathematicians who know what a gauge field is are all "physicists."

concepts. Then Chapter 4 presents the quantum model of computation with no reference to the underlying physics, before moving on to fancier notions such as density matrices, trace distance, and separability. Neither chapter is a rigorous introduction to its subject; for that there are fine textbooks—such as Papadimitriou's *Computational Complexity* [188] and Nielsen and Chuang's *Quantum Computation and Quantum Information* [182]—as well as course lecture notes available on the web. Depending on your background, you might want to skip to Chapters 3 or 4 before continuing any further, or you might want to skip past these chapters entirely.

Even the most irredeemably classical reader should take heart: of the 103 proofs in the thesis, 66 do not contain a single ket symbol.[2] Many of the proofs can be understood by simply accepting certain facts about quantum computing on faith, such as Ambainis's[3] adversary theorem [27] or Beals et al.'s polynomial lemma [45]. On the other hand, one does run the risk that after one understands the proofs, ket symbols will seem less frightening than before.

The results in the thesis have all previously appeared in published papers or preprints [1, 2, 4, 5, 7, 8, 9, 10, 11, 13], with the exception of the quantum computing based proof that PP is closed under intersection in Chapter 15. I thank Andris Ambainis for allowing me to include our joint results from [13] on quantum search of spatial regions. Results of mine that do *not* appear in the thesis include those on Boolean function query properties [3], stabilizer circuits [14] (joint work with Daniel Gottesman), and agreement complexity [6].

In writing the thesis, one of the toughest choices I faced was whether to refer to myself as 'I' or 'we.' Sometimes a personal voice seemed more appropriate, and sometimes the Voice of Scientific Truth, but I wanted to be consistent. Readers can decide whether I chose humbly or arrogantly.

2.1 Limitations of Quantum Computers

Part I studies the fundamental limitations of quantum computers within the usual model for them. With the exception of Chapter 10 on quantum advice, the contributions of Part I all deal with *black-box* or *query* complexity, meaning that one counts only the number of queries to an "oracle," not the number of computational steps. Of course, the queries can be made in quantum superposition. In Chapter 5, I explain the quantum black-box model, then offer a detailed justification for its relevance to understanding the limits of quantum computers. Some computer scientists say that black-box results should not be taken too seriously; but I argue that, within quantum computing, they are not taken seriously enough.

What follows is a (relatively) nontechnical overview of Chapters 6 to 10, which contain the results of Part I. Afterwards, Chapter 11 summarizes the conceptual lessons that I believe can be drawn from those results.

[2] To be honest, a few of those do contain density matrices—or the *theorem* contains ket symbols, but not the *proof*.

[3] Style manuals disagree about whether *Ambainis'* or *Ambainis's* is preferable, but one referee asked me to follow the latter rule with the following deadpan remark: "Exceptions to the rule generally involve religiously significant individuals, e.g., 'Jesus' lower-bound method.' "

2.1.1 The Collision Problem

Chapter 6 presents my lower bound on the quantum query complexity of the collision problem. Given a function X from $\{1,\ldots,n\}$ to $\{1,\ldots,n\}$ (where n is even), the collision problem is to decide whether X is one-to-one or two-to-one, promised that one of these is the case. Here the only way to learn about X is to call a procedure that computes $X(i)$ given i. Clearly, any deterministic classical algorithm needs to call the procedure $n/2+1$ times to solve the problem. On the other hand, a randomized algorithm can exploit the "birthday paradox": only 23 people have to enter a room before there's a 50% chance that two of them share the same birthday, since what matters is the number of *pairs* of people. Similarly, if X is two-to-one, and an algorithm queries X at $\sqrt{n}$ uniform random locations, then with constant probability it will find two locations $i \neq j$ such that $X(i) = X(j)$, thereby establishing that X is two-to-one. This bound is easily seen to be tight, meaning that the bounded-error randomized query complexity of the collision problem is $\Theta(\sqrt{n})$.

What about the *quantum* complexity? In 1997, Brassard, Høyer, and Tapp [68] gave a quantum algorithm that uses only $O\left(n^{1/3}\right)$ queries. The algorithm is simple to describe: in the first phase, query X classically at $n^{1/3}$ randomly chosen locations. In the second phase, choose $n^{2/3}$ random locations, and run Grover's algorithm on those locations, considering each location i as "marked" if $X(i) = X(j)$ for some j that was queried in the first phase. Notice that both phases use order $n^{1/3} = \sqrt{n^{2/3}}$ queries, and that the total number of comparisons is $n^{2/3}n^{1/3} = n$. So, like its randomized counterpart, the quantum algorithm finds a collision with constant probability if X is two-to-one.

What I show in Chapter 6 is that *any* quantum algorithm for the collision problem needs $\Omega\left(n^{1/5}\right)$ queries. Previously, no lower bound better than the trivial $\Omega(1)$ was known. I also show a lower bound of $\Omega\left(n^{1/7}\right)$ for the following *set comparison problem*: given oracle access to injective functions $X : \{1,\ldots,n\} \rightarrow \{1,\ldots,2n\}$ and $Y : \{1,\ldots,n\} \rightarrow \{1,\ldots,2n\}$, decide whether

$$\{X(1),\ldots,X(n),Y(1),\ldots,Y(n)\}$$

has at least $1.1n$ elements or exactly n elements, promised that one of these is the case. The set comparison problem is similar to the collision problem, except that it lacks permutation symmetry, making it harder to prove a lower bound. My results for these problems have been improved, simplified, and generalized by Shi [218], Kutin [161], Ambainis [27], and Midrijanis [176].

The implications of these results were already discussed in Chapter 1: for example, they demonstrate that a "brute-force" approach will never yield efficient quantum algorithms for the Graph Isomorphism, Approximate Shortest Vector, or Nonabelian Hidden Subgroup problems; suggest that there could be cryptographic hash functions secure against quantum attack; and imply that there exists an oracle relative to which $\mathsf{SZK} \not\subset \mathsf{BQP}$, where SZK is the class of problems having statistical zero-knowledge proof protocols, and BQP is quantum polynomial time.

Both the original lower bounds and the subsequent improvements are based on the *polynomial method*, which was introduced by Nisan and Szegedy [184], and first used to prove quantum lower bounds by Beals, Buhrman, Cleve, Mosca, and de Wolf [45]. In that method, given a quantum algorithm that makes T queries to an oracle X, we first represent

the algorithm's acceptance probability by a multilinear polynomial $p(X)$ of degree at most $2T$. We then use results from a well-developed area of mathematics called *approximation theory* to show a lower bound on the degree of p. This in turn implies a lower bound on T.

In order to apply the polynomial method to the collision problem, first I extend the collision problem's domain from one-to-one and two-to-one functions to g-to-one functions for larger values of g. Next I replace the multivariate polynomial $p(X)$ by a related *univariate* polynomial $q(g)$ whose degree is easier to lower-bound. The latter step is the real "magic" of the proof; I still have no good intuitive explanation for why it works.

The polynomial method is one of two principal methods that we have for proving lower bounds on quantum query complexity. The other is Ambainis's *quantum adversary method* [27], which can be seen as a far-reaching generalization of the "hybrid argument" that Bennett, Bernstein, Brassard, and Vazirani [51] introduced in 1994 to show that a quantum computer needs $\Omega(\sqrt{n})$ queries to search an unordered database of size n for a marked item. In the adversary method, we consider a bipartite quantum state, in which one part consists of a superposition over possible inputs, and the other part consists of a quantum algorithm's work space. We then upper-bound how much the *entanglement* between the two parts can increase as the result of a single query. This in turn implies a lower bound on the number of queries, since the two parts must be highly entangled by the end. The adversary method is more intrinsically "quantum" than the polynomial method; and as Ambainis [27] showed, it is also applicable to a wider range of problems, including those (such as game-tree search) that lack permutation symmetry. Ambainis even gave problems for which the adversary method *provably* yields a better lower bound than the polynomial method [28]. It is ironic, then, that Ambainis's original goal in developing the adversary method was to prove a lower bound for the collision problem; and in this one instance, the polynomial method succeeded while the adversary method failed.

2.1.2 Local Search

In Chapters 7, 8, and 9, however, the adversary method gets its revenge. Chapter 7 deals with the *local search* problem: given an undirected graph $G = (V, E)$ and a black-box function $f : V \rightarrow \mathbb{Z}$, find a local minimum of f—that is, a vertex v such that $f(v) \leq f(w)$ for all neighbors w of v. The graph G is known in advance, so the complexity measure is just the number of queries to f. This problem is central for understanding the performance of the quantum adiabatic algorithm, as well as classical algorithms such as simulated annealing. If G is the Boolean hypercube $\{0,1\}^n$, then previously Llewellyn, Tovey, and Trick [169] had shown that any deterministic algorithm needs $\Omega(2^n/\sqrt{n})$ queries to find a local minimum; and Aldous [24] had shown that any randomized algorithm needs $2^{n/2-o(n)}$ queries. What I show is that any quantum algorithm needs $\Omega\left(2^{n/4}/n\right)$ queries. This is the first nontrivial quantum lower bound for any local search problem; and it implies that the complexity class PLS (or "Polynomial Local Search"), defined by Johnson, Papadimitriou, and Yannakakis [149], is not in quantum polynomial time relative to an oracle.

What will be more surprising to classical computer scientists is that my proof technique, based on the quantum adversary method, also yields new *classical* lower bounds for local search. In particular, I prove a classical analogue of Ambainis's quantum adversary theorem, and show that it implies randomized lower bounds up to quadratically better

than the corresponding quantum lower bounds. I then apply my theorem to show that any randomized algorithm needs $\Omega\left(2^{n/2}/n^2\right)$ queries to find a local minimum of a function $f:\{0,1\}^n \rightarrow \mathbb{Z}$. Not only does this improve on Aldous's $2^{n/2-o(n)}$ lower bound, bringing us closer to the known upper bound of $O\left(2^{n/2}\sqrt{n}\right)$; but it does so in a simpler way that does not depend on random walk analysis. In addition, I show the first randomized *or* quantum lower bounds for finding a local minimum on a cube of constant dimension 3 or greater. Along with recent work by Bar-Yossef, Jayram, and Kerenidis [43] and by Aharonov and Regev [22], these results provide one of the earliest examples of how quantum ideas can help to resolve classical open problems. As I will discuss in Chapter 7, my results on local search have subsequently been improved by Santha and Szegedy [211] and by Ambainis [25].

2.1.3 Quantum Certificate Complexity

Chapters 8 and 9 continue to explore the power of Ambainis's lower bound method and the limitations of quantum computers. Chapter 8 is inspired by the following theorem of Beals et al. [45]: if $f:\{0,1\}^n \rightarrow \{0,1\}$ is a total Boolean function, then $\mathrm{D}(f) = O\left(\mathrm{Q}_2(f)^6\right)$, where $\mathrm{D}(f)$ is the deterministic classical query complexity of f, and $\mathrm{Q}_2(f)$ is the bounded-error quantum query complexity.[4] This theorem is noteworthy for two reasons: first, because it gives a case where quantum computers provide only a polynomial speedup, in contrast to the exponential speedup of Shor's algorithm; and second, because the exponent of 6 seems so arbitrary. The largest separation we know of is quadratic, and is achieved by the OR function on n bits: $\mathrm{D}(\mathrm{OR}) = n$, but $\mathrm{Q}_2(\mathrm{OR}) = O(\sqrt{n})$ because of Grover's search algorithm. It is a longstanding open question whether this separation is optimal. In Chapter 8, I make the best progress so far toward showing that it is. In particular I prove that

$$\mathrm{R}_2(f) = O\left(\mathrm{Q}_2(f)^2 \mathrm{Q}_0(f) \log n\right)$$

for all total Boolean functions $f:\{0,1\}^n \rightarrow \{0,1\}$. Here $\mathrm{R}_2(f)$ is the bounded-error randomized query complexity of f, and $\mathrm{Q}_0(f)$ is the zero-error quantum query complexity. To prove this result, I introduce two new query complexity measures of independent interest: the *randomized certificate complexity* $\mathrm{RC}(f)$ and the *quantum certificate complexity* $\mathrm{QC}(f)$. Using Ambainis's adversary method together with the minimax theorem, I relate these measures *exactly* to one another, showing that $\mathrm{RC}(f) = \Theta\left(\mathrm{QC}(f)^2\right)$. Then, using the polynomial method, I show that $\mathrm{R}_2(f) = O(\mathrm{RC}(f)\,\mathrm{Q}_0(f)\log n)$ for all total Boolean f, which implies the above result since $\mathrm{QC}(f) \leq \mathrm{Q}_2(f)$. Chapter 8 contains several other results of interest to researchers studying query complexity, such as a superquadratic gap between $\mathrm{QC}(f)$ and the "ordinary" certificate complexity $\mathrm{C}(f)$. But the main message is the unexpected versatility of our quantum lower bound methods: we see the first use of the adversary method to prove something about *all* total functions, not just a specific function; the first use of both the adversary and the polynomial methods at different points in a proof; and the first combination of the adversary method with a linear programming duality argument.

[4] The subscript '2' means that the error is two-sided.

2.1.4 The Need to Uncompute

Next, Chapter 9 illustrates how "the need to uncompute" imposes a fundamental limit on efficient quantum computation. Like a classical algorithm, a quantum algorithm can solve a problem recursively by calling itself as a subroutine. When this is done, though, the quantum algorithm typically needs to call itself *twice* for each subproblem to be solved. The second call's purpose is to "uncompute" garbage left over by the first call, and thereby enable interference between different branches of the computation. In a seminal paper, Bennett [52] argued[5] that uncomputation increases an algorithm's running time by only a factor of 2. Yet in the recursive setting, the increase is by a factor of 2^d, where d is the depth of recursion. Is there any way to avoid this exponential blowup?

To make the question more concrete, Chapter 9 focuses on the recursive Fourier sampling problem of Bernstein and Vazirani [55]. This is a problem that involves d levels of recursion, and that takes a Boolean function g as a parameter. What Bernstein and Vazirani showed is that for some choices of g, any classical randomized algorithm needs $n^{\Omega(d)}$ queries to solve the problem. By contrast, 2^d queries always suffice for a quantum algorithm. The question I ask is whether a quantum algorithm could get by with *fewer* than $2^{\Omega(d)}$ queries, even while the classical complexity remains large. I show that the answer is no: for every g, either Ambainis's adversary method yields a $2^{\Omega(d)}$ lower bound on the quantum query complexity, or else the classical and quantum query complexities are both 1. The lower bound proof introduces a new parameter of Boolean functions called the "nonparity coefficient," which might be of independent interest.

2.1.5 Limitations of Quantum Advice

Chapter 10 broadens the scope of Part I, to include the limitations of quantum computers equipped with "quantum advice states." Ordinarily, we assume that a quantum computer starts out in the standard "all-0" state, $|0\cdots0\rangle$. But it is perfectly sensible to drop that assumption, and consider the effects of other initial states. Most of the work doing so has concentrated on whether universal quantum computing is still possible with highly mixed initial states (see [34, 214] for example). But an equally interesting question is whether there are states that could take exponential time to prepare, but that would carry us far beyond the complexity-theoretic confines of BQP were they given to us by a wizard. For even if quantum mechanics is universally valid, we do not *really* know whether such states exist in Nature!

Let BQP/qpoly be the class of problems solvable in quantum polynomial time, with the help of a polynomial-size "quantum advice state" $|\psi_n\rangle$ that depends only on the input length n but that can otherwise be arbitrary. Then the question is whether BQP/poly = BQP/qpoly, where BQP/poly is the class of the problems solvable in quantum polynomial time using a polynomial-size *classical* advice string.[6] As usual, we could try to prove an oracle separation. But why can't we show that quantum advice is more powerful than

[5] Bennett's paper dealt with *classical* reversible computation, but this comment applies equally well to quantum computation.

[6] For clearly BQP/poly and BQP/qpoly both contain uncomputable problems not in BQP, such as whether the n^{th} Turing machine halts.

classical advice, with *no* oracle? Also, could quantum advice be used (for example) to solve NP-complete problems in polynomial time?

The results in Chapter 10 place strong limitations on the power of quantum advice. First, I show that BQP/qpoly is contained in a classical complexity class called PP/poly. This means (roughly) that quantum advice can always be replaced by classical advice, provided we're willing to use exponentially more computation time. It also means that we could not prove BQP/poly $\neq$ BQP/qpoly without showing that PP does not have polynomial-size circuits, which is believed to be an extraordinarily hard problem. To prove this result, I imagine that the advice state $|\psi_n\rangle$ is sent to the BQP/qpoly machine by a benevolent "advisor," through a one-way quantum communication channel. I then give a novel protocol for simulating that quantum channel using a classical channel. Besides showing that BQP/qpoly $\subseteq$ PP/poly, the simulation protocol also implies that for all Boolean functions $f : \{0,1\}^n \times \{0,1\}^m \rightarrow \{0,1\}$ (partial or total), we have $\mathrm{D}^1(f) = O\left(m\,\mathrm{Q}_2^1(f)\log \mathrm{Q}_2^1(f)\right)$, where $\mathrm{D}^1(f)$ is the deterministic one-way communication complexity of f, and $\mathrm{Q}_2^1(f)$ is the bounded-error quantum one-way communication complexity. This can be considered a generalization of the "dense quantum coding" lower bound due to Ambainis, Nayak, Ta-Shma, and Vazirani [32].

The second result in Chapter 10 is that there exists an oracle relative to which NP $\not\subset$ BQP/qpoly. This extends the result of Bennett et al. [51] that there exists an oracle relative to which NP $\not\subset$ BQP, to handle quantum advice. Intuitively, even though the quantum state $|\psi_n\rangle$ could in some sense encode the solutions to exponentially many NP search problems, only a miniscule fraction of that information could be extracted by measuring the advice, at least in the black-box setting that we understand today.

The proof of the oracle separation relies on another result of independent interest: a *direct product theorem* for quantum search. This theorem says that given an unordered database with n items, k of which are marked, any quantum algorithm that makes $o(\sqrt{n})$ queries[7] has probability at most $2^{-\Omega(k)}$ of finding all k of the marked items. In other words, there are no "magical" correlations by which success in finding one marked item leads to success in finding the others. This might seem intuitively obvious, but it does not follow from the $\sqrt{n}$ lower bound for Grover search, or any other previous quantum lower bound for that matter. Previously, Klauck [155] had given an incorrect proof of a direct product theorem, based on Bennett et al.'s hybrid method. I give the first correct proof by using the polynomial method, together with an inequality dealing with higher derivatives of polynomials due to V. A. Markov, the younger brother of A. A. Markov.

The third result in Chapter 10 is a new *trace distance method* for proving lower bounds on quantum one-way communication complexity. Using this method, I obtain optimal quantum lower bounds for two problems of Ambainis, for which no nontrivial lower bounds were previously known even for classical randomized protocols.

[7] Subsequently Klauck, Špalek, and de Wolf [156] improved this to $o\left(\sqrt{nk}\right)$ queries, which is tight.

2.2 Models and Reality

This thesis is concerned with the limits of efficient computation in Nature. It is not obvious that these coincide with the limits of the quantum computing model. Thus, Part II studies the relationship of the quantum computing model to physical reality. Of course, this is too grand a topic for any thesis, even a thesis as long as this one. I therefore focus on three questions that particularly interest me. First, how should we understand the arguments of "extreme" skeptics, that quantum computing is impossible not only in practice but also in principle? Second, what are the implications for quantum computing if we recognize that the speed of light is finite, and that according to widely-accepted principles, a bounded region of space can store only a finite amount of information? And third, are there reasonable changes to the quantum computing model that make it even more powerful, and if so, how much more powerful do they make it? Chapters 12 to 16 address these questions from various angles; then Chapter 17 summarizes.

2.2.1 Skepticism of Quantum Computing

Chapter 12 examines the arguments of skeptics who think that large-scale quantum computing is impossible for a fundamental physical reason. I first briefly consider the arguments of Leonid Levin and other computer scientists, that quantum computing is analogous to "extravagant" models of computation such as unit-cost arithmetic, and should be rejected on essentially the same grounds. My response emphasizes the need to grapple with the actual evidence for quantum mechanics, and to propose an alternative picture of the world that is compatible with that evidence but in which quantum computing is impossible. The bulk of the chapter, though, deals with Stephen Wolfram's *A New Kind of Science* [246], and in particular with one of that book's most surprising claims: that a deterministic cellular-automaton picture of the world is compatible with the so-called *Bell inequality violations* demonstrating the effects of quantum entanglement. To achieve compatibility, Wolfram posits "long-range threads" between spacelike-separated points. I explain in detail why this thread proposal violates Wolfram's own desiderata of relativistic and causal invariance. Nothing in Chapter 12 is very original technically, but it seems worthwhile to spell out what a scientific argument against quantum computing would have to accomplish, and why the existing arguments fail.

2.2.2 Complexity Theory of Quantum States

Chapter 13 continues the train of thought begun in Chapter 12, except that now the focus is more technical. I search for a natural *Sure/Shor separator*: a set of quantum states that can account for all experiments performed to date, but that does not contain the states arising in Shor's factoring algorithm. In my view, quantum computing skeptics would strengthen their case by proposing specific examples of Sure/Shor separators, since they could then offer testable hypotheses about *where* the assumptions of the quantum computing model break down (if not *how* they break down). So why am I doing the skeptics' work for them? Several people have wrongly inferred from this that I too am a skeptic! My goal, rather, is to illustrate what a scientific debate about the possibility of quantum computing might

look like.

Most of Chapter 13 deals with a candidate Sure/Shor separator that I call *tree states*. Any n-qubit pure state $|\psi_n\rangle$ can be represented by a tree, in which each leaf is labeled by $|0\rangle$ or $|1\rangle$, and each non-leaf vertex is labeled by either a linear combination or a tensor product of its subtrees. Then the *tree size* of $|\psi_n\rangle$ is just the minimum number of vertices in such a tree, and a "tree state" is an infinite family of states whose tree size is bounded by a polynomial in n. The idea is to keep a central axiom of quantum mechanics—that if $|\psi\rangle$ and $|\varphi\rangle$ are possible states, so are $|\psi\rangle \otimes |\varphi\rangle$ and $\alpha|\psi\rangle + \beta|\varphi\rangle$—but to limit oneself to polynomially many applications of the axiom.

The main results are *superpolynomial lower bounds* on tree size for explicit families of quantum states. Using a recent lower bound on multilinear formula size due to Raz [195, 196], I show that many states arising in quantum error correction (for example, states based on binary linear erasure codes) have tree size $n^{\Omega(\log n)}$. I show the same for the states arising in Shor's algorithm, assuming a number-theoretic conjecture. Therefore, I argue, by demonstrating such states in the lab on a large number of qubits, experimentalists could weaken[8] the hypothesis that all states in Nature are tree states.

Unfortunately, while I conjecture that the actual tree sizes are exponential, Raz's method is currently only able to show lower bounds of the form $n^{\Omega(\log n)}$. On the other hand, I do show exponential lower bounds under a restriction, called "manifest orthogonality," on the allowed linear combinations of states.

More broadly, Chapter 13 develops a complexity classification of quantum states, and—treating that classification as a subject in its own right—proves many basic results about it. To give a few examples: if a quantum computer is restricted to being in a tree state at every time step, then it can be simulated in the third level of polynomial hierarchy PH. A random state cannot even be approximated by a state with subexponential tree size. Any "orthogonal tree state" can be prepared by a polynomial-size quantum circuit. Collapses of quantum state classes would imply collapses of ordinary complexity classes, and vice versa. Many of these results involve unexpected connections between quantum computing and classical circuit complexity. For this reason, I think that the "complexity theory of quantum states" has an intrinsic computer-science motivation, besides its possible role in making debates about quantum mechanics' range of validity less philosophical and more scientific.

2.2.3 Quantum Search of Spatial Regions

A basic result in classical computer science says that Turing machines are polynomially equivalent to random-access machines. In other words, we can ignore the fact that the speed of light is finite for complexity purposes, so long as we only care about polynomial equivalence. It is easy to see that the same is true for quantum computing. Yet one of the two main quantum algorithms, Grover's algorithm, provides only a polynomial speedup.[9] So, does this speedup disappear if we consider relativity as well as quantum mechanics?

[8] Since tree size is an asymptotic notion (and for other reasons discussed in Chapter 13), strictly speaking experimentalists could never refute the hypothesis—just push it beyond all bounds of plausibility.

[9] If Grover's algorithm is applied to a combinatorial search space of size 2^n, then the speedup is by a factor of $2^{n/2}$—but in this case the speedup is only conjectured, not proven.

More concretely, suppose a "quantum robot" is searching a 2-D grid of size $\sqrt{n} \times \sqrt{n}$ for a single marked item. The robot can enter a superposition of grid locations, but moving from one location to an adjacent one takes one time step. How many steps are needed to find the marked item? If Grover's algorithm is implemented naïvely, the answer is order n—since each of the $\sqrt{n}$ Grover iterations takes $\sqrt{n}$ steps, just to move the robot across the grid and back. This yields no improvement over classical search. Benioff [50] noticed this defect of Grover's algorithm as applied to a physical database, but failed to raise the question of whether or not a faster algorithm exists.

Sadly, I was unable to prove a lower bound showing that the naïve algorithm is optimal. But in joint work with Andris Ambainis, we did the next best thing: we proved the *impossibility* of proving a lower bound, or to put it crudely, gave an algorithm. In particular, Chapter 14 shows how to search a $\sqrt{n} \times \sqrt{n}$ grid for a unique marked vertex in only $O\left(\sqrt{n} \log^{3/2} n\right)$ steps, by using a carefully-optimized recursive Grover search. It also shows how to search a d-dimensional hypercube in $O\left(\sqrt{n}\right)$ steps for $d \geq 3$. The latter result has an unexpected implication: namely, that the quantum communication complexity of the disjointness function is $O\left(\sqrt{n}\right)$. This matches a lower bound of Razborov [199], and improves previous upper bounds due to Buhrman, Cleve, and Wigderson [76] and Høyer and de Wolf [146].

Chapter 14 also generalizes our search algorithm to handle multiple marked items, as well as graphs that are not hypercubes but have sufficiently good expansion properties. More broadly, the chapter develops a new model of *quantum query complexity on graphs*, and proves basic facts about that model, such as lower bounds for search on "starfish" graphs. Of particular interest to physicists will be Section 14.3, which relates our results to fundamental limits on information processing imposed by the holographic principle. For example, we can give an approximate answer to the following question: assuming a positive cosmological constant $\Lambda > 0$, and assuming the only constraints (besides quantum mechanics) are the speed of light and the holographic principle, how large a database could ever be searched for a specific entry, before most of the database receded past one's cosmological horizon?

2.2.4 Quantum Computing and Postselection

There is at least one foolproof way to solve NP-complete problems in polynomial time: guess a random solution, then kill yourself if the solution is incorrect. Conditioned on looking at anything at all, you will be looking at a correct solution! It's a wonder that this approach is not tried more often.

The general idea, of throwing out all runs of a computation except those that yield a particular result, is called *postselection*. Chapter 15 explores the general power of postselection when combined with quantum computing. I define a new complexity class called PostBQP: the class of problems solvable in polynomial time on a quantum computer, given the ability to measure a qubit and *assume* the outcome will be $|1\rangle$ (or equivalently, discard all runs in which the outcome is $|0\rangle$). I then show that PostBQP coincides with the classical complexity class PP.

Surprisingly, this new characterization of PP yields an extremely simple, quantum

computing based proof that PP is closed under intersection. This had been an open problem for two decades, and the previous proof, due to Beigel, Reingold, and Spielman [47], used highly nontrivial ideas about rational approximations of the sign function. I also reestablish an extension of the Beigel-Reingold-Spielman result due to Fortnow and Reingold [115], that PP is closed under polynomial-time truth-table reductions. Indeed, I show that PP is closed under BQP truth-table reductions, which seems to be a new result.

The rest of Chapter 15 studies the computational effects of simple changes to the axioms of quantum mechanics. In particular, what if we allow linear but nonunitary transformations, or change the measurement probabilities from $|\alpha|^2$ to $|\alpha|^p$ (suitably normalized) for some $p \neq 2$? I show that the first change would yield exactly the power of PostBQP, and therefore of PP; while the second change would yield PP if $p \in \{4, 6, 8, \ldots\}$, and some class between PP and PSPACE otherwise.

My results complement those of Abrams and Lloyd [15], who showed that nonlinear quantum mechanics would let us solve NP- and even #P-complete problems in polynomial time; and Brun [72] and Bacon [40], who showed the same for quantum computers involving closed timelike curves. Taken together, these results lend credence to an observation of Weinberg [241]: that quantum mechanics is a "brittle" theory, in the sense that even a tiny change to it would have dramatic consequences.

2.2.5 The Power of History

Contrary to widespread belief, what makes quantum mechanics so hard to swallow is *not* indeterminism about the future trajectory of a particle. That is no more bizarre than a coin flip in a randomized algorithm. The difficulty is that quantum mechanics also seems to require indeterminism about a particle's *past* trajectory. Or rather, the very notion of a "trajectory" is undefined—for until the particle is measured, there is just an evolving wavefunction.

In spite of this, Schrödinger [213], Bohm [59], Bell [49], and others proposed *hidden-variable theories*, in which a quantum state is supplemented by "actual" values of certain observables. These actual values evolve in time by a dynamical rule, in such a way that the predictions of quantum mechanics are recovered at any individual time. On the other hand, it now makes sense to ask questions like the following: "Given that a particle was at location x_1 at time t_1 (even though it was not measured at t_1), what is the probability of it being at location x_2 at time t_2?" The answers to such questions yield a probability distribution over possible trajectories.

Chapter 16 initiates the study of hidden variables from the discrete, abstract perspective of quantum computing. For me, a hidden-variable theory is simply a way to convert a unitary matrix that maps one quantum state to another, into a stochastic matrix that maps the initial probability distribution to the final one in some fixed basis. I list five axioms that we might want such a theory to satisfy, and investigate previous hidden-variable theories of Dieks [97] and Schrödinger [213] in terms of these axioms. I also propose a new hidden-variable theory based on *network flows*, which are classic objects of study in computer science, and prove that this theory satisfies two axioms called "indifference" and "robustness." *A priori*, it was not at all obvious that these two key axioms could be satisfied simultaneously.

Next I turn to a new question: the computational complexity of simulating hidden-variable theories. I show that, if we could examine the entire history of a hidden variable, then we could efficiently solve problems that are believed to be intractable even for quantum computers. In particular, under any hidden-variable theory satisfying the indifference axiom, we could solve the Graph Isomorphism and Approximate Shortest Vector problems in polynomial time, and indeed could simulate the entire class SZK (Statistical Zero Knowledge). Combining this result with the collision lower bound of Chapter 6, we get an oracle relative to which BQP is strictly contained in DQP, where DQP (Dynamical Quantum Polynomial-Time) is the class of problems efficiently solvable by sampling histories.

Using the histories model, I also show that one could search an N-item database using $O\left(N^{1/3}\right)$ queries, as opposed to $O\left(\sqrt{N}\right)$ with Grover's algorithm. On the other hand, the $N^{1/3}$ bound is tight, meaning that one could probably *not* solve NP-complete problems in polynomial time. We thus obtain the first good example of a model of computation that appears *slightly* more powerful than the quantum computing model.

In summary, Chapter 16 ties together many of the themes of this thesis: the black-box limitations of quantum computers; the application of nontrivial computer science techniques; the obsession with the computational resources needed to simulate our universe; and finally, the use of quantum computing to shine light on the mysteries of quantum mechanics itself.

Chapter 3

Complexity Theory Cheat Sheet

"If pigs can whistle, then donkeys can fly."

(Summary of complexity theory, attributed to Richard Karp)

To most people who are not theoretical computer scientists, the theory of computational complexity—one of the great intellectual achievements of the twentieth century—is simply a meaningless jumble of capital letters. The goal of this chapter is to turn it into a meaningful jumble.

In computer science, a *problem* is ordinarily an infinite set of yes-or-no questions: for example, "Given a graph, is it connected?" Each particular graph is an *instance* of the general problem. An algorithm for the problem is *polynomial-time* if, given any instance as input, it outputs the correct answer after at most kn^c steps, where k and c are constants, and n is the *length* of the instance, or the number of bits needed to specify it. For example, in the case of a directed graph, n is just the number of vertices squared. Then P is the class of all problems for which there exists a deterministic classical polynomial-time algorithm. Examples of problems in P include graph connectivity, and (as was discovered two years ago [17]) deciding whether a positive integer written in binary is prime or composite.

Now, NP (Nondeterministic Polynomial-Time) is the class of problems for which, if the answer to a given instance is 'yes', then an omniscient wizard could provide a polynomial-size *proof* of that fact, which would enable us to verify it in deterministic polynomial time. As an example, consider the Satisfiability problem: "given a formula involving the Boolean variables $x_1, \ldots, x_n$ and the logical connectives $\wedge, \vee, \neg$ (and, or, not), is there an assignment to the variables that makes the formula true?" If there is such an assignment, then a short, easily-verified proof is just the assignment itself. On the other hand, it might be extremely difficult to *find* a satisfying assignment without the wizard's help—or for that matter, to verify the *absence* of a satisfying assignment, even given a purported proof of its absence from the wizard. The question of whether there exist polynomial-size proofs of *un*satisfiability that can be verified in polynomial time is called the NP versus coNP question. Here coNP is the class containing the *complement* of every NP problem—for example, "given a Boolean formula, is it *not* satisfiable?"

The Satisfiability problem turns out to be NP-complete, which means it is among the "hardest" problems in NP: any instance of any NP problem can be efficiently converted

into an instance of Satisfiability. The central question, of course, is whether NP-complete problems are solvable in polynomial time, or equivalently whether P = NP (it being clear that P ⊆ NP). By definition, if any NP-complete problem is solvable in polynomial time, then all of them are. One thing we know is that if P ≠ NP, as is almost universally assumed, then there are problems in NP that are neither in P nor NP-complete [162]. Candidates for such "intermediate" problems include deciding whether or not two graphs are isomorphic, and integer factoring (e.g. given integers N, M written in binary, does N have a prime factor greater than M?). The NP-intermediate problems have been a major focus of quantum algorithms research.

3.1 The Complexity Zoo Junior

I now present a glossary of 12 complexity classes besides P and NP that appear in this thesis; non-complexity-theorist readers might wish to refer back to it as needed. The known relationships among these classes are diagrammed in Figure 3.1. These classes represent a tiny sample of the more than 400 classes described on my Complexity Zoo web page (www.complexityzoo.com).

PSPACE **(Polynomial Space)** is the class of problems solvable by a deterministic classical algorithm that uses a polynomially-bounded amount of memory. Thus NP ⊆ PSPACE, since a PSPACE machine can loop through all possible proofs.

EXP **(Exponential-Time)** is the class of problems solvable by a deterministic classical algorithm that uses at most $2^{q(n)}$ time steps, for some polynomial q. Thus PSPACE ⊆ EXP.

BPP **(Bounded-Error Probabilistic Polynomial-Time)** is the class of problems solvable by a *probabilistic* classical polynomial-time algorithm, which given any instance, must output the correct answer for that instance with probability at least 2/3. Thus P ⊆ BPP ⊆ PSPACE. It is widely conjectured that BPP = P [147], but not even known that BPP ⊆ NP.

PP **(Probabilistic Polynomial-Time)** is the class of problems solvable by a probabilistic classical polynomial-time algorithm, which given any instance, need only output the correct answer for that instance with probability greater than 1/2. The following problem is PP-complete: given a Boolean formula φ, decide whether at least half of the possible truth assignments satisfy φ. We have NP ⊆ PP ⊆ PSPACE and also BPP ⊆ PP.

$\mathsf{P}^{\#\mathsf{P}}$ **(pronounced "P to the sharp-P")** is the class of problems solvable by a P machine that can access a "counting oracle." Given a Boolean formula φ, this oracle returns the number of truth assignments that satisfy φ. We have PP ⊆ $\mathsf{P}^{\#\mathsf{P}}$ ⊆ PSPACE.

BQP **(Bounded-Error Quantum Polynomial-Time)** is the class of problems solvable by a quantum polynomial-time algorithm, which given any instance, must output the correct answer for that instance with probability at least 2/3. More information is in Chapter 4. We have BPP ⊆ BQP ⊆ PP [55, 16].

EQP **(Exact Quantum Polynomial-Time)** is similar to BQP, except that the probability of correctness must be 1 instead of 2/3. This class is extremely artificial; it is not even clear how to define it independently of the choice of gate set. But for any reasonable choice, P ⊆ EQP ⊆ BQP.

$\mathsf{P/poly}$ **(P with polynomial-size advice)** is the class of problems solvable by a P algorithm that, along with a problem instance of length n, is also given an "advice string" z_n of length bounded by a polynomial in n. The only constraint is that z_n can depend only on n, and not on any other information about the instance. Otherwise the z_n's can be chosen arbitrarily to help the algorithm. It is not hard to show that $\mathsf{BPP} \subseteq \mathsf{P/poly}$. Since the z_n's can encode noncomputable problems (for example, does the n^{th} Turing machine halt?), $\mathsf{P/poly}$ is not contained in any uniform complexity class, where "uniform" means that the same information is available to an algorithm regardless of n. We can also add polynomial-size advice to other complexity classes, obtaining $\mathsf{EXP/poly}$, $\mathsf{PP/poly}$, and so on.

PH **(Polynomial-Time Hierarchy)** is the union of NP, $\mathsf{NP}^{\mathsf{NP}}$, $\mathsf{NP}^{\mathsf{NP}^{\mathsf{NP}}}$, etc. Equivalently, PH is the class of problems that are polynomial-time reducible to the following form: for all truth assignments x, does there exist an assignment y such that for all assignments $z, \ldots, \varphi(x, y, z, \ldots)$ is satisfied, where φ is a Boolean formula? Here the number of alternations between "for all" and "there exists" quantifiers is a constant independent of n. Sipser [222] and Lautemann [163] showed that $\mathsf{BPP} \subseteq \mathsf{PH}$, while Toda [228] showed that $\mathsf{PH} \subseteq \mathsf{P}^{\#\mathsf{P}}$.

MA **(Merlin Arthur)** is the class of problems for which, if the answer to a given instance is 'yes,' then an omniscient wizard could provide a polynomial-size proof of that fact, which would enable us to verify it in BPP (classical probabilistic polynomial-time, with probability at most 1/3 of accepting an invalid proof or rejecting a valid one). We have $\mathsf{NP} \subseteq \mathsf{MA} \subseteq \mathsf{PP}$.

AM **(Arthur Merlin)** is the class of problems for which, if the answer to a given instance is 'yes,' then a BPP algorithm could become convinced of that fact after a constant number of rounds of interaction with an omniscient wizard. We have $\mathsf{MA} \subseteq \mathsf{AM} \subseteq \mathsf{PH}$. There is evidence that $\mathsf{AM} = \mathsf{MA} = \mathsf{NP}$ [157].

SZK **(Statistical Zero Knowledge)** is the class of problems that possess "statistical zero-knowledge proof protocols." We have $\mathsf{BPP} \subseteq \mathsf{SZK} \subseteq \mathsf{AM}$. Although SZK contains nontrivial problems such as graph isomorphism [130], there are strong indications that it does *not* contain all of NP [63].

Other complexity classes, such as PLS, TFNP, $\mathsf{BQP/qpoly}$, and $\mathsf{BPP_{path}}$, will be introduced throughout the thesis as they are needed.

3.2 Notation

In computer science, the following symbols are used to describe asymptotic growth rates:

- $F(n) = O(G(n))$ means that $F(n)$ is at most order $G(n)$; that is, $F(n) \leq a + bG(n)$ for all $n \geq 0$ and some nonnegative constants a, b.
- $F(n) = \Omega(G(n))$ means that $F(n)$ is at least order $G(n)$; that is, $G(n) = O(F(n))$.
- $F(n) = \Theta(G(n))$ means that $F(n)$ is exactly order $G(n)$; that is, $F(n) = O(G(n))$ and $F(n) = \Omega(G(n))$.

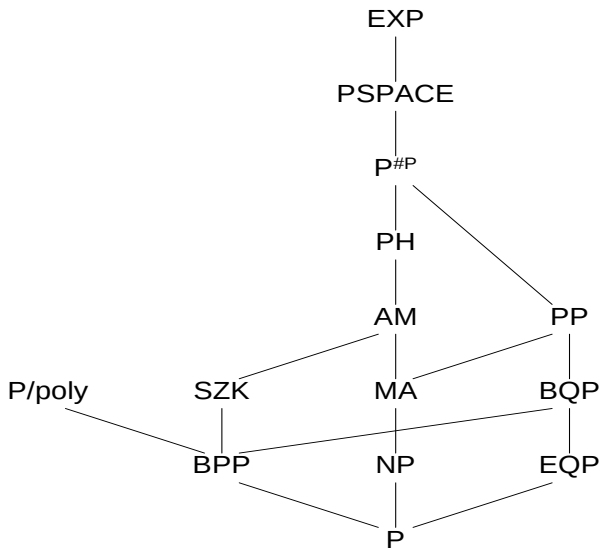

Figure 3.1: Known relations among 14 complexity classes.

- $F(n) = o(G(n))$ means that $F(n)$ is less than order $G(n)$; that is, $F(n) = O(G(n))$ but not $F(n) = \Omega(G(n))$.

The set of all n-bit strings is denoted $\{0,1\}^n$. The set of all binary strings, $\bigcup_{n \geq 0} \{0,1\}^n$, is denoted $\{0,1\}^*$.

3.3 Oracles

One complexity-theoretic concept that will be needed again and again in this thesis is that of an *oracle*. An oracle is a subroutine available to an algorithm, that is guaranteed to compute some function even if we have no idea how. Oracles are denoted using superscripts. For example, P^{NP} is the class of problems solvable by a P algorithm that, given any instance of an NP-complete problem such as Satisfiability, can instantly find the solution for that instance by calling the NP oracle. The algorithm can make multiple calls to the oracle, and these calls can be *adaptive* (that is, can depend on the outcomes of previous calls). If a quantum algorithm makes oracle calls, then unless otherwise specified we assume that the calls can be made in superposition. Further details about the quantum oracle model are provided in Chapter 5.

We identify an oracle with the function that it computes, usually a Boolean function $f : \{0,1\}^* \rightarrow \{0,1\}$. Often we think of f as defining a problem instance, or rather an infinite sequence of problem instances, one for each positive integer n. For example, "does there exist an $x \in \{0,1\}^n$ such that $f(x) = 1$?" In these cases the *oracle string*, which consists of $f(x)$ for every $x \in \{0,1\}^n$, can be thought of as an input that is 2^n bits

long instead of n bits. Of course, a classical algorithm running in polynomial time could examine only a tiny fraction of such an input, but maybe a quantum algorithm could do better. When discussing such questions, we need to be careful to distinguish between two functions: f itself, and the function of the oracle string that an algorithm is trying is to compute.

Chapter 4

Quantum Computing Cheat Sheet

> "Somebody says ... 'You know those quantum mechanical amplitudes you told me about, they're so complicated and absurd, what makes you think those are right? Maybe they aren't right.' Such remarks are obvious and are perfectly clear to anybody who is working on this problem. It does not do any good to point this out."
>
> —Richard Feynman, *The Character of Physical Law* [109]

Non-physicists often have the mistaken idea that quantum mechanics is hard. Unfortunately, many physicists have done nothing to correct that idea. But in newer textbooks, courses, and survey articles [18, 114, 175, 182, 235], the truth is starting to come out: if you wish to understand the central 'paradoxes' of quantum mechanics, together with almost the entire body of research on quantum information and computing, then you do not need to know anything about wave-particle duality, ultraviolet catastrophes, Planck's constant, atomic spectra, boson-fermion statistics, or even Schrödinger's equation. All you need to know is how to manipulate vectors whose entries are complex numbers. If that is too difficult, then positive and negative *real* numbers turn out to suffice for most purposes as well. After you have mastered these vectors, you will then have some context if you wish to learn more about the underlying physics. But the *historical* order in which the ideas were discovered is almost the reverse of the *logical* order in which they are easiest to learn!

What quantum mechanics says is that, if an object can be in either of two perfectly distinguishable states, which we denote $|0\rangle$ and $|1\rangle$, then it can also be in a linear "superposition" of those states, denoted $\alpha|0\rangle + \beta|1\rangle$. Here α and β are complex numbers called "amplitudes," which satisfy $|\alpha|^2 + |\beta|^2 = 1$. The asymmetric brackets $|\ \rangle$ are called "Dirac ket notation"; one gets used to them with time.

If we *measure* the state $\alpha|0\rangle + \beta|1\rangle$ in a standard way, then we see the "basis state" $|0\rangle$ with probability $|\alpha|^2$, and $|1\rangle$ with probability $|\beta|^2$. Also, the state *changes* to whichever outcome we see—so if we see $|0\rangle$ and then measure again, nothing having happened in the interim, we will still see $|0\rangle$. The two probabilities $|\alpha|^2$ and $|\beta|^2$ sum to 1, as they ought to. So far, we might as well have described the object using classical probabilities—for example, "this cat is alive with probability 1/2 and dead with probability 1/2; we simply don't know which."

The difference between classical probabilities and quantum amplitudes arises in how the object's state changes when we perform an operation on it. Classically, we can multiply a vector of probabilities by a *stochastic matrix*, which is a matrix of nonnegative real numbers each of whose columns sums to 1. Quantum-mechanically, we multiply the vector of amplitudes by a *unitary matrix*, which is a matrix of complex numbers that maps any unit vector to another unit vector. (Equivalently, U is unitary if and only if its inverse U^{-1} equals its conjugate transpose U^*.) As an example, suppose we start with the state $|0\rangle$, which corresponds to the vector of amplitudes

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

We then left-multiply this vector by the unitary matrix

$$U = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix},$$

which maps the vector to

$$\begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix},$$

and therefore the state $|0\rangle$ to

$$U|0\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle.$$

If we now measured, we would see $|0\rangle$ with probability $1/2$ and $|1\rangle$ with probability $1/2$. The interesting part is what happens if we apply the same operation U a second time, without measuring. We get

$$\begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

which is $|1\rangle$ with certainty (see Figure 4.1). Applying a "randomizing" operation to a "random" state produces a deterministic outcome! The reason is that, whereas probabilities are always nonnegative, amplitudes can be positive, negative, or even complex, and can therefore cancel each other out. This interference of amplitudes can be considered the source of all "quantum weirdness."

4.1 Quantum Computers: N Qubits

The above description applied to "qubits," or objects with only two distinguishable states. But it generalizes to objects with a larger number of distinguishable states. Indeed, in quantum computing we consider a system of N qubits, each of which can be $|0\rangle$ or $|1\rangle$. We then need to assign amplitudes to all 2^N possible outcomes of measuring the qubits in order from first to last. So the computer's state has the form

$$|\psi\rangle = \sum_{z \in \{0,1\}^N} \alpha_z |z\rangle$$

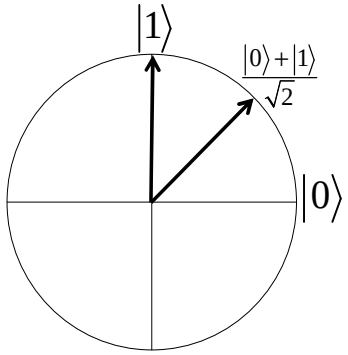

Figure 4.1: Quantum states of the form $\alpha|0\rangle + \beta|1\rangle$, with α and β real, can be represented by unit vectors in the plane. Then the operation U corresponds to a 45° counterclockwise rotation.

where

$$\sum_{z\in\{0,1\}^N} |\alpha_z|^2 = 1.$$

What was just said is remarkable—for it suggests that Nature needs to keep track of 2^N complex numbers just to describe a state of N interacting particles. If $N = 300$, then this is already more complex numbers than there are particles in the known universe. The goal of quantum computing is to exploit this strange sort of parallelism that is inherent in the laws of physics as we currently understand them.

The difficulty is that, when the computer's state is measured, we only see one of the "basis states" $|x\rangle$, not the entire collection of amplitudes. However, for a few specific problems, we might be able to arrange things so that basis states corresponding to wrong answers all have amplitudes close to 0, because of interference between positive and negative contributions. If we can do that, then basis states corresponding to right answers will be measured with high probability.

More explicitly, a quantum computer applies a sequence of unitary matrices called *gates*, each of which acts on only one or two of the N qubits (meaning that is a tensor product of the identity operation on $N-1$ or $N-2$ qubits, and the operation of interest on the remaining qubits). As an example, the *controlled-NOT* or CNOT gate is a two-qubit gate that flips a "target" qubit if a "control" qubit is 1, and otherwise does nothing:

$$|00\rangle \rightarrow |00\rangle, \quad |01\rangle \rightarrow |01\rangle, \quad |10\rangle \rightarrow |11\rangle, \quad |11\rangle \rightarrow |10\rangle.$$

The unitary matrix corresponding to the CNOT gate is

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Adleman, DeMarrais, and Huang [16] showed that the CNOT gate, together with the one-qubit gate

$$\begin{bmatrix} \frac{3}{5} & -\frac{4}{5} \\ \frac{4}{5} & \frac{3}{5} \end{bmatrix},$$

constitute a *universal* set of quantum gates, in that they can be used to approximate any other gate to any desired accuracy. Indeed, almost any set of one- and two-qubit gates is universal in this sense [94].

A *quantum circuit* is just a sequence of gates drawn from a finite universal set. Without loss of generality, we can take the circuit's output to be the result of a single measurement after all gates have been applied; that is, $z \in \{0,1\}^N$ with probability $|\alpha_z|^2$. (If a binary output is needed, we simply throw away the last $N-1$ bits of z.) It is known that allowing intermediate measurements does not yield any extra computational power [55]. The circuit is *polynomial-size* if both N and the number of gates are upper-bounded by a polynomial in the length n of the input.

We can now define the important complexity class BQP, or Bounded-Error Quantum Polynomial-Time. Given an input $x \in \{0,1\}^n$, first a polynomial-time classical algorithm A prepares a polynomial-size quantum circuit U_x. (The requirement that the circuit itself be efficiently preparable is called *uniformity*.) Then U_x is applied to the "all-0" initial state $|0\rangle^{\otimes N}$. We say a language $L \subseteq \{0,1\}^n$ is in BQP if there exists an A such that for all x,

(i) If $x \in L$ then U_x outputs '1' with probability at least 2/3.

(ii) If $x \notin L$ then U_x outputs '0' with probability at least 2/3.

By running U_x multiple times and taking the majority answer, we can boost the probability of success from 2/3 to $1 - 2^{-p(n)}$ for any polynomial p.

BQP was first defined in a 1993 paper by Bernstein and Vazirani [55].[1] That paper marked a turning point. Before, quantum computing had been an *idea*, explored in pioneering work by Deutsch [90], Feynman [108], and others. Afterward, quantum computing was a full-fledged *model* in the sense of computational complexity theory, which could be meaningfully compared against other models. For example, Bernstein and Vazirani showed that $\mathsf{BPP} \subseteq \mathsf{BQP} \subseteq \mathsf{P}^{\#\mathsf{P}}$: informally, quantum computers are *at least* as powerful as classical probabilistic computers, and at most exponentially *more* powerful. (The containment $\mathsf{BQP} \subseteq \mathsf{P}^{\#\mathsf{P}}$ was later improved to $\mathsf{BQP} \subseteq \mathsf{PP}$ by Adleman, DeMarrais, and Huang [16].)

Bernstein and Vazirani also gave an oracle problem called Recursive Fourier Sampling (RFS), and showed that it requires $n^{\Omega(\log n)}$ classical probabilistic queries but only n quantum queries. This provided the first evidence that quantum computers are strictly more powerful than classical probabilistic computers, i.e. that $\mathsf{BPP} \neq \mathsf{BQP}$. Soon afterward, Simon [220] widened the gap to polynomial versus exponential, by giving an oracle problem that requires $\Omega\left(2^{n/2}\right)$ classical probabilistic queries but only $O(n)$ quantum

[1] As a historical note, Bernstein and Vazirani [55] defined BQP in terms of "quantum Turing machines." However, Yao [248] showed that Bernstein and Vazirani's definition is equivalent to the much simpler one given here. Also, Berthiaume and Brassard [57] had implicitly defined EQP (Exact Quantum Polynomial-Time) a year earlier, and had shown that it lies outside P and even NP relative to an oracle.

queries. However, these results attracted limited attention because the problems seemed artificial.

People finally paid attention when Shor [219] showed that quantum computers could factor integers and compute discrete logarithms in polynomial time. The security of almost all modern cryptography rests on the presumed intractability of those two problems. It had long been known [177] that factoring is classically reducible to the following problem: given oracle access to a periodic function $f : \{1, \ldots, R\} \rightarrow \{1, \ldots, R\}$, where R is exponentially large, find the period of f. Shor gave an efficient quantum algorithm for this oracle problem, by exploiting the *quantum Fourier transform*, a tool that had earlier been used by Simon. (The algorithm for the discrete logarithm problem is more complicated but conceptually similar.)

Other results in the "quantum canon," such as Grover's algorithm [139] and methods for quantum error-correction and fault-tolerance [20, 80, 132, 159, 225], will be discussed in this thesis as the need arises.

4.2 Further Concepts

This section summarizes "fancier" quantum mechanics concepts, which are needed for Part II and for Chapter 10 of Part I (which deals with quantum advice). They are not needed for the other chapters in Part I.

Tensor Product. If $|\psi\rangle$ and $|\varphi\rangle$ are two quantum states, then their *tensor product*, denoted $|\psi\rangle \otimes |\varphi\rangle$ or $|\psi\rangle |\varphi\rangle$, is just a state that consists of $|\psi\rangle$ and $|\varphi\rangle$ next to each other. For example, if $|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$ and $|\varphi\rangle = \gamma |0\rangle + \delta |1\rangle$, then

$$|\psi\rangle |\varphi\rangle = (\alpha |0\rangle + \beta |1\rangle) (\gamma |0\rangle + \delta |1\rangle) = \alpha\gamma |00\rangle + \alpha\delta |01\rangle + \beta\gamma |10\rangle + \beta\delta |11\rangle .$$

Inner Product. The *inner product* between two states $|\psi\rangle = \alpha_1 |1\rangle + \cdots + \alpha_N |N\rangle$ and $|\varphi\rangle = \beta_1 |1\rangle + \cdots + \beta_N |N\rangle$ is defined as

$$\langle\psi|\varphi\rangle = \alpha_1^* \beta_1 + \cdots + \alpha_N^* \beta_N$$

where $*$ denotes complex conjugate. The inner product satisfies all the expected properties, such as $\langle\psi|\psi\rangle = 1$ and

$$\langle\psi| (|\varphi\rangle + |\phi\rangle) = \langle\psi|\varphi\rangle + \langle\psi|\phi\rangle .$$

If $\langle\psi|\varphi\rangle = 0$ then we say $|\psi\rangle$ and $|\varphi\rangle$ are *orthogonal*.

General Measurements. In principle, we can choose any orthogonal basis of states $\{|\varphi_1\rangle, \ldots, |\varphi_N\rangle\}$ in which to measure a state $|\psi\rangle$. (Whether that measurement can actually be performed *efficiently* is another matter.) Then the probability of obtaining outcome $|\varphi_j\rangle$ is $|\langle\psi|\varphi_j\rangle|^2$. We can even measure in a *non*-orthogonal basis, a concept called Positive Operator Valued Measurements (POVM's) that I will not explain here. None of these more general measurements increase the power of the quantum computing model, since we can always produce the same effect by first applying a unitary matrix (possibly using additional qubits called *ancillas*), and then measuring in a "standard" basis such as $\{|1\rangle, \ldots, |N\rangle\}$.

Mixed States. Superposition states, such as $\alpha|0\rangle + \beta|1\rangle$, are also called *pure states*. This is to distinguish them from *mixed states*, which are the most general kind of state in quantum mechanics. Mixed states are just classical probability distributions over pure states. There is a catch, though: any mixed state can be decomposed into a probability distribution over pure states in infinitely many nonequivalent ways. For example, if we have a state that is $|0\rangle$ with probability 1/2 and $|1\rangle$ with probability 1/2, then no experiment could ever distinguish it from a state that is $(|0\rangle + |1\rangle)/\sqrt{2}$ with probability 1/2 and $(|0\rangle - |1\rangle)/\sqrt{2}$ with probability 1/2. For regardless of what orthogonal basis we measured in, the two possible outcomes of measuring would both occur with probability 1/2. Therefore, this state is called the one-qubit *maximally mixed state*.

Density Matrices. We can represent mixed states using a formalism called *density matrices*. The *outer product* of $|\psi\rangle = \alpha_1|1\rangle + \cdots + \alpha_N|N\rangle$ with itself, denoted $|\psi\rangle\langle\psi|$, is an $N \times N$ complex matrix whose (i,j) entry is $\alpha_i\alpha_j^*$. Now suppose we have a state that is $|\varphi\rangle = \alpha|0\rangle + \beta|1\rangle$ with probability p, and $|\phi\rangle = \gamma|0\rangle + \delta|1\rangle$ with probability $1-p$. We represent the state by a Hermitian positive definite matrix ρ with trace 1, as follows:

$$\rho = p|\varphi\rangle\langle\varphi| + (1-p)|\phi\rangle\langle\phi| = p\begin{bmatrix} \alpha\alpha^* & \alpha\beta^* \\ \beta\alpha^* & \beta\beta^* \end{bmatrix} + (1-p)\begin{bmatrix} \gamma\gamma^* & \gamma\delta^* \\ \delta\gamma^* & \delta\delta^* \end{bmatrix}.$$

When we apply a unitary operation U, the density matrix ρ changes to $U\rho U^{-1}$. When we measure in the standard basis, the probability of outcome $|j\rangle$ is the j^{th} diagonal entry of ρ. Proving that these rules are the correct ones, and that a density matrix really is a unique description of a mixed state, are "exercises for the reader" (which as always means the author was too lazy). Density matrices will mainly be used in Chapter 10.

Trace Distance. Suppose you are given a system that was prepared in state ρ with probability 1/2, and σ with probability 1/2. After making a measurement, you must guess which state the system was prepared in. What is the maximum probability that you will be correct? The answer turns out to be

$$\frac{1 + \|\rho - \sigma\|_{\text{tr}}}{2}$$

where $\|\rho - \sigma\|_{\text{tr}}$ is the *trace distance* between ρ and σ, defined as $\frac{1}{2}\sum_i |\lambda_i|$ where $\lambda_1, \ldots, \lambda_N$ are the eigenvalues of $\rho - \sigma$.

Entanglement. Suppose ρ is a joint state of two systems. If ρ can be written as a probability distribution over pure states of the form $|\psi\rangle \otimes |\varphi\rangle$, then we say ρ is *separable*; otherwise ρ is *entangled*.

Hamiltonians. Instead of discrete unitary operations, we can imagine that a quantum state evolves in time by a continuous rotation called a *Hamiltonian*. A Hamiltonian is an $N \times N$ Hermitian matrix H. To find the unitary operation $U(t)$ that is effected by "leaving H on" for t time steps, the rule[2] is $U(t) = e^{-iHt}$. The only place I use Hamiltonians is in Chapter 14, and even there the use is incidental.

[2]Here Planck's constant is set equal to 1 as always.

Part I

Limitations of Quantum Computers

Chapter 5

Introduction

> "A quantum possibility is less real than a classical reality, but more real than a classical possibility."
>
> —Boris Tsirelson [229]

Notwithstanding accounts in the popular press, a decade of research has made it clear that quantum computers would not be a panacea. In particular, we still do not have a quantum algorithm to solve NP-complete problems in polynomial time. But can we prove that no such algorithm exists, i.e. that $\mathsf{NP} \not\subset \mathsf{BQP}$? The difficulty is that we can't even prove no *classical* algorithm exists; this is the P versus NP question. Of course, we could ask whether $\mathsf{NP} \not\subset \mathsf{BQP}$ *assuming* that $\mathsf{P} \neq \mathsf{NP}$—but unfortunately, even this conditional question seems far beyond our ability to answer. So we need to refine the question even further: can quantum computers solve NP-complete problems in polynomial time, *by brute force*?

What is meant by "brute force" is the following. In Shor's factoring algorithm [219], we prepare a superposition of the form

$$\frac{1}{\sqrt{R}} \sum_{r=1}^{R} |r\rangle \, |g(r)\rangle$$

where $g(r) = x^r \bmod N$ for some x, N. But as far as the key step of the algorithm is concerned, the function g is a "black box." Given any superposition like the one above, the algorithm will find the period of g assuming g is periodic; it does not need further information about how g was computed. So in the language of Section 3.3, we might as well say that g is computed by an oracle.

Now suppose we are given a Boolean formula φ over n variables, and are asked to decide whether φ is satisfiable. One approach would be to exploit the internal structure of φ: "let's see, if I set variable x_{37} to TRUE, then this clause here is satisfied, but those other clauses aren't satisfied any longer ... darn!" However, inspired by Shor's factoring algorithm, we might hope for a cruder quantum algorithm that treats φ merely as an oracle, mapping an input string $x \in \{0,1\}^n$ to an output bit $\varphi(x)$ that is 1 if and only if x satisfies φ. The algorithm would have to decide whether there exists an $x \in \{0,1\}^n$ such that

$\varphi(x) = 1$, using as few calls to the φ oracle as possible, and not learning about φ in any other way. This is what is meant by brute force.

A fundamental result of Bennett, Bernstein, Brassard, and Vazirani [51] says that no brute-force quantum algorithm exists to solve NP-complete problems in polynomial time. In particular, for some probability distribution over oracles, any quantum algorithm needs $\Omega\left(2^{n/2}\right)$ oracle calls to decide, with at least a 2/3 chance of being correct, whether there exists an $x \in \{0,1\}^n$ such that $\varphi(x) = 1$. On a classical computer, of course, $\Theta(2^n)$ oracle calls are necessary and sufficient. But as it turns out, Bennett et al.'s quantum lower bound is tight, since Grover's quantum search algorithm [139] can find a satisfying assignment (if it exists) quadratically faster than any classical algorithm. Amusingly, Grover's algorithm was proven optimal before it was discovered to exist!

A recurring theme in this thesis is the *pervasiveness* of Bennett et al.'s finding. I will show that, even if a problem has considerably more structure than the basic Grover search problem, even if "quantum advice states" are available, or even if we could examine the entire history of a hidden variable, still any brute-force quantum algorithm would take exponential time.

5.1 The Quantum Black-Box Model

The *quantum black-box model* formalizes the idea of a brute-force algorithm. For the time being, suppose that a quantum algorithm's goal is to evaluate $f(X)$, where $f : \{0,1\}^n \rightarrow \{0,1\}$ is a Boolean function and $X = x_1 \ldots x_n$ is an n-bit string. Then the algorithm's state at any time t has the form

$$\sum_{i,z} \alpha_{i,z}^{(t)} |i,z\rangle .$$

Here $i \in \{1,\ldots,N\}$ is the index of an oracle bit x_i to query, and z is an arbitrarily large string of bits called the "workspace," containing whatever information the algorithm wants to store there. The state evolves in time via an alternating sequence of *algorithm steps* and *query steps*. An algorithm step multiplies the vector of $\alpha_{i,z}$'s by an arbitrary unitary matrix that does not depend on X. It does not matter how many quantum gates would be needed to implement this matrix. A query step maps each basis state $|i,z\rangle$ to $|i,z \oplus x_i\rangle$, effecting the transformation $\alpha_{i,z}^{(t+1)} = \alpha_{i,z\oplus x_i}^{(t)}$. Here $z \oplus x_i$ is the string z, with x_i exclusive-OR'ed into a particular location in z called the "answer bit." The reason exclusive-OR is used is that the query step has to be reversible, or else it would not be unitary.

At the final step T, the state is measured in the standard basis, and the output of the algorithm is taken to be (say) z_1, the first bit of z. The algorithm succeeds if

$$\sum_{i,z \,:\, z_1 = f(X)} \left|\alpha_{i,z}^{(T)}\right|^2 \geq \frac{2}{3}$$

for all $X \in \{0,1\}^n$. Here the constant 2/3 is arbitrary. Then the *(bounded-error) quantum query complexity* of f, denoted $\mathrm{Q}_2(f)$, is the minimum over all quantum algorithms A that succeed at evaluating f, of the number of queries to f made by A. Here the '2' represents the fact that the error probability is two-sided. One can compare $\mathrm{Q}_2(f)$ with $\mathrm{Q}_0(f)$,

or zero-error quantum query complexity; $\mathrm{R}_2(f)$, or bounded-error classical randomized query complexity; and $\mathrm{D}(f)$, or deterministic query complexity, among other complexity measures. Chapter 8 will define many such measures and compare them in detail.

As a simple example of the black-box model, let $\mathrm{OR}(x_1, \ldots, x_n) = x_1 \vee \cdots \vee x_n$. Then Grover's algorithm [139] implies that $\mathrm{Q}_2(\mathrm{OR}) = O(\sqrt{n})$, while the lower bound of Bennett et al. [51] implies that $\mathrm{Q}_2(\mathrm{OR}) = \Omega(\sqrt{n})$. By comparison, $D(\mathrm{OR}) = R_2(\mathrm{OR}) = \Theta(n)$.

The quantum black-box model has some simple generalizations, which I will use when appropriate. First, f can be a partial function, defined only on a subset of $\{0,1\}^n$ (so we obtain what is called a *promise problem*). Second, the x_i's do not need to be bits; in Chapters 6 and 7 they will take values from a larger range. Third, in Chapter 7 the output will not be Boolean, and there will generally be more than one valid output (so we obtain what is called a *relation problem*).

5.2 Oracle Separations

"I do believe it
Against an oracle."
—Shakespeare, *The Tempest*

Several times in this thesis, I will use a lower bound on quantum query complexity to show that a complexity class is not in BQP "relative to an oracle." The method for turning query complexity lower bounds into oracle separations was invented by Baker, Gill, and Solovay [41] to show that there exists an oracle A relative to which $\mathsf{P}^A \neq \mathsf{NP}^A$. Basically, they encoded into A an infinite sequence of exponentially hard search problems, one for each input length n, such that (i) a nondeterministic machine can solve the n^{th} problem in time polynomial in n, but (ii) any deterministic machine would need time exponential in n. They guaranteed (ii) by "diagonalizing" against all possible deterministic machines, similarly to how Turing created an uncomputable real number by diagonalizing against all possible computable reals. Later, Bennett and Gill [54] showed that a simpler way to guarantee (ii) is just to choose the search problems uniformly at random. Throughout the thesis, I will cavalierly ignore such issues, proceeding immediately from a query complexity lower bound to the statement of the corresponding oracle separation.

The point of an oracle separation is to rule out certain approaches to solving an open problem in complexity theory. For example, the Baker-Gill-Solovay theorem implies that the standard techniques of computability theory, which *relativize* (that is, are "oblivious" to the presence of an oracle), cannot be powerful enough to show that $\mathsf{P} = \mathsf{NP}$. Similarly, the result of Bennett et al. [51] that $\mathrm{Q}_2(\mathrm{OR}) = \Omega(\sqrt{n})$ implies that there exists an oracle A relative to which $\mathsf{NP}^A \not\subset \mathsf{BQP}^A$. While this does not show that $\mathsf{NP} \not\subset \mathsf{BQP}$, it does show that any proof of $\mathsf{NP} \subseteq \mathsf{BQP}$ would have to use "non-relativizing" techniques that are unlike anything we understand today.

However, many computer scientists are skeptical that anything can be learned from oracles. The reason for their skepticism is that over the past 15 years, they have seen several examples of non-relativizing results in classical complexity theory. The most

famous of these is Shamir's Theorem [215, 171] that $\mathsf{PSPACE} \subseteq \mathsf{IP}$, where IP is the class of problems that have *interactive proof systems*, meaning that if the answer for some instance is "yes," then a polynomial-time verifier can become convinced of that fact to any desired level of confidence by exchanging a sequence of messages with an omniscient prover.[1] By contrast, oracles had been known relative to which not even coNP, let alone PSPACE, is contained in IP [117]. So why should we ever listen to oracles again, if they got interactive proofs so dramatically wrong?

My answer is threefold. First, essentially all quantum algorithms that we know today—from Shor's algorithm, as discussed previously, to Grover's algorithm, to the quantum adiabatic algorithm[2] [106], to the algorithms of Hallgren [141] and van Dam, Hallgren, and Ip [232]—are oracle algorithms at their core. We do not know of any non-relativizing quantum algorithm technique analogous to the arithmetization technique that was used to prove $\mathsf{PSPACE} \subseteq \mathsf{IP}$. If such a technique is ever discovered, I will be one of the first to want to learn it.

The second response is that without oracle results, we do not have even the beginnings of understanding. Once we know (for example) that $\mathsf{SZK} \not\subset \mathsf{BQP}$ relative to an oracle, we can *then* ask the far more difficult unrelativized question, knowing something about the hurdles that any proof of $\mathsf{SZK} \subseteq \mathsf{BQP}$ would have to overcome.

The third response is that "the proof of the pudding is in the proving." In other words, the real justification for the quantum black-box model is not the *a priori* plausibility of its assumptions, but the depth and nontriviality of what can be (and has been) proved in it. For example, the result that $\mathsf{coNP} \not\subset \mathsf{IP}$ relative to an oracle [117] does not tell us much about interactive proof systems. For given an exponentially long oracle string X, it is intuitively obvious that nothing a prover could say could convince a classical polynomial-time verifier that X is the all-0 string, even if the prover and verifier could interact. The only issue is how to formalize that obvious fact by diagonalizing against all possible proof systems. By contrast, the quantum oracle separations that we have are *not* intuitively obvious in the same way; or rather, the act of understanding them confers an intuition where none was previously present.

[1] Arora, Impagliazzo, and Vazirani [36] claim the Cook-Levin Theorem, that Satisfiability is NP-complete, as another non-relativizing result. But this hinges on what we mean by "non-relativizing," far more than the $\mathsf{PSPACE} \subseteq \mathsf{IP}$ example.

[2] Given an assignment x to a 3SAT formula φ, the adiabatic algorithm actually queries an oracle that returns the *number of clauses* of φ that x satisfies, not just whether x satisfies φ or not. Furthermore, van Dam, Mosca, and Vazirani [233] have shown such an oracle is sufficient to reconstruct φ. On the other hand, the adiabatic algorithm itself would be just as happy with a fitness landscape that did not correspond to any 3SAT instance, and that is what I mean by saying that it is an oracle algorithm at the core.

Chapter 6

The Collision Problem

The collision problem of size n, or Col_n, is defined as follows. Let $X = x_1 \dots x_n$ be a sequence of n integers drawn from $\{1, \dots, n\}$, with n even. We are guaranteed that either

(1) X is one-to-one (that is, a permutation of $\{1, \dots, n\}$), or

(2) X is two-to-one (that is, each element of $\{1, \dots, n\}$ appears in X twice or not at all).

The problem is to decide whether (1) or (2) holds. (A variant asks us to *find* a collision in a given two-to-one function. Clearly a lower bound for the collision problem as defined above implies an equivalent lower bound for this variant.) Because of its simplicity, the collision problem was widely considered a benchmark for our understanding of quantum query complexity.

I will show that $\mathrm{Q}_2(\mathrm{Col}_n) = \Omega\left(n^{1/5}\right)$, where $\mathrm{Q}_2(f)$ is the bounded-error quantum query complexity of function f. The best known upper bound, due to Brassard, Høyer, and Tapp [68], is $O\left(n^{1/3}\right)$ (see Section 2.1.1). Previously, though, no lower bound better than the trivial $\Omega(1)$ bound was known. How great a speedup quantum computers yield for the problem was apparently first asked by Rains [193].

Previous lower bound techniques failed for the problem because they depended on a function's being sensitive to many disjoint changes to the input. For example, Beals et al. [45] showed that for all total Boolean functions f, $\mathrm{Q}_2(f) = \Omega\left(\sqrt{\mathrm{bs}(f)}\right)$, where $\mathrm{bs}(f)$ is the block sensitivity, defined by Nisan [183] to be, informally, the maximum number of disjoint changes (to any particular input X) to which f is sensitive. In the case of the collision problem, though, every one-to-one input differs from every two-to-one input in at least $n/2$ places, so the block sensitivity is $O(1)$. Ambainis's adversary method [27] faces a related obstacle. In that method we consider the algorithm and input as a bipartite quantum state, and upper-bound how much the entanglement of the state can increase via a single query. But under the simplest measures of entanglement, it turns out that the algorithm and input can become almost maximally entangled after $O(1)$ queries, again because every one-to-one input is far from every two-to-one input.[1]

[1] More formally, the adversary method cannot prove any lower bound on $\mathrm{Q}_2(f)$ better than $\mathrm{RC}(f)$, where $\mathrm{RC}(f)$ is the *randomized certificate complexity* of f (to be defined in Chapter 8). But for the

My proof is an adaptation of the polynomial method, introduced to quantum computing by Beals et al. [45]. Their idea was to reduce questions about quantum algorithms to easier questions about multivariate polynomials. In particular, if a quantum algorithm makes T queries, then its acceptance probability is a polynomial over the input bits of degree at most $2T$. So by showing that any polynomial approximating the desired output has high degree, one obtains a lower bound on T.

To lower-bound the degree of a multivariate polynomial, a key technical trick is to construct a related *univariate* polynomial. Beals et al. [45], using a lemma due to Minsky and Papert [178], replace a polynomial $p(X)$ (where X is a bit string) by $q(|X|)$ (where $|X|$ denotes the Hamming weight of X), satisfying

$$q(k) = \underset{|X|=k}{\mathrm{EX}} p(X)$$

and $\deg(q) \leq \deg(p)$.

Here I construct the univariate polynomial in a different way. I consider a uniform distribution over g-to-one inputs, where g might be greater than 2. Even though the problem is to distinguish $g = 1$ from $g = 2$, the acceptance probability must lie in the interval $[0, 1]$ for all g, and that is a surprisingly strong constraint. I show that the acceptance probability is *close* to a univariate polynomial in g of degree at most $2T$. I then obtain a lower bound by generalizing a classical approximation theory result of Ehlich and Zeller [104] and Rivlin and Cheney [204]. Much of the proof deals with the complication that g does not divide n in general.

Shortly after this work was completed, Shi [218] improved it to give a tight lower bound of $\Omega\left(n^{1/3}\right)$ for the collision problem, when the x_i range from 1 to $3n/2$ rather than from 1 to n. For a range of size n, his bound was $\Omega\left(n^{1/4}\right)$. Subsequently Kutin [161] and Ambainis [29] showed a lower bound of $\Omega\left(n^{1/3}\right)$ for a range of size n as well. By a simple reduction, these results imply a lower bound of $\Omega\left(n^{2/3}\right)$ for the *element distinctness problem*—that of deciding whether there exist $i \neq j$ such that $x_i = x_j$. The previous best known lower bound was $\Omega\left(n^{1/2}\right)$, and at the time of Shi's work, the best known upper bound was $O\left(n^{3/4}\right)$, due to Buhrman et al. [77]. Recently, however, Ambainis [30] gave a novel algorithm based on quantum walks that matches the $n^{2/3}$ lower bound.

The chapter is organized as follows. Section 6.1 motivates the collision lower bound within quantum computing, pointing out connections to collision-resistant hash functions, the nonabelian hidden subgroup problem, statistical zero-knowledge, and information erasure. Section 6.2 gives technical preliminaries, Section 6.3 proves the crucial fact that the acceptance probability is "almost" a univariate polynomial, and Section 6.4 completes the lower bound argument. I conclude in Section 6.6 with some open problems. In Section 6.5 I show a lower bound of $\Omega\left(n^{1/7}\right)$ for the *set comparison problem*, a variant of the collision problem needed for the application to information erasure.

---

collision function, $\mathrm{RC}(\mathrm{Col}_n) = O(1)$.

6.1 Motivation

In Chapter 1 I listed seven implications of the collision lower bound; this section discusses a few of those implications in more detail. The implication that motivated me personally—concerning the computational power of so-called *hidden-variable theories*—is deferred to Chapter 16.

6.1.1 Oracle Hardness Results

The original motivation for the collision problem was to model *(strongly) collision-resistant hash functions* in cryptography. There is a large literature on collision-resistant hashing; see [201, 42] for example. When building secure digital signature schemes, it is useful to have a family of hash functions $\{H_i\}$, such that finding a distinct (x, y) pair with $H_i(x) = H_i(y)$ is computationally intractable. A quantum algorithm for finding collisions using $O(\text{polylog}(n))$ queries would render *all* hash functions insecure against quantum attack in this sense. (Shor's algorithm [219] already renders hash functions based on modular arithmetic insecure.) My result indicates that collision-resistant hashing might still be possible in a quantum setting.

The collision problem also models the *nonabelian hidden subgroup problem*, of which graph isomorphism is a special case. Given a group G and subgroup $H \leq G$, suppose we have oracle access to a function $f : G \rightarrow \mathbb{N}$ such that for all $g_1, g_2 \in G$, $f(g_1) = f(g_2)$ if and only if g_1 and g_2 belong to the same coset of H. Is there then an efficient quantum algorithm to determine H? If G is abelian, the work of Simon [220], Shor [219], and Kitaev [152] implies an affirmative answer. If G is nonabelian, though, efficient quantum algorithms are known only for special cases [105, 138]. An $O(\text{polylog}(n))$-query algorithm for the collision problem would yield a polynomial-time algorithm to distinguish $|H| = 1$ from $|H| = 2$, which does not exploit the group structure at all. My result implies that no such algorithm exists.

Finally, the collision lower bound implies that there exists an oracle relative to which $\mathsf{SZK} \not\subset \mathsf{BQP}$, where SZK is the class of problems having statistical zero-knowledge proof protocols. For suppose that a verifier V and prover P both have oracle access to a sequence $X = x_1 \ldots x_{2^n}$, which is either one-to-one or two-to-one. To verify with zero knowledge that X is one-to-one, V can repeatedly choose an $i \in_R \{1, \ldots, 2^n\}$ and send x_i to P, whereupon P must send i back to V. Thus, using standard diagonalization techniques, one can produce an oracle A such that $\mathsf{SZK}^A \not\subset \mathsf{BQP}^A$.

6.1.2 Information Erasure

Let $f : \{0,1\}^n \rightarrow \{0,1\}^m$ with $m \geq n$ be a one-to-one function. Then we can consider two kinds of quantum oracle for f:

(A) a *standard oracle*, one that maps $|x\rangle |z\rangle$ to $|x\rangle |z \oplus f(x)\rangle$, or

(B) an *erasing oracle* (as proposed by Kashefi et al. [150]), which maps $|x\rangle$ to $|f(x)\rangle$, in effect "erasing" $|x\rangle$.

Intuitively erasing oracles seem at least as strong as standard ones, though it is not clear how to simulate the latter with the former without also having access to an oracle that maps $|y\rangle$ to $\left|f^{-1}(y)\right\rangle$. The question that concerns us here is whether erasing oracles are *more* useful than standard ones for some problems. One-way functions provide a clue: if f is one-way, then (by assumption) $|x\rangle|f(x)\rangle$ can be computed efficiently, but if $|f(x)\rangle$ could be computed efficiently given $|x\rangle$ then so could $|x\rangle$ given $|f(x)\rangle$, and hence f could be inverted. But can we find, for some problem, an exponential gap between query complexity given a standard oracle and query complexity given an erasing oracle?

In Section 6.5 I extend the collision lower bound to show an affirmative answer. Define the set comparison problem of size n, or SetComp_n, as follows. We are given as input two sequences, $X = x_1 \ldots x_n$ and $Y = y_1 \ldots y_n$, such that for each i, $x_i, y_i \in \{1, \ldots, 2n\}$. A query has the form (b, i), where $b \in \{0, 1\}$ and $i \in \{1, \ldots, n\}$, and produces as output $(0, x_i)$ if $b = 0$ and $(1, y_i)$ if $b = 1$. Sequences X and Y are both one-to-one; that is, $x_i \neq x_j$ and $y_i \neq y_j$ for all $i \neq j$. We are furthermore guaranteed that either

(1) X and Y are equal as sets (that is, $\{x_1, \ldots, x_n\} = \{y_1, \ldots, y_n\}$) or

(2) X and Y are far as sets (that is, $|\{x_1, \ldots, x_n\} \cup \{y_1, \ldots, y_n\}| \geq 1.1n$).

As before the problem is to decide whether (1) or (2) holds.

This problem can be solved with high probability in a constant number of queries using an erasing oracle, by using a trick similar to that of Watrous [239] for verifying group non-membership. First, using the oracle, we prepare the uniform superposition

$$\frac{1}{\sqrt{2n}} \sum_{i \in \{1,\ldots,n\}} \left(|0\rangle|x_i\rangle + |1\rangle|y_i\rangle\right).$$

We then apply a Hadamard gate to the first register, and finally we measure the first register. If X and Y are equal as sets, then interference occurs between every $\left(|0\rangle|z\rangle, |1\rangle|z\rangle\right)$ pair and we observe $|0\rangle$ with certainty. But if X and Y are far as sets, then basis states $|b\rangle|z\rangle$ with no matching $|1-b\rangle|z\rangle$ have probability weight at least $1/10$, and hence we observe $|1\rangle$ with probability at least $1/20$.

In Section 6.5 I sketch a proof that $\text{Q}_2(\text{SetComp}_n) = \Omega\left(n^{1/7}\right)$; that is, no efficient quantum algorithm using a standard oracle exists for this problem. Recently, Midrijanis [176] gave a lower bound of $\Omega\left((n/\log n)^{1/5}\right)$ not merely for the set comparison problem, but for the *set equality problem* (where we are promised that X and Y are either equal or disjoint).

6.2 Preliminaries

Let A be a quantum query algorithm as defined in Section 5.1. A basis state of A is written $|i, z\rangle$. Then a query replaces each $|i, z\rangle$ by $|i, z \oplus x_i\rangle$, where x_i is exclusive-OR'ed into some specified location of z. Between queries, the algorithm can perform any unitary operation

that does not depend on the input. Let T be the total number of queries. Also, assume for simplicity that all amplitudes are real; this restriction is without loss of generality [55].

Let $\alpha_{i,z}^{(t)}(X)$ be the amplitude of basis state $|i,z\rangle$ after t queries when the input is X. Also, let $\Delta(x_i,h)=1$ if $x_i=h$, and $\Delta(x_i,h)=0$ if $x_i\neq h$. Let $P(X)$ be the probability that A returns "two-to-one" when the input is X. Then we obtain a simple variant of a lemma due to Beals et al. [45].

Lemma 1 *$P(X)$ is a multilinear polynomial of degree at most $2T$ over the $\Delta(x_i,h)$.*

Proof. We show, by induction on t, that for all basis states $|i,z\rangle$, the amplitude $\alpha_{i,z}^{(t)}(X)$ is a multilinear polynomial of degree at most t over the $\Delta(x_i,h)$. Since $P(X)$ is a sum of squares of $\alpha_{i,z}^{(t)}$'s, the lemma follows.

The base case ($t=0$) holds since, before making any queries, each $\alpha_{i,z}^{(t)}$ is a degree-0 polynomial over the $\Delta(x_i,h)$. A unitary transformation on the algorithm part replaces each $\alpha_{i,z}^{(t)}$ by a linear combination of $\alpha_{i,z}^{(t)}$'s, and hence cannot increase the degree. Suppose the lemma holds prior to the t^{th} query. Then

$$\alpha_{i,z}^{(t+1)}(X)=\sum_{1\leq h\leq n}\alpha_{i,z\oplus h}^{(t)}(X)\,\Delta(x_i,h),$$

and we are done. ■

6.3 Reduction to Bivariate Polynomial

Call the point $(g,N)\in\Re^2$ an (n,T)-*quasilattice point* if and only if

(1) g and N are integers, with g dividing N,

(2) $1\leq g\leq\sqrt{n}$,

(3) $n\leq N\leq n+n/(10T)$, and

(4) if $g=1$ then $N=n$.

For quasilattice point (g,N), define $\mathcal{D}_n(g,N)$ to be the uniform distribution over all size-n subfunctions of g-to-1 functions having domain $\{1,\ldots,N\}$ and range a subset of $\{1,\ldots,n\}$. More precisely: to draw an X from $\mathcal{D}_n(g,N)$, we first choose a set $S\subseteq\{1,\ldots,n\}$ with $|S|=N/g\leq n$ uniformly at random. We then choose a g-to-1 function $\widehat{X}=\widehat{x}_1\ldots\widehat{x}_N$ from $\{1,\ldots,N\}$ to S uniformly at random. Finally we let $x_i=\widehat{x}_i$ for each $1\leq i\leq n$.

Let $P(g,N)$ be the probability that algorithm A returns $z=2$ when the input is chosen from $\mathcal{D}_n(g,N)$:

$$P(g,N)=\mathop{\mathrm{EX}}_{X\in\mathcal{D}_n(g,N)}[P(X)].$$

We then have the following surprising characterization:

Lemma 2 *For all sufficiently large n and if $T \leq \sqrt{n}/3$, there exists a bivariate polynomial $q(g,N)$ of degree at most $2T$ such that if (g,N) is a quasilattice point, then*

$$|P(g,N) - q(g,N)| < 0.182$$

(where the constant 0.182 can be made arbitrarily small by adjusting parameters).

Proof. Let I be a product of $\Delta(x_i,h)$ variables, with degree $r(I)$, and let $I(X) \in \{0,1\}$ be I evaluated on input X. Then define

$$\gamma(I,g,N) = \underset{X \in \mathcal{D}_n(g,N)}{\mathrm{EX}}[I(X)]$$

to be the probability that monomial I evaluates to 1 when the input is drawn from $\mathcal{D}_n(g,N)$. Then by Lemma 1, $P(X)$ is a polynomial of degree at most $2T$ over X, so

$$\begin{aligned} P(g,N) &= \underset{X \in \mathcal{D}_n(g,N)}{\mathrm{EX}}[P(X)] \\ &= \underset{X \in \mathcal{D}_n(g,N)}{\mathrm{EX}}\left[\sum_{I:r(I)\leq 2t} \beta_I I(X)\right] \\ &= \sum_{I:r(I)\leq 2T} \beta_I \gamma(I,g,N) \end{aligned}$$

for some coefficients β_I.

We now calculate $\gamma(I,g,N)$. Assume without loss of generality that for all $\Delta(x_i,h_1), \Delta(x_j,h_2) \in I$, either $i \neq j$ or $h_1 = h_2$, since otherwise $\gamma(I,g,N) = 0$.

Define the "range" $Z(I)$ of I to be the set of all h such that $\Delta(x_i,h) \in I$. Let $w(I) = |Z(I)|$; then we write $Z(I) = \{z_1,\ldots,z_{w(I)}\}$. Clearly $\gamma(I,g,N) = 0$ unless $Z(I) \in S$, where S is the range of $\widehat{X}$. By assumption,

$$\frac{N}{g} \geq \frac{n}{\sqrt{n}} \geq 2T \geq r(I)$$

so the number of possible S is $\binom{n}{N/g}$ and, of these, the number that contain Z is $\binom{n-w(I)}{N/g-w(I)}$.

Then, conditioned on $Z \in S$, what is the probability that $\gamma(I,g,N) = 1$? The total number of g-to-1 functions with domain size N is $N!/(g!)^{N/g}$, since we can permute the N function values arbitrarily, but must not count permutations that act only within the N/g constant-value blocks of size g.

Among these functions, how many satisfy $\gamma(I,g,N) = 1$? Suppose that, for each $1 \leq j \leq w(I)$, there are $r_j(I)$ distinct i such that $\Delta(x_i,z_j) \in I$. Clearly

$$r_1(I) + \cdots + r_{w(I)}(I) = r(I).$$

Then we can permute the $(N - r(I))!$ function values outside of I arbitrarily, but must not count permutations that act only within the N/g constant-value blocks, which have size either g or $g - r_i(I)$ for some i. So the number of functions for which $\gamma(I, g, N) = 1$ is

$$\frac{(N - r(I))!}{(g!)^{N/g - w(I)} \prod_{i=1}^{w(I)} (g - r_i(I))!}.$$

Putting it all together,

$$\begin{aligned}
\gamma(I, g, N) = & \frac{\binom{n - w(I)}{N/g - w(I)}}{\binom{n}{N/g}} \cdot \frac{(N - r(I))! (g!)^{N/g}}{(g!)^{N/g - w(I)} N! \prod_{i=1}^{w(I)} (g - r_i(I))!} \\
= & \frac{(N - r(I))! (n - w(I))! (N/g)!}{N! n! (N/g - w(I))!} \cdot \frac{(g!)^{w(I)}}{\prod_{i=1}^{w(I)} (g - r_i(I))!} \\
= & \frac{(N - r(I))!}{N!} \frac{(n - w(I))!}{n!} \cdot \prod_{i=0}^{w(I)-1} \left(\frac{N}{g} - i \right) \prod_{i=1}^{w(I)} \left[g \prod_{j=1}^{r_i(I)-1} (g - j) \right] \\
= & \frac{(N - 2T)! n!}{N! (n - 2T)!} \widetilde{q}_{n,T,I}(g, N)
\end{aligned}$$

where

$$\widetilde{q}_{n,T,I}(g, N) = \frac{(n - w(I))! (n - 2T)!}{(n!)^2} \cdot \prod_{i=r(I)}^{2T-1} (N - i) \prod_{i=0}^{w(I)-1} (N - gi) \prod_{i=1}^{w(I) r_i(I) - 1} \prod_{j=1} (g - j)$$

is a bivariate polynomial of total degree at most

$$(2T - r(I)) + w(I) + (r(I) - w(I)) = 2T.$$

(Note that in the case $r_i(I) > g$ for some i, this polynomial evaluates to 0, which is what it ought to do.) Hence

$$\begin{aligned}
P(g, N) = & \sum_{I : r(I) \le 2T} \beta_I \gamma(I, g, N) \\
= & \frac{(N - 2T)! n!}{N! (n - 2T)!} q(g, N)
\end{aligned}$$

where

$$q(g, N) = \sum_{I : r(I) \le 2T} \beta_I \widetilde{q}_{n,T,I}(g, N).$$

Clearly

$$\frac{(N - 2T)! n!}{N! (n - 2T)!} \le 1.$$

Since $N \leq n + n/(10T)$ and $T \leq \sqrt{n}/3$, we also have

$$\begin{aligned}\frac{(N-2T)!n!}{N!(n-2T)!} &\geq \left(\frac{n-2T+1}{N-2T+1}\right)^{2T} \\ &\geq \exp\left\{-\frac{1}{5}\frac{n}{n-(2T+1)/n}\right\} \\ &\geq 0.818\end{aligned}$$

for all sufficiently large n. Thus, since $0 \leq P(g,N) \leq 1$,

$$|P(g,N) - q(g,N)| < 0.182$$

and we are done. ■

6.4 Lower Bound

We have seen that, if a quantum algorithm for the collision problem makes few queries, then its acceptance probability can be approximated by a low-degree bivariate polynomial. This section completes the lower bound proof by showing that no such polynomial exists. To do so, it generalizes an approximation theory result due to Rivlin and Cheney [204] and (independently) Ehlich and Zeller [104]. That result was applied to query complexity by Nisan and Szegedy [184] and later by Beals et al. [45].

Theorem 3 $\mathrm{Q}_2(\mathrm{Col}_n) = \Omega\left(n^{1/5}\right)$.

Proof. Let g have range $1 \leq g \leq G$. Then the quasilattice points (g,N) all lie in the rectangular region $R = [1,G] \times [n, n+n/(10T)]$. Recalling the polynomial $q(g,N)$ from Lemma 2, define

$$d(q) = \max_{(g,N)\in R}\left(\max\left\{\left|\frac{\partial q}{\partial g}\right|, \frac{n}{10T(G-1)}\cdot\left|\frac{\partial q}{\partial N}\right|\right\}\right).$$

Suppose without loss of generality that we require

$$P(1,n) \leq \frac{1}{10} \quad \text{and} \quad P(2,n) \geq \frac{9}{10}$$

(that is, algorithm A distinguishes 1-to-1 from 2-to-1 functions with error probability at most 1/10). Then, since

$$|P(g,N) - q(g,N)| < 0.182$$

by elementary calculus we have

$$d(q) \geq \max_{1\leq g\leq 2}\frac{\partial q}{\partial g} > 0.8 - 2(0.182) = 0.436.$$

An inequality due to Markov (see [82, 184]) states that, for a univariate polynomial p, if $b_1 \leq p(x) \leq b_2$ for all $a_1 \leq x \leq a_2$, then

$$\max_{a[1]\leq x\leq a[2]}\left|\frac{dp(x)}{dx}\right| \leq \frac{b_2-b_1}{a_2-a_1}\deg(p)^2.$$

Clearly for every point $\left(\widehat{g}, \widehat{N}\right) \in R$, there exists a quasilattice point (g, N) for which

$$|g - \widehat{g}| \leq 1 \text{ and } \left|N - \widehat{N}\right| \leq G.$$

For take $g = \lceil \widehat{g} \rceil$—or, in the special case $\widehat{g} = 1$, take $g = 2$, since there is only one quasilattice point with $g = 1$. Furthermore, since $P(g, N)$ represents an acceptance probability at such a point, we have

$$-0.182 < q(g, N) < 1.182.$$

Observe that for all $\left(\widehat{g}, \widehat{N}\right) \in R$,

$$-0.182 - \left(\frac{10TG(G-1)}{n} + 1\right) d(q) < q\left(\widehat{g}, \widehat{N}\right) < 1.182 + \left(\frac{10TG(G-1)}{n} + 1\right) d(q).$$

For consider a quasilattice point close to $\left(\widehat{g}, \widehat{N}\right)$, and note that the maximum-magnitude derivative is at most $d(q)$ in the g direction and $10T(G-1)d(q)/n$ in the N direction.

Let (g^*, N^*) be a point in R at which the weighted maximum-magnitude derivative $d(q)$ is attained. Suppose first that the maximum is attained in the g direction. Then $q(g, N^*)$ (with N^* constant) is a univariate polynomial with

$$\left|\frac{dq(g, N^*)}{dg}\right| > 0.436$$

for some $1 \leq g \leq G$. So

$$\begin{aligned} 2T &\geq \deg\left(q(g, N^*)\right) \\ &\geq \sqrt{\frac{d(q)(G-1)}{1.364 + 2d(q)(1 + 10TG(G-1)/n)}} \\ &= \Omega\left(\min\left\{\sqrt{G}, \sqrt{\frac{n}{TG}}\right\}\right). \end{aligned}$$

Similarly, suppose the maximum $d(q)$ is attained in the N direction. Then $q(g^*, N)$ (with g^* constant) is a univariate polynomial with

$$\left|\frac{dq(g^*, N)}{dN}\right| > \frac{0.436T(G-1)}{n}$$

for some $n \leq N \leq n + n/(10T)$. So

$$\begin{aligned} 2T &\geq \sqrt{\frac{(10T(G-1)/n)\, d(q)\, n/(10T)}{1.364 + 2d(q)(1 + 10TG(G-1)/n)}} \\ &\geq \Omega\left(\min\left\{\sqrt{G}, \sqrt{\frac{n}{TG}}\right\}\right). \end{aligned}$$

One can show that the lower bound on T is optimized when we take $G = n^{2/5} \leq \sqrt{n}$. Then

$$T = \Omega\left(\min\left\{n^{1/5}, \frac{\sqrt{n}}{\sqrt{T}n^{1/5}}\right\}\right),$$
$$T = \Omega\left(n^{1/5}\right)$$

and we are done. ■

6.5 Set Comparison

Here I sketch a proof that $\mathrm{Q}_2\left(\mathrm{SetComp}_n\right) = \Omega\left(n^{1/7}\right)$, where $\mathrm{SetComp}_n$ is the set comparison problem of size n as defined in Section 6.1.2.

The idea is the following. We need a distribution of inputs with a parameter g, such that the inputs are one-to-one when $g = 1$ or $g = 2$—since otherwise the problem of distinguishing $g = 1$ from $g = 2$ would be ill-defined for erasing oracles. On the other hand, the inputs must *not* be one-to-one for all $g > 2$—since otherwise the lower bound for standard oracles would apply also to erasing oracles, and we would not obtain a separation between the two. Finally, the acceptance probability must be close to a polynomial in g.

The solution is to consider $\kappa(g)$-to-one inputs, where

$$\kappa(g) = 4g^2 - 12g + 9.$$

is a quadratic with $\kappa(1) = \kappa(2) = 1$. The total range of the inputs (on sequences X and Y combined) has size roughly n/g; thus, we can tell the $g = 1$ inputs apart from the $g = 2$ inputs using an erasing oracle, even though $\kappa(g)$ is the same for both. The disadvantage is that, because $\kappa(g)$ increases quadratically rather than linearly in g, the quasilattice points become sparse more quickly. That is what weakens the lower bound from $\Omega\left(n^{1/5}\right)$ to $\Omega\left(n^{1/7}\right)$. Note that, using the ideas of Shi [218], one can improve my lower bound on $\mathrm{Q}_2\left(\mathrm{SetComp}_n\right)$ to $\Omega\left(n^{1/6}\right)$.

Call $(g, N, M) \in \Re^3$ an (n, T)-*super-quasilattice point* if and only if

(1) g is an integer in $\left[1, n^{1/3}\right]$,

(2) N and M are integers in $[n, n(1 + 1/(100T))]$,

(3) g divides N,

(4) if $g = 1$ then $N = n$,

(5) $\kappa(g)$ divides M, and

(6) if $g = 2$ then $M = n$.

For super-quasilattice point (g, N, M), we draw input $(X, Y) = (x_1 \ldots x_n, y_1 \ldots y_n)$ from distribution $\mathcal{L}_n(g, N, M)$ as follows. We first choose a set $S \subseteq \{1, \ldots, 2n\}$ with $|S| = 2N/g \leq 2n$ uniformly at random. We then choose two sets $S_X, S_Y \subseteq S$ with

$|S_X| = |S_X| = M/\kappa(g) \leq |S|$, uniformly at random and independently. Next we choose $\kappa(g)$-1 functions $\widehat{X} = \widehat{x}_1 \ldots \widehat{x}_N : \{1, \ldots, M\} \to S_X$ and $\widehat{Y} = \widehat{y}_1 \ldots \widehat{y}_N : \{1, \ldots, M\} \to S_Y$ uniformly at random and independently. Finally we let $x_i = \widehat{x}_i$ and $y_i = \widehat{y}_i$ for each $1 \leq i \leq n$.

Define sets $X_S = \{x_1, \ldots, x_n\}$ and $Y_S = \{y_1, \ldots, y_n\}$. Suppose $g = 1$ and $N = M = n$; then by Chernoff bounds,

$$\Pr_{(X,Y) \in \mathcal{L}_n(1,n,n)} [|X_S \cup Y_S| < 1.1n] \leq 2e^{-n/10}.$$

Thus, if algorithm A can distinguish $|X_S \cup Y_S| = n$ from $|X_S \cup Y_S| \geq 1.1n$ with probability at least $9/10$, then it can distinguish $(X, Y) \in \mathcal{L}_n(1, n, n)$ from $(X, Y) \in \mathcal{L}_n(2, n, n)$ with probability at least $9/10 - 2e^{-n/10}$. So a lower bound for the latter problem implies an equivalent lower bound for the former.

Define $P(X, Y)$ to be the probability that the algorithm returns that X and Y are far on input (X, Y), and let

$$P(g, N, M) = \operatorname*{EX}_{(X,Y) \in \mathcal{L}_n(g,N,M)} [P(X, Y)].$$

We then have

Lemma 4 *For all sufficiently large n and if $T \leq n^{1/3}/8$, there exists a trivariate polynomial $q(g, N, M)$ of degree at most $8T$ such that if (g, N, M) is a super-quasilattice point, then*

$$|P(g, N, M) - q(g, N, M)| < \varepsilon$$

for some constant $0 < \varepsilon < 1/2$.

Proof Sketch. By analogy to Lemma 1, $P(X, Y)$ is a multilinear polynomial of degree at most $2T$ over variables of the form $\Delta(x_i, h)$ and $\Delta(y_i, h)$. Let $I(X, Y) = I_X(X) I_Y(Y)$ where I_X is a product of $r_X(I)$ distinct $\Delta(x_i, h)$ variables and I_Y is a product of $r_Y(I)$ distinct $\Delta(y_i, h)$ variables. Let $r(I) = r_X(I) + r_Y(I)$. Define

$$\gamma(I, g, N, M) = \operatorname*{EX}_{(X,Y) \in \mathcal{L}_n(g,N,M)} [I(X, Y)];$$

then

$$P(g, N, M) = \sum_{I : r(I) \leq 2T} \beta_I \gamma(I, g, N, M)$$

for some coefficients β_I. We now calculate $\gamma(I, g, N, M)$. As before we assume there are no pairs of variables $\Delta(x_i, h_1), \Delta(x_i, h_2) \in I$ with $h_1 \neq h_2$. Let $Z_X(I)$ be the range of I_X and let $Z_Y(I)$ be the range of I_Y. Then let $Z(I) = Z_X(I) \cup Z_Y(I)$. Let $w_X(I) = |Z_X(I)|$, $w_Y(I) = |Z_Y(I)|$, and $w(I) = |Z(I)|$. By assumption

$$\frac{N}{g} \geq \frac{M}{\kappa(g)} \geq \frac{1}{4} n^{1/3} \geq 2T$$

so

$$\Pr\left[Z\left(I\right)\subseteq S\right]=\frac{\binom{2n-w\left(I\right)}{2N/g-w\left(I\right)}}{\binom{2n}{2N/g}}.$$

The probabilities that $Z_X\left(I\right)\subseteq S_X$ given $Z\left(I\right)\subseteq S$ and $Z_Y\left(I\right)\subseteq S_Y$ given $Z\left(I\right)\subseteq S$ can be calculated similarly.

Let $r_{X,1}\left(I\right),\ldots,r_{X,w[X](I)}\left(I\right)$ be the multiplicities of the range elements in $Z_X\left(I\right)$, so that

$$r_{X,1}\left(I\right)+\cdots+r_{X,w[X](I)}\left(I\right)=r_X\left(I\right).$$

Then

$$\Pr\left[I_X\left(X\right)\mid Z_X\left(I\right)\subseteq S_X\right]=\frac{\left(M-r_X\left(I\right)\right)!}{M!}\prod_{i=1}^{w[X](I)r[X,i](I)-1}\prod_{j=0}\left(\kappa\left(g\right)-j\right)$$

and similarly for $\Pr\left[I_Y\left(Y\right)\mid Z_Y\left(I\right)\subseteq S_Y\right]$.

Putting it all together and manipulating, we obtain (analogously to Lemma 1) that

$$\gamma\left(I,g,N,M\right)\approx\widetilde{q}_{n,T,I}\left(g,N,M\right)$$

where $\widetilde{q}_{n,T,I}\left(g,N,M\right)$ is a trivariate polynomial in $\left(g,N,M\right)$ of total degree at most $8T$. Thus

$$P\left(g,N,M\right)\approx q\left(g,N,M\right)$$

where $q\left(g,N,M\right)$ is a polynomial of total degree at most $8T$. The argument that q approximates P to within a constant is analogous to that of Lemma 2. ■

The remainder of the lower bound argument follows the lines of Theorem 3.

Theorem 5 $\mathrm{Q}_2\left(\mathrm{SetComp}_n\right)=\Omega\left(n^{1/7}\right)$.

Proof Sketch. Let $g\in[1,G]$ for some $G\leq n^{1/3}$. Then the super-quasilattice points $\left(g,N,M\right)$ all lie in $R=[1,G]\times\left[n,n+n/\left(100T\right)\right]^2$. Define $d(q)$ to be

$$\max_{(g,N,M)\in R}\left(\max\left\{\left|\frac{\partial q}{\partial g}\right|,\frac{n/100T}{\left(G-1\right)}\left|\frac{\partial q}{\partial N}\right|,\frac{n/100T}{\left(G-1\right)}\left|\frac{\partial q}{\partial M}\right|\right\}\right).$$

Then $d\left(q\right)\geq\delta$ for some constant $\delta>0$, by Lemma 4.

For every point $\left(\widehat{g},\widehat{N},\widehat{M}\right)\in R$, there exists a super-quasilattice point $\left(g,N,M\right)$ such that $\left|g-\widehat{g}\right|\leq1$, $\left|N-\widehat{N}\right|\leq G$, and $\left|M-\widehat{M}\right|\leq\kappa\left(G\right)$. Hence, $q\left(\widehat{g},\widehat{N},\widehat{M}\right)$ can deviate from $[0,1]$ by at most

$$O\left(\left(\frac{TG^3}{n}+1\right)d\left(q\right)\right).$$

Let (g^*, N^*, M^*) be a point in R at which $d(q)$ is attained. Suppose $d(q)$ is attained in the g direction; the cases of the N and M directions are analogous. Then $q(g, N^*, M^*)$ is a univariate polynomial in g, and

$$\begin{aligned} 8T &\geq \deg\left(q\left(g, N^*, M^*\right)\right) \\ &= \Omega\left(\min\left\{\sqrt{G}, \sqrt{\frac{n}{TG^2}}\right\}\right). \end{aligned}$$

One can show that the bound is optimized when we take $G = n^{2/7} \leq n^{1/3}$. Then

$$\begin{aligned} T &= \Omega\left(\min\left\{n^{1/7}, \frac{\sqrt{n}}{\sqrt{T}n^{2/7}}\right\}\right), \\ T &= \Omega\left(n^{1/7}\right). \end{aligned}$$

■

6.6 Open Problems

In my original paper on the collision problem, I listed four open problems: improving the collision lower bound to $\Omega\left(n^{1/3}\right)$; showing any nontrivial quantum lower bound for the set equality problem; proving a time-space tradeoff lower bound for the collision problem; and deciding whether quantum query complexity and degree as a real polynomial are always asymptotically equal. Happily, three of these problems have since been resolved [161, 176, 28], but the time-space tradeoff remains wide open. We would like to say (for example) that if a quantum computer is restricted to using $O(\log n)$ qubits, then it needs $\Theta(\sqrt{n})$ queries for the collision problem, ordinary Grover search being the best possible algorithm. Currently, we cannot show such a result for *any* problem with Boolean output, only for problems such as sorting with a large non-Boolean output [156].

Another problem is to give an oracle relative to which $\mathsf{SZK} \not\subset \mathsf{QMA}$, where QMA is Quantum Merlin Arthur as defined in [239]. In other words, show that if a function is one-to-one rather than two-to-one, then this fact cannot be verified using a small number of quantum queries, even with the help of a succinct quantum proof.

Finally, is it the case that for all (partial or total) functions f that are invariant under permutation symmetry, $\mathrm{R}_2(f)$ and $\mathrm{Q}_2(f)$ are polynomially related?

Chapter 7

Local Search

This chapter deals with the following problem.

LOCAL SEARCH. *Given an undirected graph $G = (V, E)$ and function $f : V \to \mathbb{N}$, find a local minimum of f—that is, a vertex v such that $f(v) \leq f(w)$ for all neighbors w of v.*

We will be interested in the number of queries that an algorithm needs to solve this problem, where a query just returns $f(v)$ given v. We will consider deterministic, randomized, and quantum algorithms. Section 7.1 motivates the problem theoretically and practically; this section explains the results.

First, though, we need some simple observations. If G is the complete graph of size N, then clearly $\Omega(N)$ queries are needed to find a local minimum (or $\Omega\left(\sqrt{N}\right)$ with a quantum computer). At the other extreme, if G is a line of length N, then even a deterministic algorithm can find a local minimum in $O(\log N)$ queries, using binary search: query the middle two vertices, v and w. If $f(v) \leq f(w)$, then search the line of length $(N-2)/2$ connected to v; otherwise search the line connected to w. Continue recursively in this manner until a local minimum is found.

So the interesting case is when G is a graph of 'intermediate' connectedness: for example, the Boolean hypercube $\{0,1\}^n$, with two vertices adjacent if and only if they have Hamming distance 1. For this graph, Llewellyn, Tovey, and Trick [169] showed a $\Omega(2^n/\sqrt{n})$ lower bound on the number of queries needed by any deterministic algorithm, using a simple adversary argument. Intuitively, until the set of vertices queried so far comprises a *vertex cut* (that is, splits the graph into two or more connected components), an adversary is free to return a descending sequence of f-values: $f(v_1) = 2^n$ for the first vertex v_1 queried by the algorithm, $f(v_2) = 2^n - 1$ for the second vertex queried, and so on. Moreover, once the set of queried vertices does comprise a cut, the adversary can choose the largest connected component of unqueried vertices, and restrict the problem recursively to that component. So to lower-bound the deterministic query complexity, it suffices to lower-bound the size of any cut that splits the graph into two reasonably large components.[1] For the Boolean hypercube, Llewellyn et al. showed that the best one can do is essentially to query all $\Omega(2^n/\sqrt{n})$ vertices of Hamming weight $n/2$.

[1] Llewellyn et al. actually give a tight characterization of deterministic query complexity in terms of vertex cuts.

Llewellyn et al.'s argument fails completely in the case of randomized algorithms. By Yao's minimax principle, what we want here is a fixed *distribution* $\mathcal{D}$ over functions $f : \{0,1\}^n \rightarrow \mathbb{N}$, such that any deterministic algorithm needs many queries to find a local minimum of f, with high probability if f is drawn from $\mathcal{D}$. Taking $\mathcal{D}$ to be uniform will not do, since a local minimum of a uniform random function is easily found. However, Aldous [24] had the idea of defining $\mathcal{D}$ via a *random walk*, as follows. Choose a vertex $v_0 \in \{0,1\}^n$ uniformly at random; then perform an unbiased walk[2] $v_0, v_1, v_2, \ldots$ starting from v_0. For each vertex v, set $f(v)$ equal to the first hitting time of the walk at v—that is, $f(v) = \min\{t : v_t = v\}$. Clearly any f produced in this way has a unique local minimum at v_0, since for all $t > 0$, if vertex v_t is visited for the first time at step t then $f(v_t) > f(v_{t-1})$. Using sophisticated random walk analysis, Aldous managed to show a lower bound of $2^{n/2-o(n)}$ on the expected number of queries needed by any randomized algorithm to find v_0.[3] (As we will see in Section 7.2, this lower bound is close to tight.) Intuitively, since a random walk on the hypercube mixes in $O(n \log n)$ steps, an algorithm that has not queried a v with $f(v) < 2^{n/2}$ has almost no useful information about where the unique minimum v_0 is, so its next query will just be a "stab in the dark."

However, Aldous's result leaves several questions about Local Search unanswered. What if the graph G is a 3-D cube, on which a random walk does *not* mix very rapidly? Can we still lower-bound the randomized query complexity of finding a local minimum? More generally, what parameters of G make the problem hard or easy? Also, what is the quantum query complexity of Local Search?

This chapter presents a new approach to Local Search, which I believe points the way to a complete understanding of its complexity. The approach is based on Ambainis's quantum adversary method [27]. Surprisingly, the approach yields new and simpler lower bounds for the problem's *classical* randomized query complexity, in addition to quantum lower bounds. Thus, along with recent work by Kerenidis and de Wolf [151] and by Aharonov and Regev [22], the results of this chapter illustrate how quantum ideas can help to resolve classical open problems.

The results are as follows. For the Boolean hypercube $G = \{0,1\}^n$, I show that any quantum algorithm needs $\Omega\left(2^{n/4}/n\right)$ queries to find a local minimum on G, and any randomized algorithm needs $\Omega\left(2^{n/2}/n^2\right)$ queries (improving the $2^{n/2-o(n)}$ lower bound of Aldous [24]). The proofs are elementary and do not require random walk analysis. By comparison, the best known upper bounds are $O\left(2^{n/3}n^{1/6}\right)$ for a quantum algorithm and $O\left(2^{n/2}\sqrt{n}\right)$ for a randomized algorithm. If G is a d-dimensional grid of size $N^{1/d} \times \cdots \times N^{1/d}$, where $d \geq 3$ is a constant, then I show that any quantum algorithm needs $\Omega\left(\sqrt{N^{1/2-1/d}/\log N}\right)$ queries to find a local minimum on G, and any randomized algorithm needs $\Omega\left(N^{1/2-1/d}/\log N\right)$ queries. No nontrivial lower bounds (randomized or quantum) were previously known in this case.[4]

In a preprint discussing these results, I raised as my "most ambitious" conjecture that the deterministic and quantum query complexities of local search are polynomially related for *every* family of graphs. At the time, it was not even known whether deterministic

[2] Actually, Aldous used a continuous-time random walk, so the functions would be from $\{0,1\}^n$ to $\mathbb{R}$.

[3] Independently and much later, Droste et al. [99] showed the weaker bound $2^{g(n)}$ for any $g(n) = o(n)$.

[4] A lower bound on deterministic query complexity was known for such graphs [168].

and *randomized* query complexities were polynomially related, not even for simple examples such as the 2-dimensional square grid. Subsequently Santha and Szegedy [211] spectacularly resolved the conjecture, by showing that the quantum query complexity is always at least the 19^{th} root (!) of the deterministic complexity. On the other hand, in the specific case of the hypercube, my lower bound is close to tight; Santha and Szegedy's is not. Also, I give randomized lower bounds that are quadratically better than my quantum lower bounds; Santha and Szegedy give only quantum lower bounds.

In another recent development, Ambainis [25] has improved the $\Omega\left(2^{n/4}/n\right)$ quantum lower bound for local search on the hypercube to $2^{n/3}/n^{O(1)}$, using a hybrid argument. Note that Ambainis's lower bound matches the upper bound up to a polynomial factor.

The chapter is organized as follows. Section 7.1 motivates lower bounds on Local Search, pointing out connections to simulated annealing, quantum adiabatic algorithms, and the complexity class TFNP of total function problems. Section 7.2 defines notation and reviews basic facts about Local Search, including upper bounds. In Section 7.3 I give an intuitive explanation of Ambainis's quantum adversary method, then state and prove a classical analogue of Ambainis's main lower bound theorem. Section 7.4 introduces *snakes*, a construction by which I apply the two adversary methods to Local Search. I show there that to prove lower bounds for any graph G, it suffices to upper-bound a combinatorial parameter ε of a 'snake distribution' on G. Section 7.5 applies this framework to specific examples of graphs: the Boolean hypercube in Section 7.5.1, and the d-dimensional grid in Section 7.5.2.

7.1 Motivation

Local search is the most effective weapon ever devised against hard optimization problems. For many real applications, neither backtrack search, nor approximation algorithms, nor even Grover's algorithm can compare. Furthermore, along with quantum computing, local search (broadly defined) is one of the most interesting links between computer science and Nature. It is related to evolutionary biology via genetic algorithms, and to the physics of materials via simulated annealing. Thus it is both practically and scientifically important to understand its performance.

The conventional wisdom is that, although local search performs well in practice, its central (indeed defining) flaw is a tendency to get stuck at local optima. If this were correct, one corollary would be that the reason local search performs so well is that the problem it really solves—finding a local optimum—is intrinsically easy. It would thus be unnecessary to seek further explanations for its performance. Another corollary would be that, for *unimodal* functions (which have no local optima besides the global optimum), the global optimum would be easily found.

However, the conventional wisdom is false. The results of Llewellyn et al. [169] and Aldous [24] show that even if f is unimodal, any classical algorithm that treats f as a black box needs exponential time to find the global minimum of f in general. My results extend this conclusion to quantum algorithms. In my view, the practical upshot of these results is that they force us to confront the question: What is it about 'real-world' problems that makes it easy to find a local optimum? That is, why do exponentially long chains

of descending values, such as those used for lower bounds, almost never occur in practice, even in functions with large range sizes? One possibility is that the functions that occur in practice look "globally" like random functions, but I do not know whether that is true in any meaningful sense.

The results of this chapter are also relevant for physics. Many physical systems, including folding proteins and networks of springs and pulleys, can be understood as performing 'local search' through an energy landscape to reach a locally-minimal energy configuration. A key question is, how long will the system take to reach its ground state (that is, a globally-minimal configuration)? Of course, if there are local optima, the system might *never* reach its ground state, just as a rock in a mountain crevice does not roll to the bottom by going up first. But what if the energy landscape is unimodal? And moreover, what if the physical system is quantum? My results show that, for certain energy landscapes, even a quantum system would take exponential time to reach its ground state, regardless of what external Hamiltonian is applied to "drive" it. So in particular, the quantum adiabatic algorithm proposed by Farhi et al. [106], which can be seen as a quantum analogue of simulated annealing, needs exponential time to find a local minimum in the worst case.

Finally, this chapter's results have implications for so-called *total function problems* in complexity theory. Megiddo and Papadimitriou [174] defined a complexity class TFNP, consisting (informally) of those NP search problems for which a solution always exists. For example, we might be given a function $f : \{0,1\}^n \rightarrow \{0,1\}^{n-1}$ as a Boolean circuit, and asked to find any distinct x, y pair such that $f(x) = f(y)$. This particular problem belongs to a subclass of TFNP called PPP (Polynomial Pigeonhole Principle). Notice that no promise is involved: the combinatorial nature of the problem itself forces a solution to exist, even if we have no idea how to find it. In a recent talk, Papadimitriou [187] asked broadly whether such 'nonconstructive existence problems' might be good candidates for efficient quantum algorithms. In the case of PPP problems like the one above, the collision lower bound of Chapter 6 implies a negative answer in the black-box setting. For other subclasses of TFNP, such as PODN (Polynomial Odd-Degree Node), a quantum black-box lower bound follows easily from the optimality of Grover's search algorithm.

However, there is one important subclass of TFNP for which no quantum lower bound was previously known. This is PLS (Polynomial Local Search), defined by Johnson, Papadimitriou, and Yannakakis [149] as a class of optimization problems whose cost function f and neighborhood function η (that is, the set of neighbors of a given point) are both computable in polynomial time.[5] Given such a problem, the task is to output any local minimum of the cost function: that is, a v such that $f(v) \leq f(w)$ for all $w \in \eta(v)$. The lower bound of Llewellyn et al. [169] yields an oracle A relative to which $\mathsf{FP}^A \neq \mathsf{PLS}^A$, by a standard diagonalization argument along the lines of Baker, Gill, and Solovay [41]. Likewise, the lower bound of Aldous [24] yields an oracle relative to which $\mathsf{PLS} \not\subset \mathsf{FBPP}$, where FBPP is simply the function version of BPP. The results of this chapter yield the first oracle relative to which $\mathsf{PLS} \not\subset \mathsf{FBQP}$. In light of this oracle separation, I raise an

[5] Some authors require only the *minimum* neighbor of a given point to be computable in polynomial time, which does not seem like the "right" idealization to me. In any case, for lower bound purposes we always assume the algorithm knows the whole neighborhood structure in advance, and does not need to make queries to learn about it.

admittedly vague question: is there a nontrivial "combinatorial" subclass of TFNP that we can show *is* contained in FBQP?

7.2 Preliminaries

In the Local Search problem, we are given an undirected graph $G = (V, E)$ with $N = |V|$, and oracle access to a function $f : V \rightarrow \mathbb{N}$. The goal is to find any *local minimum* of f, defined as a vertex $v \in V$ such that $f(v) \leq f(w)$ for all neighbors w of v. Clearly such a local minimum exists. We want to find one using as few queries as possible, where a query returns $f(v)$ given v. Queries can be adaptive; that is, can depend on the outcomes of previous queries. We assume G is known in advance, so that only f needs to be queried. Since we care only about query complexity, not computation time, there is no difficulty in dealing with an infinite range for f—though for lower bound purposes, it will turn out that a range of size $O\left(\sqrt{|V|}\right)$ suffices. I do not know of any case where a range larger than this makes the Local Search problem harder, but I also do not know of a general reduction from large to small range.

The model of query algorithms is the standard one. Given a graph G, the deterministic query complexity of Local Search on G, which we denote $\mathrm{DLS}(G)$, is $\min_{\Gamma} \max_f T(\Gamma, f, G)$ where the minimum ranges over all deterministic algorithms Γ, the maximum ranges over all f, and $T(\Gamma, f, G)$ is the number of queries made to f by Γ before it halts and outputs a local minimum of f (or ∞ if Γ fails to do so). The randomized query complexity $\mathrm{RLS}(G)$ is defined similarly, except that now the algorithm has access to an infinite random string R, and must only output a local minimum with probability at least $2/3$ over R. For simplicity, one can assume that the number of queries T is the same for all R; clearly this assumption changes the complexity by at most a constant factor.

In the quantum model, an algorithm's state has the form $\sum_{v,z,s} \alpha_{v,z,s} |v, z, s\rangle$, where v is the label of a vertex in G, and z and s are strings representing the answer register and workspace respectively. The $\alpha_{v,z,s}$'s are complex amplitudes satisfying $\sum_{v,z,s} |\alpha_{v,z,s}|^2 = 1$. Starting from an arbitrary (fixed) initial state, the algorithm proceeds by an alternating sequence of *queries* and *algorithm steps*. A query maps each $|v, z, s\rangle$ to $|v, z \oplus f(v), s\rangle$, where $\oplus$ denotes bitwise exclusive-OR. An algorithm step multiplies the vector of $\alpha_{v,z,s}$'s by an arbitrary unitary matrix that does not depend on f. Letting $\mathcal{M}_f$ denote the set of local minima of f, the algorithm succeeds if at the end $\sum_{v,z,s \,:\, v \in \mathcal{M}_f} |\alpha_{v,z,s}|^2 \geq \frac{2}{3}$. Then the bounded-error quantum query complexity, or $\mathrm{QLS}(G)$, is defined as the minimum number of queries used by a quantum algorithm that succeeds on every f.

It is immediate that $\mathrm{QLS}(G) \leq \mathrm{RLS}(G) \leq \mathrm{DLS}(G) \leq N$. Also, letting δ be the maximum degree of G, we have the following trivial lower bound.

Proposition 6 *$\mathrm{RLS}(G) = \Omega(\delta)$ and $\mathrm{QLS}(G) = \Omega\left(\sqrt{\delta}\right)$.*

Proof. Let v be a vertex of G with degree δ. Choose a neighbor w of v uniformly at random, and let $f(w) = 1$. Let $f(v) = 2$, and $f(u) = 3$ for all neighbors u of v other than w. Let S be the neighbor set of v (including v itself); then for all $x \notin S$, let $f(x) = 3 + \Delta(x, S)$ where $\Delta(x, S)$ is the minimum distance from x to a vertex in S.

Clearly f has a unique local minimum at w. However, finding y requires exhaustive search among the δ neighbors of v, which takes $\Omega\left(\sqrt{\delta}\right)$ quantum queries by Bennett et al. [51]. ■

A corollary of Proposition 6 is that classically, zero-error randomized query complexity is equivalent to bounded-error up to a constant factor. For given a candidate local minimum v, one can check using $O(\delta)$ queries that v is indeed a local minimum. Since $\Omega(\delta)$ queries are needed anyway, this verification step does not affect the overall complexity.

As pointed out by Aldous [24], a classical randomized algorithm can find a local minimum of f with high probability in $O\left(\sqrt{N\delta}\right)$ queries. The algorithm just queries $\sqrt{N\delta}$ vertices uniformly at random, and lets v_0 be a queried vertex for which $f(v)$ is minimal. It then follows v_0 to a local minimum by steepest descent. That is, for $t = 0, 1, 2, \ldots$, it queries all neighbors of v_t, halts if v_t is a local minimum, and otherwise sets v_{t+1} to be the neighbor w of v_t for which $f(w)$ is minimal (breaking ties by lexicographic ordering). A similar idea yields an improved quantum upper bound.

Proposition 7 *For any G, $\mathrm{QLS}(G) = O\left(N^{1/3}\delta^{1/6}\right)$.*

Proof. The algorithm first chooses $N^{2/3}\delta^{1/3}$ vertices of G uniformly at random, then uses Grover search to find a chosen vertex v_0 for which $f(v)$ is minimal. By a result of Dürr and Høyer [102], this can be done with high probability in $O\left(N^{1/3}\delta^{1/6}\right)$ queries. Next, for $t = 0, 1, 2, \ldots$, the algorithm performs Grover search over all neighbors of v_t, looking for a neighbor w such that $f(w) < f(v_t)$. If it finds such a w, then it sets $v_{t+1} := w$ and continues to the next iteration. Otherwise, it repeats the Grover search $\log(N/\delta)$ times before finally giving up and returning v_t as a claimed local minimum.

The expected number of u such that $f(u) < f(v_0)$ is at most $N/\left(N^{2/3}\delta^{1/3}\right) = (N/\delta)^{1/3}$. Since $f(v_{t+1}) < f(v_t)$ for all t, clearly the number of such u provides an upper bound on t. Furthermore, assuming there exists a w such that $f(w) < f(v_t)$, the expected number of repetitions of Grover's algorithm until such a w is found is $O(1)$. Since each repetition takes $O\left(\sqrt{\delta}\right)$ queries, by linearity of expectation the total expected number of queries used by the algorithm is therefore

$$O\left(N^{1/3}\delta^{1/6} + (N/\delta)^{1/3}\sqrt{\delta} + \log(N/\delta)\sqrt{\delta}\right)$$

or $O\left(N^{1/3}\delta^{1/6}\right)$. To see that the algorithm finds a local minimum with high probability, observe that for each t, the probability of not finding a w such that $f(w) < f(v_t)$, given that one exists, is at most $c^{-\log(N/\delta)} \leq (\delta/N)^{1/3}/10$ for a suitable constant c. So by the union bound, the probability that the algorithm returns a 'false positive' is at most $(N/\delta)^{1/3} \cdot (\delta/N)^{1/3}/10 = 1/10$. ■

7.3 Relational Adversary Method

There are essentially two known methods for proving lower bounds on quantum query complexity: the polynomial method of Beals et al. [45], and the quantum adversary method

of Ambainis [27].[6] For a few problems, such as the collision problem [2, 218], the polynomial method succeeded where the adversary method failed. However, for problems that lack permutation symmetry (such as Local Search), the adversary method has proven more effective.[7]

How could a quantum lower bound method possibly be applied classically? When proving randomized lower bounds, the tendency is to attack "bare-handed": fix a distribution over inputs, and let $x_1, \ldots, x_t$ be the locations queried so far by the algorithm. Show that for small t, the posterior distribution over inputs, *conditioned* on $x_1, \ldots, x_t$, is still 'hard' with high probability—so that the algorithm knows almost nothing even about which location x_{t+1} to query next. This is essentially the approach taken by Aldous [24] to prove a $2^{n/2-o(n)}$ lower bound on $\mathrm{RLS}\left(\{0,1\}^n\right)$.

In the quantum case, however, it is unclear how to specify what an algorithm 'knows' after a given number of queries. So we are almost *forced* to step back, and identify general combinatorial properties of input sets that make them hard to distinguish. Once we have such properties, we can then try to exhibit them in functions of interest.

We will see, somewhat surprisingly, that this "gloved" approach is useful for classical lower bounds as well as quantum ones. In the *relational adversary method*, we assume there exists a T-query randomized algorithm for function F. We consider a set $\mathcal{A}$ of 0-inputs of F, a set $\mathcal{B}$ of 1-inputs, and an arbitrary real-valued *relation function* $R(A,B) \geq 0$ for $A \in \mathcal{A}$ and $B \in \mathcal{B}$. Intuitively, $R(A,B)$ should be large if A and B differ in only a few locations. We then fix a probability distribution $\mathcal{D}$ over inputs; by Yao's minimax principle, there exists a T-query deterministic algorithm Γ^* that succeeds with high probability on inputs drawn from $\mathcal{D}$. Let W_A be the set of 0-inputs and W_B the set of 1-inputs on which Γ^* succeeds. Using the relation function R, we define a *separation measure* S between W_A and W_B, and show that (1) initially $S = 0$, (2) by the end of the computation S must be large, and (3) S increases by only a small amount as the result of each query. It follows that T must be large.

The advantage of the relational method is that converts a "dynamic" opponent—an algorithm that queries adaptively—into a relatively static one. It thereby makes it easier to focus on what is unique about a problem, aspects of query complexity that are common to all problems having been handled automatically. Furthermore, one does not need to know anything about quantum computing to understand and apply the method. On the other hand, I have no idea how one would come up with it in the first place, without Ambainis's *quantum* adversary method [27] and the reasoning about entanglement that led to it.

The starting point is the "most general" adversary theorem in Ambainis's original paper (Theorem 6 in [27]), which he introduced to prove a quantum lower bound for the problem of inverting a permutation. Here the input is a permutation $\sigma(1), \ldots, \sigma(N)$, and the task is to output 0 if $\sigma^{-1}(1) \leq N/2$ and 1 otherwise. To lower-bound this problem's query complexity, what we would like to say is this:

Given any 0*-input* σ *and any location* x*, if we choose a random* 1*-input* τ *that is*

[6] I am thinking here of the hybrid method [51] as a cousin of the adversary method.

[7] Indeed, Ambainis [28] has given problems for which the adversary method provably yields a better lower bound than the polynomial method.

'related' to σ, then the probability $\theta(\sigma,x)$ over τ that $\sigma(x)$ does not equal $\tau(x)$ is small. In other words, the algorithm is unlikely to distinguish σ from a random neighbor τ of σ by querying x.

Unfortunately, the above claim is false. Letting $x=\sigma^{-1}(1)$, we have that $\sigma(x)\neq\tau(x)$ for *every* 1-input τ, and thus $\theta(\sigma,x)=1$. Ambainis resolves this difficulty by letting us take the maximum, over all 0-inputs σ and 1-inputs τ that are related and differ at x, of the *geometric mean* $\sqrt{\theta(\sigma,x)\,\theta(\tau,x)}$. Even if $\theta(\sigma,x)=1$, the geometric mean is still small provided that $\theta(\tau,x)$ is small. More formally:

Theorem 8 (Ambainis) *Let $\mathcal{A}\subseteq F^{-1}(0)$ and $\mathcal{B}\subseteq F^{-1}(1)$ be sets of inputs to function F. Let $R(A,B)\geq 0$ be a symmetric real-valued function, and for $A\in\mathcal{A}$, $B\in\mathcal{B}$, and location x, let*

$$\theta(A,x)=\frac{\sum_{B^*\in\mathcal{B}\,:\,A(x)\neq B^*(x)}R(A,B^*)}{\sum_{B^*\in\mathcal{B}}R(A,B^*)},$$
$$\theta(B,x)=\frac{\sum_{A^*\in\mathcal{A}\,:\,A^*(x)\neq B(x)}R(A^*,B)}{\sum_{A^*\in\mathcal{A}}R(A^*,B)},$$

where the denominators are all nonzero. Then the number of quantum queries needed to evaluate F with at least $9/10$ probability is $\Omega(1/\upsilon_{\text{geom}})$, where

$$\upsilon_{\text{geom}}=\max_{A\in\mathcal{A},\ B\in\mathcal{B},\ x\,:\,R(A,B)>0,\ A(x)\neq B(x)}\sqrt{\theta(A,x)\,\theta(B,x)}.$$

The best way to understand Theorem 8 is to see it used in an example.

Proposition 9 (Ambainis) *The quantum query complexity of inverting a permutation is $\Omega\left(\sqrt{N}\right)$.*

Proof. Let $\mathcal{A}$ be the set of all permutations σ such that $\sigma^{-1}(1)\leq N/2$, and $\mathcal{B}$ be the set of permutations τ such that $\tau^{-1}(1)>N/2$. Given $\sigma\in\mathcal{A}$ and $\tau\in\mathcal{B}$, let $R(\sigma,\tau)=1$ if σ and τ differ only at locations $\sigma^{-1}(1)$ and $\tau^{-1}(1)$, and $R(\sigma,\tau)=0$ otherwise. Then given σ,τ with $R(\sigma,\tau)=1$, if $x\neq\sigma^{-1}(1)$ then $\theta(\sigma,x)=2/N$, and if $x\neq\tau^{-1}(1)$ then $\theta(\tau,x)=2/N$. So $\max_{x\,:\,\sigma(x)\neq\tau(x)}\sqrt{\theta(\sigma,x)\,\theta(\tau,x)}=\sqrt{2/N}$. ■

The only difference between Theorem 8 and my relational adversary theorem is that in the latter, we take the *minimum* of $\theta(A,x)$ and $\theta(B,x)$ instead of the geometric mean. Taking the reciprocal then gives up to a quadratically better lower bound: for example, we obtain that the randomized query complexity of inverting a permutation is $\Omega(N)$. However, the proofs of the two theorems are quite different.

Theorem 10 *Let $\mathcal{A},\mathcal{B},R,\theta$ be as in Theorem 8. Then the number of randomized queries needed to evaluate F with at least $9/10$ probability is $\Omega(1/\upsilon_{\min})$, where*

$$\upsilon_{\min}=\max_{A\in\mathcal{A},\ B\in\mathcal{B},\ x\,:\,R(A,B)>0,\ A(x)\neq B(x)}\min\left\{\theta(A,x),\theta(B,x)\right\}.$$

Proof. Let Γ be a randomized algorithm that, given an input A, returns $F(A)$ with at least 9/10 probability. Let T be the number of queries made by Γ. For all $A \in \mathcal{A}$, $B \in \mathcal{B}$, define

$$M(A) = \sum_{B^* \in \mathcal{B}} R(A, B^*),$$
$$M(B) = \sum_{A^* \in \mathcal{A}} R(A^*, B),$$
$$M = \sum_{A^* \in \mathcal{A}} M(A^*) = \sum_{B^* \in \mathcal{B}} M(B^*).$$

Now let $\mathcal{D}_A$ be the distribution over $A \in \mathcal{A}$ in which each A is chosen with probability $M(A)/M$; and let $\mathcal{D}_B$ be the distribution over $B \in \mathcal{B}$ in which each B is chosen with probability $M(B)/M$. Let $\mathcal{D}$ be an equal mixture of $\mathcal{D}_A$ and $\mathcal{D}_B$. By Yao's minimax principle, there exists a deterministic algorithm Γ^* that makes T queries, and succeeds with at least 9/10 probability given an input drawn from $\mathcal{D}$. Therefore Γ^* succeeds with at least 4/5 probability given an input drawn from $\mathcal{D}_A$ alone, or from $\mathcal{D}_B$ alone. In other words, letting W_A be the set of $A \in \mathcal{A}$ and W_B the set of $B \in \mathcal{B}$ on which Γ^* succeeds, we have

$$\sum_{A \in W_A} M(A) \geq \frac{4}{5}M, \quad \sum_{B \in W_B} M(B) \geq \frac{4}{5}M.$$

Define a predicate $P^{(t)}(A, B)$, which is true if Γ^* has distinguished $A \in \mathcal{A}$ from $B \in \mathcal{B}$ by the t^{th} query and false otherwise. (To distinguish A from B means to query an index x for which $A(x) \neq B(x)$, given either A or B as input.) Also, for all $A \in \mathcal{A}$, define a score function

$$S^{(t)}(A) = \sum_{B^* \in \mathcal{B} \,:\, P^{(t)}(A, B^*)} R(A, B^*).$$

This function measures how much "progress" has been made so far in separating A from $\mathcal{B}$-inputs, where the $\mathcal{B}$-inputs are weighted by $R(A, B)$. Similarly, for all $B \in \mathcal{B}$ define

$$S^{(t)}(B) = \sum_{A^* \in \mathcal{A} \,:\, P^{(t)}(A^*, B)} R(A^*, B).$$

It is clear that for all t,

$$\sum_{A \in \mathcal{A}} S^{(t)}(A) = \sum_{B \in \mathcal{B}} S^{(t)}(B).$$

So we can denote the above sum by $S^{(t)}$ and think of it as a global progress measure. The proof relies on the following claims about $S^{(t)}$:

(i) $S^{(0)} = 0$ initially.

(ii) $S^{(T)} \geq 3M/5$ by the end.

(iii) $\Delta S^{(t)} \leq 3 v_{\min} M$ for all t, where $\Delta S^{(t)} = S^{(t)} - S^{(t-1)}$ is the amount by which $S^{(t)}$ increases as the result of a single query.

It follows from (i)-(iii) that

$$T \geq \frac{3M/5}{3\upsilon_{\min} M} = \frac{1}{5\upsilon_{\min}}$$

which establishes the theorem. Part (i) is obvious. For part (ii), observe that for every pair (A, B) with $A \in W_A$ and $B \in W_B$, the algorithm Γ^* must query an x such that $A(x) \neq B(x)$. Thus

$$\begin{aligned} S^{(T)} &\geq \sum_{A \in W_A,\ B \in W_B} R(A, B) \\ &\geq \sum_{A \in W_A} M(A) - \sum_{B \notin W_B} M(B) \\ &\geq \frac{4}{5} M - \frac{1}{5} M. \end{aligned}$$

It remains only to show part (iii). Suppose $\Delta S^{(t)} > 3\upsilon_{\min} M$ for some t; we will obtain a contradiction. Let

$$\Delta S^{(t)}(A) = S^{(t)}(A) - S^{(t-1)}(A),$$

and let C_A be the set of $A \in \mathcal{A}$ for which $\Delta S^{(t)}(A) > \upsilon_{\min} M(A)$. Since

$$\sum_{A \in \mathcal{A}} \Delta S^{(t)}(A) = \Delta S^{(t)} > 3\upsilon_{\min} M,$$

it follows by Markov's inequality that

$$\sum_{A \in C_A} \Delta S^{(t)}(A) \geq \frac{2}{3} \Delta S^{(t)}.$$

Similarly, if we let C_B be the set of $B \in \mathcal{B}$ for which $\Delta S^{(t)}(B) > \upsilon_{\min} M(B)$, we have

$$\sum_{B \in C_B} \Delta S^{(t)}(B) \geq \frac{2}{3} \Delta S^{(t)}.$$

In other words, at least $2/3$ of the increase in $S^{(t)}$ comes from (A, B) pairs such that $A \in C_A$, and at least $2/3$ comes from (A, B) pairs such that $B \in C_B$. Hence, by a 'pigeonhole' argument, there exists an $A \in C_A$ and $B \in C_B$ with $R(A, B) > 0$ that are distinguished by the t^{th} query. In other words, there exists an x with $A(x) \neq B(x)$, such that the t^{th} index queried by Γ^* is x whether the input is A or B. Then since $A \in C_A$, we have $\upsilon_{\min} M(A) < \Delta S^{(t)}(A)$, and hence

$$\begin{aligned} \upsilon_{\min} &< \frac{\Delta S^{(t)}(A)}{M(A)} \\ &\leq \frac{\sum_{B^* \in \mathcal{B}\,:\, A(x) \neq B^*(x)} R(A, B^*)}{\sum_{B^* \in \mathcal{B}} R(A, B^*)} \end{aligned}$$

which equals $\theta(A, x)$. Similarly $\upsilon_{\min} < \theta(B, x)$ since $B \in C_B$. This contradicts the definition

$$\upsilon_{\min} = \max_{A \in \mathcal{A},\ B \in \mathcal{B},\ x\,:\ R(A,B) > 0,\ A(x) \neq B(x)} \min\left\{\theta(A, x), \theta(B, x)\right\},$$

and we are done. ■

7.4 Snakes

For the lower bounds, it will be convenient to generalize random walks to arbitrary distributions over paths, which we call *snakes*.

Definition 11 *Given a vertex h in G and a positive integer L, a snake distribution $\mathcal{D}_{h,L}$ (parameterized by h and L) is a probability distribution over paths $(x_0, \ldots, x_{L-1})$ in G, such that each x_t is either equal or adjacent to x_{t+1}, and $x_{L-1} = h$. Let $D_{h,L}$ be the support of $\mathcal{D}_{h,L}$. Then an element of $D_{h,L}$ is called a snake; the part near x_0 is the tail and the part near $x_{L-1} = h$ is the head.*

Given a snake X and integer t, we use $X[t]$ as shorthand for $\{x_0, \ldots, x_t\}$.

Definition 12 *We say a snake $X \in D_{h,L}$ is ε-good if the following holds. Choose j uniformly at random from $\{0, \ldots, L-1\}$, and let $Y = (y_0, \ldots, y_{L-1})$ be a snake drawn from $\mathcal{D}_{h,L}$ conditioned on $x_t = y_t$ for all $t > j$. Then*

(i) Letting $S_{X,Y}$ be the set of vertices v in $X \cap Y$ such that $\min\{t : x_t = v\} = \min\{t : y_t = v\}$, we have

$$\Pr_{j,Y}[X \cap Y = S_{X,Y}] \geq 9/10.$$

(ii) For all vertices v, $\Pr_{j,Y}[v \in Y[j]] \leq \varepsilon$.

The procedure above—wherein we choose a j uniformly at random, then draw a Y from $\mathcal{D}_{h,L}$ consistent with X on all steps later than j—will be important in what follows. I call it *the snake X flicking its tail*. Intuitively, a snake is good if it is spread out fairly evenly in G—so that when it flicks its tail, (1) with high probability the old and new tails do not intersect, and (2) any particular vertex is hit by the new tail with probability at most ε.

I now explain the 'snake method' for proving lower bounds for Local Search. Given a snake X, we define an input f_X with a unique local minimum at x_0, and f-values that decrease along X from head to tail. Then, given inputs f_X and f_Y with $X \cap Y = S_{X,Y}$, we let the relation function $R(f_X, f_Y)$ be proportional to the probability that snake Y is obtained by X flicking its tail. (If $X \cap Y \neq S_{X,Y}$ we let $R = 0$.) Let f_X and g_Y be inputs with $R(f_X, g_Y) > 0$, and let v be a vertex such that $f_X(v) \neq g_Y(v)$. Then if all snakes were good, there would be two mutually exclusive cases: (1) v belongs to the tail of X, or (2) v belongs to the tail of Y. In case (1), v is hit with small probability when Y flicks its tail, so $\theta(f_Y, v)$ is small. In case (2), v is hit with small probability when X flicks its tail, so $\theta(f_X, v)$ is small. In either case, then, the *geometric mean* $\sqrt{\theta(f_X, v)\,\theta(f_Y, v)}$ and *minimum* $\min\{\theta(f_X, v), \theta(f_Y, v)\}$ are small. So even though $\theta(f_X, v)$ or $\theta(f_Y, v)$ could be large individually, Theorems 8 and 10 yield a good lower bound, as in the case of inverting a permutation (see Figure 7.1).

One difficulty is that not all snakes are good; at best, a large fraction of them are. We could try deleting all inputs f_X such that X is not good, but that might ruin some remaining inputs, which would then have fewer neighbors. So we would have to delete

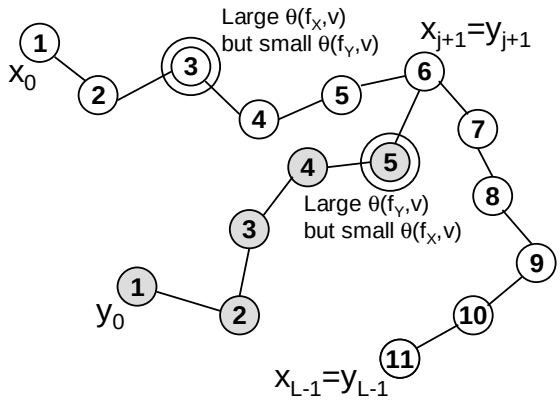

Figure 7.1: For every vertex v such that $f_X(v) \neq f_Y(v)$, either when snake X flicks its tail v is not hit with high probability, or when snake Y flicks its tail v is not hit with high probability.

those inputs as well, and so on ad infinitum. What we need is basically a way to replace "all inputs" by "most inputs" in Theorems 8 and 10.

Fortunately, a simple graph-theoretic lemma can accomplish this. The lemma (see Diestel [98, p.6] for example) says that any graph with average degree at least k contains an induced subgraph with *minimum* degree at least $k/2$. Below I prove a weighted analogue of the lemma.

Lemma 13 *Let $p(1), \ldots, p(m)$ be positive reals summing to 1. Also let $w(i,j)$ for $i,j \in \{1,\ldots,m\}$ be nonnegative reals satisfying $w(i,j) = w(j,i)$ and $\sum_{i,j} w(i,j) \geq r$. Then there exists a nonempty subset $U \subseteq \{1,\ldots,m\}$ such that for all $i \in U$, $\sum_{j \in U} w(i,j) \geq rp(i)/2$.*

Proof. If $r = 0$ then the lemma trivially holds, so assume $r > 0$. We construct U via an iterative procedure. Let $U(0) = \{1,\ldots,m\}$. Then for all t, if there exists an $i^* \in U(t)$ for which

$$\sum_{j \in U(t)} w(i^*, j) < \frac{r}{2} p(i^*),$$

then set $U(t+1) = U(t) \backslash \{i^*\}$. Otherwise halt and return $U = U(t)$. To see that the U so constructed is nonempty, observe that when we remove i^*, the sum $\sum_{i \in U(t)} p(i)$ decreases by $p(i^*)$, while $\sum_{i,j \in U(t)} w(i,j)$ decreases by at most

$$\sum_{j \in U(t)} w(i^*, j) + \sum_{j \in U(t)} w(j, i^*) < rp(i^*).$$

So since $\sum_{i,j \in U(t)} w(i,j)$ was positive to begin with, it must still be positive at the end of the procedure; hence U must be nonempty. ■

I can now prove the main result of the section.

Theorem 14 *Suppose a snake drawn from $\mathcal{D}_{h,L}$ is ε-good with probability at least 9/10. Then*

$$\mathrm{RLS}(G) = \Omega(1/\varepsilon), \qquad \mathrm{QLS}(G) = \Omega\left(\sqrt{1/\varepsilon}\right).$$

Proof. Given a snake $X \in D_{h,L}$, we construct an input function f_X as follows. For each $v \in X$, let $f_X(v) = \min\{t : x_t = v\}$; and for each $v \notin X$, let $f_X(v) = \Delta(v,h) + L$ where $\Delta(v,h)$ is the distance from v to h in G. Clearly f_X so defined has a unique local minimum at x_0. To obtain a decision problem, we stipulate that querying x_0 reveals an answer bit (0 or 1) in addition to $f_X(x_1)$; the algorithm's goal is then to return the answer bit. Obviously a lower bound for the decision problem implies a corresponding lower bound for the search problem. Let us first prove the theorem in the case that all snakes in $D_{h,L}$ are ε-good. Let $p(X)$ be the probability of drawing snake X from $\mathcal{D}_{h,L}$. Also, given snakes X, Y and $j \in \{0, \ldots, L-1\}$, let $q_j(X,Y)$ be the probability that $X^* = Y$, if X^* is drawn from $\mathcal{D}_{h,L}$ conditioned on agreeing with X on all steps later than j. Then define

$$w(X,Y) = \frac{p(X)}{L} \sum_{j=0}^{L-1} q_j(X,Y).$$

The first claim is that w is symmetric; that is, $w(X,Y) = w(Y,X)$. It suffices to show that

$$p(X)\, q_j(X,Y) = p(Y)\, q_j(Y,X)$$

for all j. We can assume X agrees with Y on all steps later than j, since otherwise $q_j(X,Y) = q_j(Y,X) = 0$. Given an $X^* \in D_{h,L}$, let A denote the event that X^* agrees with X (or equivalently Y) on all steps later than j, and let B_X (resp. B_Y) denote the event that X^* agrees with X (resp. Y) on steps 1 to j. Then

$$\begin{aligned} p(X)\, q_j(X,Y) &= \Pr[A] \Pr[B_X|A] \cdot \Pr[B_Y|A] \\ &= p(Y)\, q_j(Y,X). \end{aligned}$$

Now let $E(X,Y)$ denote the event that $X \cap Y = S_{X,Y}$, where $S_{X,Y}$ is as in Definition 12. Also, let f_X be the input obtained from X that has answer bit 0, and g_X be the input that has answer bit 1. To apply Theorems 8 and 10, take $\mathcal{A} = \{f_X : X \in D_{h,L}\}$ and $\mathcal{B} = \{g_X : X \in D_{h,L}\}$. Then take $R(f_X, g_Y) = w(X,Y)$ if $E(X,Y)$ holds, and $R(f_X, g_Y) = 0$ otherwise. Given $f_X \in \mathcal{A}$ and $g_Y \in \mathcal{B}$ with $R(f_X, g_Y) > 0$, and letting v be a vertex such that $f_X(v) \neq g_Y(v)$, we must then have either $v \notin X$ or $v \notin Y$. Suppose the former case; then

$$\sum_{f_{X^*} \in \mathcal{A} \,:\, f_{X^*}(v) \neq g_Y(v)} R(f_{X^*}, g_Y) \leq \sum_{f_{X^*} \in \mathcal{A} \,:\, f_{X^*}(v) \neq g_Y(v)} \frac{p(Y)}{L} \sum_{j=0}^{L-1} q_j(Y, X^*) \leq \varepsilon p(Y),$$

since Y is ε-good. Thus $\theta(g_Y, v)$ equals

$$\frac{\sum_{f_{X^*} \in \mathcal{A} \,:\, f_{X^*}(v) \neq g_Y(v)} R(f_{X^*}, g_Y)}{\sum_{f_{X^*} \in \mathcal{A}} R(f_{X^*}, g_Y)} \leq \frac{\varepsilon p(Y)}{9p(Y)/10}.$$

Similarly, if $v \notin Y$ then $\theta(f_X, v) \leq 10\varepsilon/9$ by symmetry. Hence

$$\upsilon_{\min} = \max_{f_X \in \mathcal{A},\ g_Y \in \mathcal{B},\ v\ :\ R(f_X, g_Y) > 0,\ f_X(v) \neq g_Y(v)} \min\{\theta(f_X, v), \theta(g_Y, v)\} \leq \frac{\varepsilon}{9/10},$$
$$\upsilon_{\text{geom}} = \max_{f_X \in \mathcal{A},\ g_Y \in \mathcal{B},\ v\ :\ R(f_X, g_Y) > 0,\ f_X(v) \neq g_Y(v)} \sqrt{\theta(f_X, v)\,\theta(g_Y, v)} \leq \sqrt{\frac{\varepsilon}{9/10}},$$

the latter since $\theta(f_X, v) \leq 1$ and $\theta(g_Y, v) \leq 1$ for all f_X, g_Y and v.

In the general case, all we know is that a snake drawn from $\mathcal{D}_{h,L}$ is ε-good with probability at least $9/10$. Let $G(X)$ denote the event that X is ε-good. Take $\mathcal{A}^* = \{f_X \in \mathcal{A} : G(X)\}$ and $\mathcal{B}^* = \{g_Y \in \mathcal{B} : G(Y)\}$, and take $R(f_X, g_Y)$ as before. Then since

$$\sum_{X,Y\ :\ E(X,Y)} w(X, Y) \geq \sum_X \frac{9}{10} p(X) \geq \frac{9}{10},$$

by the union bound we have

$$\begin{aligned}\sum_{f_X \in \mathcal{A}^*,\ g_Y \in \mathcal{B}^*} R(f_X, g_Y) &\geq \sum_{X,Y\ :\ G(X) \wedge G(Y) \wedge E(X,Y)} w(X, Y) - \sum_{X\ :\ \neg G(X)} p(X) - \sum_{Y\ :\ \neg G(Y)} p(Y) \\ &\geq \frac{9}{10} - \frac{1}{10} - \frac{1}{10} \\ &= \frac{7}{10}.\end{aligned}$$

So by Lemma 13, there exist subsets $\widetilde{\mathcal{A}} \subseteq \mathcal{A}^*$ and $\widetilde{\mathcal{B}} \subseteq \mathcal{B}^*$ such that for all $f_X \in \widetilde{\mathcal{A}}$ and $g_Y \in \widetilde{\mathcal{B}}$,

$$\sum_{g_{Y^*} \in \widetilde{\mathcal{B}}} R(f_X, g_{Y^*}) \geq \frac{7p(X)}{20},$$
$$\sum_{f_{X^*} \in \widetilde{\mathcal{A}}} R(f_{X^*}, g_Y) \geq \frac{7p(Y)}{20}.$$

So for all f_X, g_Y with $R(f_X, g_Y) > 0$, and all v such that $f_X(v) \neq g_Y(v)$, either $\theta(f_X, v) \leq 20\varepsilon/7$ or $\theta(g_Y, v) \leq 20\varepsilon/7$. Hence $\upsilon_{\min} \leq 20\varepsilon/7$ and $\upsilon_{\text{geom}} \leq \sqrt{20\varepsilon/7}$. ■

7.5 Specific Graphs

In this section I apply the 'snake method' developed in Section 7.4 to specific examples of graphs: the Boolean hypercube in Section 7.5.1, and the d-dimensional cubic grid (for $d \geq 3$) in Section 7.5.2.

7.5.1 Boolean Hypercube

Abusing notation, let $\{0,1\}^n$ denote the n-dimensional Boolean hypercube—that is, the graph whose vertices are n-bit strings, with two vertices adjacent if and only if they have

Hamming distance 1. Given a vertex $v \in \{0,1\}^n$, let $v[0], \ldots, v[n-1]$ denote the n bits of v, and let $v^{(i)}$ denote the neighbor obtained by flipping bit $v[i]$. In this section I lower-bound $\mathrm{RLS}(\{0,1\}^n)$ and $\mathrm{QLS}(\{0,1\}^n)$.

Fix a 'snake head' $h \in \{0,1\}^n$ and take $L = 2^{n/2}/100$. I define the snake distribution $\mathcal{D}_{h,L}$ via what I call a *coordinate loop*, as follows. Starting from $x_0 = h$, for each t take $x_{t+1} = x_t$ with 1/2 probability, and $x_{t+1} = x_t^{(t \bmod n)}$ with 1/2 probability. The following is a basic fact about this distribution.

Proposition 15 *The coordinate loop mixes completely in n steps, in the sense that if $t^* \geq t + n$, then x_{t^*} is a uniform random vertex conditioned on x_t.*

One could also use the random walk distribution, following Aldous [24]. However, not only is the coordinate loop distribution easier to work with (since it produces fewer self-intersections), it also yields a better lower bound (since it mixes completely in n steps, as opposed to approximately in $n \log n$ steps).

I first upper-bound the probability, over X, j, and $Y[j]$, that $X \cap Y \neq S_{X,Y}$ (where $S_{X,Y}$ is as in Definition 12).

Lemma 16 *Suppose X is drawn from $\mathcal{D}_{h,L}$, j is drawn uniformly from $\{0, \ldots, L-1\}$, and $Y[j]$ is drawn from $\mathcal{D}_{x_j,j}$. Then $\Pr_{X,j,Y[j]}[X \cap Y = S_{X,Y}] \geq 0.9999$.*

Proof. Call a *disagreement* a vertex v such that

$$\min\{t : x_t = v\} \neq \min\{t^* : y_{t^*} = v\}.$$

Clearly if there are no disagreements then $X \cap Y = S_{X,Y}$. If v is a disagreement, then by the definition of $\mathcal{D}_{h,L}$ we cannot have both $t > j - n$ and $t^* > j - n$. So by Proposition 15, either y_{t^*} is uniformly random conditioned on X, or x_t is uniformly random conditioned on $Y[j]$. Hence $\Pr_{X,j,Y[j]}[x_t = y_{t^*}] = 1/2^n$. So by the union bound,

$$\Pr_{X,j,Y[j]}[X \cap Y \neq S_{X,Y}] \leq \frac{L^2}{2^n} = 0.0001.$$

■

I now argue that, unless X spends a 'pathological' amount of time in one part of the hypercube, the probability of any vertex v being hit when X flicks its tail is small. To prove this, I define a notion of *sparseness*, and then show that (1) almost all snakes drawn from $\mathcal{D}_{h,L}$ are sparse (Lemma 18), and (2) sparse snakes are unlikely to hit any given vertex v (Lemma 19).

Definition 17 *Given vertices v, w and $i \in \{0, \ldots, n-1\}$, let $\Delta(x, v, i)$ be the number of steps needed to reach v from x by first setting $x[i] := v[i]$, then setting $x[i-1] := v[i-1]$, and so on. (After we set $x[0]$ we wrap around to $x[n-1]$.) Then X is sparse if there exists a constant c such that for all $v \in \{0,1\}^n$ and all k,*

$$|\{t : \Delta(x_t, v, t \bmod n) = k\}| \leq cn\left(n + \frac{L}{2^{n-k}}\right).$$

Lemma 18 *If X is drawn from $\mathcal{D}_{h,L}$, then X is sparse with probability $1-o(1)$.*

Proof. For each $i \in \{0,\ldots,n-1\}$, the number of $t \in \{0,\ldots,L-1\}$ such that $t \equiv i \pmod{n}$ is at most L/n. For such a t, let $E_t^{(v,i,k)}$ be the event that $\Delta(x_t, v, i) \leq k$; then $E_t^{(v,i,k)}$ holds if and only if

$$x_t[i] = v[i], \ldots, x_t[i-k+1] = v[i-k+1]$$

(where we wrap around to $x_t[n-1]$ after reaching $x_t[0]$). This occurs with probability $2^k/2^n$ over X. Furthermore, by Proposition 15, the $E_t^{(v,i,k)}$ events for different t's are independent. So let

$$\mu_k = \frac{L}{n} \cdot \frac{2^k}{2^n};$$

then for fixed v, i, k, the expected number of t's for which $E_t^{(v,i,k)}$ holds is at most μ_k. Thus by a Chernoff bound, if $\mu_k \geq 1$ then

$$\Pr_X\left[\left|\left\{t : E_t^{(v,i,k)}\right\}\right| > cn \cdot \mu_k\right] < \left(\frac{e^{cn-1}}{(cn)^{cn}}\right)^{\mu_k} < \frac{1}{2^{2n}}$$

for sufficiently large c. Similarly, if $\mu_k < 1$ then

$$\Pr_X\left[\left|\left\{t : E_t^{(v,i,k)}\right\}\right| > cn\right] < \left(\frac{e^{cn/\mu_k-1}}{(cn/\mu_k)^{cn/\mu_k}}\right)^{\mu_k} < \frac{1}{2^{2n}}$$

for sufficiently large c. By the union bound, then,

$$\begin{aligned}\left|\left\{t : E_t^{(v,i,k)}\right\}\right| &\leq cn \cdot (1+\mu_k) \\ &= c\left(n + \frac{L}{2^{n-k}}\right)\end{aligned}$$

for every v, i, k triple *simultaneously* with probability at least $1 - n^2 2^n/2^{2n} = 1 - o(1)$. Summing over all i's produces the additional factor of n. ■

Lemma 19 *If X is sparse, then for every $v \in \{0,1\}^n$,*

$$\Pr_{j,Y}[v \in Y[j]] = O\left(\frac{n^2}{L}\right).$$

Proof. By assumption, for every $k \in \{0,\ldots,n\}$,

$$\begin{aligned}\Pr_j[\Delta(x_j, v, j \bmod n) = k] &\leq \frac{|\{t : \Delta(x_t, v, t \bmod n) = k\}|}{L} \\ &\leq \frac{cn}{L}\left(n + \frac{L}{2^{n-k}}\right).\end{aligned}$$

Consider the probability that $v \in Y[j]$ in the event that $\Delta(x_j, v, j \bmod n) = k$. Clearly

$$\Pr_Y[v \in \{y_{j-n+1}, \ldots, y_j\}] = \frac{1}{2^k}.$$

Also, Proposition 15 implies that for every $t \leq j - n$, the probability that $y_t = v$ is 2^{-n}. So by the union bound,

$$\Pr_Y[v \in \{y_0, \ldots, y_{j-n}\}] \leq \frac{L}{2^n}.$$

Then $\Pr_{j,Y}[v \in Y[j]]$ equals

$$\sum_{k=0}^{n} \left(\begin{array}{c} \Pr_j[\Delta(x_j, v, j \bmod n) = k] \cdot \\ \Pr_Y[v \in Y[j] \mid \Delta(x_j, v, j \bmod n) = k] \end{array} \right) \leq \sum_{k=0}^{n} \frac{cn}{L} \left(n + \frac{L}{2^{n-k}}\right) \left(\frac{1}{2^k} + \frac{L}{2^n}\right)$$
$$= O\left(\frac{cn^2}{L}\right)$$

as can be verified by breaking the sum into cases and doing some manipulations. ■

The main result follows easily:

Theorem 20

$$\mathrm{RLS}(\{0,1\}^n) = \Omega\left(\frac{2^{n/2}}{n^2}\right), \quad \mathrm{QLS}(\{0,1\}^n) = \Omega\left(\frac{2^{n/4}}{n}\right).$$

Proof. Take $\varepsilon = n^2/2^{n/2}$. Then by Theorem 14, it suffices to show that a snake X drawn from $\mathcal{D}_{h,L}$ is $O(\varepsilon)$-good with probability at least 9/10. First, since

$$\Pr_{X,j,Y[j]}[X \cap Y = S_{X,Y}] \geq 0.9999$$

by Lemma 16, Markov's inequality shows that

$$\Pr_X\left[\Pr_{j,Y[j]}[X \cap Y = S_{X,Y}] \geq \frac{9}{10}\right] \geq \frac{19}{20}.$$

Second, by Lemma 18, X is sparse with probability $1 - o(1)$, and by Lemma 19, if X is sparse then

$$\Pr_{j,Y}[v \in Y[j]] = O\left(\frac{n^2}{L}\right) = O(\varepsilon)$$

for every v. So both requirements of Definition 12 hold simultaneously with probability at least 9/10. ■

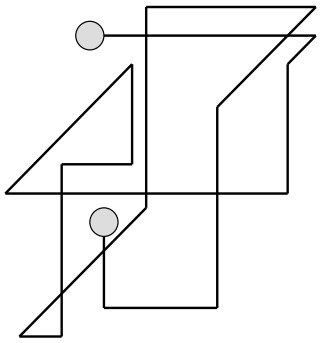

Figure 7.2: In $d = 3$ dimensions, a snake drawn from $\mathcal{D}_{h,L}$ moves a random distance left or right, then a random distance up or down, then a random distance inward or outward, etc.

7.5.2 Constant-Dimensional Grid Graph

In the Boolean hypercube case, $\mathcal{D}_{h,L}$ was defined by a 'coordinate loop' instead of the usual random walk mainly for convenience. When we move to the d-dimensional grid, though, the drawbacks of random walks become more serious: first, the mixing time is too long, and second, there are too many self-intersections, particularly if $d \leq 4$. The snake distribution will instead use straight lines of randomly chosen lengths attached at the endpoints, as in Figure 7.2. Let $G_{d,N}$ be a d-dimensional grid graph with $d \geq 3$. That is, $G_{d,N}$ has N vertices of the form $v = (v[0], \ldots, v[d-1])$, where each $v[i]$ is in $\left\{1, \ldots, N^{1/d}\right\}$ (assume for simplicity that N is a d^{th} power). Vertices v and w are adjacent if and only if $|v[i] - w[i]| = 1$ for some $i \in \{0, \ldots, d-1\}$, and $v[j] = w[j]$ for all $j \neq i$ (so $G_{d,N}$ does not wrap around at the boundaries).

Take $L = \sqrt{N}/100$, and define the snake distribution $\mathcal{D}_{h,L}$ as follows. Starting from $x_0 = h$, for each T take $x_{N^{1/d}(T+1)}$ identical to $x_{N^{1/d}T}$, but with the $(T \bmod d)^{th}$ coordinate $x_{N^{1/d}(T+1)}[T \bmod d]$ replaced by a uniform random value in $\left\{1, \ldots, N^{1/d}\right\}$. Then take the vertices $x_{N^{1/d}T+1}, \ldots, x_{N^{1/d}T+N^{1/d}-1}$ to lie along the shortest path from $x_{N^{1/d}T}$ to $x_{N^{1/d}(T+1)}$, 'stalling' at $x_{N^{1/d}(T+1)}$ once that vertex has been reached. Call

$$\Phi_T = \left(x_{N^{1/d}T}, \ldots, x_{N^{1/d}T+N^{1/d}-1}\right)$$

a *line* of vertices, whose *direction* is $T \bmod d$. As in the Boolean hypercube case, we have:

Proposition 21 *$\mathcal{D}_{h,L}$ mixes completely in $dN^{1/d}$ steps, in the sense that if $T^* \geq T + d$, then $x_{N^{1/d}T^*}$ is a uniform random vertex conditioned on $x_{N^{1/d}T}$.*

Lemma 16 in Section 7.5.1 goes through essentially without change.

Definition 22 *Letting $\Delta(x, v, i)$ be as before, we say X is sparse if there exists a constant c (possibly dependent on d) such that for all vertices v and all k,*

$$\left|\left\{t : \Delta\left(x_t, v, \left\lfloor t/N^{1/d} \right\rfloor \bmod d\right) = k\right\}\right| \leq (c \log N)\left(N^{1/d} + \frac{L}{N^{1-k/d}}\right).$$

Lemma 23 *If X is drawn from $\mathcal{D}_{h,L}$, then X is sparse with probability $1-o(1)$.*

Proof. Similar to Lemma 18. Let Φ_T be a line of vertices with direction $i = T \bmod d$, and notice that $\Delta(x_t, v, i)$ is the same for every vertex x_t in Φ_T. Let $E_T^{(v,i,k)}$ denote the event that $\Delta(x_t, v, i) \leq k$ for the x_t's in Φ_T. Then $E_T^{(v,i,k)}$ occurs with probability $N^{(k-1)/d}/N$ over X. Furthermore, if $|T - T^*| \geq d$ then $E_T^{(v,i,k)}$ and $E_{T^*}^{(v,i,k)}$ are independent events. So let

$$\mu_k = L \cdot \frac{N^{(k-1)/d}}{N};$$

then for fixed v, i, k, the expected number of lines for which $E_T^{(v,i,k)}$ holds is at most μ_k. Thus, by a Chernoff bound, if $\mu_k \geq 1$ then

$$\Pr_X\left[\left|\left\{T : E_T^{(v,i,k)}\right\}\right| > c\log N \cdot \mu_k\right] < \left(\frac{e^{c\log N - 1}}{(c\log N)^{c\log N}}\right)^{\mu_k}$$

which is at most $1/N^2$ for sufficiently large c. Similarly, if $\mu_k < 1$ then letting $m = (c\log N)/\mu_k$,

$$\Pr_X\left[\left|\left\{T : E_T^{(v,i,k)}\right\}\right| > c\log N\right] < \left(\frac{e^{m-1}}{m^m}\right)^{\mu_k} < \frac{1}{N^2}$$

for sufficiently large c. So with probability $1-o(1)$ it holds that for all v, k, letting $i_t = \left\lfloor t/N^{1/d}\right\rfloor \bmod d$,

$$\begin{aligned}\left|\{t : \Delta(x_t, v, i_t) = k\}\right| &\leq c\log N \cdot (1+\mu_k) \cdot N^{1/d} \\ &= (c\log N)\left(N^{1/d} + \frac{L}{N^{1-k/d}}\right).\end{aligned}$$

■

Lemma 24 *If X is sparse, then for every $v \in G_{d,N}$,*

$$\Pr_{j,Y}\left[v \in Y[j]\right] = O\left(\frac{N^{1/d}\log N}{L}\right),$$

where the big-O hides a constant dependent on d.

Proof. As in Lemma 19, setting $i_j = \left\lfloor j/N^{1/d}\right\rfloor \bmod d$ we obtain that $\Pr_{j,Y}\left[v \in Y[j]\right]$ equals

$$\begin{aligned}&\sum_{k=1}^{d} \Pr_j\left[\Delta(x_j, v, i_j) = k\right] \Pr_Y\left[v \in Y[j] \;\middle|\; \Delta(x_j, v, i_j) = k\right] \\ &\leq \sum_{k=1}^{d} \frac{c\log N}{L}\left(N^{1/d} + \frac{L}{N^{1-k/d}}\right)\left(\frac{1}{N^{(k-1)/d}} + \frac{L}{N}\right) \\ &= O\left(\frac{N^{1/d}\log N}{L}\right).\end{aligned}$$

■

By the same proof as for Theorem 20, taking $\varepsilon = (\log N)/N^{1/2-1/d}$ yields the following:

Theorem 25 *Neglecting a constant dependent on d, for all $d \geq 3$*

$$\mathrm{RLS}(G_{d,N}) = \Omega\left(\frac{N^{1/2-1/d}}{\log N}\right),$$
$$\mathrm{QLS}(G_{d,N}) = \Omega\left(\sqrt{\frac{N^{1/2-1/d}}{\log N}}\right).$$

Chapter 8

Quantum Certificate Complexity

This chapter studies the relationships between classical and quantum measures of query complexity. Let $f : \mathcal{S} \to \{0,1\}$ be a Boolean function with $\mathcal{S} \subseteq \{0,1\}^n$, that takes input $Y = y_1 \dots y_n$. Then the deterministic query complexity $\mathrm{D}(f)$ is the minimum number of queries to the y_i's needed to evaluate f, if Y is chosen adversarially and if queries can be adaptive (that is, can depend on the outcomes of previous queries). Also, the bounded-error randomized query complexity, $\mathrm{R}_2(f)$, is the minimum expected number of queries needed by a randomized algorithm that, for each Y, outputs $f(Y)$ with probability at least $2/3$. Here the '2' refers to two-sided error; if instead we require $f(Y)$ to be output with probability 1 for every Y, we obtain $\mathrm{R}_0(f)$, or zero-error randomized query complexity.

Analogously, $\mathrm{Q}_2(f)$ is the minimum number of queries needed by a quantum algorithm that outputs $f(Y)$ with probability at least $2/3$ for all Y. Also, for $k \in \{0,1\}$ let $\mathrm{Q}_0^k(f)$ be the minimum number of queries needed by a quantum algorithm that outputs $f(Y)$ with probability 1 if $f(Y) = k$, and with probability at least $1/2$ if $f(Y) \neq k$. Then let $\mathrm{Q}_0(f) = \max\left\{\mathrm{Q}_0^0(f), \mathrm{Q}_0^1(f)\right\}$. If we require a single algorithm that succeeds with probability 1 for all Y, we obtain $\mathrm{Q}_E(f)$, or exact quantum query complexity. See Buhrman and de Wolf [78] for a more detailed survey of these measures.

It is immediate that

$$\mathrm{Q}_2(f) \leq \mathrm{R}_2(f) \leq \mathrm{R}_0(f) \leq \mathrm{D}(f) \leq n,$$

that $\mathrm{Q}_0(f) \leq \mathrm{R}_0(f)$, and that $\mathrm{Q}_E(f) \leq \mathrm{D}(f)$. If f is partial (i.e. $\mathcal{S} \neq \{0,1\}^n$), then $\mathrm{Q}_2(f)$ can be superpolynomially smaller than $\mathrm{R}_2(f)$; this is what makes Shor's period-finding algorithm [219] possible. For total f, by contrast, the largest known gap even between $\mathrm{D}(f)$ and $\mathrm{Q}_2(f)$ is quadratic, and is achieved by the OR function on n bits: $\mathrm{D}(OR) = n$ (indeed $\mathrm{R}_2(\mathrm{OR}) = \Omega(n)$), whereas $\mathrm{Q}_2(\mathrm{OR}) = \Theta(\sqrt{n})$ because of Grover's search algorithm [139]. Furthermore, for total f, Beals et al. [45] showed that $\mathrm{D}(f) = O\left(\mathrm{Q}_2(f)^6\right)$, while de Wolf [244] showed that $\mathrm{D}(f) = O\left(\mathrm{Q}_2(f)^2 \mathrm{Q}_0(f)^2\right)$.

The result of Beals et al. [45] relies on two intermediate complexity measures, the *certificate complexity* $\mathrm{C}(f)$ and *block sensitivity* $\mathrm{bs}(f)$, which are defined as follows.

Definition 26 *A certificate for an input X is a set $S \subseteq \{1, \dots, n\}$ such that for all inputs*

	Deterministic	Randomized	Quantum
Query complexity	$\mathrm{D}(f)$	$\mathrm{R}_2(f)$	$\mathrm{Q}_2(f)$
Certificate complexity	$\mathrm{C}(f)$	$\mathrm{RC}(f)$	$\mathrm{QC}(f)$

Table 8.1: Query complexity measures and their certificate complexity analogues.

Y of f, if $y_i = x_i$ for all $i \in S$ then $f(Y) = f(X)$. Then $\mathrm{C}^X(f)$ is the minimum size of a certificate for X, and $\mathrm{C}(f)$ is the maximum of $\mathrm{C}^X(f)$ over all X.

Definition 27 *A sensitive block on input X is a set $B \subseteq \{1,\ldots,n\}$ such that $f\left(X^{(B)}\right) \neq f(X)$, where $X^{(B)}$ is obtained from X by flipping x_i for each $i \in B$. Then $\mathrm{bs}^X(f)$ is the maximum number of disjoint sensitive blocks on X, and $\mathrm{bs}(f)$ is the maximum of $\mathrm{bs}^X(f)$ over all X.*

Clearly $\mathrm{bs}(f) \leq \mathrm{C}(f) \leq \mathrm{D}(f)$. For total f, these measures are all polynomially related: Nisan [183] showed that $\mathrm{C}(f) \leq \mathrm{bs}(f)^2$, while Beals et al. [45] showed that $\mathrm{D}(f) \leq \mathrm{C}(f)\,\mathrm{bs}(f)$. Combining these results with $\mathrm{bs}(f) = O\left(\mathrm{Q}_2(f)^2\right)$ (from the optimality of Grover's algorithm), one obtains $\mathrm{D}(f) = O\left(\mathrm{Q}_2(f)^6\right)$.

8.1 Summary of Results

I investigate $\mathrm{RC}(f)$ and $\mathrm{QC}(f)$, the bounded-error randomized and quantum generalizations of the certificate complexity $\mathrm{C}(f)$ (see Table 8.1). My motivation is that, just as $\mathrm{C}(f)$ was used to show a polynomial relation between $\mathrm{D}(f)$ and $\mathrm{Q}_2(f)$, so $\mathrm{RC}(f)$ and $\mathrm{QC}(f)$ can lead to new relations among fundamental query complexity measures.

What the certificate complexity $\mathrm{C}(f)$ measures is the number of *queries* used to verify a certificate, not the number of *bits* used to communicate it. Thus, if we want to generalize $\mathrm{C}(f)$, we should assume the latter is unbounded. A consequence is that without loss of generality, a certificate is just a claimed value X for the input Y[1]—since any additional information that a prover might provide, the verifier can compute for itself. The verifier's job is to check that $f(Y) = f(X)$. With this in mind I define $\mathrm{RC}(f)$ as follows.

Definition 28 *A randomized verifier for input X is a randomized algorithm that, on input Y to f, (i) accepts with probability 1 if $Y = X$, and (ii) rejects with probability at least $1/2$ if $f(Y) \neq f(X)$. (If $Y \neq X$ but $f(Y) = f(X)$, the acceptance probability can be arbitrary.) Then $\mathrm{RC}^X(f)$ is the minimum expected number of queries used by a randomized verifier for X, and $\mathrm{RC}(f)$ is the maximum of $\mathrm{RC}^X(f)$ over all X.*

I define $\mathrm{QC}(f)$ analogously, with quantum instead of randomized algorithms. The following justifies the definition (the $\mathrm{RC}(f)$ part was originally shown by Raz et al. [197]).

[1]Throughout this chapter, I use Y to denote the 'actual' input being queried, and X to denote the 'claimed' input.

Proposition 29 *Making the error probability two-sided rather than one-sided changes* $\mathrm{RC}(f)$ *and* $\mathrm{QC}(f)$ *by at most a constant factor.*

Proof. For $\mathrm{RC}(f)$, let r_V^Y be the event that verifier V rejects on input Y, and let d_V^Y be the event that V encounters a disagreement with X on Y. We may assume $\Pr\left[r_V^Y \mid d_V^Y\right] = 1$. Suppose that $\Pr\left[r_V^Y\right] \leq \varepsilon_0$ if $Y = X$ and $\Pr\left[r_V^Y\right] \geq 1 - \varepsilon_1$ if $f(Y) \neq f(X)$. We wish to lower-bound $\Pr\left[d_V^Y\right]$ for all Y such that $f(Y) \neq f(X)$. Observe that

$$\Pr\left[r_V^Y \wedge \neg d_V^Y \mid f(Y) \neq f(X)\right] \leq \Pr\left[r_V^X \wedge \neg d_V^X\right] = \Pr\left[r_V^X\right] \leq \varepsilon_0.$$

Hence for $f(Y) \neq f(X)$,

$$\Pr\left[d_V^Y\right] \geq \Pr\left[r_V^Y\right] - \Pr\left[r_V^Y \wedge \neg d_V^Y\right] \geq 1 - \varepsilon_1 - \varepsilon_0.$$

Now let V^* be identical to V except that, whenever V rejects despite having found no disagreement with X, V^* accepts. Clearly $\Pr\left[r_{V^*}^X\right] = 0$. Also, in the case $f(Y) \neq f(X)$,

$$\Pr\left[r_{V^*}^Y\right] = \Pr\left[d_V^Y\right] \geq 1 - \varepsilon_1 - \varepsilon_0.$$

The result follows since $O(1)$ repetitions suffice to boost any constant error probability to any other constant error probability.

For $\mathrm{QC}(f)$, suppose the verifier's final state given input Y is

$$\sum_z \alpha_z^Y |z\rangle \left(\beta_z^Y |0\rangle + \gamma_z^Y |1\rangle\right)$$

where $|0\rangle$ is the reject state, $|1\rangle$ is the accept state, and $\left|\beta_z^Y\right|^2 + \left|\gamma_z^Y\right|^2 = 1$ for all z. Suppose also that $A^X \geq 1 - \varepsilon_0$ and that $A^Y \leq \varepsilon_1$ whenever $f(Y) \neq f(X)$, where $A^Y = \sum_z \left|\alpha_z^Y \gamma_z^Y\right|^2$ is the probability of accepting. Then the verifier can make $A^X = 1$ by performing the conditional rotation

$$\begin{pmatrix} \gamma_z^X & -\beta_z^X \\ \beta_z^X & \gamma_z^X \end{pmatrix}$$

on the second register prior to measurement. In the case $f(Y) \neq f(X)$, this produces

$$\begin{aligned} A^Y &= \sum_z \left|\alpha_z^Y\right|^2 \left|\beta_z^X \beta_z^Y + \gamma_z^X \gamma_z^Y\right|^2 \\ &\leq 2 \sum_z \left|\alpha_z^Y\right|^2 \left(\left|\beta_z^X\right|^2 + \left|\gamma_z^Y\right|^2\right) \\ &\leq 2(\varepsilon_0 + \varepsilon_1). \end{aligned}$$

■

It is immediate that $\mathrm{QC}(f) \leq \mathrm{RC}(f) \leq \mathrm{C}(f)$, that $\mathrm{QC}(f) = O(\mathrm{Q}_2(f))$, and that $\mathrm{RC}(f) = O(\mathrm{R}_2(f))$. We also have $\mathrm{RC}(f) = \Omega(\mathrm{bs}(f))$, since a randomized verifier for X must query each sensitive block on X with $1/2$ probability. This suggests viewing $\mathrm{RC}(f)$ as an 'alloy' of block sensitivity and certificate complexity, an interpretation for which Section 8.5 gives some justification.

The results of this chapter are as follows. In Section 8.3 I show that $\mathrm{QC}(f) = \Theta\left(\sqrt{\mathrm{RC}(f)}\right)$ for all f (partial or total), precisely characterizing quantum certificate complexity in terms of randomized certificate complexity. To do this, I first give a nonadaptive characterization of $\mathrm{RC}(f)$, and then apply the adversary method of Ambainis [27] to lower-bound $\mathrm{QC}(f)$ in terms of this characterization. Then, in Section 8.4, I extend results on polynomials due to de Wolf [244] and to Nisan and Smolensky (as described by Buhrman and de Wolf [78]), to show that $\mathrm{R}_0(f) = O(\mathrm{RC}(f)\,\mathrm{ndeg}(f)\log n)$ for all total f, where $\mathrm{ndeg}(f)$ is the minimum degree of a polynomial p such that $p(X) \neq 0$ if and only if $f(X) \neq 0$. Combining the results of Sections 8.3 and 8.4 leads to a new lower bound on quantum query complexity: that $\mathrm{R}_0(f) = O\left(\mathrm{Q}_2(f)^2\,\mathrm{Q}_0(f)\log n\right)$ for all total f. To my knowledge, this is the first quantum lower bound to use both the adversary method and the polynomial method at different points in the argument.

Finally, in Section 8.5, I exhibit asymptotic gaps between $\mathrm{RC}(f)$ and other query complexity measures, including a total f for which $\mathrm{C}(f) = \Theta\left(\mathrm{QC}(f)^{2.205}\right)$, and a symmetric partial f for which $\mathrm{QC}(f) = O(1)$ yet $\mathrm{Q}_2(f) = \Omega(n/\log n)$. I conclude in Section 8.6 with some open problems.

8.2 Related Work

Raz et al. [197] studied a query complexity measure they called $\mathrm{ma}(f)$, for Merlin-Arthur. In my notation, $\mathrm{ma}(f)$ equals the maximum of $\mathrm{RC}^X(f)$ over all X with $f(X) = 1$. Raz et al. observed that $\mathrm{ma}(f) = \mathrm{ip}(f)$, where $\mathrm{ip}(f)$ is the number of queries needed given arbitrarily many rounds of interaction with a prover. They also used error-correcting codes to construct a total f for which $\mathrm{ma}(f) = O(1)$ but $\mathrm{C}(f) = \Omega(n)$. This has similarities to the construction, in Section 8.5.2, of a symmetric partial f for which $\mathrm{QC}(f) = O(1)$ but $\mathrm{Q}_2(f) = \Omega(n/\log n)$. Aside from that and from Proposition 29, Raz et al.'s results do not overlap with the results here.

Watrous [239] has investigated a different notion of 'quantum certificate complexity'—whether certificates that are quantum states can be superpolynomially smaller than any classical certificate. Also, de Wolf [245] has investigated 'nondeterministic quantum query complexity' in the alternate sense of algorithms that accept with zero probability when $f(Y) = 0$, and with positive probability when $f(Y) = 1$.

8.3 Characterization of Quantum Certificate Complexity

We wish to show that $\mathrm{QC}(f) = \Theta\left(\sqrt{\mathrm{RC}(f)}\right)$, precisely characterizing quantum certificate complexity in terms of randomized certificate complexity. The first step is to give a simpler characterization of $\mathrm{RC}(f)$.

Lemma 30 *Call a randomized verifier for X nonadaptive if, on input Y, it queries each y_i with independent probability λ_i, and rejects if and only if it encounters a disagreement with X. (Thus, we identify such a verifier with the vector $(\lambda_1, \ldots, \lambda_n)$.) Let $\mathrm{RC}^X_{na}(f)$*

be the minimum of $\lambda_1 + \cdots + \lambda_n$ *over all nonadaptive verifiers for* X. *Then* $\mathrm{RC}^X_{na}(f) = \Theta\left(\mathrm{RC}^X(f)\right)$.

Proof. Clearly $\mathrm{RC}^X_{na}(f) = \Omega\left(\mathrm{RC}^X(f)\right)$. For the upper bound, we can assume that a randomized verifier rejects immediately on finding a disagreement with X, and accepts if it finds no disagreement. Let $\mathcal{Y} = \{Y : f(Y) \neq f(X)\}$. Let V be an optimal randomized verifier, and let $p_t(Y)$ be the probability that V, when given input $Y \in \mathcal{Y}$, finds a disagreement with X on the t^{th} query. By Markov's inequality, V must have found a disagreement with probability at least $1/2$ after $T = \left\lceil 2\,\mathrm{RC}^X(f) \right\rceil$ queries. So by the union bound

$$p_1(Y) + \cdots + p_T(Y) \geq \frac{1}{2}$$

for each $Y \in \mathcal{Y}$. Suppose we choose $t \in \{1, \ldots, T\}$ uniformly at random and simulate the t^{th} query, pretending that queries $1, \ldots, t-1$ have already been made and have returned agreement with X. Then we must find a disagreement with probability at least $1/2T$. By repeating this procedure $4T$ times, we can boost the probability to $1 - e^{-2}$. For $i \in \{1, \ldots, n\}$, let λ_i be the probability that y_i is queried at least once. Then $\lambda_1 + \cdots + \lambda_n \leq 4T$, whereas for each $Y \in \mathcal{Y}$,

$$\sum_{i : y_i \neq x_i} \lambda_i \geq 1 - e^{-2}.$$

It follows that, if each y_i is queried with independent probability λ_i, then the probability that at least one y_i disagrees with X is at least

$$1 - \prod_{i : y_i \neq x_i} (1 - \lambda_i) \geq 1 - \left(1 - \frac{1 - e^{-2}}{n}\right)^n > 0.57.$$

■

To obtain a lower bound on $\mathrm{QC}(f)$, I will use the following simple reformulation of Ambainis's adversary method [27].

Theorem 31 (Ambainis) *Given a function* $f : \mathcal{S} \to \{0,1\}$ *with* $\mathcal{S} \subseteq \{0,1\}^n$, *let* β *be a function from* $\mathcal{S}$ *to nonnegative reals, and let* $R : \mathcal{S}^2 \to \{0,1\}$ *be a relation such that* $R(X,Y) = R(Y,X)$ *for all* X, Y *and* $R(X,Y) = 0$ *whenever* $f(X) = f(Y)$. *Let* $\delta_0, \delta_1 \in (0,1]$ *be such that for every* $X \in \mathcal{S}$ *and* $i \in \{1, \ldots, n\}$,

$$\sum_{Y\,:\,R(X,Y)=1} \beta(Y) \geq 1,$$

$$\sum_{Y\,:\,R(X,Y)=1, x_i \neq y_i} \beta(Y) \leq \delta_{f(X)}.$$

Then $\mathrm{Q}_2(f) = \Omega\left(\sqrt{\frac{1}{\delta_0 \delta_1}}\right)$.

I now prove the main result of the section.

Theorem 32 *For all f (partial or total) and all X,*

$$\mathrm{QC}^X(f) = \Theta\left(\sqrt{\mathrm{RC}^X(f)}\right).$$

Proof. Let $(\lambda_1, \ldots, \lambda_n)$ be an optimal nonadaptive randomized verifier for X, and let

$$S = \lambda_1 + \cdots + \lambda_n.$$

First, $\mathrm{QC}^X(f) = O\left(\sqrt{S}\right)$. We can run a "weighted Grover search," in which the proportion of basis states querying index i is within a constant factor of λ_i/S. (It suffices to use n^2 basis states.) Let $\mathcal{Y} = \{Y : f(Y) \neq f(X)\}$; then for any $Y \in \mathcal{Y}$, $O\left(\sqrt{S}\right)$ iterations suffice to find a disagreement with X with probability $\Omega(1)$. Second, $\mathrm{QC}^X(f) = \Omega\left(\sqrt{S}\right)$. Consider a matrix game in which Alice chooses an index i to query and Bob chooses $Y \in \mathcal{Y}$; Alice wins if and only if $y_i \neq x_i$. If both players are rational, then Alice wins with probability $O(1/S)$, since otherwise Alice's strategy would yield a verifier $(\lambda'_1, \ldots, \lambda'_n)$ with

$$\lambda'_1 + \cdots + \lambda'_n = o(S).$$

Hence by the minimax theorem, there exists a distribution μ over $\mathcal{Y}$ such that for every i,

$$\Pr_{Y \in \mu}[y_i \neq x_i] = O\left(\frac{1}{S}\right).$$

Let $\beta(X) = 1$ and let $\beta(Y) = \mu(Y)$ for each $Y \in \mathcal{Y}$. Also, let $R(Y, Z) = 1$ if and only if $Z = X$ for each $Y \in \mathcal{Y}$ and $Z \notin \mathcal{Y}$. Then we can take $\delta_{f(Y)} = 1$ and $\delta_{f(X)} = O(1/S)$ in Theorem 31. So the quantum query complexity of distinguishing X from an arbitrary $Y \in \mathcal{Y}$ is $\Omega\left(\sqrt{S}\right)$. ■

8.4 Quantum Lower Bound for Total Functions

The goal of this section is to show that

$$\mathrm{R}_0(f) = O\left(\mathrm{Q}_2(f)^2 \mathrm{Q}_0(f) \log n\right)$$

for all total f. Say that a real multilinear polynomial $p(x_1, \ldots, x_n)$ nondeterministically represents f if for all $X \in \{0,1\}^n$, $p(X) \neq 0$ if and only if $f(X) \neq 0$. Let $\mathrm{ndeg}(f)$ be the minimum degree of a nondeterministic polynomial for f. Also, given such a polynomial p, say that a monomial $M_1 \in p$ is *covered* by $M_2 \in p$ if M_2 contains every variable in M_1. A monomial M is called a *maxonomial* if it is not covered by any other monomial of p. The following is a simple generalization of a lemma attributed in [78] to Nisan and Smolensky.

Lemma 33 (Nisan-Smolensky) *Let p nondeterministically represent f. Then for every maxonomial M of p and $X \in f^{-1}(0)$, there is a set B of variables in M such that $f\left(X^{(B)}\right) \neq f(X)$, where $X^{(B)}$ is obtained from X by flipping the variables in B.*

Proof. Obtain a restricted function g from f, and a restricted polynomial q from p, by setting each variable outside of M to x_i. Then g cannot be constant, since its representing polynomial q contains M as a monomial. Thus there is a subset B of variables in M such that $g\left(X^{(B)}\right)=1$, and hence $f\left(X^{(B)}\right)=1$. ■

Using Lemma 33, de Wolf [244] showed that $\mathrm{D}(f) \leq \mathrm{C}(f)\,\mathrm{ndeg}(f)$ for all total f, slightly improving the result $\mathrm{D}(f) \leq \mathrm{C}(f)\deg(f)$ due to Buhrman and de Wolf [78]. In Theorem 35, I will give an analogue of this result for *randomized* query and certificate complexities. However, I first need a probabilistic lemma.

Lemma 34 *Suppose we repeatedly apply the following procedure: first identify the set B of maxonomials of p, then 'shrink' each $M \in B$ with (not necessarily independent) probability at least $1/2$. Shrinking M means replacing it by an arbitrary monomial of degree $\deg(M)-1$. Then with high probability p is a constant polynomial after $O(\deg(p)\log n)$ iterations.*

Proof. For any set A of monomials, consider the weighting function

$$\omega(A)=\sum_{M\in A}\deg(M)!$$

Let S be the set of monomials of p. Initially $\omega(S) \leq n^{\deg(p)}\deg(p)!$, and we are done when $\omega(S)=0$. The claim is that at every iteration, $\omega(B) \geq \frac{1}{e}\omega(S)$. For every $M^* \in S \setminus B$ is covered by some $M \in B$, but a given $M \in B$ can cover at most $\binom{\deg(M)}{\ell}$ distinct M^* with $\deg(M^*)=\ell$. Hence

$$\begin{aligned}\omega(S\setminus B) &\leq \sum_{M\in B}\sum_{\ell=0}^{\deg(M)-1}\tbinom{\deg(M)}{\ell}\ell! \\ &\leq \sum_{M\in B}\deg(M)!\left(\frac{1}{1!}+\frac{1}{2!}+\cdots\right) \\ &\leq (e-1)\,\omega(B).\end{aligned}$$

At every iteration, the contribution of each $M \in B$ to $\omega(A)$ has at least $1/2$ probability of shrinking from $\deg(M)!$ to $(\deg(M)-1)!$ (or to 0 if $\deg(M)=1$). When this occurs, the contribution of M is at least halved. Hence $\omega(S)$ decreases by an expected amount at least $\frac{1}{4e}\omega(S)$. Thus after

$$\log_{4e/(4e-1)}\left(2n^{\deg(p)}\deg(p)!\right)=O(\deg(p)\log n)$$

iterations, the expectation of $\omega(S)$ is less than $1/2$, so S is empty with probability at least $1/2$. ■

I can now prove the main result.[2]

Theorem 35 *For total f,*

$$\mathrm{R}_0(f)=O(\mathrm{RC}(f)\,\mathrm{ndeg}(f)\log n).$$

[2]The proof of Theorem 35 that I gave previously [4] makes a claim that is both superfluous for proving the theorem and false. I am grateful to Gatis Midrijanis for pointing this out to me.

Proof. The algorithm is as follows.

```
Repeat
    Choose a 0-input X compatible with all queries made so far[3]
    Query a randomized 0-certificate for X
Until f has been restricted to a constant function
```

Let p be a polynomial that nondeterministically represents f. Then the key fact is that for every 0-input X, when we query a randomized 0-certificate for X we "hit" each maxonomial M of p with probability at least $1/2$. Here hitting M means querying a variable in M. This is because, by Lemma 33, it is possible to change $f(X)$ from 0 to 1 just by flipping variables in M. So a randomized certificate would be incorrect if it probed those variables with probability less than $1/2$.

Therefore, each iteration of the algorithm shrinks each maxonomial of p with probability at least $1/2$. It follows from Lemma 34 that the algorithm terminates after an expected number of iterations $O(\deg(p) \log n)$. ■

Buhrman et al. [45] showed that $\mathrm{ndeg}(f) \leq 2\,\mathrm{Q}_0(f)$. Combining this with Theorems 32 and 35 yields a new relation between classical and quantum query complexity.

Corollary 36 *For all total f,*

$$\mathrm{R}_0(f) = O\left(\mathrm{Q}_2(f)^2\,\mathrm{Q}_0(f) \log n\right).$$

The best previous relation of this kind was $\mathrm{R}_0(f) = O\left(\mathrm{Q}_2(f)^2\,\mathrm{Q}_0(f)^2\right)$, due to de Wolf [244]. It is worth mentioning another corollary of Theorems 32 and 35, this one purely classical:

Corollary 37 *For all total f,*

$$\mathrm{R}_0(f) = O\left(\mathrm{R}_2(f)\,\mathrm{ndeg}(f) \log n\right)$$

Previously, no relation between R_0 and R_2 better than $\mathrm{R}_0(f) = O\left(\mathrm{R}_2(f)^3\right)$ was known (although no asymptotic gap between R_0 and R_2 is known either [210]).

8.5 Asymptotic Gaps

Having related $\mathrm{RC}(f)$ and $\mathrm{QC}(f)$ to other query complexity measures in Section 8.4, in what follows I seek the largest possible asymptotic gaps among the measures. In particular, I give a total f for which $\mathrm{RC}(f) = \Theta\left(\mathrm{C}(f)^{0.907}\right)$ and hence $\mathrm{C}(f) = \Theta\left(\mathrm{QC}(f)^{2.205}\right)$, as well as a total f for which $\mathrm{bs}(f) = \Theta\left(\mathrm{RC}(f)^{0.922}\right)$. Although these gaps are the largest

[3] Clearly, as long as f is not a constant function, there *exists* a 0-input X compatible with all queries made so far.

of which I know, Section 8.5.1 shows that no 'local' technique can improve the relations $\mathrm{C}(f) = O\left(\mathrm{RC}(f)^2\right)$ and $\mathrm{RC}(f) = O\left(\mathrm{bs}(f)^2\right)$. Finally, Section 8.5.2 uses combinatorial designs to construct a symmetric partial f for which $\mathrm{RC}(f)$ and $\mathrm{QC}(f)$ are $O(1)$, yet $\mathrm{Q}_2(f) = \Omega(n/\log n)$.

Wegener and Zádori [240] exhibited total Boolean functions with asymptotic gaps between $\mathrm{C}(f)$ and $\mathrm{bs}(f)$. In similar fashion, I give a function family $\{g_t\}$ with an asymptotic gap between $\mathrm{C}(g_t)$ and $\mathrm{RC}(g_t)$. Let $g_1(x_1, \ldots, x_{29})$ equal 1 if and only if the Hamming weight of its input is 13, 14, 15, or 16. (The parameter 29 was found via computer search to produce a maximal separation.) Then for $t > 1$, let

$$g_t(x_1, \ldots, x_{29^t}) = g_0\left[g_{t-1}(X_1), \ldots, g_{t-1}(X_{29})\right]$$

where X_1 is the first 29^{t-1} input bits, X_2 is the second 29^{t-1}, and so on. For $k \in \{0, 1\}$, let

$$\mathrm{bs}^k(f) = \max_{f(X)=k} \mathrm{bs}^X(f),$$
$$\mathrm{C}^k(f) = \max_{f(X)=k} \mathrm{C}^X(f).$$

Then since $\mathrm{bs}^0(g_1) = \mathrm{bs}^1(g_1) = 17$, we have $\mathrm{bs}(g_t) = 17^t$. On the other hand, $\mathrm{C}^0(g_1) = 17$ but $\mathrm{C}^1(g_1) = 26$, so

$$\mathrm{C}^1(g_t) = 13\,\mathrm{C}^1(g_{t-1}) + 13\,\mathrm{C}^0(g_{t-1}),$$
$$\mathrm{C}^0(g_t) = 17 \max\left\{\mathrm{C}^1(g_{t-1}), \mathrm{C}^0(g_{t-1})\right\}.$$

Solving this recurrence yields $\mathrm{C}(g_t) = \Theta\left(22.725^t\right)$. We can now show a gap between C and RC.

Proposition 38 $\mathrm{RC}(g_t) = \Theta\left(\mathrm{C}(g_t)^{0.907}\right)$.

Proof. Since $\mathrm{bs}(g_t) = \Omega\left(\mathrm{C}(g_t)^{0.907}\right)$, it suffices to show that $\mathrm{RC}(g_t) = O(\mathrm{bs}(g_t))$. The randomized verifier V chooses an input variable to query as follows. Let X be the claimed input, and let $K = \sum_{i=1}^{29} g_{t-1}(X_i)$. Let $I_0 = \{i : g_{t-1}(X_i) = 0\}$ and $I_1 = \{i : g_{t-1}(X_i) = 1\}$. With probability p_K, V chooses an $i \in I_1$ uniformly at random; otherwise A chooses an $i \in I_0$ uniformly at random. Here p_K is as follows.

K	$[0,12]$	13	14	15	16	$[17,29]$
p_K	0	$\frac{13}{17}$	$\frac{7}{12}$	$\frac{5}{12}$	$\frac{4}{17}$	1

Once i is chosen, V repeats the procedure for X_i, and continues recursively in this manner until reaching a variable y_j to query. One can check that if $g_t(X) \neq g_t(Y)$, then $g_{t-1}(X_i) \neq g_{t-1}(Y_i)$ with probability at least $1/17$. Hence $x_j \neq y_j$ with probability at least $1/17^t$, and $\mathrm{RC}(g_t) = O\left(17^t\right)$. ■

By Theorem 32, it follows that $\mathrm{C}(g_t) = \Theta\left(\mathrm{QC}(g_t)^{2.205}\right)$. This offers a surprising contrast with the query complexity setting, where the best known gap between the deterministic and quantum measures is quadratic ($\mathrm{D}(f) = \Theta\left(\mathrm{Q}_2(f)^2\right)$).

The family $\{g_t\}$ happens *not* to yield an asymptotic gap between $\mathrm{bs}(f)$ and $\mathrm{RC}(f)$. The reason is that any input to g_0 can be covered perfectly by sensitive blocks of minimum size, with no variables left over. In general, though, one can have $\mathrm{bs}(f) = o(\mathrm{RC}(f))$. As reported by Bublitz et al. [74], M. Paterson found a total Boolean function $h_1(x_1, \ldots, x_6)$ such that $\mathrm{C}^X(h_1) = 5$ and $\mathrm{bs}^X(h_1) = 4$ for all X. Composing h_1 recursively yields $\mathrm{bs}(h_t) = \Theta\left(\mathrm{C}(h_t)^{0.861}\right)$ and $\mathrm{bs}(h_t) = \Theta\left(\mathrm{RC}(h_t)^{0.922}\right)$, both of which are the largest such gaps of which I know.

8.5.1 Local Separations

It is a longstanding open question whether the relation $\mathrm{C}(f) \leq \mathrm{bs}(f)^2$ due to Nisan [183] is tight. As a first step, one can ask whether the relations $\mathrm{C}(f) = O\left(\mathrm{RC}(f)^2\right)$ and $\mathrm{RC}(f) = O\left(\mathrm{bs}(f)^2\right)$ are tight. In this section I introduce a notion of *local proof* in query complexity, and then show there is no local proof that $\mathrm{C}(f) = o\left(\mathrm{RC}(f)^2\right)$ or that $\mathrm{RC}(f) = o\left(\mathrm{bs}(f)^2\right)$. This implies that proving either result would require techniques unlike those that are currently known. My inspiration comes from computational complexity, where researchers first formalized known methods of proof, including *relativizable proofs* [41] and *natural proofs* [200], and then argued that these methods were not powerful enough to resolve the field's outstanding problems.

Let $G(f)$ and $H(f)$ be query complexity measures obtained by maximizing over all inputs—that is,

$$G(f) = \max_X G^X(f),$$
$$H(f) = \max_X H^X(f).$$

Call $B \subseteq \{1, \ldots, n\}$ a *minimal block* on X if B is sensitive on X (meaning $f\left(X^{(B)}\right) \neq f(X)$), and no sub-block $B' \subset B$ is sensitive on X. Also, let X's *neighborhood* $\mathcal{N}(X)$ consist of X together with $X^{(B)}$ for every minimal block B of X. Consider a proof that $G(f) = O(t(H(f)))$ for some nondecreasing t. I call the proof *local* if it proceeds by showing that for every input X,

$$G^X(f) = O\left(\max_{Y \in \mathcal{N}(X)} \left\{t\left(H^Y(f)\right)\right\}\right).$$

As a canonical example, Nisan's proof [183] that $\mathrm{C}(f) \leq \mathrm{bs}(f)^2$ is local. For each X, Nisan observes that (i) a maximal set of disjoint minimal blocks is a certificate for X, (ii) such a set can contain at most $\mathrm{bs}^X(f)$ blocks, and (iii) each block can have size at most $\max_{Y \in \mathcal{N}(X)} \mathrm{bs}^Y(f)$. Another example of a local proof is the proof in Section 8.3 that $\mathrm{RC}(f) = O\left(\mathrm{QC}(f)^2\right)$.

Proposition 39 *There is no local proof showing that* $\mathrm{C}(f) = o\left(\mathrm{RC}(f)^2\right)$ *or that* $\mathrm{RC}(f) = o\left(\mathrm{bs}(f)^2\right)$ *for all total* f.

Proof. The first part is easy: let $f(X) = 1$ if $|X| \geq \sqrt{n}$ (where $|X|$ denotes the Hamming weight of X), and $f(X) = 0$ otherwise. Consider the all-zero input 0^n. We have $\mathrm{C}^{0^n}(f) = n - \lceil \sqrt{n} \rceil + 1$, but $\mathrm{RC}^{0^n}(f) = O(\sqrt{n})$, and indeed $\mathrm{RC}^Y(f) = O(\sqrt{n})$ for all $Y \in \mathcal{N}(0^n)$. For the second part, arrange the input variables in a lattice of size $\sqrt{n} \times \sqrt{n}$. Take $m = \Theta\left(n^{1/3}\right)$, and let $g(X)$ be the monotone Boolean function that outputs 1 if and only if X contains a 1-*square* of size $m \times m$. This is a square of 1's that can wrap around the edges of the lattice; note that only the variables along the sides must be set to 1, not those in the interior. An example input, with a 1-square of size 3×3, is shown below.

$$\begin{array}{ccccc} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 \end{array}$$

Clearly $\mathrm{bs}^{0^n}(g) = \Theta\left(n^{1/3}\right)$, since there can be at most n/m^2 disjoint 1-squares of size $m \times m$. Also, $\mathrm{bs}^Y(g) = \Theta\left(n^{1/3}\right)$ for any Y that is 0 except for a single 1-square. On the other hand, if we choose uniformly at random among all such Y's, then at any lattice site i, $\Pr_Y[y_i = 1] = \Theta\left(n^{-2/3}\right)$. Hence $\mathrm{RC}^{0^n}(g) = \Omega\left(n^{2/3}\right)$. ■

8.5.2 Symmetric Partial Functions

If f is partial, then $\mathrm{QC}(f)$ can be much smaller than $\mathrm{Q}_2(f)$. This is strikingly illustrated by the collision problem: let $\mathrm{Col}(Y) = 0$ if $Y = y_1 \ldots y_n$ is a one-to-one sequence and $\mathrm{Col}(Y) = 1$ if Y is a two-to-one sequence, promised that one of these is the case. Then $\mathrm{RC}(\mathrm{Col}) = \mathrm{QC}(\mathrm{Col}) = O(1)$, since every one-to-one input differs from every two-to-one input on at least $n/2$ of the y_i's. On the other hand, Chapter 6 showed that $\mathrm{Q}_2(\mathrm{Col}) = \Omega\left(n^{1/5}\right)$.

From the example of the collision problem, it is tempting to conjecture that (say) $\mathrm{Q}_2(f) = O\left(n^{1/3}\right)$ whenever $\mathrm{QC}(f) = O(1)$—that is, 'if every 0-input is far from every 1-input, then the quantum query complexity is sublinear.' Here I disprove this conjecture, even for the special case of symmetric functions such as Col. (Given a finite set $\mathcal{H}$, a function $f : \mathcal{S} \to \{0,1\}$ where $\mathcal{S} \subseteq \mathcal{H}^n$ is called symmetric if $x_1 \ldots x_n \in \mathcal{S}$ implies $x_{\sigma(1)} \ldots x_{\sigma(n)} \in \mathcal{S}$ and $f(x_1 \ldots x_n) = f\left(x_{\sigma(1)} \ldots x_{\sigma(n)}\right)$ for every permutation σ.)

The proof uses the following lemma, which can be found in Nisan and Wigderson [185] for example.

Lemma 40 (Nisan-Wigderson) *For any $\gamma > 1$, there exists a family of sets*

$$A_1, \ldots, A_m \subseteq \{1, \ldots, \lceil \gamma n \rceil\}$$

such that $m = \Omega\left(2^{n/\gamma}\right)$, $|A_i| = n$ for all i, and $|A_i \cap A_j| \leq n/\gamma$ for all $i \neq j$.

A lemma due to Ambainis [26] is also useful. Let $f : \mathcal{S} \to \{0,1\}$ where $\mathcal{S} \subseteq \{0,1\}^n$ be a partial Boolean function, and let $p : \{0,1\}^n \to \mathbb{R}$ be a real-valued multilinear polynomial. We say that p approximates f if (i) $p(X) \in [0,1]$ for every input $X \in \{0,1\}^n$ (not merely those in $\mathcal{S}$), and (ii) $|p(X) - g(X)| \leq 1/3$ for every $X \in \mathcal{S}$.

Lemma 41 (Ambainis) *At most* $2^{O\left(\Delta(n,d)dn^2\right)}$ *distinct Boolean functions (partial or total) can be approximated by polynomials of degree* d*, where* $\Delta(n,d) = \sum_{i=0}^{d} \binom{n}{i}$.

The result is an easy consequence of Lemmas 40 and 41.

Theorem 42 *There exists a symmetric partial* f *for which* $\mathrm{QC}(f) = O(1)$ *and* $\mathrm{Q}_2(f) = \Omega(n/\log n)$.

Proof. Let $f : \mathcal{S} \to \{0,1\}$ where $\mathcal{S} \subseteq \{1,\ldots,3n\}^n$, and let $m = \Omega\left(2^{n/3}\right)$. Let $A_1,\ldots,A_m \subseteq \{1,\ldots,3n\}$ be as in Lemma 40. We put $x_1,\ldots,x_n$ in $\mathcal{S}$ if and only if $\{x_1,\ldots,x_n\} = A_j$ for some j. Clearly $\mathrm{QC}(f) = O(1)$, since if $i \neq j$ then every permutation of A_i differs from every permutation of A_j on at least $n/3$ indices. The number of symmetric f with $\mathcal{S}$ as above is $2^m = 2^{\Omega\left(2^{n/3}\right)}$. We can convert any such f to a Boolean function g on $O(n\log n)$ variables. But Beals et al. [45] showed that, if $\mathrm{Q}_2(g) = T$, then g is approximated by a polynomial of degree at most $2T$. So by Lemma 41, if $\mathrm{Q}_2(g) \leq T$ for every g then

$$2T \cdot \Delta(n\log n, 2T) \cdot (n\log n)^2 = \Omega\left(2^{n/3}\right)$$

and we solve to obtain $T = \Omega(n/\log n)$. ■

8.6 Open Problems

Is $\widetilde{\deg}(f) = \Omega\left(\sqrt{\mathrm{RC}(f)}\right)$, where $\widetilde{\deg}(f)$ is the minimum degree of a polynomial approximating f? In other words, can one lower-bound $\mathrm{QC}(f)$ using the polynomial method of Beals et al. [45], rather than the adversary method of Ambainis [27]?

Also, is $\mathrm{R}_0(f) = O\left(\mathrm{RC}(f)^2\right)$? If so we obtain the new relations $\mathrm{R}_0(f) = O\left(\mathrm{Q}_2(f)^4\right)$ and $\mathrm{R}_0(f) = O\left(\mathrm{R}_2(f)^2\right)$.

Chapter 9

The Need to Uncompute

Like a classical algorithm, a quantum algorithm can solve problems recursively by calling itself as a subroutine. When this is done, though, the algorithm typically needs to call itself *twice* for each subproblem to be solved. The second call's purpose is to uncompute 'garbage' left over by the first call, and thereby enable interference between different branches of the computation. Of course, a factor of 2 increase in running time hardly seems like a big deal, when set against the speedups promised by quantum computing. The problem is that these factors of 2 multiply, with each level of recursion producing an additional factor. Thus, one might wonder whether the uncomputing step is really necessary, or whether a cleverly designed algorithm might avoid it. This chapter gives the first nontrivial example in which recursive uncomputation is provably necessary.

The example concerns a long-neglected problem called *Recursive Fourier Sampling* (henceforth RFS), which was introduced by Bernstein and Vazirani [55] in 1993 to prove the first oracle separation between BPP and BQP. Many surveys on quantum computing pass directly from the Deutsch-Jozsa algorithm [95] to the dramatic results of Simon [220] and Shor [219], without even mentioning RFS. There are two likely reasons for this neglect. First, the RFS problem seems artificial. It was introduced for the sole purpose of proving an oracle result, and is unlike all other problems for which a quantum speedup is known. (I will define RFS in Section 9.1; but for now, it involves a tree of depth $\log n$, where each vertex is labeled with a function to be evaluated via a Fourier transform.) Second, the speedup for RFS is only quasipolynomial (n versus $n^{\log n}$), rather than exponential as for the period-finding and hidden subgroup problems.

Nevertheless, I believe that RFS merits renewed attention—for it serves as an important link between quantum computing and the ideas of classical complexity theory. One reason is that, although other problems in BQP—such as the factoring, discrete logarithm, and 'shifted Legendre symbol' problems [232]—are thought to be classically intractable, these problems are quite low-level by complexity-theoretic standards. They, or their associated decision problems, are in $\mathsf{NP} \cap \mathsf{coNP}$.[1] By contrast, Bernstein and Vazirani [55] showed that, as an oracle problem, RFS lies outside NP and even MA (the latter result is unpublished, though not difficult). Subsequently Watrous [239] gave an oracle A, based

[1] For the shifted Legendre symbol problem, this is true assuming a number-theoretic conjecture of Boneh and Lipton [61].

on an unrelated problem, for which $\mathsf{BQP}^A \not\subset \mathsf{MA}^A$.[2] Also, Green and Pruim [135] gave an oracle B for which $\mathsf{BQP}^B \not\subset \mathsf{P}^{\mathsf{NP}^B}$. However, Watrous' problem was shown by Babai [38] to be in AM, while Green and Pruim's problem is in BPP. Thus, neither problem can be used to place BQP outside higher levels of the polynomial hierarchy.

On the other hand, Umesh Vazirani and others have conjectured that RFS is not in PH, from which it would follow that there exists an oracle A relative to which $\mathsf{BQP}^A \not\subset \mathsf{PH}^A$. Proving this is, in my view, one of the central open problems in quantum complexity theory. Its solution seems likely to require novel techniques for constant-depth circuit lower bounds.[3]

In this chapter I examine the RFS problem from a different angle. Could Bernstein and Vazirani's quantum algorithm for RFS be improved even further, to give an *exponential* speedup over the classical algorithm? And could we use RFS, not merely to place BQP outside of PH relative to an oracle, but to place it outside of PH with (say) a logarithmic number of alternations?

My answer to both questions is a strong 'no.' I study a large class of variations on RFS, and show that all of them fall into one of two classes:

(1) a trivial class, for which there exists a classical algorithm making only one query, or

(2) a nontrivial class, for which any quantum algorithm needs $2^{\Omega(h)}$ queries, where h is the height of the tree to be evaluated. (By comparison, the Bernstein-Vazirani algorithm uses 2^h queries, because of its need to uncompute garbage recursively at each level of the tree.)

Since n^h queries always suffice classically, this dichotomy theorem implies that the speedup afforded by quantum computers is at most quasipolynomial. It also implies that (nontrivial) RFS is solvable in quantum polynomial time only when $h = O(\log n)$.

The plan is as follows. In Section 9.1 I define the RFS problem, and give Bernstein and Vazirani's quantum algorithm for solving it. In Section 9.2, I use the adversary method of Ambainis [27] to prove a lower bound on the quantum query complexity of any RFS variant. This bound, however, requires a parameter that I call the "nonparity coefficient" to be large. Intuitively, given a Boolean function $g : \{0,1\}^n \rightarrow \{0,1\}$, the nonparity coefficient measures how far g is from being the parity of some subset of its input bits—not under the uniform distribution over inputs (the standard assumption in Fourier analysis), but under an adversarial distribution. The crux of the argument is that *either* the nonparity coefficient is zero (meaning the RFS variant in question is trivial), or else it is bounded below by a positive constant. This statement is proved in Section 9.2, and seems like it might be of independent interest. Section 9.3 concludes with some open problems.

[2] Actually, to place BQP outside MA relative to an oracle, it suffices to consider the complement of Simon's problem ("Does $f(x) = f(x \oplus s)$ only when $s = 0$?").

[3] For the RFS function can be represented by a low-degree real polynomial—this follows from the existence of a polynomial-time quantum algorithm for RFS, together with the result of Beals et al. [45] relating quantum algorithms to low-degree polynomials. As a result, the circuit lower bound technique of Razborov [198] and Smolensky [223], which is based on the nonexistence of low-degree polynomials, seems unlikely to work. Even the random restriction method of Furst et al. [120] can be related to low-degree polynomials, as shown by Linial et al. [166].

9.1 Preliminaries

In ordinary Fourier sampling, we are given oracle access to a Boolean function $A : \{0,1\}^n \rightarrow \{0,1\}$, and are promised that there exists a secret string $s \in \{0,1\}^n$ such that $A(x) = s \cdot x \pmod 2$ for all x. The problem is to find s—or rather, since we need a problem with Boolean output, the problem is to return $g(s)$, where $g : \{0,1\}^n \rightarrow \{0,1\}$ is some known Boolean function. We can think of $g(s)$ as the "hard-core bit" of s, and can assume that g itself is efficiently computable, or else that we are given access to an oracle for g.

To obtain a height-2 recursive Fourier sampling tree, we simply compose this problem. That is, we are no longer given direct access to $A(x)$, but instead are promised that $A(x) = g(s_x)$, where $s_x \in \{0,1\}^n$ is the secret string for another Fourier sampling problem. A query then takes the form (x, y), and produces as output $A_x(y) = s_x \cdot y \pmod 2$. As before, we are promised that there exists an s such that $A(x) = s \cdot x \pmod 2$ for all x, meaning that the s_x strings must be chosen consistent with this promise. Again we must return $g(s)$.

Continuing, we can define height-h recursive Fourier sampling, or RFS_h, recursively as follows. We are given oracle access to a function $A(x_1, \ldots, x_h)$ for all $x_1, \ldots, x_h \in \{0,1\}^n$, and are promised that

(1) for each fixed x_1^*, $A(x_1^*, x_2, \ldots, x_h)$ is an instance of RFS_{h-1} on $x_2, \ldots, x_h$, having answer bit $b(x_1^*) \in \{0,1\}$; and

(2) there exists a secret string $s \in \{0,1\}^n$ such that $b(x_1^*) = s \cdot x_1^* \pmod 2$ for each x_1^*.

Again the answer bit to be returned is $g(s)$. Note that g is assumed to be the same everywhere in the tree—though using the techniques in this chapter, it would be straightforward to generalize to the case of different g's. As an example that will be used later, we could take $g(s) = g_{\text{mod}\,3}(s)$, where $g_{\text{mod}\,3}(s) = 0$ if $|s| \equiv 0 \pmod 3$ and $g_{\text{mod}\,3}(s) = 1$ otherwise, and $|s|$ denotes the Hamming weight of s. We do not want to take g to be the parity of s, for if we did then $g(s)$ could be evaluated using a single query. To see this, observe that if x is the all-1's string, then $s \cdot x \pmod 2$ is the parity of s.

By an 'input,' I will mean a complete assignment for the RFS oracle (that is, $A(x_1, \ldots, x_h)$ for all $x_1, \ldots, x_h$). I will sometimes refer also to an 'RFS tree,' where each vertex at distance ℓ from the root has a label $x_1, \ldots, x_\ell$. If $\ell = h$ then the vertex is a leaf; otherwise it has 2^n children, each with a label $x_1, \ldots, x_\ell, x_{\ell+1}$ for some $x_{\ell+1}$. The subtrees of the tree just correspond to the sub-instances of RFS.

Bernstein and Vazirani [55] showed that $\text{RFS}_{\log n}$, or RFS with height $\log n$ (all logarithms are base 2), is solvable on a quantum computer in time polynomial in n. I include a proof for completeness. Let $A = (A_n)_{n \geq 0}$ be an oracle that, for each n, encodes an instance of $\text{RFS}_{\log n}$ whose answer is Ψ_n. Then let L_A be the unary language $\{0^n : \Psi_n = 1\}$.

Lemma 43 *$L_A \in \mathsf{EQP}^A \subseteq \mathsf{BQP}^A$ for any choice of A.*

Proof. RFS_1 can be solved exactly in four queries, with no garbage bits left over. The algorithm is as follows: first prepare the state

$$2^{-n/2} \sum_{x \in \{0,1\}^n} |x\rangle |A(x)\rangle ,$$

using one query to A. Then apply a phase flip conditioned on $A(x) = 1$, and uncompute $A(x)$ using a second query, obtaining

$$2^{-n/2} \sum_{x \in \{0,1\}^n} (-1)^{A(x)} |x\rangle .$$

Then apply a Hadamard gate to each bit of the $|x\rangle$ register. It can be checked that the resulting state is simply $|s\rangle$. One can then compute $|s\rangle |g(s)\rangle$ and uncompute $|s\rangle$ using two more queries to A, to obtain $|g(s)\rangle$. To solve $RFS_{\log n}(n)$, we simply apply the above algorithm recursively at each level of the tree. The total number of queries used is $4^{\log n} = n^2$.

One can further reduce the number of queries to $2^{\log n} = n$ by using the "one-call kickback trick," described by Cleve et al. [87]. Here one prepares the state

$$2^{-n/2} \sum_{x \in \{0,1\}^n} |x\rangle \otimes \frac{|1\rangle - |0\rangle}{\sqrt{2}}$$

and then exclusive-OR's $A(x)$ into the second register. This induces the desired phase $(-1)^{A(x)}$ without the need to uncompute $A(x)$. However, one still needs to uncompute $|s\rangle$ after computing $|g(s)\rangle$. ■

A remark on notation: to avoid confusion with subscripts, I denote the i^{th} bit of string x by $x[i]$.

9.2 Quantum Lower Bound

In this section I prove a lower bound on the quantum query complexity of RFS. Crucially, the bound should hold for any nontrivial one-bit function of the secret strings, not just a specific function such as $g_{\mod 3}(s)$ defined in Section 9.1. Let RFS_h^g be height-h recursive Fourier sampling in which the problem at each vertex is to return $g(s)$. The following notion turns out to be essential.

Definition 44 *Given a Boolean function $g : \{0,1\}^n \rightarrow \{0,1\}$ (partial or total), the nonparity coefficient $\mu(g)$ is the largest μ^* for which there exist distributions D_0 over the 0-inputs of g, and D_1 over the 1-inputs, such that for all $z \in \{0,1\}^n$, all 0-inputs $\widehat{s}_0$, and all 1-inputs $\widehat{s}_1$, we have*

$$\Pr_{s_0 \in D_0, s_1 \in D_1} [s_0 \cdot z \equiv \widehat{s}_1 \cdot z \pmod 2 \quad \vee \quad s_1 \cdot z \equiv \widehat{s}_0 \cdot z \pmod 2] \geq \mu^*.$$

Loosely speaking, the nonparity coefficient is high if there exist distributions over 0-inputs and 1-inputs that make g far from being a parity function of a subset of input bits. The following proposition develops some intuition about $\mu(g)$.

Proposition 45

(i) $\mu(g) \leq 3/4$ for all nonconstant g.

(ii) $\mu(g) = 0$ *if and only if* g *can be written as the parity (or the NOT of the parity) of a subset* B *of input bits.*

Proof.

(i) Given any $s_0 \neq \widehat{s}_1$ and $s_1 \neq \widehat{s}_0$, a uniform random z will satisfy

$$\Pr_z [s_0 \cdot z \not\equiv \widehat{s}_1 \cdot z \,(\text{mod}\, 2) \;\; \wedge \;\; s_1 \cdot z \not\equiv \widehat{s}_0 \cdot z \,(\text{mod}\, 2)] \geq \frac{1}{4}.$$

(If $s_0 \oplus \widehat{s}_1 = s_1 \oplus \widehat{s}_0$ then this probability will be 1/2; otherwise it will be 1/4.) So certainly there is a fixed choice of z that works for random s_0 and s_1.

(ii) For the 'if' direction, take $z[i] = 1$ if and only if $i \in B$, and choose $\widehat{s}_0$ and $\widehat{s}_1$ arbitrarily. This ensures that $\mu^* = 0$. For the 'only if' direction, if $\mu(g) = 0$, we can choose D_0 to have support on all 0-inputs, and D_1 to have support on all 1-inputs. Then there must be a z such that $s_0 \cdot z$ is constant as we range over 0-inputs, and $s_1 \cdot z$ is constant as we range over 1-inputs. Take $i \in B$ if and only if $z[i] = 1$.

■

If $\mu(g) = 0$, then RFS_h^g is easily solvable using a single classical query. Theorem 47 will show that for all g (partial or total),

$$\text{Q}_2\left(\text{RFS}_h^g\right) = \Omega\left(\left(\frac{1}{1-\mu(g)}\right)^{h/2}\right),$$

where Q_2 is bounded-error quantum query complexity as defined in Section 5.1. In other words, any RFS problem with μ bounded away from 0 requires a number of queries exponential in the tree height h.

However, there is an essential further part of the argument, which restricts the values of $\mu(g)$ itself. Suppose there existed a family $\{g_n\}$ of 'pseudoparity' functions: that is, $\mu(g_n) > 0$ for all n, yet $\mu(g_n) = O(1/\log n)$. Then the best bound obtainable from Theorem 47 would be $\Omega\left((1+1/\log n)^{h/2}\right)$, suggesting that $\text{RFS}_{\log^2 n}^g$ might still be solvable in quantum polynomial time. On the other hand, it would be unclear a priori how to solve $\text{RFS}_{\log^2 n}^g$ classically with a logarithmic number of alternations. Theorem 49 will rule out this scenario by showing that pseudoparity functions do not exist: if $\mu(g) < 0.146$ then g is a parity function, and hence $\mu(g) = 0$.

The theorem of Ambainis that we need is his "most general" lower bound from [27], which he introduced to show that the quantum query complexity of inverting a permutation is $\Omega(\sqrt{n})$, and which we used already in Chapter 7. Let us restate the theorem in the present context.

Theorem 46 (Ambainis) *Let* $X \subseteq f^{-1}(0)$ *and* $Y \subseteq f^{-1}(1)$ *be sets of inputs to function* f. *Let* $R(x,y) \geq 0$ *be a symmetric real-valued relation function, and for* $x \in X$, $y \in Y$,

and index i, let

$$\theta(x,i) = \frac{\sum_{y^* \in Y \,:\, x[i] \neq y^*[i]} R(x,y^*)}{\sum_{y^* \in Y} R(x,y^*)},$$
$$\theta(y,i) = \frac{\sum_{x^* \in X \,:\, x^*[i] \neq y[i]} R(x^*,y)}{\sum_{y^* \in Y} R(x^*,y)},$$

where the denominators are all nonzero. Then $\mathrm{Q}_2(f) = O(1/\upsilon)$ where

$$\upsilon = \max_{x \in X,\ y \in Y,\ i \,:\, R(x,y)>0,\ x[i] \neq y[i]} \sqrt{\theta(x,i)\,\theta(y,i)}.$$

We are now ready to prove a lower bound for RFS.

Theorem 47 *For all g (partial or total), $\mathrm{Q}_2\left(\mathrm{RFS}_h^g\right) = \Omega\left(\left(1-\mu(g)\right)^{-h/2}\right)$.*

Proof. Let X be the set of all 0-inputs to RFS_h^g, and let Y be the set of all 1-inputs. We will weight the inputs using the distributions D_0, D_1 from the definition of the nonparity coefficient $\mu(g)$. For all $x \in X$, let $p(x)$ be the product, over all vertices v in the RFS tree for x, of the probability of the secret string s at v, if s is drawn from $D_{g(s)}$ (where we condition on v's output bit, $g(s)$). Next, say that $x \in X$ and $y \in Y$ *differ minimally* if, for all vertices v of the RFS tree, the subtrees rooted at v are identical in x and in y whenever the answer bit $g(s)$ at v is the same in x and in y. If x and y differ minimally, then we will set $R(x,y) = p(x)\,p(y)$; otherwise we will set $R(x,y) = 0$. Clearly $R(x,y) = R(y,x)$ for all $x \in X, y \in Y$. Furthermore, we claim that $\theta(x,i)\,\theta(y,i) \leq (1-\mu(g))^h$ for all x, y that differ minimally and all i such that $x[i] \neq y[i]$. For suppose $y^* \in Y$ is chosen with probability proportional to $R(x,y^*)$, and $x^* \in X$ is chosen with probability proportional to $R(x^*,y)$. Then $\theta(x,i)\,\theta(y,i)$ equals the probability that we would notice the switch from x to y^* by monitoring i, times the probability that we would notice the switch from y to x^*.

Let v_j be the j^{th} vertex along the path in the RFS tree from the root to the leaf vertex i, for all $j \in \{1,\ldots,h\}$. Also, let $z_j \in \{0,1\}^n$ be the label of the edge between v_{j-1} and v_j, and let $s_{x,j}$ and $s_{y,j}$ be the secret strings at v_j in x and y respectively. Then since x and y differ minimally, we must have $g(s_{x,j}) \neq g(s_{y,j})$ for all j—for otherwise the subtrees rooted at v_j would be identical, which contradicts the assumption $x[i] \neq y[i]$. So we can think of the process of choosing y^* as first choosing a random $s'_{x,1}$ from D_1 so that $1 = g\left(s'_{x,1}\right) \neq g(s_{x,1}) = 0$, then choosing a random $s'_{x,2}$ from $D_{1-g(s_{x,2})}$ so that $g\left(s'_{x,2}\right) \neq g(s_{x,2})$, and so on. Choosing x^* is analogous, except that whenever we used D_0 in choosing y^* we use D_1, and vice versa. Since the $2h$ secret strings $s_{x,1},\ldots,s_{x,h},s_{y,1},\ldots,s_{y,h}$ to be updated are independent of one another, it follows that

$$\begin{aligned}
\Pr[y^*[i] \neq x[i]] \Pr[x^*[i] \neq y[i]] &= \prod_{j=1}^{h} \Pr_{s \in D_0}[s \cdot z_j \not\equiv s_{x,j} \cdot z_j] \Pr_{s \in D_1}[s \cdot z_j \not\equiv s_{y,j} \cdot z_j] \\
&\leq \prod_{j=1}^{h} (1-\mu(g)) \\
&= (1-\mu(g))^h
\end{aligned}$$

by the definition of $\mu(g)$. Therefore

$$\mathrm{Q}_2\left(\mathrm{RFS}_h^g\right) = \Omega\left(\left(1-\mu(g)\right)^{-h/2}\right)$$

by Theorem 46. ■

Before continuing further, let me show that there is a natural, explicit choice of g—the function $g_{\mathrm{mod}\,3}(s)$ from Section 9.1—for which the nonparity coefficient is almost 3/4. Thus, for $g = g_{\mathrm{mod}\,3}$, the algorithm of Lemma 43 is essentially optimal.

Proposition 48 $\mu(g_{\mathrm{mod}\,3}) = 3/4 - O(1/n)$.

Proof. Let $n \geq 6$. Let D_0 be the uniform distribution over all s with $|s| = 3\lfloor n/6 \rfloor$ (so $g_{\mathrm{mod}\,3}(s) = 0$); likewise let D_1 be the uniform distribution over s with $|s| = 3\lfloor n/6 \rfloor + 2$ ($g_{\mathrm{mod}\,3}(s) = 1$). We consider only the case of s drawn from D_0; the D_1 case is analogous. We will show that for any z,

$$\left| \Pr_{s \in D_0}\left[s \cdot z \equiv 0\right] - \frac{1}{2} \right| = O\left(\frac{1}{n}\right)$$

(all congruences are mod 2). The theorem then follows, since by the definition of the nonparity coefficient, given any z the choices of $s_0 \in D_0$ and $s_1 \in D_1$ are independent.

Assume without loss of generality that $1 \leq |z| \leq n/2$ (if $|z| > n/2$, then replace z by its complement). We apply induction on $|z|$. If $|z| = 1$, then clearly

$$\Pr\left[s \cdot z \equiv 0\right] = 3\lfloor n/6 \rfloor / n = \frac{1}{2} \pm O\left(\frac{1}{n}\right).$$

For $|z| \geq 2$, let $z = z_1 \oplus z_2$, where z_2 contains only the rightmost 1 of z and z_1 contains all the other 1's. Suppose the proposition holds for $|z| - 1$. Then

$$\begin{aligned}\Pr\left[s \cdot z \equiv 0\right] = &\Pr\left[s \cdot z_1 \equiv 0\right] \Pr\left[s \cdot z_2 \equiv 0 | s \cdot z_1 \equiv 0\right] + \\ &\Pr\left[s \cdot z_1 \equiv 1\right] \Pr\left[s \cdot z_2 \equiv 1 | s \cdot z_1 \equiv 1\right],\end{aligned}$$

where

$$\Pr\left[s \cdot z_1 \equiv 0\right] = \frac{1}{2} + \alpha, \;\; \Pr\left[s \cdot z_1 \equiv 1\right] = \frac{1}{2} - \alpha$$

for some $|\alpha| = O(1/n)$. Furthermore, even conditioned on $s \cdot z_1$, the expected number of 1's in s outside of z_1 is $(n - |z_1|)/2 \pm O(1)$ and they are uniformly distributed. Therefore

$$\Pr\left[s \cdot z_2 \equiv b | s \cdot z_1 \equiv b\right] = \frac{1}{2} + \beta_b$$

for some $|\beta_0|, |\beta_1| = O(1/n)$. So

$$\begin{aligned}\Pr\left[s \cdot z \equiv 0\right] &= \frac{1}{2} + \frac{\beta_0}{2} + \alpha\beta_0 - \frac{\beta_1}{2} - \alpha\beta_1 \\ &= \frac{1}{2} \pm O\left(\frac{1}{n}\right).\end{aligned}$$

■

Finally it must be shown that pseudoparity functions do not exist. That is, if g is too close to a parity function for the bound of Theorem 47 to apply, then g actually *is* a parity function, from which it follows that RFS_h^g admits an efficient classical algorithm.

Theorem 49 *Suppose $\mu(g) < 0.146$. Then g is a parity function (equivalently, $\mu(g) = 0$).*

Proof. By linear programming duality, there exists a joint distribution $\mathcal{D}$ over $z \in \{0,1\}^n$, 0-inputs $\widehat{s}_0 \in g^{-1}(0)$, and 1-inputs $\widehat{s}_1 \in g^{-1}(1)$, such that for all $s_0 \in g^{-1}(0)$ and $s_1 \in g^{-1}(1)$,

$$\Pr_{(z,\widehat{s}_0,\widehat{s}_1)\in\mathcal{D}}\left[s_0 \cdot z \equiv \widehat{s}_1 \cdot z \pmod 2 \quad \vee \quad s_1 \cdot z \equiv \widehat{s}_0 \cdot z \pmod 2\right] < \mu(g).$$

Furthermore $\widehat{s}_0 \cdot z \not\equiv \widehat{s}_1 \cdot z \pmod 2$, since otherwise we could violate the hypothesis by taking $s_0 = \widehat{s}_0$ or $s_1 = \widehat{s}_1$. It follows that there exists a joint distribution $\mathcal{D}'$ over $z \in \{0,1\}^n$ and $b \in \{0,1\}$ such that

$$\Pr_{(z,b)\in\mathcal{D}'}\left[s \cdot z \equiv b \pmod 2\right] > 1 - \mu(g)$$

for all $s \in g^{-1}(0)$, and

$$\Pr_{(z,b)\in\mathcal{D}'}\left[s \cdot z \not\equiv b \pmod 2\right] > 1 - \mu(g)$$

for all $s \in g^{-1}(1)$. But this implies that g is a bounded-error threshold function of parity functions. More precisely, there exist probabilities p_z, summing to 1, as well as $b_z \in \{0,1\}$ such that for all $s \in \{0,1\}^n$,

$$\Psi(s) = \sum_{z\in\{0,1\}^n} p_z\left((s \cdot z) \oplus b_z\right) \text{ is } \begin{cases} > 1-\mu(g) & \text{if } g(s) = 1 \\ < \mu(g) & \text{if } g(s) = 0. \end{cases}$$

We will consider $\mathrm{var}(\Psi)$, the variance of the above quantity $\Psi(s)$ if s is drawn uniformly at random from $\{0,1\}^n$. First, if $p_z \geq 1/2$ for any z, then $g(s) = (s \cdot z) \oplus b_z$ is a parity function and hence $\mu(g) = 0$. So we can assume without loss of generality that $p_z < 1/2$ for all z. Then since s is uniform, for each $z_1 \neq z_2$ we know that $(s \cdot z_1) \oplus b_{z_1}$ and $(s \cdot z_2) \oplus b_{z_2}$ are pairwise independent $\{0,1\}$ random variables, both with expectation $1/2$. So

$$\mathrm{var}(\Psi) = \frac{1}{4}\sum_z p_z^2 < \frac{1}{4}\left(\left(\frac{1}{2}\right)^2 + \left(\frac{1}{2}\right)^2\right) = \frac{1}{8}.$$

On the other hand, since $\Psi(s)$ is always less than μ or greater than $1-\mu$,

$$\mathrm{var}(\Psi) > \left(\frac{1}{2} - \mu\right)^2.$$

Combining,

$$\mu > \frac{2-\sqrt{2}}{4} > 0.146.$$

■

9.3 Open Problems

An intriguing open problem is whether Theorem 47 can be proved using the polynomial method of Beals et al. [45], rather than the adversary method of Ambainis [27]. It is known that one can lower-bound polynomial degree in terms of block sensitivity, or the maximum number of disjoint changes to an input that change the output value. The trouble is that the RFS function has block sensitivity 1—the "sensitive blocks" of each input tend to have small intersection, but are not disjoint. For this reason, I implicitly used the quantum certificate complexity of Chapter 8 rather than block sensitivity to prove a lower bound.

I believe the constant of Theorem 49 can be improved. The smallest nonzero $\mu(g)$ value I know of is attained when $n = 2$ and $g = \mathrm{OR}(s[1], s[2])$:

Proposition 50 $\mu(\mathrm{OR}) = 1/3$.

Proof. First, $\mu(\mathrm{OR}) \geq 1/3$, since D_1 can choose $s[1]\,s[2]$ to be 01, 10, or 11 each with probability 1/3; then for any $z \neq 0$ and the unique 0-input $\widehat{s}_0 = 00$, we have $s_1 \cdot z \not\equiv \widehat{s}_0 \cdot z$ with probability at most 2/3. Second, $\mu(\mathrm{OR}) \leq 1/3$, since applying linear programming duality, we can let the pair $(z, \widehat{s}_1)$ equal $(01, 01)$, $(10, 10)$, or $(11, 10)$ each with probability 1/3. Then $0 \equiv s_0 \cdot z \not\equiv \widehat{s}_1 \cdot z \equiv 1$ always, and for any 1-input s_1, we have $s_1 \cdot z \equiv 1 \not\equiv \widehat{s}_0 \cdot z$ with probability 2/3. ■

Finally, I conjecture that uncomputation is unavoidable not just for RFS but for many other recursive problems, such as game-tree evaluation. Formally, the conjecture is that the quantum query complexity of evaluating a game tree increases exponentially with depth as the number of leaves is held constant, even if there is at most one winning move per vertex (so that the tree can be evaluated with zero probability of error).

Chapter 10

Limitations of Quantum Advice

How many classical bits can "really" be encoded into n qubits? Is it n, because of Holevo's Theorem [145]; $2n$, because of dense quantum coding [78] and quantum teleportation [53]; exponentially many, because of quantum fingerprinting [75]; or infinitely many, because amplitudes are continuous? The best general answer to this question is probably *mu*, the Zen word that "unasks" a question.[1]

To a computer scientist, however, it is natural to formalize the question in terms of *quantum one-way communication complexity* [43, 75, 154, 250]. The setting is as follows: Alice has an n-bit string x, Bob has an m-bit string y, and together they wish to evaluate $f(x,y)$ where $f : \{0,1\}^n \times \{0,1\}^m \rightarrow \{0,1\}$ is a Boolean function. After examining her input $x = x_1 \ldots x_n$, Alice can send a single quantum message ρ_x to Bob, whereupon Bob, after examining his input $y = y_1 \ldots y_m$, can choose some basis in which to measure ρ_x. He must then output a claimed value for $f(x,y)$. We are interested in how long Alice's message needs to be, for Bob to succeed with high probability on any x, y pair. Ideally the length will be much smaller than if Alice had to send a classical message.

Communication complexity questions have been intensively studied in theoretical computer science (see the book of Kushilevitz and Nisan [160] for example). In both the classical and quantum cases, though, most attention has focused on *two-way* communication, meaning that Alice and Bob get to send messages back and forth. I believe that the study of one-way quantum communication presents two main advantages. First, many open problems about two-way communication look gruesomely difficult—for example, are the randomized and quantum communication complexities of every total Boolean function polynomially related? We might gain insight into these problems by tackling their one-way analogues first. And second, because of its greater simplicity, the one-way model more directly addresses our opening question: how much "useful stuff" can be packed into a quantum state? Thus, results on one-way communication fall into the quantum information theory tradition initiated by Holevo [145] and others, as much as the communication complexity tradition initiated by Yao [247].

Related to quantum one-way communication is the notion of *quantum advice*. As pointed out by Nielsen and Chuang [182, p.203], there is no compelling physical reason to

[1] Another *mu*-worthy question is, "Where does the power of quantum computing come from? Superposition? Interference? The large size of Hilbert space?"

assume that the starting state of a quantum computer is a computational basis state:[2]

> [W]e know that many systems in Nature 'prefer' to sit in highly entangled states of many systems; might it be possible to exploit this preference to obtain extra computational power? It might be that having access to certain states allows particular computations to be done much more easily than if we are constrained to start in the computational basis.

One way to interpret Nielsen and Chuang's provocative question is as follows. Suppose we could request the *best possible* starting state for a quantum computer, knowing the language to be decided and the input length n but not knowing the input itself.[3] Denote the class of languages that we could then decide by BQP/qpoly—meaning quantum polynomial time, given an arbitrarily-entangled but polynomial-size quantum advice state.[4] How powerful is this class? If BQP/qpoly contained (for example) the NP-complete problems, then we would need to rethink our most basic assumptions about the power of quantum computing. We will see later that quantum advice is closely related to quantum one-way communication, since we can think of an advice state as a one-way message sent to an algorithm by a benevolent "advisor."

This chapter is about the *limitations* of quantum advice and one-way communication. It presents three contributions which are basically independent of one another.

First, Section 10.2 shows that $\mathrm{D}^1(f) = O\left(mQ_2^1(f)\log \mathrm{Q}_2^1(f)\right)$ for any Boolean function f, partial or total. Here $\mathrm{D}^1(f)$ is deterministic one-way communication complexity, $\mathrm{Q}_2^1(f)$ is bounded-error one-way quantum communication complexity, and m is the length of Bob's input. Intuitively, whenever the set of Bob's possible inputs is not too large, Alice can send him a short classical message that lets him learn the outcome of any measurement he would have wanted to make on the quantum message ρ_x. It is interesting that a slightly tighter bound for total functions—$\mathrm{D}^1(f) = O\left(mQ_2^1(f)\right)$—follows easily from a result of Klauck [154] together with a lemma of Sauer [212] about VC-dimension. However, the proof of the latter bound is highly nonconstructive, and seems to fail for partial f.

Using my communication complexity result, Section 10.2.1 shows that BQP/qpoly $\subseteq$ PP/poly—in other words, BQP with polynomial-size quantum advice can be simulated in PP with polynomial-size classical advice.[5] This resolves a question of Harry Buhrman (personal communication), who asked whether quantum advice can be simulated in *any* classical complexity class with short classical advice. A corollary of this containment is that we cannot hope to show an unrelativized separation between quantum and classical advice (that is, that BQP/poly $\neq$ BQP/qpoly), without also showing that PP does not have polynomial-size circuits.

[2] One might object that the starting state is itself the outcome of some computational process, which began no earlier than the Big Bang. However, (1) for all we know highly entangled states were created in the Big Bang, and (2) 14 billion years is a long time.

[3] If we knew the input, we would simply request a starting state that contains the right answer!

[4] BQP/qpoly might remind readers of a better-studied class called QMA (Quantum Merlin-Arthur). But there are two key differences: first, advice can be trusted while proofs cannot; second, proofs can be tailored to a particular input while advice cannot.

[5] Given a complexity class C, the class C/poly consists of all languages decidable by a C machine, given a polynomial-size classical advice string that depends only on the input length. See Chapter 3 for more information about the complexity classes mentioned in this chapter.

What makes this result surprising is that, in the minds of many computer scientists, a quantum state is basically an exponentially long vector. Indeed, this belief seems to fuel skepticism of quantum computing (see Goldreich [128] for example). But given an exponentially long advice string, even a classical computer could decide any language whatsoever. So one might imagine naïvely that quantum advice would let us solve problems that are not even recursively enumerable given classical advice of a similar size! The failure of this naïve intuition supports the view that a quantum superposition over n-bit strings is "more similar" to a probability distribution over n-bit strings than to a 2^n-bit string.

The second contribution of the chapter, in Section 10.3, is an oracle relative to which NP is not contained in BQP/qpoly. Underlying this oracle separation is the first correct proof of a *direct product theorem* for quantum search. Given an N-item database with K marked items, the direct product theorem says that if a quantum algorithm makes $o\left(\sqrt{N}\right)$ queries, then the probability that the algorithm finds all K of the marked items decreases exponentially in K. Notice that such a result does not follow from any existing quantum lower bound. Earlier Klauck [155] had claimed a weaker direct product theorem, based on the hybrid method of Bennett et al. [51], in a paper on quantum time-space tradeoffs for sorting. Unfortunately, Klauck's proof is incorrect. The proof uses the polynomial method of Beals et al. [45], with the novel twist that we examine all *higher* derivatives of a polynomial (not just the first derivative). The proof has already been improved by Klauck, Špalek, and de Wolf [156], who were able to recover and even extend Klauck's original claims about quantum sorting.

The final contribution, in Section 10.4, is a new *trace distance method* for proving lower bounds on quantum one-way communication complexity. Previously there was only one basic lower bound technique: the VC-dimension method of Klauck [154], which relied on lower bounds for quantum random access codes due to Ambainis et al. [32] and Nayak [180]. Using VC-dimension one can show, for example, that $\mathrm{Q}_2^1(\mathrm{DISJ}) = \Omega(n)$, where the *disjointness function* $\mathrm{DISJ} : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}$ is defined by $\mathrm{DISJ}(x,y) = 1$ if and only if $x_i y_i = 0$ for all $i \in \{1,\ldots,n\}$.

For some problems, however, the VC-dimension method yields no nontrivial quantum lower bound. Seeking to make this point vividly, Ambainis posed the following problem. Alice is given two elements x, y of a finite field $\mathbb{F}_p$ (where p is prime); Bob is given another two elements $a, b \in \mathbb{F}_p$. Bob's goal is to output 1 if $y \equiv ax + b \pmod{p}$ and 0 otherwise. For this problem, the VC-dimension method yields no randomized *or* quantum lower bound better than constant. On the other hand, the well-known fingerprinting protocol for the equality function [192] seems to fail for Ambainis' problem, because of the interplay between addition and multiplication. So it is natural to conjecture that the randomized and even quantum one-way complexities are $\Theta(\log p)$—that is, that no nontrivial protocol exists for this problem.

Ambainis posed a second problem in the same spirit. Here Alice is given $x \in \{1,\ldots,N\}$, Bob is given $y \in \{1,\ldots,N\}$, and both players know a subset $S \subset \{1,\ldots,N\}$. Bob's goal is to decide whether $x - y \in S$ where subtraction is modulo N. The conjecture is that if S is chosen uniformly at random with $|S|$ about $\sqrt{N}$, then with high probability the randomized and quantum one-way complexities are both $\Theta(\log N)$.

Using the trace distance method, I am able to show optimal quantum lower bounds

for both of Ambainis' problems. Previously, no nontrivial lower bounds were known even for randomized protocols. The key idea is to consider two probability distributions over Alice's quantum message ρ_x. The first distribution corresponds to x chosen uniformly at random; the second corresponds to x chosen uniformly conditioned on $f(x,y)=1$. These distributions give rise to two mixed states ρ and ρ_y, which Bob must be able to distinguish with non-negligible bias assuming he can evaluate $f(x,y)$. I then show an upper bound on the trace distance $\|\rho-\rho_y\|_{\text{tr}}$, which implies that Bob cannot distinguish the distributions.

Theorem 65 gives a very general condition under which the trace distance method works; Corollaries 66 and 67 then show that the condition is satisfied for Ambainis' two problems. Besides showing a significant limitation of the VC-dimension method, I hope the new method is a non-negligible step towards proving that $\mathrm{R}_2^1(f)=O\left(\mathrm{Q}_2^1(f)\right)$ for all total Boolean functions f, where $\mathrm{R}_2^1(f)$ is randomized one-way complexity. I conclude in Section 10.5 with some open problems.

10.1 Preliminaries

Following standard conventions, I denote by $\mathrm{D}^1(f)$ the deterministic one-way complexity of f, or the minimum number of bits that Alice must send if her message is a function of x. Also, $\mathrm{R}_2^1(f)$, the bounded-error randomized one-way complexity, is the minimum k such that for every x,y, if Alice sends Bob a k-bit message drawn from some distribution $\mathcal{D}_x$, then Bob can output a bit a such that $a=f(x,y)$ with probability at least $2/3$. (The subscript 2 means that the error is two-sided.) The zero-error randomized complexity $\mathrm{R}_0^1(f)$ is similar, except that Bob's answer can never be wrong: he must output $f(x,y)$ with probability at least $1/2$ and otherwise declare failure.

The bounded-error quantum one-way complexity $\mathrm{Q}_2^1(f)$ is the minimum k such that, if Alice sends Bob a mixed state ρ_x of k qubits, there exists a joint measurement of ρ_x and y enabling Bob to output an a such that $a=f(x,y)$ with probability at least $2/3$. The zero-error and exact complexities $\mathrm{Q}_0^1(f)$ and $\mathrm{Q}_E^1(f)$ are defined analogously. Requiring Alice's message to be a pure state would increase these complexities by at most a factor of 2, since by Kraus' Theorem, every k-qubit mixed state can be realized as half of a $2k$-qubit pure state. (Winter [243] has shown that this factor of 2 is tight.) See Klauck [154] for more detailed definitions of quantum and classical one-way communication complexity measures.

It is immediate that $\mathrm{D}^1(f)\geq\mathrm{R}_0^1(f)\geq\mathrm{R}_2^1(f)\geq\mathrm{Q}_2^1(f)$, that $\mathrm{R}_0^1(f)\geq\mathrm{Q}_0^1(f)\geq\mathrm{Q}_2^1(f)$, and that $\mathrm{D}^1(f)\geq\mathrm{Q}_E^1(f)$. Also, for total f, Duriš et al. [101] showed that $\mathrm{R}_0^1(f)=\Theta\left(\mathrm{D}^1(f)\right)$, while Klauck [154] showed that $\mathrm{Q}_E^1(f)=\mathrm{D}^1(f)$ and that $\mathrm{Q}_0^1(f)=\Theta\left(\mathrm{D}^1(f)\right)$. In other words, randomized and quantum messages yield no improvement for total functions if one is unwilling to tolerate a bounded probability of error. This remains true even if Alice and Bob share arbitrarily many EPR pairs [154]. As is often the case, the situation is dramatically different for partial functions: there it is easy to see that $\mathrm{R}_0^1(f)$ can be constant even though $\mathrm{D}^1(f)=\Omega(n)$: let $f(x,y)=1$ if $x_1y_1+\cdots+x_{n/2}y_{n/2}\geq n/4$ and $x_{n/2+1}y_{n/2+1}+\cdots+x_ny_n=0$ and $f(x,y)=0$ if $x_1y_1+\cdots+x_{n/2}y_{n/2}=0$ and $x_{n/2+1}y_{n/2+1}+\cdots+x_ny_n\geq n/4$, promised that one of these is the case.

Moreover, Bar-Yossef, Jayram, and Kerenidis [43] have *almost* shown that $\mathrm{Q}_E^1(f)$

can be exponentially smaller than $\mathrm{R}_2^1(f)$. In particular, they proved that separation for a *relation*, meaning a problem for which Bob has many possible valid outputs. For a partial function f based on their relation, they also showed that $\mathrm{Q}_E^1(f) = \Theta(\log n)$ whereas $\mathrm{R}_0^1(f) = \Theta(\sqrt{n})$; and they conjectured (but did not prove) that $\mathrm{R}_2^1(f) = \Theta(\sqrt{n})$.

10.1.1 Quantum Advice

Informally, BQP/qpoly is the class of languages decidable in polynomial time on a quantum computer, given a polynomial-size quantum advice state that depends only on the input length. I now make the definition more formal.

Definition 51 *A language L is in* BQP/qpoly *if there exists a polynomial-size quantum circuit family $\{C_n\}_{n\geq 1}$, and a polynomial-size family of quantum states $\{|\psi_n\rangle\}_{n\geq 1}$, such that for all $x \in \{0,1\}^n$,*

(i) If $x \in L$ then $q(x) \geq 2/3$, where $q(x)$ is the probability that the first qubit is measured to be $|1\rangle$, after C_n is applied to the starting state $|x\rangle \otimes |0\cdots 0\rangle \otimes |\psi_n\rangle$.

(ii) If $x \notin L$ then $q(x) \leq 1/3$.[6]

The central open question about BQP/qpoly is whether it equals BQP/poly, or BQP with polynomial-size *classical* advice. We do have a candidate for an oracle problem separating the two classes: the *group membership problem* of Watrous [239], which I will describe for completeness. Let G_n be a black box group[7] whose elements are uniquely labeled by n-bit strings, and let H_n be a subgroup of G_n. Both G_n and H_n depend only on the input length n, so we can assume that a nonuniform algorithm knows generating sets for both of them. Given an element $x \in G_n$ as input, the problem is to decide whether $x \in H_n$.

If G_n is "sufficiently nonabelian" and H_n is exponentially large, we do not know how to solve this problem in BQP or even BQP/poly. On the other hand, we can solve it in BQP/qpoly as follows. Let the quantum advice state be an equal superposition over all elements of H_n:

$$|H_n\rangle = \frac{1}{\sqrt{|H_n|}} \sum_{y \in H_n} |y\rangle$$

We can transform $|H_n\rangle$ into

$$|xH_n\rangle = \frac{1}{\sqrt{|H_n|}} \sum_{y \in H_n} |xy\rangle$$

by mapping $|y\rangle|0\rangle$ to $|y\rangle|xy\rangle$ to $\left|y \oplus x^{-1}xy\right\rangle|xy\rangle = |0\rangle|xy\rangle$ for each $y \in H_n$. Our algorithm will first prepare the state $(|0\rangle|H_n\rangle + |1\rangle|xH_n\rangle)/\sqrt{2}$, then apply a Hadamard gate to the

[6] If the starting state is $|x\rangle \otimes |0\cdots 0\rangle \otimes |\varphi\rangle$ for some $|\varphi\rangle \neq |\psi_n\rangle$, then the acceptance probability is not required to lie in $[0,1/3] \cup [2/3,1]$. Therefore, what I call BQP/qpoly corresponds to what Nishimura and Yamakami [186] call BQP/*Qpoly. Also, it does not matter whether the circuit family $\{C_n\}_{n\geq 1}$ is uniform, since we are giving it advice anyway.

[7] In other words, we have a quantum oracle available that given $x, y \in G_n$ outputs xy (i.e. exclusive-OR's xy into an answer register), and that given $x \in G_n$ outputs x^{-1}.

first qubit, and finally measure the first qubit in the standard basis, in order to distinguish the cases $|H_n\rangle = |xH_n\rangle$ and $\langle H_n|xH_n\rangle = 0$ with constant bias. The first case occurs whenever $x \in H_n$, and the second occurs whenever $x \notin H_n$.

Although the group membership problem provides intriguing evidence for the power of quantum advice, we have no idea how to show that it is not also solvable using classical advice. Indeed, apart from a result of Nishimura and Yamakami [186] that $\mathsf{EESPACE} \not\subset \mathsf{BQP/qpoly}$, essentially nothing was known about the class $\mathsf{BQP/qpoly}$ before the work reported here.

10.1.2 The Almost As Good As New Lemma

The following simple lemma, which was implicit in [32], is used three times in this chapter—in Theorems 56, 57, and 64. It says that, if the outcome of measuring a quantum state ρ could be predicted with near-certainty given knowledge of ρ, then measuring ρ will damage it only slightly. Recall that the trace distance $\|\rho - \sigma\|_{\text{tr}}$ between two mixed states ρ and σ equals $\frac{1}{2}\sum_i |\lambda_i|$, where $\lambda_1, \ldots, \lambda_N$ are the eigenvalues of $\rho - \sigma$.

Lemma 52 *Suppose a 2-outcome measurement of a mixed state ρ yields outcome 0 with probability $1-\varepsilon$. Then after the measurement, we can recover a state $\widetilde{\rho}$ such that $\|\widetilde{\rho} - \rho\|_{\text{tr}} \leq \sqrt{\varepsilon}$. This is true even if the measurement is a POVM (that is, involves arbitrarily many ancilla qubits).*

Proof. Let $|\psi\rangle$ be a purification of the entire system (ρ plus ancilla). We can represent any measurement as a unitary U applied to $|\psi\rangle$, followed by a 1-qubit measurement. Let $|\varphi_0\rangle$ and $|\varphi_1\rangle$ be the two possible pure states after the measurement; then $\langle\varphi_0|\varphi_1\rangle = 0$ and $U|\psi\rangle = \alpha|\varphi_0\rangle + \beta|\varphi_1\rangle$ for some α, β such that $|\alpha|^2 = 1 - \varepsilon$ and $|\beta|^2 = \varepsilon$. Writing the measurement result as $\sigma = (1-\varepsilon)|\varphi_0\rangle\langle\varphi_0| + \varepsilon|\varphi_1\rangle\langle\varphi_1|$, it is easy to show that

$$\left\|\sigma - U|\psi\rangle\langle\psi|U^{-1}\right\|_{\text{tr}} = \sqrt{\varepsilon(1-\varepsilon)}.$$

So applying U^{-1} to σ,

$$\left\|U^{-1}\sigma U - |\psi\rangle\langle\psi|\right\|_{\text{tr}} = \sqrt{\varepsilon(1-\varepsilon)}.$$

Let $\widetilde{\rho}$ be the restriction of $U^{-1}\sigma U$ to the original qubits of ρ. Theorem 9.2 of Nielsen and Chuang [182] shows that tracing out a subsystem never increases trace distance, so $\|\widetilde{\rho} - \rho\|_{\text{tr}} \leq \sqrt{\varepsilon(1-\varepsilon)} \leq \sqrt{\varepsilon}$. ■

10.2 Simulating Quantum Messages

Let $f : \{0,1\}^n \times \{0,1\}^m \rightarrow \{0,1\}$ be a Boolean function. In this section I first combine existing results to obtain the relation $\mathrm{D}^1(f) = O\left(mQ_2^1(f)\right)$ for total f, and then prove using a new method that $\mathrm{D}^1(f) = O\left(mQ_2^1(f)\log \mathrm{Q}_2^1(f)\right)$ for all f (partial or total).

Define the *communication matrix* M_f to be a $2^n \times 2^m$ matrix with $f(x,y)$ in the x^{th} row and y^{th} column. Then letting $\text{rows}(f)$ be the number of distinct rows in M_f, the following is immediate.

Proposition 53 *For total f,*

$$\mathrm{D}^1(f) = \lceil \log_2 \mathrm{rows}(f) \rceil,$$
$$\mathrm{Q}_2^1(f) = \Omega(\log \log \mathrm{rows}(f)).$$

Also, let the VC-dimension $\mathrm{VC}(f)$ equal the maximum k for which there exists a $2^n \times k$ submatrix M_g of M_f with $\mathrm{rows}(g) = 2^k$. Then Klauck [154] observed the following, based on a lower bound for quantum random access codes due to Nayak [180].

Proposition 54 (Klauck) $\mathrm{Q}_2^1(f) = \Omega(\mathrm{VC}(f))$ *for total f.*

Now let $\mathrm{cols}(f)$ be the number of distinct columns in M_f. Then Proposition 54 yields the following general lower bound:

Corollary 55 $\mathrm{D}^1(f) = O\left(mQ_2^1(f)\right)$ *for total f, where m is the size of Bob's input.*

Proof. It follows from a lemma of Sauer [212] that

$$\mathrm{rows}(f) \leq \sum_{i=0}^{\mathrm{VC}(f)} \binom{\mathrm{cols}(f)}{i} \leq \mathrm{cols}(f)^{\mathrm{VC}(f)+1}.$$

Hence $\mathrm{VC}(f) \geq \log_{\mathrm{cols}(f)} \mathrm{rows}(f) - 1$, so

$$\begin{aligned}\mathrm{Q}_2^1(f) = \Omega(\mathrm{VC}(f)) &= \Omega\left(\frac{\log \mathrm{rows}(f)}{\log \mathrm{cols}(f)}\right) \\ &= \Omega\left(\frac{\mathrm{D}^1(f)}{m}\right).\end{aligned}$$

■

In particular, $\mathrm{D}^1(f)$ and $\mathrm{Q}_2^1(f)$ are polynomially related for total f, whenever Bob's input is polynomially smaller than Alice's, and Alice's input is not "padded." More formally, $\mathrm{D}^1(f) = O\left(\mathrm{Q}_2^1(f)^{1/(1-c)}\right)$ whenever $m = O(n^c)$ for some $c < 1$ and $\mathrm{rows}(f) = 2^n$ (i.e. all rows of M_f are distinct). For then $\mathrm{D}^1(f) = n$ by Proposition 53, and $\mathrm{Q}_2^1(f) = \Omega\left(\mathrm{D}^1(f)/n^c\right) = \Omega\left(n^{1-c}\right)$ by Corollary 55.

I now give a new method for replacing quantum messages by classical ones when Bob's input is small. Although the best bound I know how to obtain with this method—$\mathrm{D}^1(f) = O\left(mQ_2^1(f) \log \mathrm{Q}_2^1(f)\right)$—is slightly weaker than the $\mathrm{D}^1(f) = O\left(mQ_2^1(f)\right)$ of Corollary 55, our method works for *partial* Boolean functions as well as total ones. It also yields a (relatively) efficient procedure by which Bob can reconstruct Alice's quantum message, a fact I will exploit in Section 10.2.1 to show $\mathsf{BQP/qpoly} \subseteq \mathsf{PP/poly}$. By contrast, the method based on Sauer's Lemma seems to be nonconstructive.

Theorem 56 $\mathrm{D}^1(f) = O\left(mQ_2^1(f) \log \mathrm{Q}_2^1(f)\right)$ *for all f (partial or total).*

Proof. Let $f : \mathcal{D} \rightarrow \{0,1\}$ be a partial Boolean function with $\mathcal{D} \subseteq \{0,1\}^n \times \{0,1\}^m$, and for all $x \in \{0,1\}^n$, let $\mathcal{D}_x = \{y \in \{0,1\}^m : (x,y) \in \mathcal{D}\}$. Suppose Alice can send Bob a quantum state with $\mathrm{Q}_2^1(f)$ qubits, that enables him to compute $f(x,y)$ for any $y \in \mathcal{D}_x$ with error probability at most $1/3$. Then she can also send him a boosted state ρ with $K = O\left(\mathrm{Q}_2^1(f) \log \mathrm{Q}_2^1(f)\right)$ qubits, such that for all $y \in \mathcal{D}_x$,

$$|P_y(\rho) - f(x,y)| \leq \frac{1}{\mathrm{Q}_2^1(f)^{10}},$$

where $P_y(\rho)$ is the probability that some measurement $\Lambda[y]$ yields a '1' outcome when applied to ρ. We can assume for simplicity that ρ is a pure state $|\psi\rangle\langle\psi|$; as discussed in Section 10.1, this increases the message length by at most a factor of 2.

Let $\mathcal{Y}$ be any subset of $\mathcal{D}_x$ satisfying $|\mathcal{Y}| \leq \mathrm{Q}_2^1(f)^2$. Then starting with ρ, Bob can measure $\Lambda[y]$ for each $y \in \mathcal{Y}$ in lexicographic order, reusing the same message state again and again but uncomputing whatever garbage he generates while measuring. Let ρ_t be the state after the t^{th} measurement; thus $\rho_0 = \rho = |\psi\rangle\langle\psi|$. Since the probability that Bob outputs the wrong value of $f(x,y)$ on any given y is at most $1/\mathrm{Q}_2^1(f)^{10}$, Lemma 52 implies that

$$\|\rho_t - \rho_{t-1}\|_{\mathrm{tr}} \leq \sqrt{\frac{1}{\mathrm{Q}_2^1(f)^{10}}} = \frac{1}{\mathrm{Q}_2^1(f)^5}.$$

Since trace distance satisfies the triangle inequality, this in turn implies that

$$\|\rho_t - \rho\|_{\mathrm{tr}} \leq \frac{t}{\mathrm{Q}_2^1(f)^5} \leq \frac{1}{\mathrm{Q}_2^1(f)^3}.$$

Now imagine an "ideal scenario" in which $\rho_t = \rho$ for every t; that is, the measurements do not damage ρ at all. Then the maximum bias with which Bob could distinguish the actual from the ideal scenario is

$$\left\|\rho_0 \otimes \cdots \otimes \rho_{|\mathcal{Y}|-1} - \rho^{\otimes|\mathcal{Y}|}\right\|_{\mathrm{tr}} \leq \frac{|\mathcal{Y}|}{\mathrm{Q}_2^1(f)^3} \leq \frac{1}{\mathrm{Q}_2^1(f)}.$$

So by the union bound, Bob will output $f(x,y)$ for every $y \in \mathcal{Y}$ simultaneously with probability at least

$$1 - \frac{|\mathcal{Y}|}{\mathrm{Q}_2^1(f)^{10}} - \frac{1}{\mathrm{Q}_2^1(f)} \geq 0.9$$

for sufficiently large $\mathrm{Q}_2^1(f)$.

Now imagine that the communication channel is blocked, so Bob has to guess what message Alice wants to send him. He does this by using the K-qubit maximally mixed state I in place of ρ. We can write I as

$$I = \frac{1}{2^K} \sum_{j=1}^{2^K} |\psi_j\rangle\langle\psi_j|,$$

where $|\psi_1\rangle, \ldots, |\psi_{2^K}\rangle$ are orthonormal vectors such that $|\psi_1\rangle = |\psi\rangle$. So if Bob uses the same procedure as above except with I instead of ρ, then for any $\mathcal{Y} \subseteq \mathcal{D}_x$ with $|\mathcal{Y}| \leq \mathrm{Q}_2^1(f)^2$, he will output $f(x,y)$ for every $y \in \mathcal{Y}$ simultaneously with probability at least $0.9/2^K$.

The classical simulation of the quantum protocol is now as follows. Alice's message to Bob consists of $T \leq K$ inputs $y_1, \ldots, y_T \in \mathcal{D}_x$, together with $f(x, y_1), \ldots, f(x, y_T)$.[8] Thus the message length is $mT + T = O\left(mQ_2^1(f) \log Q_2^1(f)\right)$. Here are the semantics of Alice's message: *"Bob, suppose you looped over all $y \in \mathcal{D}_x$ in lexicographic order; and for each one, guessed that $f(x,y) = \text{round}(P_y(I))$, where $\text{round}(p)$ is 1 if $p \geq 1/2$ and 0 if $p < 1/2$. Then y_1 is the first y for which you would guess the wrong value of $f(x,y)$. In general, let I_t be the state obtained by starting from I and then measuring $\Lambda[y_1], \ldots, \Lambda[y_t]$ in that order, given that the outcomes of the measurements are $f(x, y_1), \ldots, f(x, y_t)$ respectively. (Note that I_t is not changed by measurements of every $y \in \mathcal{D}_x$ up to y_t, only by measurements of $y_1, \ldots, y_t$.) If you looped over all $y \in \mathcal{D}_x$ in lexicographic order beginning from y_t, then y_{t+1} is the first y you would encounter for which $\text{round}(P_y(I_t)) \neq f(x,y)$."*

Given the sequence of y_t's as defined above, it is obvious that Bob can compute $f(x,y)$ for any $y \in \mathcal{D}_x$. First, if $y = y_t$ for some t, then he simply outputs $f(x, y_t)$. Otherwise, let t^* be the largest t for which $y_t < y$ lexicographically. Then Bob prepares a classical description of the state I_{t^*}—which he can do since he knows $y_1, \ldots, y_{t^*}$ and $f(x, y_1), \ldots, f(x, y_{t^*})$—and then outputs $\text{round}(P_y(I_{t^*}))$ as his claimed value of $f(x,y)$. Notice that, although Alice uses her knowledge of $\mathcal{D}_x$ to prepare her message, Bob does not need to know $\mathcal{D}_x$ in order to interpret the message. That is why the simulation works for partial as well as total functions.

But why can we assume that the sequence of y_t's stops at y_T for some $T \leq K$? Suppose $T > K$; we will derive a contradiction. Let $\mathcal{Y} = \{y_1, \ldots, y_{K+1}\}$. Then $|\mathcal{Y}| = K + 1 \leq Q_2^1(f)^2$, so we know from previous reasoning that if Bob starts with I and then measures $\Lambda[y_1], \ldots, \Lambda[y_{K+1}]$ in that order, he will observe $f(x, y_1), \ldots, f(x, y_{K+1})$ simultaneously with probability at least $0.9/2^K$. But by the definition of y_t, the probability that $\Lambda[y_t]$ yields the correct outcome is at most $1/2$, conditioned on $\Lambda[y_1], \ldots, \Lambda[y_{t-1}]$ having yielded the correct outcomes. Therefore $f(x, y_1), \ldots, f(x, y_{K+1})$ are observed simultaneously with probability at most $1/2^{K+1} < 0.9/2^K$, contradiction. ■

10.2.1 Simulating Quantum Advice

I now apply the new simulation method to upper-bound the power of quantum advice.

Theorem 57 $\mathsf{BQP/qpoly} \subseteq \mathsf{PP/poly}$.

Proof. For notational convenience, let $L_n(x) = 1$ if input $x \in \{0,1\}^n$ is in language L, and $L_n(x) = 0$ otherwise. Suppose L_n is computed by a BQP machine using quantum advice of length $p(n)$. We will give a PP machine that computes L_n using classical advice of length $O(np(n) \log p(n))$. Because of the close connection between advice and one-way communication, the simulation method will be essentially identical to that of Theorem 56.

By using a boosted advice state on $K = O(p(n) \log p(n))$ qubits, a polynomial-time quantum algorithm A can compute $L_n(x)$ with error probability at most $1/p(n)^{10}$. Now the classical advice to the PP machine consists of $T \leq K$ inputs $x_1, \ldots, x_T \in \{0,1\}^n$,

[8] Strictly speaking, Bob will be able to compute $f(x, y_1), \ldots, f(x, y_T)$ for himself given $y_1, \ldots, y_T$; he does not need Alice to tell him the f values.

together with $L_n(x_1), \ldots, L_n(x_T)$. Let I be the maximally mixed state on K qubits. Also, let $P_x(\rho)$ be the probability that A outputs '1' on input x, given ρ as its advice state. Then x_1 is the lexicographically first input x for which $\text{round}(P_x(I)) \neq L_n(x)$. In general, let I_t be the state obtained by starting with I as the advice and then running A on $x_1, \ldots, x_t$ in that order (uncomputing garbage along the way), if we postselect on A correctly outputting $L_n(x_1), \ldots, L_n(x_t)$. Then x_{t+1} is the lexicographically first $x > x_t$ for which $\text{round}(P_x(I_t)) \neq L_n(x)$.

Given the classical advice, we can compute $L_n(x)$ as follows: if $x \in \{x_1, \ldots, x_T\}$ then output $L_n(x_t)$. Otherwise let t^* be the largest t for which $x_t < x$ lexicographically, and output $\text{round}(P_x(I_{t^*}))$. The proof that this algorithm works is the same as in Theorem 56, and so is omitted for brevity. All that needs to be shown is that the algorithm can be implemented in PP.

Adleman, DeMarrais, and Huang [16] (see also Fortnow and Rogers [116]) showed that BQP $\subseteq$ PP, by using what physicists would call a "Feynman sum-over-histories." Specifically, let C be a polynomial-size quantum circuit that starts in the all-0 state, and that consists solely of Toffoli and Hadamard gates (Shi [217] has shown that this gate set is universal). Also, let α_z be the amplitude of basis state $|z\rangle$ after all gates in C have been applied. We can write α_z as a sum of exponentially many contributions, $a_1 + \cdots + a_N$, where each a_i is a rational real number computable in classical polynomial time. So by evaluating the sum

$$|\alpha_z|^2 = \sum_{i,j=1}^{N} a_i a_j,$$

putting positive and negative terms on "opposite sides of the ledger," a PP machine can check whether $|\alpha_z|^2 > \beta$ for any rational constant β. It follows that a PP machine can also check whether

$$\sum_{z \,:\, S_1(z)} |\alpha_z|^2 > \sum_{z \,:\, S_0(z)} |\alpha_z|^2$$

(or equivalently, whether $\Pr[S_1] > \Pr[S_0]$) for any classical polynomial-time predicates S_1 and S_0.

Now suppose the circuit C does the following, in the case $x \notin \{x_1, \ldots, x_T\}$. It first prepares the K-qubit maximally mixed state I (as half of a $2K$-qubit pure state), and then runs A on $x_1, \ldots, x_{t^*}, x$ in that order, using I as its advice state. The claimed values of $L_n(x_1), \ldots, L_n(x_{t^*}), L_n(x)$ are written to output registers but not measured. For $i \in \{0, 1\}$, let the predicate $S_i(z)$ hold if and only if basis state $|z\rangle$ contains the output sequence $L_n(x_1), \ldots, L_n(x_{t^*}), i$. Then it is not hard to see that

$$P_x(I_{t^*}) = \frac{\Pr[S_1]}{\Pr[S_1] + \Pr[S_0]},$$

so $P_x(I_{t^*}) > 1/2$ and hence $L_n(x) = 1$ if and only if $\Pr[S_1] > \Pr[S_0]$. Since the case $x \in \{x_1, \ldots, x_T\}$ is trivial, this shows that $L_n(x)$ is computable in PP/poly. ■

Let me make five remarks about Theorem 57. First, for the same reason that Theorem 56 works for partial as well as total functions, one actually obtains the stronger

result that PromiseBQP/qpoly $\subseteq$ PromisePP/poly, where PromiseBQP and PromisePP are the promise-problem versions of BQP and PP respectively.

Second, as pointed out to me by Lance Fortnow, a corollary of Theorem 57 is that we cannot hope to show an unrelativized separation between BQP/poly and BQP/qpoly, without also showing that PP does not have polynomial-size circuits. For BQP/poly $\neq$ BQP/qpoly clearly implies that P/poly $\neq$ PP/poly. But the latter then implies that PP $\not\subset$ P/poly, since assuming PP $\subset$ P/poly we could also obtain polynomial-size circuits for a language $L \in$ PP/poly by defining a new language $L' \in$ PP, consisting of all (x, a) pairs such that the PP machine would accept x given advice string a. The reason this works is that PP is a syntactically defined class.

Third, initially I showed that BQP/qpoly $\subseteq$ EXP/poly, by using a simulation in which an EXP machine keeps track of a subspace H of the advice Hilbert space to which the 'true' advice state must be close. In that simulation, the classical advice specifies inputs $x_1, \ldots, x_T$ for which $\dim(H)$ is at least halved; the observation that $\dim(H)$ must be at least 1 by the end then implies that $T \leq K = O(p(n) \log p(n))$, meaning that the advice is of polynomial size. The huge improvement from EXP to PP came solely from working with *measurement outcomes* and their *probabilities* instead of with *subspaces* and their *dimensions*. We can compute the former using the same "Feynman sum-over-histories" that Adleman et al. [16] used to show BQP $\subseteq$ PP, but I could not see any way to compute the latter without explicitly storing and diagonalizing exponentially large matrices.

Fourth, assuming BQP/poly $\neq$ BQP/qpoly, Theorem 57 is *almost* the best result of its kind that one could hope for, since the only classes known to lie between BQP and PP and not known to equal either are obscure ones such as AWPP [116]. Initially the theorem seemed to me to prove something stronger, namely that BQP/qpoly $\subseteq$ PostBQP/poly. Here PostBQP is the class of languages decidable by polynomial-size quantum circuits with *post-selection*—meaning the ability to measure a qubit that has a nonzero probability of being $|1\rangle$, and then *assume* that the measurement outcome will be $|1\rangle$. Clearly PostBQP lies somewhere between BQP and PP; one can think of it as a quantum analogue of the classical complexity class $\mathsf{BPP}_{\mathsf{path}}$ [142]. It turns out, however, that PostBQP = PP (see Chapter 15).

Fifth, it is clear that Adleman et al.'s BQP $\subseteq$ PP result [16] can be extended to show that PQP = PP. Here PQP is the quantum analogue of PP—that is, quantum polynomial time but where the probability of a correct answer need only be bounded above 1/2, rather than above 2/3. It has been asked whether Theorem 57 could similarly be extended to show that PQP/qpoly = PP/poly. The answer is no—for indeed, PQP/qpoly contains every language whatsoever! To see this, given any function $L_n : \{0,1\}^n \rightarrow \{0,1\}$, let the quantum advice state be

$$|\psi_n\rangle = \frac{1}{2^{n/2}} \sum_{x \in \{0,1\}^n} |x\rangle |L_n(x)\rangle .$$

Then a PQP algorithm to compute L_n is as follows: given an input $x \in \{0,1\}^n$, first measure $|\psi_n\rangle$ in the standard basis. If $|x\rangle |L_n(x)\rangle$ is observed, output $L_n(x)$; otherwise output a uniform random bit.

10.3 A Direct Product Theorem for Quantum Search

Can quantum computers solve NP-complete problems in polynomial time? In the early days of quantum computing, Bennett et al. [51] gave an oracle relative to which $\mathsf{NP} \not\subset \mathsf{BQP}$, providing what is still the best evidence we have that the answer is no. It is easy to extend Bennett et al.'s result to give an oracle relative to which $\mathsf{NP} \not\subset \mathsf{BQP/poly}$; that is, NP is hard even for nonuniform quantum algorithms. But when we try to show $\mathsf{NP} \not\subset \mathsf{BQP/qpoly}$ relative to an oracle, a new difficulty arises: even if the oracle encodes 2^n exponentially hard search problems for each input length n, the quantum advice, being an "exponentially large object" itself, might somehow encode information about all 2^n problems. We need to argue that even if so, only a miniscule fraction of that information can be extracted by measuring the advice.

How does one prove such a statement? As it turns out, the task can be reduced to proving a *direct product theorem* for quantum search. This is a theorem that in its weakest form says the following: given N items, K of which are marked, if we lack enough time to find even *one* marked item, then the probability of finding all K items decreases exponentially in K. For intuitively, suppose there were a quantum advice state that let us efficiently find any one of K marked items. Then by "guessing" the advice (i.e. replacing it by a maximally mixed state), and then using the guessed advice multiple times, we could efficiently find all K of the items with a success probability that our direct product theorem shows is impossible. This reduction is formalized in Theorem 64.

But what about the direct product theorem itself? It seems like it should be trivial to prove—for surely there are no devious correlations by which success in finding one marked item leads to success in finding all the others! So it is surprising that even a weak direct product theorem eluded proof for years. In 2001, Klauck [155] gave an attempted proof using the hybrid method of Bennett et al. [51]. His motivation was to show a limitation of space-bounded quantum sorting algorithms. Unfortunately, Klauck's proof is fallacious.[9]

In this section I give the first correct proof of a direct product theorem, based on the polynomial method of Beals et al. [45]. Besides showing that $\mathsf{NP} \not\subset \mathsf{BQP/qpoly}$ relative to an oracle, my result can be used to recover the conclusions in [155] about the hardness of quantum sorting (see Klauck, Špalek, and de Wolf [156] for details). I expect the result to have other applications as well.

I will need the following lemma of Beals et al. [45].

Lemma 58 (Beals et al.) *Suppose a quantum algorithm makes T queries to an oracle string $X \in \{0,1\}^N$, and accepts with probability $A(X)$. Then there exists a real polynomial p, of degree at most $2T$, such that*

$$p(i) = \underset{|X|=i}{\mathrm{EX}} [A(X)]$$

for all integers $i \in \{0,\ldots,N\}$, where $|X|$ denotes the Hamming weight of X.

[9] Specifically, the last sentence in the proof of Lemma 5 in [155] ("Clearly this probability is at least $\mathrm{Q}_x(p_x - \alpha)$") is not justified by what precedes it.

Lemma 58 implies that, to lower-bound the number of queries T made by a quantum algorithm, it suffices to lower-bound $\deg(p)$, where p is a real polynomial representing the algorithm's expected acceptance probability. As an example, any quantum algorithm that computes the OR function on N bits, with success probability at least $2/3$, yields a polynomial p such that $p(0) \in [0,1/3]$ and $p(i) \in [2/3,1]$ for all integers $i \in \{1,\ldots,N\}$. To lower-bound the degree of such a polynomial, one can use an inequality proved by A. A. Markov in 1890 ([172]; see also [203]):

Theorem 59 (A. A. Markov) *Given a real polynomial p and constant $N > 0$, let $r^{(0)} = \max_{x\in[0,N]} |p(x)|$ and $r^{(1)} = \max_{x\in[0,N]} |p'(x)|$. Then*

$$\deg(p) \geq \sqrt{\frac{Nr^{(1)}}{2r^{(0)}}}.$$

Theorem 59 deals with the entire range $[0,N]$, whereas in our setting $p(x)$ is constrained only at the integer points $x \in \{0,\ldots,N\}$. But as shown in [104, 184, 204], this is not a problem. For by elementary calculus, $p(0) \leq 1/3$ and $p(1) \geq 2/3$ imply that $p'(x) \geq 1/3$ for some real $x \in [0,1]$, and therefore $r^{(1)} \geq 1/3$. Furthermore, let x^* be a point in $[0,N]$ where $|p(x^*)| = r^{(0)}$. Then $p(\lfloor x^* \rfloor) \in [0,1]$ and $p(\lceil x^* \rceil) \in [0,1]$ imply that $r^{(1)} \geq 2\left(r^{(0)} - 1\right)$. Thus

$$\deg(p) \geq \sqrt{\frac{Nr^{(1)}}{2r^{(0)}}} \geq \sqrt{\frac{N \max\left\{1/3, 2\left(r^{(0)}-1\right)\right\}}{2r^{(0)}}} = \Omega\left(\sqrt{N}\right).$$

This is the proof of Beals et al. [45] that quantum search requires $\Omega\left(\sqrt{N}\right)$ queries.

When proving a direct product theorem, one can no longer apply Theorem 59 so straightforwardly. The reason is that the success probabilities in question are extremely small, and therefore the maximum derivative $r^{(1)}$ could also be extremely small. Fortunately, though, one can still prove a good lower bound on the degree of the relevant polynomial p. The key is to look not just at the first derivative of p, but at higher derivatives.

To start, we need a lemma about the behavior of functions under repeated differentiation.

Lemma 60 *Let $f : \mathbb{R} \rightarrow \mathbb{R}$ be an infinitely differentiable function such that for some positive integer K, we have $f(i) = 0$ for all $i \in \{0,\ldots,K-1\}$ and $f(K) = \delta > 0$. Also, let $r^{(m)} = \max_{x\in[0,N]} \left|f^{(m)}(x)\right|$, where $f^{(m)}(x)$ is the m^{th} derivative of f evaluated at x (thus $f^{(0)} = f$). Then $r^{(m)} \geq \delta/m!$ for all $m \in \{0,\ldots,K\}$.*

Proof. We claim, by induction on m, that there exist $K-m+1$ points $0 \leq x_0^{(m)} < \cdots < x_{K-m}^{(m)} \leq K$ such that $f^{(m)}\left(x_i^{(m)}\right) = 0$ for all $i \leq K-m-1$ and $f^{(m)}\left(x_{K-m}^{(m)}\right) \geq \delta/m!$. If we define $x_i^{(0)} = i$, then the base case $m = 0$ is immediate from the conditions of the lemma. Suppose the claim is true for m; then by elementary calculus, for all $i \leq K-m-2$

there exists a point $x_i^{(m+1)} \in \left(x_i^{(m)}, x_{i+1}^{(m)}\right)$ such that $f^{(m+1)}\left(x_i^{(m+1)}\right) = 0$. Notice that $x_i^{(m+1)} \geq x_i^{(m)} \geq \cdots \geq x_i^{(0)} = i$. So there is also a point $x_{K-m-1}^{(m+1)} \in \left(x_{K-m-1}^{(m)}, x_{K-m}^{(m)}\right)$ such that

$$\begin{aligned} f^{(m+1)}\left(x_{K-m-1}^{(m+1)}\right) &\geq \frac{f^{(m)}\left(x_{K-m}^{(m)}\right) - f^{(m)}\left(x_{K-m-1}^{(m)}\right)}{x_{K-m}^{(m)} - x_{K-m-1}^{(m)}} \\ &\geq \frac{\delta/m! - 0}{K - (K-m-1)} \\ &= \frac{\delta}{(m+1)!}. \end{aligned}$$

■

With the help of Lemma 60, one can sometimes lower-bound the degree of a real polynomial even its first derivative is small throughout the region of interest. To do so, I will use the following generalization of A. A. Markov's inequality (Theorem 59), which was proved by A. A. Markov's younger brother V. A. Markov in 1892 ([173]; see also [203]).

Theorem 61 (V. A. Markov) *Given a real polynomial p of degree d and positive real number N, let $r^{(m)} = \max_{x\in[0,N]} \left|p^{(m)}(x)\right|$. Then for all $m \in \{1, \ldots, d\}$,*

$$\begin{aligned} r^{(m)} &\leq \left(\frac{2r^{(0)}}{N}\right)^m T_d^{(m)}(1) \\ &\leq \left(\frac{2r^{(0)}}{N}\right)^m \frac{d^2\left(d^2-1^2\right)\left(d^2-2^2\right)\cdot\cdots\cdot\left(d^2-(m-1)^2\right)}{1\cdot 3\cdot 5\cdot\cdots\cdot(2m-1)}. \end{aligned}$$

Here $T_d(x) = \cos(d \arccos x)$ is the d^{th} Chebyshev polynomial of the first kind.

As demonstrated below, combining Theorem 61 with Lemma 60 yields a lower bound on $\deg(p)$.

Lemma 62 *Let p be a real polynomial such that*

(i) $p(x) \in [0,1]$ at all integer points $x \in \{0, \ldots, N\}$, and

(ii) for some positive integer $K \leq N$ and real $\delta > 0$, we have $p(K) = \delta$ and $p(i) = 0$ for all $i \in \{0, \ldots, K-1\}$.

Then $\deg(p) = \Omega\left(\sqrt{N\delta^{1/K}}\right)$.

Proof. Let $p^{(m)}$ and $r^{(m)}$ be as in Theorem 61. Then for all $m \in \{1, \ldots, \deg(p)\}$, Theorem 61 yields

$$r^{(m)} \leq \left(\frac{2r^{(0)}}{N}\right)^m \frac{\deg(p)^{2m}}{1\cdot 3\cdot 5\cdot\cdots\cdot(2m-1)}$$

Rearranging,

$$\deg(p) \geq \sqrt{\frac{N}{2r^{(0)}} \left(1 \cdot 3 \cdot 5 \cdot \cdots \cdot (2m-1) \cdot r^{(m)}\right)^{1/m}}$$

for all $m \geq 1$ (if $m > \deg(p)$ then $r^{(m)} = 0$ so the bound is trivial).

There are now two cases. First suppose $r^{(0)} \geq 2$. Then as discussed previously, condition (i) implies that $r^{(1)} \geq 2\left(r^{(0)} - 1\right)$, and hence that

$$\deg(p) \geq \sqrt{\frac{Nr^{(1)}}{2r^{(0)}}} \geq \sqrt{\frac{N\left(r^{(0)}-1\right)}{r^{(0)}}} = \Omega\left(\sqrt{N}\right)$$

by Theorem 59. Next suppose $r^{(0)} < 2$. Then $r^{(m)} \geq \delta/m!$ for all $m \leq K$ by Lemma 60. So setting $m = K$ yields

$$\deg(p) \geq \sqrt{\frac{N}{4}\left(1 \cdot 3 \cdot 5 \cdot \cdots \cdot (2K-1) \cdot \frac{\delta}{K!}\right)^{1/K}} = \Omega\left(\sqrt{N\delta^{1/K}}\right).$$

Either way we are done. ■

Strictly speaking, one does not need the full strength of Theorem 61 to prove a lower bound on $\deg(p)$ that suffices for an oracle separation between NP and BQP/qpoly. For one can show a "rough-and-ready" version of V. A. Markov's inequality by applying A. A. Markov's inequality (Theorem 59) repeatedly, to $p, p^{(1)}, p^{(2)}$, and so on. This yields

$$r^{(m)} \leq \frac{2}{N}\deg(p)^2 r^{(m-1)} \leq \left(\frac{2}{N}\deg(p)^2\right)^m r^{(0)}$$

for all m. If $\deg(p)$ is small, then this upper bound on $r^{(m)}$ contradicts the lower bound of Lemma 60. However, the lower bound on $\deg(p)$ that one gets from A. A. Markov's inequality is only $\Omega\left(\sqrt{N\delta^{1/K}/K}\right)$, as opposed to $\Omega\left(\sqrt{N\delta^{1/K}}\right)$ from Lemma 62.[10]

Shortly after seeing my proof of a weak direct product theorem, Klauck, Špalek, and de Wolf [156] managed to improve the lower bound on $\deg(p)$ to the essentially tight $\Omega\left(\sqrt{NK\delta^{1/K}}\right)$. In particular, their bound implies that δ decreases exponentially in K whenever $\deg(p) = o\left(\sqrt{NK}\right)$. They obtained this improvement by *factoring* p instead of differentiating it as in Lemma 60.

In any case, a direct product theorem follows trivially from what has already been said.

Theorem 63 (Direct Product Theorem) *Suppose a quantum algorithm makes T queries to an oracle string $X \in \{0,1\}^N$. Let δ be the minimum probability, over all X with Hamming weight $|X| = K$, that the algorithm finds all K of the '1' bits. Then $\delta \leq \left(cT^2/N\right)^K$ for some constant c.*

[10] An earlier version of this chapter claimed to prove $\deg(p) = \Omega\left(\sqrt{NK}/\log^{3/2}(1/\delta)\right)$, by applying *Bernstein's inequality* [56] rather than A. A. Markov's to all derivatives $p^{(m)}$. I have since discovered a flaw in that argument. In any case, the Bernstein lower bound is both unnecessary for an oracle separation, and superseded by the later results of Klauck et al. [156].

Proof. Have the algorithm accept if it finds K or more '1' bits and reject otherwise. Let $p(i)$ be the expected probability of acceptance if X is drawn uniformly at random subject to $|X| = i$. Then we know the following about p:

(i) $p(i) \in [0,1]$ at all integer points $i \in \{0,\ldots,N\}$, since $p(i)$ is a probability.

(ii) $p(i) = 0$ for all $i \in \{0,\ldots,K-1\}$, since there are not K marked items to be found.

(iii) $p(K) \geq \delta$.

Furthermore, Lemma 58 implies that p is a polynomial in i satisfying $\deg(p) \leq 2T$. It follows from Lemma 62 that $T = \Omega\left(\sqrt{N\delta^{1/K}}\right)$, or rearranging, that $\delta \leq \left(cT^2/N\right)^K$. ■

The desired oracle separation can now be proven using standard complexity theory tricks.

Theorem 64 *There exists an oracle relative to which* $\mathsf{NP} \not\subset \mathsf{BQP/qpoly}$.

Proof. Given an oracle $A : \{0,1\}^* \rightarrow \{0,1\}$, define the language L_A by $(y,z) \in L_A$ if and only if $y \leq z$ lexicographically and there exists an x such that $y \leq x \leq z$ and $A(x) = 1$. Clearly $L_A \in \mathsf{NP}^A$ for all A. We argue that for some A, no $\mathsf{BQP/qpoly}$ machine M with oracle access to A can decide L_A. Without loss of generality we assume M is fixed, so that only the advice states $\{|\psi_n\rangle\}_{n\geq 1}$ depend on A. We also assume the advice is boosted, so that M's error probability on any input (y,z) is $2^{-\Omega\left(n^2\right)}$.

Choose a set $S \subset \{0,1\}^n$ subject to $|S| = 2^{n/10}$; then for all $x \in \{0,1\}^n$, set $A(x) = 1$ if and only if $x \in S$. We claim that by using M, an algorithm could find all $2^{n/10}$ elements of S with high probability after only $2^{n/10}\,\mathrm{poly}(n)$ queries to A. Here is how: first use binary search (repeatedly halving the distance between y and z) to find the lexicographically first element of S. By Lemma 52, the boosted advice state $|\psi_n\rangle$ is good for $2^{\Omega\left(n^2\right)}$ uses, so this takes only $\mathrm{poly}(n)$ queries. Then use binary search to find the lexicographically second element, and so on until all elements have been found.

Now replace $|\psi_n\rangle$ by the maximally mixed state as in Theorem 56. This yields an algorithm that uses no advice, makes $2^{n/10}\,\mathrm{poly}(n)$ queries, and finds all $2^{n/10}$ elements of S with probability $2^{-O(\mathrm{poly}(n))}$. But taking $\delta = 2^{-O(\mathrm{poly}(n))}$, $T = 2^{n/10}\,\mathrm{poly}(n)$, $N = 2^n$, and $K = 2^{n/10}$, such an algorithm would satisfy $\delta \gg \left(cT^2/N\right)^K$, which violates the bound of Theorem 63. ■

Indeed one can show that $\mathsf{NP} \not\subset \mathsf{BQP/qpoly}$ relative a random oracle with probability 1.[11]

10.4 The Trace Distance Method

This section introduces a new method for proving lower bounds on quantum one-way communication complexity. Unlike in Section 10.2, here I do not try to simulate quantum

[11] First group the oracle bits into polynomial-size blocks as Bennett and Gill [54] do, then use the techniques of Chapter 6 to show that the acceptance probability is a low-degree univariate polynomial in the number of all-0 blocks. The rest of the proof follows Theorem 64.

protocols using classical ones. Instead I prove lower bounds for quantum protocols directly, by reasoning about the trace distance between two possible distributions over Alice's quantum message (that is, between two mixed states). The result is a method that works even if Alice's and Bob's inputs are the same size.

I first state the method as a general theorem; then, in Section 10.4.1, I apply the theorem to prove lower bounds for two problems of Ambainis. Let $\|\mathcal{D} - \mathcal{E}\|$ denote the variation distance between probability distributions $\mathcal{D}$ and $\mathcal{E}$.

Theorem 65 *Let $f : \{0,1\}^n \times \{0,1\}^m \rightarrow \{0,1\}$ be a total Boolean function. For each $y \in \{0,1\}^m$, let $\mathcal{A}_y$ be a distribution over $x \in \{0,1\}^n$ such that $f(x,y) = 1$. Let $\mathcal{B}$ be a distribution over $y \in \{0,1\}^m$, and let $\mathcal{D}_k$ be the distribution over $(\{0,1\}^n)^k$ formed by first choosing $y \in \mathcal{B}$ and then choosing k samples independently from $\mathcal{A}_y$. Suppose that $\Pr_{x \in \mathcal{D}_1, y \in \mathcal{B}}[f(x,y) = 0] = \Omega(1)$ and that $\left\|\mathcal{D}_2 - \mathcal{D}_1^2\right\| \leq \delta$. Then $\mathrm{Q}_2^1(f) = \Omega(\log 1/\delta)$.*

Proof. Suppose that if Alice's input is x, then she sends Bob the ℓ-qubit mixed state ρ_x. Suppose also that for every $x \in \{0,1\}^n$ and $y \in \{0,1\}^m$, Bob outputs $f(x,y)$ with probability at least $2/3$. Then by amplifying a constant number of times, Bob's success probability can be made $1 - \varepsilon$ for any constant $\varepsilon > 0$. So with $L = O(\ell)$ qubits of communication, Bob can distinguish the following two cases with constant bias:

Case I. y was drawn from $\mathcal{B}$ and x from $\mathcal{D}_1$.

Case II. y was drawn from $\mathcal{B}$ and x from $\mathcal{A}_y$.

For in Case I, we assumed that $f(x,y) = 0$ with constant probability, whereas in Case II, $f(x,y) = 1$ always. An equivalent way to say this is that with constant probability over y, Bob can distinguish the mixed states $\rho = \mathrm{EX}_{x \in \mathcal{D}_1}[\rho_x]$ and $\rho_y = \mathrm{EX}_{x \in \mathcal{A}_y}[\rho_x]$ with constant bias. Therefore

$$\mathop{\mathrm{EX}}_{y \in \mathcal{B}}\left[\|\rho - \rho_y\|_{\mathrm{tr}}\right] = \Omega(1).$$

We need an upper bound on the trace distance $\|\rho - \rho_y\|_{\mathrm{tr}}$ that is more amenable to analysis. Let $\lambda_1, \ldots, \lambda_{2^L}$ be the eigenvalues of $\rho - \rho_y$. Then

$$\begin{aligned}
\|\rho - \rho_y\|_{\mathrm{tr}} &= \frac{1}{2}\sum_{i=1}^{2^L} |\lambda_i| \\
&\leq \frac{1}{2}\sqrt{2^L \sum_{i=1}^{2^L} \lambda_i^2} \\
&= 2^{L/2-1}\sqrt{\sum_{i,j=1}^{2^L} \left|(\rho)_{ij} - (\rho_y)_{ij}\right|^2}
\end{aligned}$$

where $(\rho)_{ij}$ is the (i,j) entry of ρ. Here the second line uses the Cauchy-Schwarz inequality, and the third line uses the unitary invariance of the Frobenius norm.

We claim that

$$\mathop{\mathrm{EX}}_{y \in \mathcal{B}}\left[\sum_{i,j=1}^{2^L} \left|(\rho)_{ij} - (\rho_y)_{ij}\right|^2\right] \leq 2\delta.$$

From this claim it follows that

$$\begin{aligned}\underset{y\in\mathcal{B}}{\mathrm{EX}}\left[\left\|\rho-\rho_y\right\|_{\mathrm{tr}}\right] &\leq 2^{L/2-1}\underset{y\in\mathcal{B}}{\mathrm{EX}}\left[\sqrt{\sum_{i,j=1}^{2^L}\left|(\rho)_{ij}-(\rho_y)_{ij}\right|^2}\right]\\ &\leq 2^{L/2-1}\sqrt{\underset{y\in\mathcal{B}}{\mathrm{EX}}\left[\sum_{i,j=1}^{2^L}\left|(\rho)_{ij}-(\rho_y)_{ij}\right|^2\right]}\\ &\leq \sqrt{2^{L-1}\delta}.\end{aligned}$$

Therefore the message length L must be $\Omega\left(\log 1/\delta\right)$ to ensure that $\mathrm{EX}_{y\in\mathcal{B}}\left[\left\|\rho-\rho_y\right\|_{\mathrm{tr}}\right]=\Omega\left(1\right)$.

Let us now prove the claim. We have

$$\begin{aligned}\underset{y\in\mathcal{B}}{\mathrm{EX}}\left[\sum_{i,j=1}^{2^L}\left|(\rho)_{ij}-(\rho_y)_{ij}\right|^2\right] &= \sum_{i,j=1}^{2^L}\left(\left|(\rho)_{ij}\right|^2-2\,\mathrm{Re}\left((\rho)_{ij}^*\underset{y\in\mathcal{B}}{\mathrm{EX}}\left[(\rho_y)_{ij}\right]\right)+\underset{y\in\mathcal{B}}{\mathrm{EX}}\left[\left|(\rho_y)_{ij}\right|^2\right]\right)\\ &= \sum_{i,j=1}^{2^L}\left(\underset{y\in\mathcal{B}}{\mathrm{EX}}\left[\left|(\rho_y)_{ij}\right|^2\right]-\left|(\rho)_{ij}\right|^2\right),\end{aligned}$$

since $\mathrm{EX}_{y\in\mathcal{B}}\left[(\rho_y)_{ij}\right]=(\rho)_{ij}$. For a given (i,j) pair,

$$\begin{aligned}\underset{y\in\mathcal{B}}{\mathrm{EX}}\left[\left|(\rho_y)_{ij}\right|^2\right]-\left|(\rho)_{ij}\right|^2 &= \underset{y\in\mathcal{B}}{\mathrm{EX}}\left[\left|\underset{x\in\mathcal{A}_y}{\mathrm{EX}}\left[(\rho_x)_{ij}\right]\right|^2\right]-\left|\underset{x\in\mathcal{D}_1}{\mathrm{EX}}\left[(\rho_x)_{ij}\right]\right|^2\\ &= \underset{y\in\mathcal{B},x,z\in\mathcal{A}_y}{\mathrm{EX}}\left[(\rho_x)_{ij}^*\,(\rho_z)_{ij}\right]-\underset{x,z\in\mathcal{D}_1}{\mathrm{EX}}\left[(\rho_x)_{ij}^*\,(\rho_z)_{ij}\right]\\ &= \sum_{x,z}\left(\Pr_{\mathcal{D}_2}\left[x,z\right]-\Pr_{\mathcal{D}_1^2}\left[x,z\right]\right)(\rho_x)_{ij}^*\,(\rho_z)_{ij}\,.\end{aligned}$$

Now for all x,z,

$$\left|\sum_{i,j=1}^{2^L}(\rho_x)_{ij}^*\,(\rho_z)_{ij}\right|\leq\sum_{i,j=1}^{2^L}\left|(\rho_x)_{ij}\right|^2\leq 1.$$

Hence

$$\begin{aligned}\sum_{x,z}\left(\Pr_{\mathcal{D}_2}\left[x,z\right]-\Pr_{\mathcal{D}_1^2}\left[x,z\right]\right)\sum_{i,j=1}^{2^L}(\rho_x)_{ij}^*\,(\rho_z)_{ij} &\leq \sum_{x,z}\left(\Pr_{\mathcal{D}_2}\left[x,z\right]-\Pr_{\mathcal{D}_1^2}\left[x,z\right]\right)\\ &= 2\left\|\mathcal{D}_2-\mathcal{D}_1^2\right\|\\ &\leq 2\delta,\end{aligned}$$

and we are done. ■

The difficulty in extending Theorem 65 to partial functions is that the distribution $\mathcal{D}_1$ might not make sense, since it might assign a nonzero probability to some x for which $f\left(x,y\right)$ is undefined.

10.4.1 Applications

In this subsection I apply Theorem 65 to prove lower bounds for two problems of Ambainis. To facilitate further research and to investigate the scope of our method, I state the problems in a more general way than Ambainis did. Given a group G, the *coset problem* $\mathrm{Coset}(G)$ is defined as follows. Alice is given a left coset C of a subgroup in G, and Bob is given an element $y \in G$. Bob must output 1 if $y \in C$ and 0 otherwise. By restricting the group G, we obtain many interesting and natural problems. For example, if p is prime then $\mathrm{Coset}(\mathbb{Z}_p)$ is just the equality problem, so the protocol of Rabin and Yao [192] yields $\mathrm{Q}_2^1(\mathrm{Coset}(\mathbb{Z}_p)) = \Theta(\log\log p)$.

Theorem 66 $\mathrm{Q}_2^1\left(\mathrm{Coset}\left(\mathbb{Z}_p^2\right)\right) = \Theta(\log p)$.

Proof. The upper bound is obvious. For the lower bound, it suffices to consider a function f_p defined as follows. Alice is given $\langle x, y\rangle \in \mathbb{F}_p^2$ and Bob is given $\langle a, b\rangle \in \mathbb{F}_p^2$; then

$$f_p(x, y, a, b) = \begin{cases} 1 & \text{if } y \equiv ax + b \pmod p \\ 0 & \text{otherwise.} \end{cases}$$

Let $\mathcal{B}$ be the uniform distribution over $\langle a, b\rangle \in \mathbb{F}_p^2$, and let $\mathcal{A}_{a,b}$ be the uniform distribution over $\langle x, y\rangle$ such that $y \equiv ax + b \pmod p$. Thus $\mathcal{D}_1$ is the uniform distribution over $\langle x, y\rangle \in \mathbb{F}_p^2$; note that

$$\Pr_{\langle x,y\rangle \in \mathcal{D}_1, \langle a,b\rangle \in \mathcal{B}}\left[f_p(x, y, a, b) = 0\right] = 1 - \frac{1}{p}.$$

But what about the distribution $\mathcal{D}_2$, which is formed by first drawing $\langle a, b\rangle \in \mathcal{B}$, and then drawing $\langle x, y\rangle$ and $\langle z, w\rangle$ independently from $\mathcal{A}_{a,b}$? Given a pair $\langle x, y\rangle, \langle z, w\rangle \in \mathbb{F}_p^2$, there are three cases regarding the probability of its being drawn from $\mathcal{D}_2$:

(1) $\langle x, y\rangle = \langle z, w\rangle$ (p^2 pairs). In this case

$$\begin{aligned} \Pr_{\mathcal{D}_2}\left[\langle x, y\rangle, \langle z, w\rangle\right] &= \sum_{\langle a,b\rangle \in \mathbb{F}_p^2} \Pr\left[\langle a, b\rangle\right] \Pr\left[\langle x, y\rangle, \langle z, w\rangle \mid \langle a, b\rangle\right] \\ &= p\left(\frac{1}{p^2} \cdot \frac{1}{p^2}\right) = \frac{1}{p^3}. \end{aligned}$$

(2) $x \neq z$ ($p^4 - p^3$ pairs). In this case there exists a unique $\langle a^*, b^*\rangle$ such that $y \equiv a^*x + b^* \pmod p$ and $w \equiv a^*z + b^* \pmod p$, so

$$\begin{aligned} \Pr_{\mathcal{D}_2}\left[\langle x, y\rangle, \langle z, w\rangle\right] &= \Pr\left[\langle a^*, b^*\rangle\right] \Pr\left[\langle x, y\rangle, \langle z, w\rangle \mid \langle a^*, b^*\rangle\right] \\ &= \frac{1}{p^2} \cdot \frac{1}{p^2} = \frac{1}{p^4}. \end{aligned}$$

(3) $x = z$ but $y \neq w$ ($p^3 - p^2$ pairs). In this case $\Pr_{\mathcal{D}_2}\left[\langle x, y\rangle, \langle z, w\rangle\right] = 0$.

Putting it all together,

$$\begin{aligned}\left\|\mathcal{D}_2 - \mathcal{D}_1^2\right\| &= \frac{1}{2}\left(p^2\left|\frac{1}{p^3} - \frac{1}{p^4}\right| + \left(p^4 - p^3\right)\left|\frac{1}{p^4} - \frac{1}{p^4}\right| + \left(p^3 - p^2\right)\left|0 - \frac{1}{p^4}\right|\right) \\ &= \frac{1}{p} - \frac{1}{p^2}.\end{aligned}$$

So taking $\delta = 1/p - 1/p^2$, we have $\mathrm{Q}_2^1\left(\mathrm{Coset}\left(\mathbb{Z}_p^2\right)\right) = \Omega\left(\log\left(1/\delta\right)\right) = \Omega\left(\log p\right)$ by Theorem 65. ■

I now consider Ambainis' second problem. Given a group G and nonempty set $S \subset G$ with $|S| \leq |G|/2$, the *subset problem* $\mathrm{Subset}\left(G, S\right)$ is defined as follows. Alice is given $x \in G$ and Bob is given $y \in G$; then Bob must output 1 if $xy \in S$ and 0 otherwise.

Let $\mathcal{M}$ be the distribution over $st^{-1} \in G$ formed by drawing s and t uniformly and independently from S. Then let $\Delta = \left\|\mathcal{M} - \mathcal{D}_1\right\|$, where $\mathcal{D}_1$ is the uniform distribution over G.

Proposition 67 *For all G, S such that $|S| \leq |G|/2$,*

$$\mathrm{Q}_2^1\left(\mathrm{Subset}\left(G, S\right)\right) = \Omega\left(\log 1/\Delta\right).$$

Proof. Let $\mathcal{B}$ be the uniform distribution over $y \in G$, and let $\mathcal{A}_y$ be the uniform distribution over x such that $xy \in S$. Thus $\mathcal{D}_1$ is the uniform distribution over $x \in G$; note that

$$\Pr_{x \in \mathcal{D}_1, y \in \mathcal{B}}\left[xy \notin S\right] = 1 - \frac{|S|}{|G|} \geq \frac{1}{2}.$$

We have

$$\begin{aligned}\left\|\mathcal{D}_2 - \mathcal{D}_1^2\right\| &= \frac{1}{2}\sum_{x,z \in G}\left|\frac{\left|\left\{y \in G, s, t \in S : xy = s, zy = t\right\}\right|}{|G|\,|S|^2} - \frac{1}{|G|^2}\right| \\ &= \frac{1}{2}\sum_{x,z \in G}\left|\frac{\left|\left\{s, t \in S : xz^{-1} = st^{-1}\right\}\right|}{|S|^2} - \frac{1}{|G|^2}\right| \\ &= \frac{1}{2}\sum_{x \in G}\left|\frac{\left|\left\{s, t \in S : x = st^{-1}\right\}\right|}{|S|^2} - \frac{1}{|G|}\right| \\ &= \frac{1}{2}\sum_{x \in G}\left|\Pr_{\mathcal{M}}\left[x\right] - \frac{1}{|G|}\right| \\ &= \left\|\mathcal{M} - \mathcal{D}_1\right\| \\ &= \Delta.\end{aligned}$$

Therefore $\log\left(1/\delta\right) = \Omega\left(\log 1/\Delta\right)$. ■

Having lower-bounded $\mathrm{Q}_2^1\left(\mathrm{Subset}\left(G, S\right)\right)$ in terms of $1/\Delta$, it remains only to upper-bound the variation distance Δ. The following proposition implies that for all constants $\varepsilon > 0$, if S is chosen uniformly at random subject to $|S| = |G|^{1/2+\varepsilon}$, then $\mathrm{Q}_2^1\left(\mathrm{Subset}\left(G, S\right)\right) = \Omega\left(\log\left(|G|\right)\right)$ with constant probability over S.

Theorem 68 *For all groups G and integers $K \in \{1, \ldots, |G|\}$, if $S \subset G$ is chosen uniformly at random subject to $|S| = K$, then $\Delta = O\left(\sqrt{|G|}/K\right)$ with $\Omega(1)$ probability over S.*

Proof. We have

$$\Delta = \frac{1}{2} \sum_{x \in G} \left| \Pr_{\mathcal{M}}[x] - \frac{1}{|G|} \right| \leq \frac{\sqrt{|G|}}{2} \sqrt{\sum_{x \in G} \left(\Pr_{\mathcal{M}}[x] - \frac{1}{|G|} \right)^2}$$

by the Cauchy-Schwarz inequality. We claim that

$$\mathop{\mathrm{EX}}_{S} \left[\sum_{x \in G} \left(\Pr_{\mathcal{M}}[x] - \frac{1}{|G|} \right)^2 \right] \leq \frac{c}{K^2}$$

for some constant c. From this it follows by Markov's inequality that

$$\Pr_{S} \left[\sum_{x \in G} \left(\Pr_{\mathcal{M}}[x] - \frac{1}{|G|} \right)^2 \geq \frac{2c}{K^2} \right] \leq \frac{1}{2}$$

and hence

$$\Delta \leq \frac{\sqrt{|G|}}{2} \sqrt{\frac{2c}{K^2}} = O\left(\frac{\sqrt{|G|}}{K} \right)$$

with probability at least $1/2$.

Let us now prove the claim. We have

$$\Pr_{\mathcal{M}}[x] = \Pr_{i,j} \left[s_i s_j^{-1} = x \right] = \Pr_{i,j}[s_i = x s_j],$$

where $S = \{s_1, \ldots, s_K\}$ and i, j are drawn uniformly and independently from $\{1, \ldots, K\}$. So by linearity of expectation,

$$\begin{aligned} \mathop{\mathrm{EX}}_{S} \left[\sum_{x \in G} \left(\Pr_{\mathcal{M}}[x] - \frac{1}{|G|} \right)^2 \right] &= \mathop{\mathrm{EX}}_{S} \left[\sum_{x \in G} \left(\left(\Pr_{i,j}[s_i = x s_j] \right)^2 - \frac{2}{|G|} \Pr_{i,j}[s_i = x s_j] + \frac{1}{|G|^2} \right) \right] \\ &= \sum_{x \in G} \left(\frac{1}{K^4} \sum_{i,j,k,l=1}^{K} p_{x,ijkl} \right) - \frac{2}{|G|} \sum_{x \in G} \left(\frac{1}{K^2} \sum_{i,j=1}^{K} p_{x,ij} \right) + \frac{1}{|G|} \end{aligned}$$

where

$$\begin{aligned} p_{x,ij} &= \Pr_{S}[s_i = x s_j], \\ p_{x,ijkl} &= \Pr_{S}[s_i = x s_j \wedge s_k = x s_l]. \end{aligned}$$

First we analyze $p_{x,ij}$. Let $\mathrm{ord}(x)$ be the order of x in G. Of the K^2 possible ordered pairs (i, j), there are K pairs with the "pattern" ii (meaning that $i = j$), and $K(K-1)$ pairs with the pattern ij (meaning that $i \neq j$). If $\mathrm{ord}(x) = 1$ (that is, x is the

Pattern	Number of such 4-tuples	$\mathrm{ord}\,(x)=1$	$\mathrm{ord}\,(x)=2$	$\mathrm{ord}\,(x)>2$
iiii,iikk	K^2	1	0	0
ijij	$K\,(K-1)$	0	$\frac{1}{\lvert G\rvert-1}$	$\frac{1}{\lvert G\rvert-1}$
ijji	$K\,(K-1)$	0	$\frac{1}{\lvert G\rvert-1}$	0
iiil,iiki,ijii,ijjj	$4K\,(K-1)$	0	0	0
ijki,ijjk	$2K\,(K-1)\,(K-2)$	0	0	$\frac{1}{(\lvert G\rvert-1)(\lvert G\rvert-2)}$
iikl,ijkk,ijik,ijkj	$4K\,(K-1)\,(K-2)$	0	0	0
ijkl	$K\,(K-1)\,(K-2)\,(K-3)$	0	$\frac{1}{(\lvert G\rvert-1)(\lvert G\rvert-3)}$	$\frac{1}{(\lvert G\rvert-1)(\lvert G\rvert-3)}$

Table 10.1: Expressions for $p_{x,ijkl}$

identity), then we have $p_{x,ij}=\Pr_S\left[s_i=s_j\right]$, so $p_{x,ij}=1$ under the pattern ii, and $p_{x,ij}=0$ under the pattern ij. On the other hand, if $\mathrm{ord}\,(x)>1$, then $p_{x,ij}=0$ under the pattern ii, and $p_{x,ij}=\frac{1}{|G|-1}$ under the pattern ij. So

$$\frac{1}{K^2}\sum_{x\in G}\sum_{i,j=1}^{K}p_{x,ij}=\frac{1}{K^2}\left(K+(|G|-1)\,\frac{K\,(K-1)}{|G|-1}\right)=1.$$

Though unnecessarily cumbersome, the above analysis was a warmup for the more complicated case of $p_{x,ijkl}$. Table 10.1 lists the expressions for $p_{x,ijkl}$, given $\mathrm{ord}\,(x)$ and the pattern of (i,j,k,l).

Let r be the number of $x\in G$ such that $\mathrm{ord}\,(x)=2$, and let $r'=|G|-r-1$ be the number such that $\mathrm{ord}\,(x)>2$. Then

$$\begin{aligned}\frac{1}{K^4}\sum_{x\in G}\sum_{i,j,k,l=1}^{K}p_{x,ijkl}&=\frac{1}{K^4}\left(\begin{array}{c}K^2+(2r+r')\,\frac{K(K-1)}{|G|-1}+2r'\frac{K(K-1)(K-2)}{(|G|-1)(|G|-2)}\\+(r+r')\,\frac{K(K-1)(K-2)(K-3)}{(|G|-1)(|G|-3)}\end{array}\right)\\&\leq\frac{1}{|G|-3}+O\left(\frac{1}{K^2}\right)\end{aligned}$$

using the fact that $K\leq|G|$.

Putting it all together,

$$\mathop{\mathrm{EX}}_{S}\left[\sum_{x\in G}\left(\Pr_{\mathcal{M}}\left[x\right]-\frac{1}{|G|}\right)^2\right]\leq\frac{1}{|G|-3}+O\left(\frac{1}{K^2}\right)-\frac{2}{|G|}+\frac{1}{|G|}=O\left(\frac{1}{K^2}\right)$$

and we are done. ■

From fingerprinting one also has the following upper bound. Let q be the periodicity of S, defined as the number of distinct sets $gS=\{gs:s\in S\}$ where $g\in G$.

Proposition 69 $\mathrm{R}_2^1\left(\mathrm{Subset}\,(G,S)\right)=O\left(\log|S|+\log\log q\right)$.

Proof. Assume for simplicity that $q=|G|$; otherwise we could reduce to a subgroup $H\leq G$ with $|H|=q$. The protocol is as follows: Alice draws a uniform random prime

p from the range $\left[|S|^2 \log^2 |G|, 2|S|^2 \log^2 |G|\right]$; she then sends Bob the pair $(p, x \bmod p)$ where x is interpreted as an integer. This takes $O(\log |S| + \log\log |G|)$ bits. Bob outputs 1 if and only if there exists a $z \in G$ such that $zy \in S$ and $x \equiv z \pmod p$. To see the protocol's correctness, observe that if $x \neq z$, then there at most $\log |G|$ primes p such that $x - z \equiv 0 \pmod p$, whereas the relevant range contains $\Omega\left(\frac{|S|^2 \log^2 |G|}{\log(|S| \log |G|)}\right)$ primes. Therefore, if $xy \notin S$, then by the union bound

$$\Pr_p [\exists z : zy \in S, x \equiv z \pmod p] = O\left(|S| \log |G| \frac{\log(|S| \log |G|)}{|S|^2 \log^2 |G|}\right) = o(1).$$

■

10.5 Open Problems

Are $\mathrm{R}_2^1(f)$ and $\mathrm{Q}_2^1(f)$ polynomially related for every total Boolean function f? Also, can we exhibit *any* asymptotic separation between these measures? The best separation I know of is a factor of 2: for the equality function we have $\mathrm{R}_2^1(\mathrm{EQ}) \geq (1 - o(1)) \log_2 n$, whereas Winter [243] has shown that $\mathrm{Q}_2^1(\mathrm{EQ}) \leq (1/2 + o(1)) \log_2 n$ using a protocol involving mixed states.[12] This factor-2 savings is tight for equality: a simple counting argument shows that $\mathrm{Q}_2^1(\mathrm{EQ}) \geq (1/2 - o(1)) \log_2 n$; and although the usual randomized protocol for equality [192] uses $(2 + o(1)) \log_2 n$ bits, there exist protocols based on error-correcting codes that use only $\log_2(cn) = \log_2 n + O(1)$ bits. All of this holds for any constant error probability $0 < \varepsilon < 1/2$.

Can we lower-bound $\mathrm{Q}_2^1(\mathrm{Coset}(G))$ for groups other than $\mathbb{Z}_p^2$ (such as $\mathbb{Z}_2^n$, or nonabelian groups)? Also, can we characterize $\mathrm{Q}_2^1(\mathrm{Subset}(G, S))$ for all sets S, closing the gap between the upper and lower bounds?

Is there an oracle relative to which $\mathsf{BQP/poly} \neq \mathsf{BQP/qpoly}$?

Can we give oracles relative to which $\mathsf{NP} \cap \mathsf{coNP}$ and SZK are not contained in $\mathsf{BQP/qpoly}$? Even more ambitiously, can we prove a direct product theorem for quantum query complexity that applies to any partial or total function (not just search)?

For all f (partial or total), is $\mathrm{R}_2^1(f) = O(\sqrt{n})$ whenever $\mathrm{Q}_2^1(f) = O(\log n)$? In other words, is the separation of Bar-Yossef et al. [43] the best possible?

Can the result $\mathrm{D}^1(f) = O\left(m\mathrm{Q}_2^1(f) \log \mathrm{Q}_2^1(f)\right)$ for partial f be improved to $\mathrm{D}^1(f) = O\left(m\mathrm{Q}_2^1(f)\right)$? I do not even know how to rule out $\mathrm{D}^1(f) = O\left(m + \mathrm{Q}_2^1(f)\right)$.

In the Simultaneous Messages (SM) model, there is no direct communication between Alice and Bob; instead, Alice and Bob both send messages to a third party called the *referee*, who then outputs the function value. The complexity measure is the sum of the two message lengths. Let $\mathrm{R}_2^{||}(f)$ and $\mathrm{Q}_2^{||}(f)$ be the randomized and quantum bounded-error SM complexities of f respectively, and let $\mathrm{R}_2^{||,\mathrm{pub}}(f)$ be the randomized SM complexity if Alice and Bob share an arbitrarily long random string. Building on work by Buhrman et al. [75], Yao [250] showed that $\mathrm{Q}_2^{||}(f) = O(\log n)$ whenever $\mathrm{R}_2^{||,\mathrm{pub}}(f) = O(1)$. He then

[12] If we restrict ourselves to pure states, then $(1 - o(1)) \log_2 n$ qubits are needed. Based on that fact, a previous version of this chapter claimed incorrectly that $\mathrm{Q}_2^1(\mathrm{EQ}) \geq (1 - o(1)) \log_2 n$.

asked about the other direction: for some $\varepsilon > 0$, does $\mathrm{R}_2^{||,\mathrm{pub}}(f) = O\left(n^{1/2-\varepsilon}\right)$ whenever $\mathrm{Q}_2^{||}(f) = O(\log n)$, and does $\mathrm{R}_2^{||}(f) = O\left(n^{1-\varepsilon}\right)$ whenever $\mathrm{Q}_2^{||}(f) = O(\log n)$? In an earlier version of this chapter, I showed that $\mathrm{R}_2^{||}(f) = O\left(\sqrt{n}\left(\mathrm{R}_2^{||,\mathrm{pub}}(f) + \log n\right)\right)$, which means that a positive answer to Yao's first question would imply a positive answer to the second. Later I learned that Yao independently proved the same result [249]. Here I ask a related question: can $\mathrm{Q}_2^{||}(f)$ ever be exponentially smaller than $\mathrm{R}_2^{||,\mathrm{pub}}(f)$? (Buhrman et al. [75] showed that $\mathrm{Q}_2^{||}(f)$ can be exponentially smaller than $\mathrm{R}_2^{||}(f)$.) Iordanis Kerenidis has pointed out to me that, based on the hidden matching problem of Bar-Yossef et al. [43] discussed in Section 10.1, one can define a *relation* for which $\mathrm{Q}_2^{||}(f)$ is exponentially smaller than $\mathrm{R}_2^{||,\mathrm{pub}}(f)$. However, as in the case of $\mathrm{Q}_2^{1}(f)$ versus $\mathrm{R}_2^{1}(f)$, it remains to extend that result to functions.

Chapter 11

Summary of Part I

From my unbiased perspective, quantum lower bounds are some of the deepest results to have emerged from the study of quantum computing and information. These results tell us that many problems we thought were intractable based on classical intuition, really *are* intractable according to our best theory of the physical world. On the other hand, the *reasons* for intractability are much more subtle than in the classical case. In some sense, this has to be true—for otherwise the reasons would apply even to those problems for which dramatic quantum speedups exist.

We currently have two methods for proving lower bounds on quantum query complexity: the polynomial method of Beals et al. [45], and the adversary method of Ambainis [27]. The preceding chapters have illustrated what, borrowing from Wigner [242], we might call the "unreasonable effectiveness" of these methods. Both continue to work far outside of their original design specs—whether by proving *classical* lower bounds, lower bounds for exponentially small success probabilities (as in the direct product theorem), or polynomial lower bounds for quantities that have "no right" to be polynomials (as in the collision and set comparison problems). Yet the two methods also have complementary limitations. The adversary method is useless when the relevant probability gaps are small, or when every 0-input differs from every 1-input in a constant fraction of locations. Likewise, the polynomial method cannot be applied to problems that lack permutation symmetry, at least using the techniques we currently know. Thus, perhaps the most important open problem in quantum lower bounds is to develop a new method that overcomes the limitations of both the polynomial and the adversary methods.[1]

In keeping with the theme of this thesis, I end Part I by listing some classical intuitions about computing, that a hypothetical being from Conway's Game of Life could safely carry into the quantum universe.

- The collision problem is not that much easier than unordered search. For despite being extremely far from any one-to-one function, a random two-to-one function still *looks* one-to-one unless we do an expensive search for collisions.
- Finding a local minimum of a function is not that much easier than finding a global

[1] Along these lines, Barnum, Saks, and Szegedy [44] have given what in some sense is a provably optimal method, but their method (based on semidefinite programming) seems too difficult to apply directly.

minimum. This is because the paths leading to local minima could be exponentially long.

- If we want to distinguish an input X from the set of all Y such that $f(Y) \neq f(X)$, then there is nothing much better to do than to query nonadaptively according to the minimax strategy.
- The difficulty of recursive Fourier sampling increases exponentially with the height of the tree.
- Given n unrelated instances of a problem, but only enough time to solve $o(n)$ of them, the probability of succeeding on all n instances decreases exponentially with n.
- NP-complete problems are probably hard, even with the help of polynomial-size advice.

Part II

Models and Reality

LS: So you believe quantum mechanics?

Me: Of course I do!

LS: So a thousand years from now, people will still be doing quantum mechanics?

Me: Well... um... I guess so...

—Conversation between me and Lee Smolin

Chapter 12

Skepticism of Quantum Computing

> "QC of the sort that factors long numbers seems firmly rooted in science fiction ... The present attitude would be analogous to, say, Maxwell selling the Daemon of his famous thought experiment as a path to cheaper electricity from heat."
>
> —Leonid Levin [165]

Quantum computing presents a dilemma: is it reasonable to study a type of computer that has never been built, and might never be built in one's lifetime? Some researchers strongly believe the answer is 'no.' Their objections generally fall into four categories:

(A) There is a fundamental physical reason why large quantum computers can never be built.

(B) Even if (A) fails, large quantum computers will never be built in practice.

(C) Even if (A) and (B) fail, the speedup offered by quantum computers is of limited theoretical interest.

(D) Even if (A), (B), and (C) fail, the speedup is of limited practical value.[1]

The objections can be classified along two axes, as in Table 12.1.

This chapter focuses on objection (A), that quantum computing is impossible for a fundamental physical reason. Among computer scientists, this objection is most closely

[1]Because of the 'even if' clauses, the objections seem to me logically independent, so that there are 16 possible positions regarding them (or 15 if one is against quantum computing). I ignore the possibility that no speedup exists, in other words that BPP = BQP. By 'large quantum computer' I mean any computer much faster than its best classical simulation, as a result of asymptotic complexity rather than the speed of elementary operations. Such a computer need not be universal; it might be specialized for (say) factoring.

	Theoretical	**Practical**
Physical	(A)	(B)
Algorithmic	(C)	(D)

Table 12.1: Four objections to quantum computing.

associated with Leonid Levin [165].[2] The following passage captures much of the flavor of his critique:

> The major problem [with quantum computing] is the requirement that basic quantum equations hold to multi-hundredth if not millionth decimal positions where the significant digits of the relevant quantum amplitudes reside. We have never seen a physical law valid to over a dozen decimals. Typically, every few new decimal places require major rethinking of most basic concepts. Are quantum amplitudes still complex numbers to such accuracies or do they become quaternions, colored graphs, or sick-humored gremlins? [165]

Among other things, Levin argues that quantum computing is analogous to the unit-cost arithmetic model, and should be rejected for essentially the same reasons; that claims to the contrary rest on a confusion between metric and topological approximation; that quantum fault-tolerance theorems depend on extravagant assumptions; and that even if a quantum computer failed, we could not measure its state to prove a breakdown of quantum mechanics, and thus would be unlikely to learn anything new.

A few responses to Levin's arguments can be offered immediately. First, even classically, one can flip a coin a thousand times to produce probabilities of order 2^{-1000}. Should one dismiss such probabilities as unphysical? At the very least, it is not obvious that amplitudes should behave differently than probabilities with respect to error—since both evolve linearly, and neither is directly observable.

Second, if Levin believes that quantum mechanics will fail, but is agnostic about what will replace it, then his argument can be turned around. How do we know that the successor to quantum mechanics will limit us to BPP, rather than letting us solve (say) PSPACE-complete problems? This is more than a logical point. Abrams and Lloyd [15] argue that a wide class of nonlinear variants of the Schrödinger equation would allow NP-complete and even #P-complete problems to be solved in polynomial time. And Penrose [189], who proposed a model for 'objective collapse' of the wavefunction, believes that his proposal takes us outside the set of computable functions entirely!

Third, to falsify quantum mechanics, it would suffice to show that a quantum computer evolved to *some* state far from the state that quantum mechanics predicts. Measuring the exact state is unnecessary. Nobel prizes have been awarded in the past 'merely' for falsifying a previously held theory, rather than replacing it by a new one. An example is the physics Nobel awarded to Fitch [110] and Cronin [89] in 1980 for discovering CP symmetry violation.

Perhaps the key to understanding Levin's unease about quantum computing lies in his remark that "we have never seen a physical law valid to over a dozen decimals." Here he touches on a serious epistemological question: *How far should we extrapolate from today's experiments to where quantum mechanics has never been tested?* I will try to address this question by reviewing the evidence for quantum mechanics. For my purposes it will not

[2] More recently, Oded Goldreich [128] has also put forward an argument against quantum computing. Compared to Levin's arguments, Goldreich's is easily understood: he believes that states arising in Shor's algorithm have exponential "non-degeneracy" and therefore take exponential time to prepare, and that there is no burden on those who hold this view to suggest a definition of non-degeneracy.

suffice to declare the predictions of quantum mechanics "verified to one part in a trillion," because we have to distinguish at least three different *types* of prediction: *interference*, *entanglement*, and *Schrödinger cats*. Let us consider these in turn.

(1) **Interference.** If the different paths that an electron could take in its orbit around a nucleus did not interfere destructively, canceling each other out, then electrons would not have quantized energy levels. So being accelerating electric charges, they would lose energy and spiral into their respective nuclei, and all matter would disintegrate. That this has not happened—together with the results of (for example) single-photon double-slit experiments—is compelling evidence for the reality of quantum interference.

(2) **Entanglement.** One might accept that a single particle's position is described by a wave in three-dimensional phase space, but deny that two particles are described by a wave in *six*-dimensional phase space. However, the Bell inequality experiments of Aspect et al. [37] and successors have convinced all but a few physicists that quantum entanglement exists, can be maintained over large distances, and cannot be explained by local hidden-variable theories.

(3) **Schrödinger Cats.** Accepting two- and three-particle entanglement is not the same as accepting that whole molecules, cats, humans, and galaxies can be in coherent superposition states. However, recently Arndt et al. [35] have performed the double-slit interference experiment using C_{60} molecules (buckyballs) instead of photons; while Friedman et al. [119] have found evidence that a superconducting current, consisting of billions of electrons, can enter a coherent superposition of flowing clockwise around a coil and flowing counterclockwise (see Leggett [164] for a survey of such experiments). Though short of cats, these experiments at least allow us to say the following: *if we could build a general-purpose quantum computer with as many components as have already been placed into coherent superposition, then on certain problems, that computer would outperform any computer in the world today.*

Having reviewed some of the evidence for quantum mechanics, we must now ask what alternatives have been proposed that might also explain the evidence. The simplest alternatives are those in which quantum states "spontaneously collapse" with some probability, as in the GRW (Ghirardi-Rimini-Weber) theory [123].[3] The drawbacks of the GRW theory include violations of energy conservation, and parameters that must be fine-tuned to avoid conflicting with experiments. More relevant for us, though, is that the collapses postulated by the theory are only in the position basis, so that quantum information stored in internal degrees of freedom (such as spin) is unaffected. Furthermore, even if we extended the theory to collapse those internal degrees, large quantum computers could still be built. For the theory predicts roughly one collapse per particle per 10^{15} seconds, with a collapse affecting everything in a 10^{-7}-meter vicinity. So even in such a vicinity, one could perform a computation involving (say) 10^{10} particles for 10^5 seconds. Finally, as pointed

[3]Penrose [189] has proposed another such theory, but as mentioned earlier, his theory suggests that the quantum computing model is *too* restrictive.

out to me by Rob Spekkens, standard quantum error-correction techniques might be used to overcome even GRW-type decoherence.

A second class of alternatives includes those of 't Hooft [227] and Wolfram [246], in which something like a deterministic cellular automaton underlies quantum mechanics. On the basis of his theory, 't Hooft predicts that "[i]t will never be possible to construct a 'quantum computer' that can factor a large number faster, and within a smaller region of space, than a classical machine would do, if the latter could be built out of parts at least as large and as slow as the Planckian dimensions" [227]. Similarly, Wolfram states that "[i]ndeed within the usual formalism [of quantum mechanics] one can construct quantum computers that may be able to solve at least a few specific problems exponentially faster than ordinary Turing machines. But particularly after my discoveries ... I strongly suspect that even if this is formally the case, it will still not turn out to be a true representation of ultimate physical reality, but will instead just be found to reflect various idealizations made in the models used so far" [246, p.771].

The obvious question then is how these theories account for Bell inequality violations. I confess to being unable to understand 't Hooft's answer to this question, except that he believes that the usual notions of causality and locality might no longer apply in quantum gravity. As for Wolfram's theory, which involves "long-range threads" to account for Bell inequality violations, I will show in Section 12.1 below that it fails Wolfram's own desiderata of causal and relativistic invariance.

12.1 Bell Inequalities and Long-Range Threads

> This section is excerpted from my review [1] of Stephen Wolfram's *A New Kind of Science* [246].

The most interesting chapter of *A New Kind of Science* is the ninth, on 'Fundamental Physics.' Here Wolfram confronts general relativity and quantum mechanics, arguably the two most serious challenges to the deterministic, cellular-automaton-based view of nature that he espouses. Wolfram conjectures that spacetime is discrete at the Planck scale, of about 10^{-33} centimeters or 10^{-43} seconds. This conjecture is not new, and has received considerable attention recently in connection with the holographic principle [65] from black hole thermodynamics, which Wolfram does not discuss. But are new ideas offered to substantiate the conjecture?

For Wolfram, spacetime is a *causal network*, in which events are vertices and edges specify the dependence relations between events. Pages 486–496 and 508–515 discuss in detail how to generate such a network from a simple set of rules. In particular, we could start with a finite undirected 'space graph' G. We then posit a set of update rules, each of which replaces a subgraph by another subgraph with the same number of outgoing edges. The new subgraph must preserve any symmetries of the old one. Then each event in the causal network corresponds to an application of an update rule. If updating event B becomes possible as a result of event A, then we draw an edge from A to B.

Properties of space are defined in terms of G. For example, if the number of vertices in G at distance at most n from any given vertex grows as n^D, then space can be

said to have dimension D. (As for formalizing this definition, Wolfram says only that there are "some subtleties. For example, to find a definite volume growth rate one does still need to take some kind of limit—and one needs to avoid sampling too many or too few" vertices (p. 1030).) Similarly, Wolfram argues that the curvature information needed for general relativity, in particular the Ricci tensor, can be read from the connectivity pattern of G. Interestingly, to make the model as simple as possible, Wolfram does not associate a bit to each vertex of G, representing (say) the presence or absence of a particle. Instead particles are localized structures, or 'tangles,' in G.

An immediate problem is that one might obtain many nonequivalent causal networks, depending on the order in which update rules are applied to G. Wolfram calls a set of rules that allows such nondeterministic evolution a 'multiway system.' He recognizes, but rejects, a possible connection to quantum mechanics:

> The notion of 'many-figured time' has been discussed since the 1950s in the context of the many-worlds interpretation of quantum mechanics. There are some similarities to the multiway systems that I consider here. But an important difference is that while in the many-worlds approach, branchings are associated with possible observation or measurement events, what I suggest here is that they could be an intrinsic feature of even the very lowest-level rules for the universe (p. 1035-6).

It is unclear exactly what distinction is being drawn: is there any physical event that is not associated with a *possible* observation or measurement? In any case, Wolfram opts instead for rule sets that are 'causal invariant': that is, that yield the same causal network regardless of the order in which rules are applied. As noted by Wolfram, a sufficient (though not necessary) condition for causal invariance is that no 'replaceable' subgraph overlaps itself or any other replaceable subgraph.

Wolfram points out an immediate analogy to special relativity, wherein observers do not in general agree on the order in which spacelike separated events occur, yet agree on any final outcome of the events. He is vague, though, about how (say) the Lorentz transformations might be derived in a causal network model:

> There are many subtleties here, and indeed to explain the details of what is going on will no doubt require quite a few new and rather abstract concepts. But the general picture that I believe will emerge is that when particles move faster they will appear to have more nodes associated with them (p. 529).

Wolfram is "certainly aware that many physicists will want to know more details," he says in the endnotes, about how a discrete model of the sort he proposes can reproduce known features of physics. But, although he chose to omit technical formalism from the presentation, "[g]iven my own personal background in theoretical physics it will come as no surprise that I have often used such formalism in the process of working out what I describe in these sections" (p. 1043). The paradox is obvious: if technical formalism would help convince physicists of his ideas, then what could Wolfram lose by including it, say in the endnotes? If, on the other hand, such formalism is irrelevant, then why does Wolfram even mention having used it?

Physicists' hunger for details will likely grow further when they read the section on 'Quantum Phenomena' (p. 537–545). Here Wolfram maintains that quantum mechanics is only an approximation to an underlying classical (and most likely deterministic) theory. Many physicists have sought such a theory, from Einstein to (in modern times) 't Hooft [227]. But a series of results, beginning in the 1960's, has made it clear that such a theory comes at a price. I will argue that, although Wolfram discusses these results, he has not understood what they actually entail.

To begin, Wolfram is *not* advocating a hidden-variable approach such as Bohmian mechanics, in which the state vector is supplemented by an 'actual' eigenstate of a particular observable. Instead he thinks that, at the lowest level, the state vector is not needed at all; it is merely a useful construct for describing some (though presumably not all) higher-level phenomena. Indeterminacy arises because of one's inability to know the exact state of a system:

> [I]f one knew all of the underlying details of the network that makes up our universe, it should always be possible to work out the result of any measurement. I strongly believe that the initial conditions for the universe were quite simple. But like many of the processes we have seen in this book, the evolution of the universe no doubt intrinsically generates apparent randomness. And the result is that most aspects of the network that represents the current state of our universe will seem essentially random (p. 543).

Similarly, Wolfram explains as follows why an electron has wave properties: "... a network which represents our whole universe must also include us as observers. And this means that there is no way that we can look at the network from the outside and see the electron as a definite object" (p. 538). An obvious question then is how Wolfram accounts for the possibility of quantum computing, assuming $\mathsf{BPP} \neq \mathsf{BQP}$. He gives an answer in the final chapter:

> Indeed within the usual formalism [of quantum mechanics] one can construct quantum computers that may be able to solve at least a few specific problems exponentially faster than ordinary Turing machines. But particularly after my discoveries in Chapter 9 ['Fundamental Physics'], I strongly suspect that even if this is formally the case, it will still not turn out to be a true representation of ultimate physical reality, but will instead just be found to reflect various idealizations made in the models used so far (p. 771).

In the endnotes, though, where he explains quantum computing in more detail, Wolfram seems to hedge about which idealizations he has in mind:

> It does appear that only modest precision is needed for the initial amplitudes. And it seems that perturbations from the environment can be overcome using versions of error-correcting codes. But it remains unclear just what might be needed actually to perform for example the final measurements required (p. 1148).

One might respond that, with or without quantum computing, Wolfram's proposals can be ruled out on the simpler ground that they disallow Bell inequality violations. However, Wolfram puts forward an imaginative hypothesis to account for bipartite entanglement. When two particles (or 'tangles' in the graph G) collide, long-range 'threads' may form between them, which remain in place even if the particles are later separated:

> The picture that emerges is then of a background containing a very large number of connections that maintain an approximation to three-dimensional space, together with a few threads that in effect go outside of that space to make direct connections between particles (p. 544).

The threads can produce Bell correlations, but are somehow too small (i.e. contain too few edges) to transmit information in a way that violates causality.

There are several objections one could raise against this thread hypothesis. What I will show is that, *if* one accepts two of Wolfram's own desiderata—determinism and causal invariance—then the hypothesis fails. First, though, let me remark that Wolfram says little about what, to me, is a more natural possibility than the thread hypothesis. This is an explicitly *quantum* cellular automaton or causal network, with a unitary transition rule. The reason seems to be that he does not want continuity anywhere in a model, not even in probabilities or amplitudes. In the notes, he describes an experiment with a quantum cellular automaton as follows:

> One might hope to be able to get an ordinary cellular automaton with a limited set of possible values by choosing a suitable [phase rotation] θ [$\theta = \pi/4$ and $\theta = \pi/3$ are given as examples in an illustration]. But in fact in non-trivial cases most of the cells generated at each step end up having distinct values (p. 1060).

This observation is unsurprising, given the quantum computing results mentioned in Chapter 4, to the effect that almost any nontrivial gate set is universal (that is, can approximate any unitary matrix to any desired precision, or any orthogonal matrix in case one is limited to reals). Indeed, Shi [217] has shown that a Toffoli gate, plus any gate that does not preserve the computational basis, or a controlled-NOT gate plus any gate whose *square* does not preserve the computational basis, are both universal gate sets. In any case, Wolfram does not address the fact that continuity in amplitudes seems more 'benign' than continuity in measurable quantities: the former, unlike the latter, does not enable an infinite amount of computation to be performed in a finite time. Also, as observed by Bernstein and Vazirani [55], the linearity of quantum mechanics implies that tiny errors in amplitudes will not be magnified during a quantum computation.

I now proceed to the argument that Wolfram's thread hypothesis is inconsistent with causal invariance and relativity. Let $\mathcal{R}$ be a set of graph updating rules, which might be probabilistic. Then consider the following four assertions (which, though not mathematically precise, will be clarified by subsequent discussion).

(1) $\mathcal{R}$ satisfies causal invariance. That is, given any initial graph (and choice of randomness if $\mathcal{R}$ is probabilistic), $\mathcal{R}$ yields a unique causal network.

(2) $\mathcal{R}$ satisfies the relativity postulate. That is, assuming the causal network approximates a flat Minkowski spacetime at a large enough scale, there are no preferred inertial frames.

(3) $\mathcal{R}$ permits Bell inequality violations.

(4) Any updating rule in $\mathcal{R}$ is always considered to act on a fixed graph, not on a distribution or superposition over graphs. This is true even if parts of the initial graph are chosen at random, and even if $\mathcal{R}$ is probabilistic.

The goal is to show that, for any $\mathcal{R}$, at least one of these assertions is false. Current physical theory would suggest that (1)-(3) are true and that (4) is false. Wolfram, if I understand him correctly, starts with (4) as a premise, and then introduces causal invariance to satisfy (1) and (2), and long-range threads to satisfy (3). Of course, even to state the two-party Bell inequalities requires some notion of randomness. And on pages 299–326, Wolfram discusses three mechanisms for introducing randomness into a system: randomness in initial conditions, randomness from the environment (i.e. probabilistic updating rules), and intrinsic randomness (i.e. deterministic rules that produce pseudorandom output). However, all of these mechanisms are compatible with (4), and so my argument will show that they are inadequate assuming (1)-(3). The conclusion is that, in a model of the sort Wolfram considers, randomness must play a more fundamental role than he allows.

In a standard Bell experiment, Alice and Bob are given input bits x_A and x_B respectively, chosen uniformly and independently at random. Their goal is, without communicating, to output bits y_A and y_B respectively such that $y_A \oplus y_B = x_A \wedge x_B$. Under any 'local hidden variable' theory, Alice and Bob can succeed with probability at most $3/4$; the optimal strategy is for them to ignore their inputs and output (say) $y_A = 0$ and $y_B = 0$. However, suppose Alice has a qubit ρ_A and Bob a ρ_B, that are jointly in the Bell state $\left(|00\rangle + |11\rangle\right)/\sqrt{2}$. Then there is a protocol[4] by which they can succeed with probability $\left(5+\sqrt{2}\right)/8 \approx 0.802$.

To model this situation, let A and B, corresponding to Alice and Bob, be disjoint subgraphs of a graph G. Suppose that, at a large scale, G approximates a Euclidean space of some dimension; and that any causal network obtained by applying updates to G approximates a Minkowski spacetime. One can think of G as containing long-range threads from A to B, though the nature of the threads will not affect the conclusions. Encode Alice's input x_A by (say) placing an edge between two specific vertices in A if and only if $x_A = 1$. Encode x_B similarly, and also supply Alice and Bob with arbitrarily many correlated random bits. Finally, let us stipulate that at the end of the protocol, there is an edge between two specific vertices in A if and only if $y_A = 1$, and similarly for y_B. A technicality is that we need to be able to identify which vertices correspond to x_A, y_A, and so on, even as G evolves over time. We could do this by stipulating that (say) "the x_A vertices are the ones that are roots of complete binary trees of depth 3," and then choosing the rule set to guarantee that, throughout the protocol, exactly two vertices have this property.

[4] If $x_A = 1$ then Alice applies a $\pi/8$ phase rotation to ρ_A, and if $x_B = 1$ then Bob applies a $-\pi/8$ rotation to ρ_B. Both parties then measure in the standard basis and output whatever they observe.

Call a variable 'touched' after an update has been applied to a subgraph containing any of the variable's vertices. Also, let Z be an assignment to all random variables: that is, x_A, x_B, the correlated random bits, and the choice of randomness if $\mathcal{R}$ is probabilistic. Then for all Z we need the following, based on what observers in different inertial frames could perceive:

(i) There exists a sequence of updates under which y_A is output before any of Bob's variables are touched.

(ii) There exists another sequence under which y_B is output before any of Alice's variables are touched.

Now it is easy to see that, if a Bell inequality violation occurs, then causal invariance must be violated. Given Z, let $y_A^{(1)}(Z)$, $y_B^{(1)}(Z)$ be the values of y_A, y_B that are output under rule sequence (i), and let $y_A^{(2)}(Z)$, $y_B^{(2)}(Z)$ be the values output under sequence (ii). Then there must exist some Z for which either $y_A^{(1)}(Z) \neq y_A^{(2)}(Z)$ or $y_B^{(1)}(Z) \neq y_B^{(2)}(Z)$—for if not, then the entire protocol could be simulated under a local hidden variable model. It follows that the outcome of the protocol can depend on the order in which updates are applied.

To obtain a Bell inequality violation, something like the following seems to be needed. We can encode 'hidden variables' into G, representing the outcomes of the possible measurements Bob could make on ρ_B. (We can imagine, if we like, that the update rules are such that observing any one of these variables destroys all the others. Also, we make no assumption of contextuality.) Then, after Alice measures ρ_A, using the long-range threads she updates Bob's hidden variables conditioned on her measurement outcome. Similarly, Bob updates Alice's hidden variables conditioned on his outcome. Since at least one party must access its hidden variables for there to be Bell inequality violations, causal invariance is still violated. But a sort of probabilistic causal invariance holds, in the sense that if we marginalize out A (the 'Alice' part of G), then the *distribution* of values for each of Bob's hidden variables is the same before and after Alice's update. The lesson is that, if we want both causal invariance and Bell inequality violations, then we need to introduce probabilities at a fundamental level—not merely to represent Alice and Bob's subjective uncertainty about the state of G, but even to define whether a set of rules is or is not causal invariant.

Note that I made no assumption about how the random bits were generated—i.e. whether they were 'truly random' or were the pseudorandom output of some updating rule. The conclusion is also unaffected if we consider a 'deterministic' variant of Bell's theorem due to Greenberger, Horne, and Zeilinger [136]. There three parties, Alice, Bob, and Charlie, are given input bits x_A, x_B, and x_C respectively, satisfying the promise that $x_A \oplus x_B \oplus x_C = 0$. The goal is to output bits y_A, y_B, and y_C such that $y_A \oplus y_B \oplus y_C = x_A \vee x_B \vee x_C$. Under a local hidden variable model, there is no protocol that succeeds on all four possible inputs; but if the parties share the GHZ state $(|011\rangle + |101\rangle + |110\rangle - |000\rangle)/2$, then such a protocol exists. However, although the *output* is correct with certainty, assuming causal invariance one cannot *implement* the protocol without introducing randomness into the underlying rules, exactly as in the two-party case.

After a version of the above argument was sent to Wolfram, Todd Rowland, an employee of Wolfram, sent me email claiming that the argument fails for the following reason. I assumed that there exist two sequences of updating events, one in which Alice's measurement precedes Bob's and one in which Bob's precedes Alice's. But I neglected the possibility that a *single* update, call it E, is applied to a subgraph that straddles the long-range threads. The event E would encompass both Alice and Bob's measurements, so that neither would precede the other in any sequence of updates. We could thereby obtain a rule set $\mathcal{R}$ satisfying assertions (1), (3), and (4).

I argue that such an $\mathcal{R}$ would nevertheless fail to satisfy (2). For in effect we start with a flat Minkowski spacetime, and then take two distinct events that are simultaneous in a particular inertial frame, and identify them as being the *same* event E. This can be visualized as 'pinching together' two horizontally separated points on a spacetime diagram. (Actually a whole 'V' of points must be pinched together, since otherwise entanglement could not have been created.) However, what happens in a different inertial frame? It would seem that E, a single event, is perceived to occur at two separate times. That by itself might be thought acceptable, but it implies that there exists a class of preferred inertial frames: those in which E is perceived to occur only once. Of course, even in a flat spacetime, one could designate as 'preferred' those frames in which Alice and Bob's measurements are perceived to be simultaneous. A crucial distinction, though, is that there one only obtains a class of preferred frames after deciding which event at Alice's location, *and* which at Bob's location, should count as the 'measurement.' Under Rowland's hypothesis, by contrast, once one decides what counts as the measurement at Alice's location, the decision at Bob's location is made automatically, because of the identification of events that would otherwise be far apart.

Chapter 13

Complexity Theory of Quantum States

In my view, the central weakness in the arguments of quantum computing skeptics is their failure to suggest any answer the following question: *Exactly what property separates the quantum states we are sure we can create, from the states that suffice for Shor's factoring algorithm?*

I call such a property a "Sure/Shor separator." The purpose of this chapter is to develop a mathematical theory of Sure/Shor separators, and thereby illustrate what I think a scientific discussion about the possibility of quantum computing might look like. In particular, I will introduce *tree states*, which informally are those states $|\psi\rangle \in \mathcal{H}_2^{\otimes n}$ expressible by a polynomial-size 'tree' of addition and tensor product gates. For example, $\alpha|0\rangle^{\otimes n} + \beta|1\rangle^{\otimes n}$ and $(\alpha|0\rangle + \beta|1\rangle)^{\otimes n}$ are both tree states. Section 13.1 provides the philosophical motivation for thinking of tree states as a possible Sure/Shor separator; then Section 13.2 formally defines tree states and many related classes of quantum states. Next, Section 13.3 investigates basic properties of tree states. Among other results, it shows that any tree state is representable by a tree of polynomial size and logarithmic depth; and that most states do not even have large inner product with any tree state. Then Section 13.4 shows relationships among tree size, circuit size, bounded-depth tree size, Vidal's χ complexity [236], and several other measures. It also relates questions about quantum state classes to more traditional questions about computational complexity classes.

But the main results of the chapter, proved in Section 13.5, are lower bounds on tree size for various natural families of quantum states. In particular, Section 13.5.1 analyzes "subgroup states," which are uniform superpositions $|S\rangle$ over all elements of a subgroup $S \leq \mathbb{Z}_2^n$. The importance of these states arises from their central role in stabilizer codes, a type of quantum error-correcting code. I first show that if S is chosen uniformly at random, then with high probability $|S\rangle$ cannot be represented by any tree of size $n^{o(\log n)}$. This result has a corollary of independent complexity-theoretic interest: the first superpolynomial gap between the formula size and the multilinear formula size of a function $f : \{0,1\}^n \rightarrow \mathbb{R}$. I then present two improvements of the basic lower bound. First, I show that a random subgroup state cannot even be *approximated* well in trace distance by any tree of size $n^{o(\log n)}$. Second, I "derandomize" the lower bound, by using Reed-Solomon

codes to construct an *explicit* subgroup state with tree size $n^{\Omega(\log n)}$.

Section 13.5.2 analyzes the states that arise in Shor's factoring algorithm—for example, a uniform superposition over all multiples of a fixed positive integer p, written in binary. Originally, I had hoped to show a superpolynomial tree size lower bound for these states as well. However, I am only able to show such a bound assuming a number-theoretic conjecture.

The lower bounds use a sophisticated recent technique of Raz [195, 196], which was introduced to show that the permanent and determinant of a matrix require superpolynomial-size multilinear formulas. Currently, Raz's technique is only able to show lower bounds of the form $n^{\Omega(\log n)}$, but I conjecture that $2^{\Omega(n)}$ lower bounds hold in all of the cases discussed above.

One might wonder how superpolynomial tree size relates to more *physical* properties of a quantum state. Section 13.5.3 addresses this question, by pointing out how Raz's lower bound technique is connected to a notion that physicists call "persistence of entanglement" [71, 100]. On the other hand, I also give examples showing that the connection is not exact.

Section 13.6 studies a weakening of tree size called "manifestly orthogonal tree size," and shows that this measure can sometimes be characterized *exactly*, enabling us to prove *exponential* lower bounds. The techniques in Section 13.6 might be of independent interest to complexity theorists—one reason being that they do not obviously "naturalize" in the sense of Razborov and Rudich [200].

Section 13.7 addresses the following question. If the state of a quantum computer at every time step is a tree state, then can the computer be simulated classically? In other words, letting $\mathsf{TreeBQP}$ be the class of languages accepted by such a machine, does $\mathsf{TreeBQP} = \mathsf{BPP}$? A positive answer would make tree states more attractive as a Sure/Shor separator. For once we admit any states incompatible with the polynomial-time Church-Turing thesis, it seems like we might as well go all the way, and admit *all* states preparable by polynomial-size quantum circuits! Although I leave this question open, I do show that $\mathsf{TreeBQP} \subseteq \Sigma_3^{\mathsf{P}} \cap \Pi_3^{\mathsf{P}}$, where $\Sigma_3^{\mathsf{P}} \cap \Pi_3^{\mathsf{P}}$ is the third level of the polynomial hierarchy PH. By contrast, it is conjectured that $\mathsf{BQP} \not\subset \mathsf{PH}$, though admittedly not on strong evidence.

Section 13.8 discusses the implications of these results for experimental physics. It advocates a dialectic between theory and experiment, in which theorists would propose a class of quantum states that encompasses everything seen so far, and then experimenters would try to prepare states not in that class. It also asks whether states with superpolynomial tree size have already been observed in condensed-matter systems; and more broadly, what sort of evidence is needed to establish a state's existence. Other issues addressed in Section 13.8 include how to deal with mixed states and particle position and momentum states, and the experimental relevance of asymptotic bounds. I conclude in Section 13.9 with some open problems.

13.1 Sure/Shor Separators

Given the discussion in Chapter 12, I believe that the challenge for quantum computing skeptics is clear. Ideally, come up with an alternative to quantum mechanics—even an

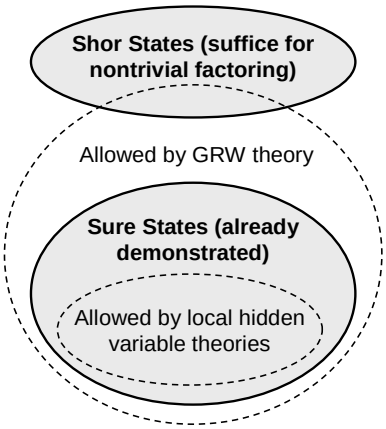

Figure 13.1: A Sure/Shor separator must contain all Sure states but no Shor states. That is why neither local hidden variables nor the GRW theory yields a Sure/Shor separator.

idealized toy theory—that can account for all present-day experiments, yet would not allow large-scale quantum computation. Failing that, *at least say what you take quantum mechanics' domain of validity to be.* One way to do this would be to propose a set S of quantum states that you believe corresponds to possible physical states of affairs.[1] The set S must contain all "Sure states" (informally, the states that have already been demonstrated in the lab), but no "Shor states" (again informally, the states that can be shown to suffice for factoring, say, 500-digit numbers). If S satisfies both of these constraints, then I call S a *Sure/Shor separator* (see Figure 13.1).

Of course, an alternative theory need not involve a sharp cutoff between possible and impossible states. So it is perfectly acceptable for a skeptic to define a "complexity measure" $C(|\psi\rangle)$ for quantum states, and then say something like the following: *If $|\psi_n\rangle$ is a state of n spins, and $C(|\psi_n\rangle)$ is at most, say, n^2, then I predict that $|\psi_n\rangle$ can be prepared using only "polynomial effort." Also, once prepared, $|\psi_n\rangle$ will be governed by standard quantum mechanics to extremely high precision. All states created to date have had small values of $C(|\psi_n\rangle)$. However, if $C(|\psi_n\rangle)$ grows as, say, 2^n, then I predict that $|\psi_n\rangle$ requires "exponential effort" to prepare, or else is not even approximately governed by quantum mechanics, or else does not even make sense in the context of an alternative theory. The states that arise in Shor's factoring algorithm have exponential values of $C(|\psi_n\rangle)$. So as my Sure/Shor separator, I propose the set of all infinite families of states $\{|\psi_n\rangle\}_{n\geq 1}$, where $|\psi_n\rangle$ has n qubits, such that $C(|\psi_n\rangle) \leq p(n)$ for some polynomial p.*

To understand the importance of Sure/Shor separators, it is helpful to think through some examples. A major theme of Levin's arguments was that exponentially small amplitudes are somehow unphysical. However, clearly we cannot reject *all* states with tiny amplitudes—for would anyone dispute that the state $2^{-5000}(|0\rangle + |1\rangle)^{\otimes 10000}$ is

[1] A skeptic might also specify what happens if a state $|\psi\rangle \in S$ is acted on by a unitary U such that $U|\psi\rangle \notin S$, but this will not be insisted upon.

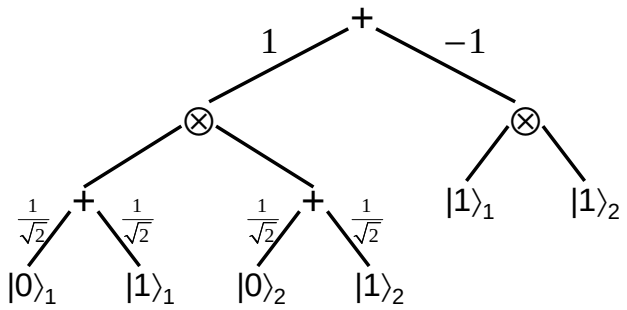

Figure 13.2: Expressing $(|00\rangle + |01\rangle + |10\rangle - |11\rangle)/2$ by a tree of linear combination and tensor product gates, with scalar multiplication along edges. Subscripts denote the identity of a qubit.

formed whenever 10,000 photons are each polarized at 45°? Indeed, once we accept $|\psi\rangle$ and $|\varphi\rangle$ as Sure states, we are almost *forced* to accept $|\psi\rangle \otimes |\varphi\rangle$ as well—since we can imagine, if we like, that $|\psi\rangle$ and $|\varphi\rangle$ are prepared in two separate laboratories.[2] So considering a Shor state such as

$$|\Phi\rangle = \frac{1}{2^{n/2}} \sum_{r=0}^{2^n-1} |r\rangle \, |x^r \bmod N\rangle \, ,$$

what property of this state could quantum computing skeptics latch onto as being physically extravagant? They might complain that $|\Phi\rangle$ involves entanglement across hundreds or thousands of particles; but as mentioned in Chapter 12, there are other states with that same property, namely the "Schrödinger cats" $\left(|0\rangle^{\otimes n} + |1\rangle^{\otimes n}\right)/\sqrt{2}$, that should be regarded as Sure states. Alternatively, the skeptics might object to the *combination* of exponentially small amplitudes with entanglement across hundreds of particles. However, simply viewing a Schrödinger cat state in the Hadamard basis produces an equal superposition over all strings of even parity, which has both properties. We seem to be on a slippery slope leading to all of quantum mechanics! Is there any defensible place to draw a line?

The dilemma above is what led me to propose *tree states* as a possible Sure/Shor separator. The idea, which might seem more natural to logicians than to physicists, is this. Once we accept the linear combination and tensor product rules of quantum mechanics—allowing $\alpha |\psi\rangle + \beta |\varphi\rangle$ and $|\psi\rangle \otimes |\varphi\rangle$ into our set S of possible states whenever $|\psi\rangle, |\varphi\rangle \in S$—one of our few remaining hopes for keeping S a proper subset of the set of *all* states is to impose some restriction on how those two rules can be iteratively applied. In particular, we could let S be the closure of $\{|0\rangle, |1\rangle\}$ under a *polynomial number* of linear combinations and tensor products. That is, S is the set of all infinite families of states $\{|\psi_n\rangle\}_{n \geq 1}$ with $|\psi_n\rangle \in \mathcal{H}_2^{\otimes n}$, such that $|\psi_n\rangle$ can be expressed as a "tree" involving at most $p(n)$ addition, tensor product, $|0\rangle$, and $|1\rangle$ gates for some polynomial p (see Figure 13.2).

[2] It might be objected that in some theories, such as Chern-Simons theory, there is no clear tensor product decomposition. However, the relevant question is whether $|\psi\rangle \otimes |\varphi\rangle$ is a Sure state, *given* that $|\psi\rangle$ and $|\varphi\rangle$ are both Sure states that are well-described in tensor product Hilbert spaces.

To be clear, I am *not* advocating that "all states in Nature are tree states" as a serious physical hypothesis. Indeed, even if I believed firmly in a breakdown of quantum mechanics,[3] there are other choices for the set S that seem equally reasonable. For example, define *orthogonal tree states* similarly to tree states, except that we can only form the linear combination $\alpha|\psi\rangle + \beta|\varphi\rangle$ if $\langle\psi|\varphi\rangle = 0$. Rather than choose among tree states, orthogonal tree states, and the other candidate Sure/Shor separators that occurred to me, my approach will be to prove everything I can about all of them. If I devote more space to tree states than to others, that is simply because tree states are the subject of the most interesting results. On the other hand, if one shows (for example) that $\{|\psi_n\rangle\}$ is not a tree state, then one has also shown that $\{|\psi_n\rangle\}$ is not an orthogonal tree state. So many candidate separators are related to each other; and indeed, their relationships will be a major theme of the chapter.

In summary, to debate whether quantum computing is fundamentally impossible, we need at least one proposal for how it *could* be impossible. Since even skeptics admit that quantum mechanics is valid within some "regime," a key challenge for any such proposal is to separate the regime of acknowledged validity from the quantum computing regime. Though others will disagree, I do not see any choice but to *identify those two regimes with classes of quantum states*. For gates and measurements that suffice for quantum computing have already been demonstrated experimentally. Thus, if we tried to identify the two regimes with classes of gates or measurements, then we could equally well talk about the class of *states* on which all 1- and 2-qubit operations behave as expected. A similar argument would apply if we identified the two regimes with classes of quantum circuits—since any "memory" that a quantum system retains of the previous gates in a circuit, is part of the system's state by definition. So: states, gates, measurements, circuits—what else is there?

I should stress that none of the above depends on the interpretation of quantum mechanics. In particular, it is irrelevant whether we regard quantum states as "really out there" or as representing subjective knowledge—since in either case, the question is whether there can exist systems that we would *describe* by $|\psi\rangle$ based on their observed behavior.

Once we agree to seek a Sure/Shor separator, we quickly find that the obvious ideas—based on precision in amplitudes, or entanglement across of hundreds of particles—are nonstarters. The only idea that seems plausible is to limit the class of allowed quantum states to those with some kind of succinct representation. That still leaves numerous possibilities; and for each one, it might be a difficult problem to decide whether a given $|\psi\rangle$ is succinctly representable or not. Thus, constructing a useful theory of Sure/Shor separators is a nontrivial task. This chapter represents a first attempt.

13.2 Classifying Quantum States

In both quantum and classical complexity theory, the objects studied are usually sets of languages or Boolean functions. However, a generic n-qubit quantum state requires exponentially many classical bits to describe, and this suggests looking at *the complexity of quantum states themselves*. That is, which states have polynomial-size classical descriptions of various kinds? This question has been studied from several angles by Aharonov

[3] which I don't

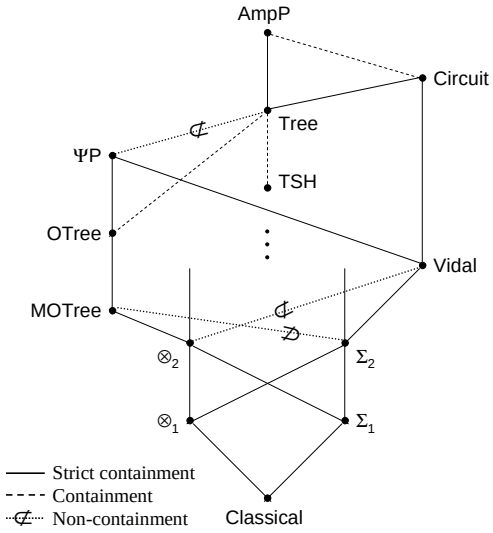

Figure 13.3: Known relations among quantum state classes.

and Ta-Shma [23]; Janzing, Wocjan, and Beth [148]; Vidal [236]; and Green et al. [134]. Here I propose a general framework for the question. For simplicity, I limit myself to pure states $|\psi_n\rangle \in \mathcal{H}_2^{\otimes n}$ with the fixed orthogonal basis $\{|x\rangle : x \in \{0,1\}^n\}$. Also, by 'states' I mean infinite families of states $\{|\psi_n\rangle\}_{n\geq 1}$.

Like complexity classes, pure quantum states can be organized into a hierarchy (see Figure 13.3). At the bottom are the classical basis states, which have the form $|x\rangle$ for some $x \in \{0,1\}^n$. We can generalize classical states in two directions: to the class $\otimes_1$ of separable states, which have the form $(\alpha_1 |0\rangle + \beta_1 |1\rangle) \otimes \cdots \otimes (\alpha_n |0\rangle + \beta_n |1\rangle)$; and to the class Σ_1, which consists of all states $|\psi_n\rangle$ that are superpositions of at most $p(n)$ classical states, where p is a polynomial. At the next level, $\otimes_2$ contains the states that can be written as a tensor product of Σ_1 states, with qubits permuted arbitrarily. Likewise, Σ_2 contains the states that can be written as a linear combination of a polynomial number of $\otimes_1$ states. We can continue indefinitely to Σ_3, $\otimes_3$, etc. Containing the whole 'tensor-sum hierarchy' $\cup_k \Sigma_k = \cup_k \otimes_k$ is the class Tree, of all states expressible by a polynomial-size tree of additions and tensor products nested arbitrarily. Formally, Tree consists of all states $|\psi_n\rangle$ such that $\mathrm{TS}(|\psi_n\rangle) \leq p(n)$ for some polynomial p, where the *tree size* $\mathrm{TS}(|\psi_n\rangle)$ is defined as follows.

Definition 70 *A quantum state tree over $\mathcal{H}_2^{\otimes n}$ is a rooted tree where each leaf vertex is labeled with $\alpha |0\rangle + \beta |1\rangle$ for some $\alpha, \beta \in \mathsf{C}$, and each non-leaf vertex (called a gate) is labeled with either $+$ or $\otimes$. Each vertex v is also labeled with a set $S(v) \subseteq \{1,\ldots,n\}$, such that*

(i) If v is a leaf then $|S(v)| = 1$,

(ii) If v is the root then $S(v) = \{1,\ldots,n\}$,

(iii) *If v is a $+$ gate and w is a child of v, then $S(w) = S(v)$,*

(iv) *If v is a $\otimes$ gate and $w_1, \ldots, w_k$ are the children of v, then $S(w_1), \ldots, S(w_k)$ are pairwise disjoint and form a partition of $S(v)$.*

Finally, if v is a $+$ gate, then the outgoing edges of v are labeled with complex numbers. For each v, the subtree rooted at v represents a quantum state of the qubits in $S(v)$ in the obvious way. We require this state to be normalized for each v.[4]

We say a tree is *orthogonal* if it satisfies the further condition that if v is a $+$ gate, then any two children w_1, w_2 of v represent $|\psi_1\rangle, |\psi_2\rangle$ with $\langle\psi_1|\psi_2\rangle = 0$. If the condition $\langle\psi_1|\psi_2\rangle = 0$ can be replaced by the stronger condition that for all basis states $|x\rangle$, either $\langle\psi_1|x\rangle = 0$ or $\langle\psi_2|x\rangle = 0$, then we say the tree is *manifestly orthogonal*. Manifest orthogonality is an extremely unphysical definition; I introduce it because it turns out to be interesting from a lower bounds perspective.

For reasons of convenience, let us define the *size* $|T|$ of a tree T to be the number of leaf vertices. Then given a state $|\psi\rangle \in \mathcal{H}_2^{\otimes n}$, the *tree size* $\mathrm{TS}(|\psi\rangle)$ is the minimum size of a tree that represents $|\psi\rangle$. The *orthogonal tree size* $\mathrm{OTS}(|\psi\rangle)$ and *manifestly orthogonal tree size* $\mathrm{MOTS}(|\psi\rangle)$ are defined similarly. Then OTree is the class of $|\psi_n\rangle$ such that $\mathrm{OTS}(|\psi_n\rangle) \leq p(n)$ for some polynomial p, and MOTree is the class such that $\mathrm{MOTS}(|\psi_n\rangle) \leq p(n)$ for some p.

It is easy to see that

$$n \leq \mathrm{TS}(|\psi\rangle) \leq \mathrm{OTS}(|\psi\rangle) \leq \mathrm{MOTS}(|\psi\rangle) \leq n2^n$$

for every $|\psi\rangle$, and that the set of $|\psi\rangle$ such that $\mathrm{TS}(|\psi\rangle) < 2^n$ has measure 0 in $\mathcal{H}_2^{\otimes n}$. Two other important properties of TS and OTS are as follows:

Proposition 71

(i) TS *and* OTS *are invariant under local*[5] *basis changes, up to a constant factor of* 2.

(ii) *If $|\phi\rangle$ is obtained from $|\psi\rangle$ by applying a k-qubit unitary, then* $\mathrm{TS}(|\phi\rangle) \leq k4^k\,\mathrm{TS}(|\psi\rangle)$ *and* $\mathrm{OTS}(|\phi\rangle) \leq k4^k\,\mathrm{OTS}(|\psi\rangle)$.

Proof.

(i) Simply replace each occurrence of $|0\rangle$ in the original tree by a tree for $\alpha|0\rangle + \beta|1\rangle$, and each occurrence of $|1\rangle$ by a tree for $\gamma|0\rangle + \delta|1\rangle$, as appropriate.

(ii) Suppose without loss of generality that the gate is applied to the first k qubits. Let T be a tree representing $|\psi\rangle$, and let T_y be the restriction of T obtained by setting the first k qubits to $y \in \{0,1\}^k$. Clearly $|T_y| \leq |T|$. Furthermore, we can express $|\phi\rangle$ in the form $\sum_{y\in\{0,1\}^k} S_y T_y$, where each S_y represents a k-qubit state and hence is expressible by a tree of size $k2^k$.

[4]Requiring only the *whole* tree to represent a normalized state clearly yields no further generality.

[5]Several people told me that a reasonable complexity measure must be invariant under *all* basis changes. Alas, this would imply that all pure states have the same complexity!

■

One can also define the *ε-approximate tree size* $\mathrm{TS}_\varepsilon(|\psi\rangle)$ to be the minimum size of a tree representing a state $|\varphi\rangle$ such that $|\langle\psi|\varphi\rangle|^2 \geq 1-\varepsilon$, and define $\mathrm{OTS}_\varepsilon(|\psi\rangle)$ and $\mathrm{MOTS}_\varepsilon(|\psi\rangle)$ similarly.

Definition 72 *An arithmetic formula (over the ring $\mathbb{C}$ and n variables) is a rooted binary tree where each leaf vertex is labeled with either a complex number or a variable in $\{x_1,\ldots,x_n\}$, and each non-leaf vertex is labeled with either $+$ or $\times$. Such a tree represents a polynomial $p(x_1,\ldots,x_n)$ in the obvious way. We call a polynomial multilinear if no variable appears raised to a higher power than 1, and an arithmetic formula multilinear if the polynomials computed by each of its subtrees are multilinear.*

The *size* $|\Phi|$ of a multilinear formula Φ is the number of leaf vertices. Given a multilinear polynomial p, the multilinear formula size $\mathrm{MFS}(p)$ is the minimum size of a multilinear formula that represents p. Then given a function $f:\{0,1\}^n \rightarrow \mathbb{C}$, we define

$$\mathrm{MFS}(f) = \min_{p\,:\,p(x)=f(x)\ \forall x\in\{0,1\}^n} \mathrm{MFS}(p).$$

(Actually p turns out to be unique [184].) We can also define the ε-approximate multilinear formula size of f,

$$\mathrm{MFS}_\varepsilon(f) = \min_{p\,:\,\|p-f\|_2^2\leq\varepsilon} \mathrm{MFS}(p)$$

where $\|p-f\|_2^2 = \sum_{x\in\{0,1\}^n} |p(x)-f(x)|^2$. (This metric is closely related to the inner product $\sum_x p(x)^* f(x)$, but is often more convenient to work with.) Now given a state $|\psi\rangle = \sum_{x\in\{0,1\}^n} \alpha_x |x\rangle$ in $\mathcal{H}_2^{\otimes n}$, let f_ψ be the function from $\{0,1\}^n$ to $\mathbb{C}$ defined by $f_\psi(x) = \alpha_x$.

Theorem 73 *For all $|\psi\rangle$,*

(i) $\mathrm{MFS}(f_\psi) = O(\mathrm{TS}(|\psi\rangle))$.

(ii) $\mathrm{TS}(|\psi\rangle) = O(\mathrm{MFS}(f_\psi)+n)$.

(iii) $\mathrm{MFS}_\delta(f_\psi) = O(\mathrm{TS}_\varepsilon(|\psi\rangle))$ *where* $\delta = 2-2\sqrt{1-\varepsilon}$.

(iv) $\mathrm{TS}_{2\varepsilon}(|\psi\rangle) = O(\mathrm{MFS}_\varepsilon(f_\psi)+n)$.

Proof.

(i) Given a tree representing $|\psi\rangle$, replace every unbounded fan-in gate by a collection of binary gates, every $\otimes$ by $\times$, every $|1\rangle_i$ vertex by x_i, and every $|0\rangle_i$ vertex by a formula for $1-x_i$. Push all multiplications by constants at the edges down to $\times$ gates at the leaves.

(ii) Given a multilinear formula Φ for f_ψ, let $p(v)$ be the polynomial computed at vertex v of Φ, and let $S(v)$ be the set of variables that appears in $p(v)$. First, call Φ *syntactic* if at every $\times$ gate with children v and w, $S(v) \cap S(w) = \varnothing$. A lemma of Raz [195] states that we can always make Φ syntactic without increasing its size.

Second, at every $+$ gate u with children v and w, enlarge both $S(v)$ and $S(w)$ to $S(v) \cup S(w)$, by multiplying $p(v)$ by $x_i + (1 - x_i)$ for every $x_i \in S(w) \setminus S(v)$, and multiplying $p(w)$ by $x_i + (1 - x_i)$ for every $x_i \in S(v) \setminus S(w)$. Doing this does not invalidate any $\times$ gate that is an ancestor of u, since by the assumption that Φ is syntactic, $p(u)$ is never multiplied by any polynomial containing variables in $S(v) \cup S(w)$. Similarly, enlarge $S(r)$ to $\{x_1, \ldots, x_n\}$ where r is the root of Φ.

Third, call v *max-linear* if $|S(v)| = 1$ but $|S(w)| > 1$ where w is the parent of v. If v is max-linear and $p(v) = a + bx_i$, then replace the tree rooted at v by a tree computing $a|0\rangle_i + (a+b)|1\rangle_i$. Also, replace all multiplications by constants higher in Φ by multiplications at the edges. (Because of the second step, there are no additions by constants higher in Φ.) Replacing every $\times$ by $\otimes$ then gives a tree representing $|\psi\rangle$, whose size is easily seen to be $O(|\Phi| + n)$.

(iii) Apply the reduction from part (i). Let the resulting multilinear formula compute polynomial p; then

$$\sum_{x \in \{0,1\}^n} |p(x) - f_\psi(x)|^2 = 2 - 2 \sum_{x \in \{0,1\}^n} p(x) \overline{f_\psi(x)} \leq 2 - 2\sqrt{1-\varepsilon} = \delta.$$

(iv) Apply the reduction from part (ii). Let $(\beta_x)_{x \in \{0,1\}^n}$ be the resulting amplitude vector; since this vector might not be normalized, divide each β_x by $\sum_x |\beta_x|^2$ to produce β'_x. Then

$$\begin{aligned}
\left| \sum_{x \in \{0,1\}^n} \beta'_x \overline{\alpha_x} \right|^2 &= 1 - \frac{1}{2} \sum_{x \in \{0,1\}^n} \left| \beta'_x - \alpha_x \right|^2 \\
&\geq 1 - \frac{1}{2} \left(\sqrt{\sum_{x \in \{0,1\}^n} |\beta'_x - \beta_x|^2} + \sqrt{\sum_{x \in \{0,1\}^n} |\beta_x - \alpha_x|^2} \right)^2 \\
&\geq 1 - \frac{1}{2} \left(2\sqrt{\varepsilon} \right)^2 = 1 - 2\varepsilon.
\end{aligned}$$

■

Besides Tree, OTree, and MOTree, four other classes of quantum states deserve mention:

Circuit, a circuit analog of Tree, contains the states $|\psi_n\rangle = \sum_x \alpha_x |x\rangle$ such that for all n, there exists a multilinear arithmetic circuit of size $p(n)$ over the complex numbers that outputs α_x given x as input, for some polynomial p. (Multilinear circuits are the same as multilinear trees, except that they allow unbounded fanout—that is, polynomials computed at intermediate points can be reused arbitrarily many times.)

AmpP contains the states $|\psi_n\rangle = \sum_x \alpha_x |x\rangle$ such that for all n, b, there exists a classical circuit of size $p(n+b)$ that outputs α_x to b bits of precision given x as input, for some polynomial p.

Vidal contains the states that are 'polynomially entangled' in the sense of Vidal [236]. Given a partition of $\{1,\ldots,n\}$ into A and B, let $\chi_A(|\psi_n\rangle)$ be the minimum k for which $|\psi_n\rangle$ can be written as $\sum_{i=1}^k \alpha_i |\varphi_i^A\rangle \otimes |\varphi_i^B\rangle$, where $|\varphi_i^A\rangle$ and $|\varphi_i^B\rangle$ are states of qubits in A and B respectively. ($\chi_A(|\psi_n\rangle)$ is known as the *Schmidt rank*; see [182] for more information.) Let $\chi(|\psi_n\rangle) = \max_A \chi_A(|\psi_n\rangle)$. Then $|\psi_n\rangle \in$ Vidal if and only if $\chi(|\psi_n\rangle) \leq p(n)$ for some polynomial p.

ΨP contains the states $|\psi_n\rangle$ such that for all n and $\varepsilon > 0$, there exists a quantum circuit of size $p(n + \log(1/\varepsilon))$ that maps the all-0 state to a state some part of which has trace distance at most $1-\varepsilon$ from $|\psi_n\rangle$, for some polynomial p. Because of the Solovay-Kitaev Theorem [153, 182], ΨP is invariant under the choice of universal gate set.

13.3 Basic Results

Before studying the tree size of specific quantum states, it would be nice to know in general how tree size behaves as a complexity measure. In this section I prove three rather nice properties of tree size.

Theorem 74 *For all $\varepsilon > 0$, there exists a tree representing $|\psi\rangle$ of size $O\left(\mathrm{TS}(|\psi\rangle)^{1+\varepsilon}\right)$ and depth $O(\log \mathrm{TS}(|\psi\rangle))$, as well as a manifestly orthogonal tree of size $O\left(\mathrm{MOTS}(|\psi\rangle)^{1+\varepsilon}\right)$ and depth $O(\log \mathrm{MOTS}(|\psi\rangle))$.*

Proof. A classical theorem of Brent [70] says that given an arithmetic formula Φ, there exists an equivalent formula of depth $O(\log|\Phi|)$ and size $O(|\Phi|^c)$, where c is a constant. Bshouty, Cleve, and Eberly [73] (see also Bonet and Buss [62]) improved Brent's theorem to show that c can be taken to be $1+\varepsilon$ for any $\varepsilon > 0$. So it suffices to show that, for 'division-free' formulas, these theorems preserve multilinearity (and in the MOTS case, preserve manifest orthogonality).

Brent's theorem is proven by induction on $|\Phi|$. Here is a sketch: choose a subformula I of Φ size between $|\Phi|/3$ and $2|\Phi|/3$ (which one can show always exists). Then identifying a subformula with the polynomial computed at its root, $\Phi(x)$ can be written as $G(x) + H(x)\, I(x)$ for some formulas G and H. Furthermore, G and H are both obtainable from Φ by removing I and then applying further restrictions. So $|G|$ and $|H|$ are both at most $|\Phi| - |I| + O(1)$. Let $\widehat{\Phi}$ be a formula equivalent to Φ that evaluates G, H, and I separately, and then returns $G(x) + H(x)\, I(x)$. Then $\left|\widehat{\Phi}\right|$ is larger than $|\Phi|$ by at most a constant factor, while by the induction hypothesis, we can assume the formulas for G, H, and I have logarithmic depth. Since the number of induction steps is $O(\log|\Phi|)$, the total depth is logarithmic and the total blowup in formula size is polynomial in $|\Phi|$. Bshouty, Cleve, and Eberly's improvement uses a more careful decomposition of Φ, but the basic idea is the same.

Now, if Φ is syntactic multilinear, then clearly G, H, and I are also syntactic multilinear. Furthermore, H cannot share variables with I, since otherwise a subformula

of Φ containing I would have been multiplied by a subformula containing variables from I. Thus multilinearity is preserved. To see that manifest orthogonality is preserved, suppose we are evaluating G and H 'bottom up,' and let G_v and H_v be the polynomials computed at vertex v of Φ. Let $v_0 = \text{root}(I)$, let v_1 be the parent of v_0, let v_2 be the parent of v_1, and so on until $v_k = \text{root}(\Phi)$. It is clear that, for every x, either $G_{v_0}(x) = 0$ or $H_{v_0}(x) = 0$. Furthermore, suppose that property holds for $G_{v_{i-1}}, H_{v_{i-1}}$; then by induction it holds for G_{v_i}, H_{v_i}. If v_i is a $\times$ gate, then this follows from multilinearity (if $|\psi\rangle$ and $|\varphi\rangle$ are manifestly orthogonal, then $|0\rangle \otimes |\psi\rangle$ and $|0\rangle \otimes |\varphi\rangle$ are also manifestly orthogonal). If v_i is a $+$ gate, then letting $\text{supp}(p)$ be the set of x such that $p(x) \neq 0$, any polynomial p added to $G_{v_{i-1}}$ or $H_{v_{i-1}}$ must have

$$\text{supp}(p) \cap \left(\text{supp}\left(G_{v_{i-1}}\right) \cup \text{supp}\left(H_{v_{i-1}}\right)\right) = \emptyset,$$

and manifest orthogonality follows. ■

Theorem 75 *Any $|\psi\rangle$ can be prepared by a quantum circuit of size polynomial in $\text{OTS}(|\psi\rangle)$. Thus $\mathsf{OTree} \subseteq \mathsf{\Psi P}$.*

Proof. Let $\Gamma(|\psi\rangle)$ be the minimum size of a circuit needed to prepare $|\psi\rangle \in \mathcal{H}_2^{\otimes n}$ starting from $|0\rangle^{\otimes n}$. The claim, by induction on $\Gamma(|\psi\rangle)$, is that $\Gamma(|\psi\rangle) \leq q(\text{OTS}(|\psi\rangle))$ for some polynomial q. The base case $\text{OTS}(|\psi\rangle) = 1$ is clear. Let T be an orthogonal state tree for $|\psi\rangle$, and assume without loss of generality that every gate has fan-in 2 (this increases $|T|$ by at most a constant factor). Let T_1 and T_2 be the subtrees of $\text{root}(T)$, representing states $|\psi_1\rangle$ and $|\psi_2\rangle$ respectively; note that $|T| = |T_1| + |T_2|$. First suppose $\text{root}(T)$ is a $\otimes$ gate; then clearly $\Gamma(|\psi\rangle) \leq \Gamma(|\psi_1\rangle) + \Gamma(|\psi_2\rangle)$.

Second, suppose $\text{root}(T)$ is a $+$ gate, with $|\psi\rangle = \alpha|\psi_1\rangle + \beta|\psi_2\rangle$ and $\langle\psi_1|\psi_2\rangle = 0$. Let U be a quantum circuit that prepares $|\psi_1\rangle$, and V be a circuit that prepares $|\psi_2\rangle$. Then we can prepare $\alpha|0\rangle|0\rangle^{\otimes n} + \beta|1\rangle U^{-1}V|0\rangle^{\otimes n}$. Observe that $U^{-1}V|0\rangle^{\otimes n}$ is orthogonal to $|0\rangle^{\otimes n}$, since $|\psi_1\rangle = U|0\rangle^{\otimes n}$ is orthogonal to $|\psi_2\rangle = V|0\rangle^{\otimes n}$. So applying a NOT to the first register, conditioned on the OR of the bits in the second register, yields $|0\rangle \otimes \left(\alpha|0\rangle^{\otimes n} + \beta U^{-1}V|0\rangle^{\otimes n}\right)$, from which we obtain $\alpha|\psi_1\rangle + \beta|\psi_2\rangle$ by applying U to the second register. The size of the circuit used is $O(|U| + |V| + n)$, with a possible constant-factor blowup arising from the need to condition on the first register. If we are more careful, however, we can combine the 'conditioning' steps across multiple levels of the recursion, producing a circuit of size $|V| + O(|U| + n)$. By symmetry, we can also reverse the roles of U and V to obtain a circuit of size $|U| + O(|V| + n)$. Therefore

$$\Gamma(|\psi\rangle) \leq \min\left\{\Gamma(|\psi_1\rangle) + c\Gamma(|\psi_2\rangle) + cn,\ c\Gamma(|\psi_2\rangle) + \Gamma(|\psi_1\rangle) + cn\right\}$$

for some constant $c \geq 2$. Solving this recurrence we find that $\Gamma(|\psi\rangle)$ is polynomial in $\text{OTS}(|\psi\rangle)$. ■

Theorem 76 *If $|\psi\rangle \in \mathcal{H}_2^{\otimes n}$ is chosen uniformly at random under the Haar measure, then $\text{TS}_{1/16}(|\psi\rangle) = 2^{\Omega(n)}$ with probability $1 - o(1)$.*

Proof. To generate a uniform random state $|\psi\rangle = \sum_{x\in\{0,1\}^n} \alpha_x |x\rangle$, we can choose $\widehat{\alpha}_x, \widehat{\beta}_x \in \mathbb{R}$ for each x independently from a Gaussian distribution with mean 0 and variance 1, then let $\alpha_x = \left(\widehat{\alpha}_x + i\widehat{\beta}_x\right)/\sqrt{R}$ where $R = \sum_{x\in\{0,1\}^n} \left(\widehat{\alpha}_x^2 + \widehat{\beta}_x^2\right)$. Let

$$\Lambda_\psi = \left\{x : (\operatorname{Re}\alpha_x)^2 < \frac{1}{4\cdot 2^n}\right\},$$

and let $\mathcal{G}$ be the set of $|\psi\rangle$ for which $|\Lambda_\psi| < 2^n/5$. The claim is that $\Pr_{|\psi\rangle}[|\psi\rangle \in \mathcal{G}] = 1 - o(1)$. First, $\mathrm{EX}[R] = 2^{n+1}$, so by a standard Hoeffding-type bound, $\Pr[R < 2^n]$ is doubly-exponentially small in n. Second, assuming $R \geq 2^n$, for each x

$$\Pr[x \in \Lambda_\psi] \leq \Pr\left[\widehat{\alpha}_x^2 < \frac{1}{4}\right] = \operatorname{erf}\left(\frac{1}{4\sqrt{2}}\right) < 0.198,$$

and the claim follows by a Chernoff bound.

For $g : \{0,1\}^n \to \mathbb{R}$, let $A_g = \{x : \operatorname{sgn}(g(x)) \neq \operatorname{sgn}(\operatorname{Re}\alpha_x)\}$, where $\operatorname{sgn}(y)$ is 1 if $y \geq 0$ and -1 otherwise. Then if $|\psi\rangle \in \mathcal{G}$, clearly

$$\sum_{x\in\{0,1\}^n} |g(x) - f_\psi(x)|^2 \geq \frac{|A_g| - |\Lambda_\psi|}{4\cdot 2^n}$$

where $f_\psi(x) = \operatorname{Re}\alpha_x$, and thus

$$|A_g| \leq \left(4\|g - f_\psi\|_2^2 + \frac{1}{5}\right) 2^n.$$

Therefore to show that $\mathrm{MFS}_{1/15}(f_\psi) = 2^{\Omega(n)}$ with probability $1 - o(1)$, we need only show that for almost all Boolean functions $f : \{0,1\}^n \to \{-1,1\}$, there is no arithmetic formula Φ of size $2^{o(n)}$ such that

$$|\{x : \operatorname{sgn}(\Phi(x)) \neq f(x)\}| \leq 0.49 \cdot 2^n.$$

Here an arithmetic formula is real-valued, and can include addition, subtraction, and multiplication gates of fan-in 2 as well as constants. We do not need to assume multilinearity, and it is easy to see that the assumption of bounded fan-in is without loss of generality. Let W be the set of Boolean functions *sign-represented* by an arithmetic formula Φ of size $2^{o(n)}$, in the sense that $\operatorname{sgn}(\Phi(x)) = f(x)$ for all x. Then it suffices to show that $|W| = 2^{2^{o(n)}}$, since the number of functions sign-represented on an 0.51 fraction of inputs is at most $|W| \cdot 2^{2^n H(0.51)}$. (Here H denotes the binary entropy function.)

Let Φ be an arithmetic formula that takes as input the binary string $x = (x_1, \ldots, x_n)$ as well as constants $c_1, c_2, \ldots$. Let Φ_c denote Φ under a particular assignment c to $c_1, c_2, \ldots$. Then a result of Gashkov [121] (see also Turán and Vatan [230]), which follows from Warren's Theorem [237] in real algebraic geometry, shows that as we range over all c, Φ_c sign-represents at most $\left(2^{n+4}|\Phi|\right)^{|\Phi|}$ distinct Boolean functions, where $|\Phi|$ is the size of Φ. Furthermore, excluding constants, the number of distinct arithmetic formulas of size $|\Phi|$ is

at most $\left(3\left|\Phi\right|^2\right)^{\left|\Phi\right|}$. When $\left|\Phi\right| = 2^{o(n)}$, this gives $\left(3\left|\Phi\right|^2\right)^{\left|\Phi\right|} \cdot \left(2^{n+4}\left|\Phi\right|\right)^{\left|\Phi\right|} = 2^{2^{o(n)}}$. Therefore $\mathrm{MFS}_{1/15}\left(f_\psi\right) = 2^{\Omega(n)}$; by Theorem 73, part (iii), this implies that $\mathrm{TS}_{1/16}\left(\left|\psi\right\rangle\right) = 2^{\Omega(n)}$. ■

A corollary of Theorem 76 is the following 'nonamplification' property: there exist states that can be approximated to within, say, 1% by trees of polynomial size, but that require exponentially large trees to approximate to within a smaller margin (say 0.01%).

Corollary 77 *For all $\delta \in (0,1]$, there exists a state $\left|\psi\right\rangle$ such that $\mathrm{TS}_\delta\left(\left|\psi\right\rangle\right) = n$ but $\mathrm{TS}_\varepsilon\left(\left|\psi\right\rangle\right) = 2^{\Omega(n)}$ where $\varepsilon = \delta/32 - \delta^2/4096$.*

Proof. It is clear from Theorem 76 that there exists a state $\left|\varphi\right\rangle = \sum_{x\in\{0,1\}^n} \alpha_x \left|x\right\rangle$ such that $\mathrm{TS}_{1/16}\left(\left|\varphi\right\rangle\right) = 2^{\Omega(n)}$ and $\alpha_{0^n} = 0$. Take $\left|\psi\right\rangle = \sqrt{1-\delta}\left|0\right\rangle^{\otimes n} + \sqrt{\delta}\left|\varphi\right\rangle$. Since $\left|\left\langle\psi\right|\left.0\right\rangle^{\otimes n}\right|^2 = 1-\delta$, we have $\mathrm{MOTS}_\delta\left(\left|\psi\right\rangle\right) = n$. On the other hand, suppose some $\left|\phi\right\rangle = \sum_{x\in\{0,1\}^n} \beta_x \left|x\right\rangle$ with $\mathrm{TS}\left(\left|\phi\right\rangle\right) = 2^{o(n)}$ satisfies $\left|\left\langle\phi\right|\left.\psi\right\rangle\right|^2 \geq 1-\varepsilon$. Then

$$\sum_{x\neq 0^n} \left(\sqrt{\delta}\alpha_x - \beta_x\right)^2 \leq 2 - 2\sqrt{1-\varepsilon}.$$

Thus, letting $f_\varphi\left(x\right) = \alpha_x$, we have $\mathrm{MFS}_c\left(f_\varphi\right) = O\left(\mathrm{TS}\left(\left|\phi\right\rangle\right)\right)$ where $c = \left(2 - 2\sqrt{1-\varepsilon}\right)/\delta$. By Theorem 73, part (iv), this implies that $\mathrm{TS}_{2c}\left(\left|\varphi\right\rangle\right) = O\left(\mathrm{TS}\left(\left|\phi\right\rangle\right)\right)$. But $2c = 1/16$ when $\varepsilon = \delta/32 - \delta^2/4096$, contradiction. ■

13.4 Relations Among Quantum State Classes

This section presents some results about the quantum state hierarchy introduced in Section 13.2. Theorem 78 shows simple inclusions and separations, while Theorem 79 shows that separations higher in the hierarchy would imply major complexity class separations (and vice versa).

Theorem 78

(i) $\mathsf{Tree} \cup \mathsf{Vidal} \subseteq \mathsf{Circuit} \subseteq \mathsf{AmpP}$.

(ii) All states in Vidal *have tree size* $n^{O(\log n)}$.

(iii) $\Sigma_2 \subseteq \mathsf{Vidal}$ *but* $\otimes_2 \not\subset \mathsf{Vidal}$.

(iv) $\otimes_2 \subsetneq \mathsf{MOTree}$.

(v) Σ_1, Σ_2, Σ_3, $\otimes_1$, $\otimes_2$, *and* $\otimes_3$ *are all distinct. Also,* $\otimes_3 \neq \Sigma_4 \cap \otimes_4$.

Proof.

(i) $\mathsf{Tree} \subseteq \mathsf{Circuit}$ since any multilinear tree is also a multilinear circuit. $\mathsf{Circuit} \subseteq \mathsf{AmpP}$ since the circuit yields a polynomial-time algorithm for computing the amplitudes. For $\mathsf{Vidal} \subseteq \mathsf{Circuit}$, we use an idea of Vidal [236]: given $|\psi_n\rangle \in \mathsf{Vidal}$, for all $j \in \{1,\ldots,n\}$ we can express $|\psi_n\rangle$ as

$$\sum_{i=1}^{\chi(|\psi\rangle)} \alpha_{ij} \left|\phi_i^{[1\ldots j]}\right\rangle \otimes \left|\phi_i^{[j+1\ldots n]}\right\rangle$$

where $\chi(|\psi_n\rangle)$ is polynomially bounded. Furthermore, Vidal showed that each $\left|\phi_i^{[1\ldots j]}\right\rangle$ can be written as a linear combination of states of the form $\left|\phi_i^{[1\ldots j-1]}\right\rangle \otimes |0\rangle$ and $\left|\phi_i^{[1\ldots j-1]}\right\rangle \otimes |1\rangle$—the point being that the set of $\left|\phi_i^{[1\ldots j-1]}\right\rangle$ states is the same, independently of $\left|\phi_i^{[1\ldots j]}\right\rangle$. This immediately yields a polynomial-size multilinear circuit for $|\psi_n\rangle$.

(ii) Given $|\psi_n\rangle \in \mathsf{Vidal}$, we can decompose $|\psi_n\rangle$ as

$$\sum_{i=1}^{\chi(|\psi\rangle)} \alpha_i \left|\phi_i^{[1\ldots n/2]}\right\rangle \otimes \left|\phi_i^{[n/2+1\ldots n]}\right\rangle.$$

Then $\chi\left(\left|\phi_i^{[1\ldots n/2]}\right\rangle\right) \leq \chi(|\psi_n\rangle)$ and $\chi\left(\left|\phi_i^{[n/2+1\ldots n]}\right\rangle\right) \leq \chi(|\psi_n\rangle)$ for all i, so we can recursively decompose these states in the same manner. It follows that $\mathrm{TS}(|\psi_n\rangle) \leq 2\chi(|\psi\rangle)\,\mathrm{TS}\left(\left|\psi_{n/2}\right\rangle\right)$; solving this recurrence relation yields $\mathrm{TS}(|\psi_n\rangle) \leq (2\chi(|\psi\rangle))^{\log n} = n^{O(\log n)}$.

(iii) $\Sigma_2 \subseteq \mathsf{Vidal}$ follows since a sum of t separable states has $\chi \leq t$, while $\otimes_2 \not\subseteq \mathsf{Vidal}$ follows from the example of $n/2$ Bell pairs: $2^{-n/4}\left(|00\rangle + |11\rangle\right)^{\otimes n/2}$.

(iv) $\otimes_2 \subseteq \mathsf{MOTree}$ is obvious, while $\mathsf{MOTree} \not\subseteq \otimes_2$ follows from the example of $\left|P_n^i\right\rangle$, an equal superposition over all n-bit strings of parity i. The following recursive formulas imply that $\mathrm{MOTS}\left(\left|P_n^i\right\rangle\right) \leq 4\,\mathrm{MOTS}\left(\left|P_{n/2}^i\right\rangle\right) = O\left(n^2\right)$:

$$\left|P_n^0\right\rangle = \frac{1}{\sqrt{2}}\left(\left|P_{n/2}^0\right\rangle\left|P_{n/2}^0\right\rangle + \left|P_{n/2}^1\right\rangle\left|P_{n/2}^1\right\rangle\right),$$
$$\left|P_n^1\right\rangle = \frac{1}{\sqrt{2}}\left(\left|P_{n/2}^0\right\rangle\left|P_{n/2}^1\right\rangle + \left|P_{n/2}^1\right\rangle\left|P_{n/2}^0\right\rangle\right).$$

On the other hand, $|P_n\rangle \notin \otimes_2$ follows from $|P_n\rangle \notin \Sigma_1$ together with the fact that $|P_n\rangle$ has no nontrivial tensor product decomposition.

(v) $\otimes_1 \not\subseteq \Sigma_1$ and $\Sigma_1 \not\subseteq \otimes_1$ are obvious. $\otimes_2 \not\subseteq \Sigma_2$ (and hence $\otimes_1 \neq \otimes_2$) follows from part (iii). $\Sigma_2 \not\subseteq \otimes_2$ (and hence $\Sigma_1 \neq \Sigma_2$) follows from part (iv), together with the fact that $|P_n\rangle$ has a Σ_2 formula based on the Fourier transform:

$$|P_n\rangle = \frac{1}{\sqrt{2}}\left(\left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)^{\otimes n} + \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)^{\otimes n}\right).$$

$\Sigma_2 \neq \Sigma_3$ follows from $\otimes_2 \not\subset \Sigma_2$ and $\otimes_2 \subseteq \Sigma_3$. Also, $\Sigma_3 \not\subset \otimes_3$ follows from $\Sigma_2 \neq \Sigma_3$, together with the fact that we can easily construct states in $\Sigma_3 \setminus \Sigma_2$ that have no nontrivial tensor product decomposition—for example,

$$\frac{1}{\sqrt{2}}\left(|0\rangle^{\otimes n} + \left(\frac{|01\rangle + |10\rangle}{\sqrt{2}}\right)^{\otimes n/2}\right).$$

$\otimes_2 \neq \otimes_3$ follows from $\Sigma_2 \not\subset \otimes_2$ and $\Sigma_2 \subseteq \otimes_3$. Finally, $\otimes_3 \neq \Sigma_4 \cap \otimes_4$ follows from $\Sigma_3 \not\subset \otimes_3$ and $\Sigma_3 \subseteq \Sigma_4 \cap \otimes_4$.

■

Theorem 79

(i) $\mathsf{BQP} = \mathsf{P}^{\#\mathsf{P}}$ *implies* $\mathsf{AmpP} \subseteq \Psi\mathsf{P}$.

(ii) $\mathsf{AmpP} \subseteq \Psi\mathsf{P}$ *implies* $\mathsf{NP} \subseteq \mathsf{BQP/poly}$.

(iii) $\mathsf{P} = \mathsf{P}^{\#\mathsf{P}}$ *implies* $\Psi\mathsf{P} \subseteq \mathsf{AmpP}$.

(iv) $\Psi\mathsf{P} \subseteq \mathsf{AmpP}$ *implies* $\mathsf{BQP} \subseteq \mathsf{P/poly}$.

Proof.

(i) First, $\mathsf{BQP} = \mathsf{P}^{\#\mathsf{P}}$ implies $\mathsf{BQP/poly} = \mathsf{P}^{\#\mathsf{P}}\mathsf{/poly}$, since given a $\mathsf{P}^{\#\mathsf{P}}\mathsf{/poly}$ machine M, the language consisting of all (x, a) such that M accepts on input x and advice a is clearly in BQP. So assume $\mathsf{BQP/poly} = \mathsf{P}^{\#\mathsf{P}}\mathsf{/poly}$, and consider a state $|\psi\rangle = \sum_{x\in\{0,1\}^n} \alpha_x |x\rangle$ with $|\psi\rangle \in \mathsf{AmpP}$. By the result of Bernstein and Vazirani [55] that $\mathsf{BQP} \subseteq \mathsf{P}^{\#\mathsf{P}}$, for all b there exists a quantum circuit of size polynomial in n and b that approximates $p_0 = \sum_{y\in\{0,1\}^{n-1}} |\alpha_{0y}|^2$, or the probability that the first qubit is measured to be 0, to b bits of precision. So by uncomputing garbage, we can prepare a state close to $\sqrt{p_0}\,|0\rangle + \sqrt{1-p_0}\,|1\rangle$. Similarly, given a superposition over length-k prefixes of x, we can prepare a superposition over length-$(k+1)$ prefixes of x by approximating the conditional measurement probabilities. We thus obtain a state close to $\sum_x |\alpha_x|\,|x\rangle$. The last step is to approximate the phase of each $|x\rangle$, apply that phase, and uncompute to obtain a state close to $\sum_x \alpha_x |x\rangle$.

(ii) Given a *SAT* instance φ, first use Valiant-Vazirani [231] to produce a formula φ' that (with non-negligible probability) has one satisfying assignment if φ is satisfiable and zero otherwise. Then let $\alpha_x = 1$ if x is a satisfying assignment for φ' and $\alpha_x = 0$ otherwise; clearly $|\psi\rangle = \sum_x \alpha_x |x\rangle$ is in AmpP. By the assumption $\mathsf{AmpP} \subseteq \Psi\mathsf{P}$, there exists a polynomial-size quantum circuit that approximates $|\psi\rangle$, and thereby finds the unique satisfying assignment for φ' if it exists.

(iii) As in part (i), $\mathsf{P} = \mathsf{P}^{\#\mathsf{P}}$ implies $\mathsf{P/poly} = \mathsf{P}^{\#\mathsf{P}}\mathsf{/poly}$. The containment $\Psi\mathsf{P} \subseteq \mathsf{AmpP}$ follows since we can approximate amplitudes to polynomially many bits of precision in $\#\mathsf{P}$.

(iv) As is well known [55], any quantum computation can be made 'clean' in the sense that it accepts if and only if a particular basis state (say $|0\rangle^{\otimes n}$) is measured. The implication follows easily.

■

13.5 Lower Bounds

We want to show that certain quantum states of interest to us are not represented by trees of polynomial size. At first this seems like a hopeless task. Proving superpolynomial formula-size lower bounds for 'explicit' functions is a notoriously hard open problem, as it would imply complexity class separations such as $\mathsf{NC}^1 \neq \mathsf{P}$.

Here, though, we are only concerned with *multilinear* formulas. Could this make it easier to prove a lower bound? The answer is not obvious, but very recently, for reasons unrelated to quantum computing, Raz [195, 196] showed the first superpolynomial lower bounds on multilinear formula size. In particular, he showed that multilinear formulas computing the permanent or determinant of an $n \times n$ matrix over any field have size $n^{\Omega(\log n)}$.

Raz's technique is a beautiful combination of the Furst-Saxe-Sipser method of random restrictions [120], with matrix rank arguments as used in communication complexity. I now outline the method. Given a function $f : \{0,1\}^n \rightarrow \mathbb{C}$, let P be a partition of the input variables $x_1, \ldots, x_n$ into two collections $y = \left(y_1, \ldots, y_{n/2}\right)$ and $z = \left(z_1, \ldots, z_{n/2}\right)$. This yields a function $f_P(y,z) : \{0,1\}^{n/2} \times \{0,1\}^{n/2} \rightarrow \mathbb{C}$. Then let $M_{f|P}$ be a $2^{n/2} \times 2^{n/2}$ matrix whose rows are labeled by assignments $y \in \{0,1\}^{n/2}$, and whose columns are labeled by assignments $z \in \{0,1\}^{n/2}$. The (y,z) entry of $M_{f|P}$ is $f_P(y,z)$. Let $\mathrm{rank}\left(M_{f|P}\right)$ be the rank of $M_{f|P}$ over the complex numbers. Finally, let $\mathcal{P}$ be the uniform distribution over all partitions P.

The following, Corollary 3.6 in [196], is one statement of Raz's main theorem; recall that $\mathrm{MFS}(f)$ is the minimum size of a multilinear formula for f.

Theorem 80 ([196]) *Suppose that*

$$\Pr_{P \in \mathcal{P}}\left[\mathrm{rank}\left(M_{f|P}\right) \geq 2^{n/2-(n/2)^{1/8}/2}\right] = n^{-o(\log n)}.$$

Then $\mathrm{MFS}(f) = n^{\Omega(\log n)}$.

An immediate corollary yields lower bounds on *approximate* multilinear formula size. Given an $N \times N$ matrix $M = (m_{ij})$, let $\mathrm{rank}_\varepsilon(M) = \min_{L \,:\, \|L-M\|_2^2 \leq \varepsilon} \mathrm{rank}(L)$ where $\|L-M\|_2^2 = \sum_{i,j=1}^N |\ell_{ij} - m_{ij}|^2$.

Corollary 81 *Suppose that*

$$\Pr_{P \in \mathcal{P}}\left[\mathrm{rank}_\varepsilon\left(M_{f|P}\right) \geq 2^{n/2-(n/2)^{1/8}/2}\right] = n^{-o(\log n)}.$$

Then $\mathrm{MFS}_\varepsilon(f) = n^{\Omega(\log n)}$.

Proof. Suppose $\mathrm{MFS}_\varepsilon(f) = n^{o(\log n)}$. Then for all g such that $\|f-g\|_2^2 \leq \varepsilon$, we would have $\mathrm{MFS}(g) = n^{o(\log n)}$, and therefore

$$\Pr_{P\in\mathcal{P}}\left[\mathrm{rank}\left(M_{g|P}\right) \geq 2^{n/2-(n/2)^{1/8}/2}\right] = n^{-\Omega(\log n)}.$$

by Theorem 80. But $\mathrm{rank}_\varepsilon\left(M_{f|P}\right) \leq \mathrm{rank}\left(M_{g|P}\right)$, and hence

$$\Pr_{P\in\mathcal{P}}\left[\mathrm{rank}_\varepsilon\left(M_{f|P}\right) \geq 2^{n/2-(n/2)^{1/8}/2}\right] = n^{-\Omega(\log n)},$$

contradiction. ■

Another simple corollary gives lower bounds in terms of *restrictions* of f. Let $\mathcal{R}_\ell$ be the following distribution over restrictions R: choose 2ℓ variables of f uniformly at random, and rename them $y = (y_1, \ldots, y_\ell)$ and $z = (z_1, \ldots, z_\ell)$. Set each of the remaining $n - 2\ell$ variables to 0 or 1 uniformly and independently at random. This yields a restricted function $f_R(y,z)$. Let $M_{f|R}$ be a $2^\ell \times 2^\ell$ matrix whose (y,z) entry is $f_R(y,z)$.

Corollary 82 *Suppose that*

$$\Pr_{R\in\mathcal{R}_\ell}\left[\mathrm{rank}\left(M_{f|R}\right) \geq 2^{\ell-\ell^{1/8}/2}\right] = n^{-o(\log n)}$$

where $\ell = n^\delta$ *for some constant* $\delta \in (0,1]$. *Then* $\mathrm{MFS}(f) = n^{\Omega(\log n)}$.

Proof. Under the hypothesis, clearly there exists a *fixed* restriction $g : \{0,1\}^{2\ell} \rightarrow \mathbb{C}$ of f, which leaves 2ℓ variables unrestricted, such that

$$\Pr_{P\in\mathcal{P}}\left[\mathrm{rank}\left(M_{g|P}\right) \geq 2^{\ell-\ell^{1/8}/2}\right] = n^{-o(\log n)} = \ell^{-o(\log \ell)}.$$

Then by Theorem 80,

$$\mathrm{MFS}(f) \geq \mathrm{MFS}(g) = \ell^{\Omega(\log \ell)} = n^{\Omega(\log n)}.$$

■

The following sections apply Raz's theorem to obtain $n^{\Omega(\log n)}$ tree size lower bounds for two classes of quantum states: states arising in quantum error-correction in Section 13.5.1, and (assuming a number-theoretic conjecture) states arising in Shor's factoring algorithm in Section 13.5.2.

13.5.1 Subgroup States

Let the elements of $\mathbb{Z}_2^n$ be labeled by n-bit strings. Given a subgroup $S \leq \mathbb{Z}_2^n$, we define the *subgroup state* $|S\rangle$ as follows:

$$|S\rangle = \frac{1}{\sqrt{|S|}}\sum_{x\in S}|x\rangle.$$

Coset states arise as codewords in the class of quantum error-correcting codes known as stabilizer codes [80, 133, 225]. Our interest in these states, however, arises from their large tree size rather than their error-correcting properties.

Let $\mathcal{E}$ be the following distribution over subgroups S. Choose an $n/2 \times n$ matrix A by setting each entry to 0 or 1 uniformly and independently. Then let $S = \{x \mid Ax \equiv 0 \,(\mathrm{mod}\, 2)\}$. By Theorem 73, part (i), it suffices to lower-bound the multilinear formula size of the function $f_S(x)$, which is 1 if $x \in S$ and 0 otherwise.

Theorem 83 *If S is drawn from $\mathcal{E}$, then* $\mathrm{MFS}(f_S) = n^{\Omega(\log n)}$ *(and hence* $\mathrm{TS}(|S\rangle) = n^{\Omega(\log n)}$*), with probability $\Omega(1)$ over S.*

Proof. Let P be a uniform random partition of the inputs $x_1, \ldots, x_n$ of f_S into two sets $y = (y_1, \ldots, y_{n/2})$ and $z = (z_1, \ldots, z_{n/2})$. Let $M_{S|P}$ be the $2^{n/2} \times 2^{n/2}$ matrix whose (y, z) entry is $f_{S|P}(y, z)$; then we need to show that $\mathrm{rank}\left(M_{S|P}\right)$ is large with high probability. Let A_y be the $n/2 \times n/2$ submatrix of the $n/2 \times n$ matrix A consisting of all rows that correspond to y_i for some $i \in \{1, \ldots, n/2\}$, and similarly let A_z be the $n/2 \times n/2$ submatrix corresponding to z. Then it is easy to see that, so long as A_y and A_z are both invertible, for all $2^{n/2}$ settings of y there exists a *unique* setting of z for which $f_{S|P}(y, z) = 1$. This then implies that $M_{S|P}$ is a permutation of the identity matrix, and hence that $\mathrm{rank}\left(M_{S|P}\right) = 2^{n/2}$. Now, the probability that a random $n/2 \times n/2$ matrix over $\mathbb{Z}_2$ is invertible is

$$\frac{1}{2} \cdot \frac{3}{4} \cdot \cdots \cdot \frac{2^{n/2} - 1}{2^{n/2}} > 0.288.$$

So the probability that A_y and A_z are both invertible is at least 0.288^2. By Markov's inequality, it follows that for at least an 0.04 fraction of S's, $\mathrm{rank}\left(M_{S|P}\right) = 2^{n/2}$ for at least an 0.04 fraction of P's. Theorem 80 then yields the desired result. ■

Aaronson and Gottesman [14] showed how to prepare any n-qubit subgroup state using a quantum circuit of size $O\left(n^2/\log n\right)$. So a corollary of Theorem 83 is that $\mathsf{\Psi P} \not\subset \mathsf{Tree}$. Since f_S clearly has a (non-multilinear) arithmetic formula of size $O(nk)$, a second corollary is the following.

Corollary 84 *There exists a family of functions $f_n : \{0,1\}^n \to \mathbb{R}$ that has polynomial-size arithmetic formulas, but no polynomial-size multilinear formulas.*

The reason Corollary 84 does not follow from Raz's results is that polynomial-size formulas for the permanent and determinant are not known; the smallest known formulas for the determinant have size $n^{O(\log n)}$ (see [79]).

We have shown that not all subgroup states are tree states, but it is still conceivable that all subgroup states are extremely well *approximated* by tree states. Let us now rule out the latter possibility. We first need a lemma about matrix rank, which follows from the Hoffman-Wielandt inequality.

Lemma 85 *Let M be an $N \times N$ complex matrix, and let I_N be the $N \times N$ identity matrix. Then $\|M - I_N\|_2^2 \geq N - \mathrm{rank}(M)$.*

Proof. The Hoffman-Wielandt inequality [144] (see also [33]) states that for any two $N \times N$ matrices M, P,

$$\sum_{i=1}^{N} \left(\sigma_i\left(M\right) - \sigma_i\left(P\right)\right)^2 \leq \left\|M - P\right\|_2^2,$$

where $\sigma_i\left(M\right)$ is the i^{th} singular value of M (that is, $\sigma_i\left(M\right) = \sqrt{\lambda_i\left(MM\right)}$, where $\lambda_1\left(M\right) \geq \cdots \geq \lambda_N\left(M\right) \geq 0$ are the eigenvalues of MM^*, and M^* is the conjugate transpose of M). Clearly $\sigma_i\left(I_N\right) = 1$ for all i. On the other hand, M has only $\mathrm{rank}\left(M\right)$ nonzero singular values, so

$$\sum_{i=1}^{N} \left(\sigma_i\left(M\right) - \sigma_i\left(I_N\right)\right)^2 \geq N - \mathrm{rank}\left(M\right).$$

■

Let $\widehat{f}_S\left(x\right) = f_S\left(x\right)/\sqrt{|S|}$ be $f_S\left(x\right)$ normalized to have $\left\|\widehat{f}_S\right\|_2^2 = 1$.

Theorem 86 *For all constants $\varepsilon \in [0,1)$, if S is drawn from $\mathcal{E}$, then $\mathrm{MFS}_\varepsilon\left(\widehat{f}_S\right) = n^{\Omega(\log n)}$ with probability $\Omega\left(1\right)$ over S.*

Proof. As in Theorem 83, we look at the matrix $M_{S|P}$ induced by a random partition $P = (y, z)$. We already know that for at least an 0.04 fraction of S's, the y and z variables are in one-to-one correspondence for at least an 0.04 fraction of P's. In that case $|S| = 2^{n/2}$, and therefore $M_{S|P}$ is a permutation of $I/\sqrt{|S|} = I/2^{n/4}$ where I is the identity. It follows from Lemma 85 that for all matrices M such that $\left\|M - M_{S|P}\right\|_2^2 \leq \varepsilon$,

$$\mathrm{rank}\left(M\right) \geq 2^{n/2} - \left\|\sqrt{|S|}\left(M - M_{S|P}\right)\right\|_2^2 \geq \left(1-\varepsilon\right)2^{n/2}$$

and therefore $\mathrm{rank}_\varepsilon\left(M_{S|P}\right) \geq \left(1-\varepsilon\right)2^{n/2}$. Hence

$$\Pr_{P \in \mathcal{P}}\left[\mathrm{rank}_\varepsilon\left(M_{f|P}\right) \geq 2^{n/2-(n/2)^{1/8}/2}\right] \geq 0.04,$$

and the result follows from Corollary 81. ■

A corollary of Theorem 86 and of Theorem 73, part (iii), is that $\mathrm{TS}_\varepsilon\left(|S\rangle\right) = n^{\Omega(\log n)}$ with probability $\Omega\left(1\right)$ over S, for all $\varepsilon < 1$.

Finally, let me show how to derandomize the lower bound for subgroup states, using ideas pointed out to me by Andrej Bogdanov. In the proof of Theorem 83, all we needed about the matrix A was that a random $k \times k$ submatrix has full rank with $\Omega\left(1\right)$ probability, where $k = n/2$. If we switch from the field $\mathbb{F}_2$ to $\mathbb{F}_{2^d}$ for some $d \geq \log_2 n$, then it is easy to construct explicit $k \times n$ matrices with this same property. For example, let

$$V = \begin{pmatrix} 1^0 & 1^1 & \cdots & 1^{k-1} \\ 2^0 & 2^1 & \cdots & 2^{k-1} \\ \vdots & \vdots & & \vdots \\ n^0 & n^1 & \cdots & n^{k-1} \end{pmatrix}$$

be the $n \times k$ Vandermonde matrix, where $1, \ldots, n$ are labels of elements in $\mathbb{F}_{2^d}$. *Any* $k \times k$ submatrix of V has full rank, because the Reed-Solomon (RS) code that V represents is a perfect erasure code.[6] Hence, there exists an explicit state of n "qupits" with $p = 2^d$ that has tree size $n^{\Omega(\log n)}$—namely the uniform superposition over all elements of the set $\left\{x \mid V^T x = 0\right\}$, where V^T is the transpose of V.

To replace qupits by qubits, we concatenate the RS and Hadamard codes to obtain a *binary* linear erasure code with parameters almost as good as those of the original RS code. More explicitly, interpret $\mathbb{F}_{2^d}$ as the field of polynomials over $\mathbb{F}_2$, modulo some irreducible of degree d. Then let $m(a)$ be the $d \times d$ Boolean matrix that maps $q \in \mathbb{F}_{2^d}$ to $aq \in \mathbb{F}_{2^d}$, where q and aq are encoded by their $d \times 1$ vectors of coefficients. Let H map a length-d vector to its length-2^d Hadamard encoding. Then $Hm(a)$ is a $2^d \times d$ Boolean matrix that maps $q \in \mathbb{F}_{2^d}$ to the Hadamard encoding of aq. We can now define an $n2^d \times kd$ "binary Vandermonde matrix" as follows:

$$V_{\text{bin}} = \begin{pmatrix} Hm\left(1^0\right) & Hm\left(1^1\right) & \cdots & Hm\left(1^{k-1}\right) \\ Hm\left(2^0\right) & Hm\left(2^1\right) & \cdots & Hm\left(2^{k-1}\right) \\ \vdots & \vdots & & \vdots \\ Hm\left(n^0\right) & Hm\left(n^1\right) & \cdots & Hm\left(n^{k-1}\right) \end{pmatrix}.$$

For the remainder of the section, fix $k = n^\delta$ for some $\delta < 1/2$ and $d = O(\log n)$.

Lemma 87 *A $(kd + c) \times kd$ submatrix of V_{bin} chosen uniformly at random has rank kd (that is, full rank) with probability at least $2/3$, for c a sufficiently large constant.*

Proof. The first claim is that $|V_{\text{bin}} u| \geq (n-k)\, 2^{d-1}$ for all nonzero vectors $u \in \mathbb{F}_2^{kd}$, where $|\ \ |$ represents the number of '1' bits. To see this, observe that for all nonzero u, the "codeword vector" $Vu \in \mathbb{F}_{2^d}^n$ must have at least $n-k$ nonzero entries by the Fundamental Theorem of Algebra, where here u is interpreted as an element of $\mathbb{F}_{2^d}^k$. Furthermore, the Hadamard code maps any nonzero entry in Vu to 2^{d-1} nonzero bits in $V_{\text{bin}} u \in \mathbb{F}_2^{n2^d}$.

Now let W be a uniformly random $(kd + c) \times kd$ submatrix of V_{bin}. By the above claim, for any fixed nonzero vector $u \in \mathbb{F}_2^{kd}$,

$$\Pr_W [Wu = 0] \leq \left(1 - \frac{(n-k)\, 2^{d-1}}{n2^d}\right)^{kd+c} = \left(\frac{1}{2} + \frac{k}{2n}\right)^{kd+c}.$$

So by the union bound, Wu is nonzero for all nonzero u (and hence W is full rank) with probability at least

$$1 - 2^{kd}\left(\frac{1}{2} + \frac{k}{2n}\right)^{kd+c} = 1 - \left(1 + \frac{k}{n}\right)^{kd}\left(\frac{1}{2} + \frac{k}{2n}\right)^{c}.$$

Since $k = n^{1/2-\Omega(1)}$ and $d = O(\log n)$, the above quantity is at least $2/3$ for sufficiently large c. ■

Given an $n2^d \times 1$ Boolean vector x, let $f(x) = 1$ if $V_{\text{bin}}^T x = 0$ and $f(x) = 0$ otherwise. Then:

[6] In other words, because a degree-$(k-1)$ polynomial is determined by its values at any k points.

Theorem 88 $\mathrm{MFS}(f) = n^{\Omega(\log n)}$.

Proof. Let V_y and V_z be two disjoint $kd \times (kd+c)$ submatrices of V_{bin}^T chosen uniformly at random. Then by Lemma 87 together with the union bound, V_y and V_z both have full rank with probability at least $1/3$. Letting $\ell = kd + c$, it follows that

$$\Pr_{R \in \mathcal{R}_\ell}\left[\mathrm{rank}\left(M_{f|R}\right) \geq 2^{\ell-c}\right] \geq \frac{1}{3} = n^{-o(\log n)}$$

by the same reasoning as in Theorem 83. Therefore $\mathrm{MFS}(f) = n^{\Omega(\log n)}$ by Corollary 82. ■

Let $|S\rangle$ be a uniform superposition over all x such that $f(x) = 1$; then a corollary of Theorem 88 is that $\mathrm{TS}(|S\rangle) = n^{\Omega(\log n)}$. Naturally, using the ideas of Theorem 86 one can also show that $\mathrm{TS}_\varepsilon(|S\rangle) = n^{\Omega(\log n)}$ for all $\varepsilon < 1$.

13.5.2 Shor States

Since the motivation for this work was to study possible Sure/Shor separators, an obvious question is, *do states arising in Shor's algorithm have superpolynomial tree size?* Unfortunately, I am only able to answer this question assuming a number-theoretic conjecture. To formalize the question, let

$$\frac{1}{2^{n/2}} \sum_{r=0}^{2^n-1} |r\rangle\,|x^r \bmod N\rangle$$

be a Shor state. It will be convenient to measure the second register, so that the state of the first register has the form

$$|a + p\mathbb{Z}\rangle = \frac{1}{\sqrt{I}} \sum_{i=0}^{I} |a + pi\rangle$$

for some integers $a < p$ and $I = \lfloor (2^n - a - 1)/p \rfloor$. Here $a+pi$ is written out in binary using n bits. Clearly a lower bound on $\mathrm{TS}(|a + p\mathbb{Z}\rangle)$ would imply an equivalent lower bound for the joint state of the two registers.

To avoid some technicalities, assume p is prime (since the goal is to prove a *lower* bound, this assumption is without loss of generality). Given an n-bit string $x = x_{n-1} \ldots x_0$, let $f_{n,p,a}(x) = 1$ if $x \equiv a \pmod p$ and $f_{n,p,a}(x) = 0$ otherwise. Then $\mathrm{TS}(|a + p\mathbb{Z}\rangle) = \Theta(\mathrm{MFS}(f_{n,p,a}))$ by Theorem 73, so from now on we will focus attention on $f_{n,p,a}$.

Proposition 89

(i) Let $f_{n,p} = f_{n,p,0}$. *Then* $\mathrm{MFS}(f_{n,p,a}) \leq \mathrm{MFS}(f_{n+\log p,p})$, *meaning that we can set* $a = 0$ *without loss of generality.*

(ii) $\mathrm{MFS}(f_{n,p}) = O(\min\{n2^n/p, np\})$.

Proof.

(i) Take the formula for $f_{n+\log p,p}$, and restrict the most significant $\log p$ bits to sum to a number congruent to $-a \bmod p$ (this is always possible since $x \rightarrow 2^n x$ is an isomorphism of $\mathbb{Z}_p$).

(ii) For $\text{MFS}(f_{n,p}) = O(n2^n/p)$, write out the x's for which $f_{n,p}(x) = 1$ explicitly. For $\text{MFS}(f_{n,p}) = O(np)$, use the Fourier transform, similarly to Theorem 78, part (v):

$$f_{n,p}(x) = \frac{1}{p} \sum_{h=0}^{p-1} \prod_{j=0}^{n-1} \exp\left(\frac{2\pi i h}{p} \cdot 2^j x_j\right).$$

This immediately yields a sum-of-products formula of size $O(np)$.

■

I now state the number-theoretic conjecture.

Conjecture 90 *There exist constants $\gamma, \delta \in (0,1)$ and a prime $p = \Omega\left(2^{n^\delta}\right)$ for which the following holds. Let the set A consist of n^δ elements of $\left\{2^0, \ldots, 2^{n-1}\right\}$ chosen uniformly at random. Let S consist of all 2^{n^δ} sums of subsets of A, and let $S \bmod p = \{x \bmod p : x \in S\}$. Then*

$$\Pr_A\left[|S \bmod p| \geq (1+\gamma)\frac{p}{2}\right] = n^{-o(\log n)}.$$

Theorem 91 *Conjecture 90 implies that $\text{MFS}(f_{n,p}) = n^{\Omega(\log n)}$ and hence $\text{TS}(|p\mathbb{Z}\rangle) = n^{\Omega(\log n)}$.*

Proof. Let $f = f_{n,p}$ and $\ell = n^\delta$. Let R be a restriction of f that renames 2ℓ variables $y_1, \ldots, y_\ell, z_1, \ldots, z_\ell$, and sets each of the remaining $n - 2\ell$ variables to 0 or 1. This leads to a new function, $f_R(y,z)$, which is 1 if $y + z + c \equiv 0 \pmod p$ and 0 otherwise for some constant c. Here we are defining $y = 2^{a_1} y_1 + \cdots + 2^{a_\ell} y_\ell$ and $z = 2^{b_1} z_1 + \cdots + 2^{b_\ell} z_\ell$ where $a_1, \ldots, a_\ell, b_1, \ldots, b_\ell$ are the appropriate place values. Now suppose $y \bmod p$ and $z \bmod p$ both assume at least $(1+\gamma)p/2$ distinct values as we range over all $x \in \{0,1\}^n$. Then by the pigeonhole principle, for at least γp possible values of $y \bmod p$, there exists a unique possible value of $z \bmod p$ for which $y + z + c \equiv 0 \pmod p$ and hence $f_R(y,z) = 1$. So $\text{rank}\left(M_{f|R}\right) \geq \gamma p$, where $M_{f|R}$ is the $2^\ell \times 2^\ell$ matrix whose (y,z) entry is $f_R(y,z)$. It follows that assuming Conjecture 90,

$$\Pr_{R \in \mathcal{R}_\ell}\left[\text{rank}\left(M_{f|R}\right) \geq \gamma p\right] = n^{-o(\log n)}.$$

Furthermore, $\gamma p \geq 2^{\ell - \ell^{1/8}/2}$ for sufficiently large n since $p = \Omega\left(2^{n^\delta}\right)$. Therefore $\text{MFS}(f) = n^{\Omega(\log n)}$ by Corollary 82. ■

Using the ideas of Theorem 86, one can show that under the same conjecture, $\text{MFS}_\varepsilon(f_{n,p}) = n^{\Omega(\log n)}$ and $\text{TS}_\varepsilon(|p\mathbb{Z}\rangle) = n^{\Omega(\log n)}$ for all $\varepsilon < 1$—in other words, there exist Shor states that cannot be approximated by polynomial-size trees.

Originally, I had stated Conjecture 90 without any restriction on how the set S is formed. The resulting conjecture was far more general than I needed, and indeed was falsified by Carl Pomerance (personal communication).

13.5.3 Tree Size and Persistence of Entanglement

In this section I pursue a deeper understanding of the tree size lower bounds, by discussing a *physical* property of quantum states that is related to error-correction as well as to super-polynomial tree size. Dür and Briegel [100] call a quantum state "persistently entangled," if (roughly speaking) it remains highly entangled even after a limited amount of interaction with its environment. As an illustration, the Schrödinger cat state $\left(|0\rangle^{\otimes n} + |1\rangle^{\otimes n}\right)/\sqrt{2}$ is in some sense highly entangled, but it is *not* persistently entangled, since measuring a single qubit in the standard basis destroys all entanglement.

By contrast, consider the "cluster states" defined by Briegel and Raussendorf [71]. These states have attracted a great deal of attention because of their application to quantum computing via 1-qubit measurements only [194]. For our purposes, a two-dimensional cluster state is an equal superposition over all settings of a $\sqrt{n} \times \sqrt{n}$ array of bits, with each basis state having a phase of $(-1)^r$, where r is the number of horizontally or vertically adjacent pairs of bits that are both '1'. Dür and Briegel [100] showed that such states are persistently entangled in a precise sense: one can distill n-partite entanglement from them even after each qubit has interacted with a heat bath for an amount of time independent of n.

Persistence of entanglement seems related to how one shows tree size lower bounds using Raz's technique. For to apply Corollary 82, one basically "measures" most of a state's qubits, then partitions the unmeasured qubits into two subsystems of equal size, and argues that with high probability those two subsystems are still almost maximally entangled. The connection is not perfect, though. For one thing, setting most of the qubits to 0 or 1 uniformly at random is not the same as measuring them. For another, Theorem 80 yields $n^{\Omega(\log n)}$ tree size lower bounds without the need to trace out a subset of qubits. It suffices for the *original* state to be almost maximally entangled, no matter how one partitions it into two subsystems of equal size.

But what about 2-D cluster states—do *they* have tree size $n^{\Omega(\log n)}$? I strongly conjecture that the answer is 'yes.' However, proving this conjecture will almost certainly require going beyond Theorem 80. One will want to use random restrictions that respect the 2-D neighborhood structure of cluster states—similar to the restrictions used by Raz [195] to show that the permanent and determinant have multilinear formula size $n^{\Omega(\log n)}$.

I end this section by showing that there exist states that are persistently entangled in the sense of Dür and Briegel [100], but that have polynomial tree size. In particular, Dür and Briegel showed that even *one*-dimensional cluster states are persistently entangled. On the other hand:

Proposition 92 *Let*

$$|\psi\rangle = \frac{1}{2^{n/2}} \sum_{x \in \{0,1\}^n} (-1)^{x_1 x_2 + x_2 x_3 + \cdots + x_{n-1} x_n} |x\rangle .$$

Then $\mathrm{TS}(|\psi\rangle) = O\left(n^4\right)$.

Proof. Given bits i, j, k, let $\left|P_n^{ijk}\right\rangle$ be an equal superposition over all n-bit strings

$x_1 \dots x_n$ such that $x_1 = i$, $x_n = k$, and $x_1 x_2 + \cdots + x_{n-1} x_n \equiv j \pmod 2$. Then

$$\left|P_n^{i0k}\right\rangle = \frac{1}{\sqrt{8}} \left(\begin{array}{c} \left|P_{n/2}^{i00}\right\rangle \left|P_{n/2}^{00k}\right\rangle + \left|P_{n/2}^{i10}\right\rangle \left|P_{n/2}^{01k}\right\rangle + \left|P_{n/2}^{i00}\right\rangle \left|P_{n/2}^{10k}\right\rangle + \left|P_{n/2}^{i10}\right\rangle \left|P_{n/2}^{11k}\right\rangle + \\ \left|P_{n/2}^{i01}\right\rangle \left|P_{n/2}^{00k}\right\rangle + \left|P_{n/2}^{i11}\right\rangle \left|P_{n/2}^{01k}\right\rangle + \left|P_{n/2}^{i01}\right\rangle \left|P_{n/2}^{11k}\right\rangle + \left|P_{n/2}^{i11}\right\rangle \left|P_{n/2}^{10k}\right\rangle \end{array} \right),$$

$$\left|P_n^{i1k}\right\rangle = \frac{1}{\sqrt{8}} \left(\begin{array}{c} \left|P_{n/2}^{i00}\right\rangle \left|P_{n/2}^{01k}\right\rangle + \left|P_{n/2}^{i10}\right\rangle \left|P_{n/2}^{00k}\right\rangle + \left|P_{n/2}^{i00}\right\rangle \left|P_{n/2}^{11k}\right\rangle + \left|P_{n/2}^{i10}\right\rangle \left|P_{n/2}^{10k}\right\rangle + \\ \left|P_{n/2}^{i01}\right\rangle \left|P_{n/2}^{01k}\right\rangle + \left|P_{n/2}^{i11}\right\rangle \left|P_{n/2}^{00k}\right\rangle + \left|P_{n/2}^{i01}\right\rangle \left|P_{n/2}^{10k}\right\rangle + \left|P_{n/2}^{i11}\right\rangle \left|P_{n/2}^{11k}\right\rangle \end{array} \right).$$

Therefore $\mathrm{TS}\left(\left|P_n^{ijk}\right\rangle\right) \leq 16\,\mathrm{TS}\left(\left|P_{n/2}^{ijk}\right\rangle\right)$, and solving this recurrence relation yields

$$\mathrm{TS}\left(\left|P_n^{ijk}\right\rangle\right) = O\left(n^4\right).$$

Finally observe that

$$|\psi\rangle = \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)^{\otimes n} - \frac{\left|P_n^{010}\right\rangle + \left|P_n^{011}\right\rangle + \left|P_n^{110}\right\rangle + \left|P_n^{111}\right\rangle}{\sqrt{2}}.$$

■

13.6 Manifestly Orthogonal Tree Size

This section studies the manifestly orthogonal tree size of coset states:[7] states having the form

$$|C\rangle = \frac{1}{\sqrt{|C|}} \sum_{x \in C} |x\rangle$$

where $C = \{x \mid Ax \equiv b\}$ is a coset in $\mathbb{Z}_2^n$. In particular, I present a *tight* characterization of $\mathrm{MOTS}\left(|C\rangle\right)$, which enables me to prove *exponential* lower bounds on it, in contrast to the $n^{\Omega(\log n)}$ lower bounds for ordinary tree size. This characterization also yields a separation between orthogonal and manifestly orthogonal tree size; and an algorithm for computing $\mathrm{MOTS}\left(|C\rangle\right)$ whose complexity is only singly exponential in n. My proof technique is independent of Raz's, and is highly tailored to take advantage of manifest orthogonality. However, even if this technique finds no other application, it has two features that I hope will make it of independent interest to complexity theorists. First, it yields tight lower bounds, and second, it does not obviously "naturalize" in the sense of Razborov and Rudich [200]. Rather, it takes advantage of certain structural properties of coset states that do not seem to hold for random states.

Given a state $|\psi\rangle$, recall that the manifestly orthogonal tree size $\mathrm{MOTS}\left(|\psi\rangle\right)$ is the minimum size of a tree representing $|\psi\rangle$, in which all additions are of two states $|\psi_1\rangle, |\psi_2\rangle$ with "disjoint supports"—that is, either $\langle\psi_1|x\rangle = 0$ or $\langle\psi_2|x\rangle = 0$ for every basis state $|x\rangle$. Here the size $|T|$ of T is the number of leaf vertices. We can assume without loss

[7] All results apply equally well to the subgroup states of Section 13.5.1; the greater generality of coset states is just for convenience.

of generality that every + or $\otimes$ vertex has at least one child, and that every child of a + vertex is a $\otimes$ vertex and vice versa. Also, given a set $S \subseteq \{0,1\}^n$, let

$$|S\rangle = \frac{1}{\sqrt{|S|}} \sum_{x \in S} |x\rangle$$

be a uniform superposition over the elements of S, and let $M(S) := \text{MOTS}(|S\rangle)$.

Let $C = \{x : Ax \equiv b\}$ be a subgroup in $\mathbb{Z}_2^n$, for some $A \in \mathbb{Z}_2^{k \times n}$ and $b \in \mathbb{Z}_2^k$. Let $[n] = \{1, \ldots, n\}$, and let (I, J) be a nontrivial partition of $[n]$ (one where I and J are both nonempty). Then clearly there exist distinct cosets $C_I^{(1)}, \ldots, C_I^{(H)}$ in the I subsystem, and distinct cosets $C_J^{(1)}, \ldots, C_J^{(H)}$ in the J subsystem, such that

$$C = \bigcup_{h \in [H]} C_I^{(h)} \otimes C_J^{(h)}.$$

The $C_I^{(h)}$'s and $C_J^{(h)}$'s are unique up to ordering. Furthermore, the quantities $\left|C_I^{(h)}\right|$, $\left|C_J^{(h)}\right|$, $M\left(C_I^{(h)}\right)$, and $M\left(C_J^{(h)}\right)$ remain unchanged as we range over $h \in [H]$. For this reason I suppress the dependence on h when mentioning them.

For various sets S, the strategy will be to analyze $M(S)/|S|$, the ratio of tree size to cardinality. We can think of this ratio as the "price per pound" of S: the number of vertices that we have to pay per basis state that we cover. The following lemma says that, under that cost measure, a coset is "as good a deal" as any of its subsets:

Lemma 93 *For all cosets C,*

$$\frac{M(C)}{|C|} = \min\left(\frac{M(S)}{|S|}\right)$$

where the minimum is over nonempty $S \subseteq C$.

Proof. By induction on n. The base case $n = 1$ is obvious, so assume the lemma true for $n-1$. Choose $S^* \subseteq C$ to minimize $M(S^*)/|S^*|$. Let T be a manifestly orthogonal tree for $|S^*\rangle$ of minimum size, and let v be the root of T. We can assume without loss of generality that v is a $\otimes$ vertex, since otherwise v has some $\otimes$ child representing a set $R \subset S^*$ such that $M(R)/|R| \leq M(S^*)/|S^*|$. Therefore for some nontrivial partition (I, J) of $[n]$, and some $S_I^* \subseteq \{0,1\}^{|I|}$ and $S_J^* \subseteq \{0,1\}^{|J|}$, we have

$$\begin{aligned} |S^*\rangle &= |S_I^*\rangle \otimes |S_J^*\rangle, \\ |S^*| &= |S_I^*|\,|S_J^*|, \\ M(S^*) &= M(S_I^*) + M(S_J^*), \end{aligned}$$

where the last equality holds because if $M(S^*) < M(S_I^*) + M(S_J^*)$, then T was not a minimal tree for $|S^*\rangle$. Then

$$\frac{M(S^*)}{|S^*|} = \frac{M(S_I^*) + M(S_J^*)}{|S_I^*|\,|S_J^*|} = \min\left(\frac{M(S_I) + M(S_J)}{|S_I|\,|S_J|}\right)$$

where the minimum is over nonempty $S_I \subseteq \{0,1\}^{|I|}$ and $S_J \subseteq \{0,1\}^{|J|}$ such that $S_I \otimes S_J \subseteq C$. Now there must be an h such that $S_I^* \subseteq C_I^{(h)}$ and $S_J^* \subseteq C_J^{(h)}$, since otherwise some $x \notin C$ would be assigned nonzero amplitude. By the induction hypothesis,

$$\frac{M(C_I)}{|C_I|} = \min\left(\frac{M(S_I)}{|S_I|}\right), \qquad \frac{M(C_J)}{|C_J|} = \min\left(\frac{M(S_J)}{|S_J|}\right),$$

where the minima are over nonempty $S_I \subseteq C_I^{(h)}$ and $S_J \subseteq C_J^{(h)}$ respectively. Define $\beta = |S_I| \cdot |S_J| / M(S_J)$ and $\gamma = |S_J| \cdot |S_I| / M(S_I)$. Then since setting $S_I := C_I^{(h)}$ and $S_J := C_J^{(h)}$ maximizes the four quantities $|S_I|$, $|S_J|$, $|S_I| / M(S_I)$, and $|S_J| / M(S_J)$ simultaneously, this choice also maximizes β and γ simultaneously. Therefore it maximizes their harmonic mean,

$$\frac{\beta\gamma}{\beta+\gamma} = \frac{|S_I|\,|S_J|}{M(S_I)+M(S_J)} = \frac{|S|}{M(S)}.$$

We have proved that setting $S := C_I^{(h)} \otimes C_J^{(h)}$ maximizes $|S| / M(S)$, or equivalently minimizes $M(S) / |S|$. The one remaining observation is that taking the disjoint sum of $C_I^{(h)} \otimes C_J^{(h)}$ over all $h \in [H]$ leaves the ratio $M(S) / |S|$ unchanged. So setting $S := C$ also minimizes $M(S) / |S|$, and we are done. ■

I can now give a recursive characterization of $M(C)$.

Theorem 94 *If $n \geq 2$, then*

$$M(C) = |C| \min\left(\frac{M(C_I) + M(C_J)}{|C_I|\,|C_J|}\right)$$

where the minimum is over nontrivial partitions (I, J) of $[n]$.

Proof. The upper bound is obvious; let us prove the lower bound. Let T be a manifestly orthogonal tree for $|C\rangle$ of minimum size, and let $v^{(1)}, \ldots, v^{(L)}$ be the topmost $\otimes$ vertices in T. Then there exists a partition $\left(S^{(1)}, \ldots, S^{(L)}\right)$ of C such that the subtree rooted at $v^{(i)}$ represents $\left|S^{(i)}\right\rangle$. We have

$$|T| = M\left(S^{(1)}\right) + \cdots + M\left(S^{(L)}\right) = \left|S^{(1)}\right| \frac{M\left(S^{(1)}\right)}{\left|S^{(1)}\right|} + \cdots + \left|S^{(L)}\right| \frac{M\left(S^{(L)}\right)}{\left|S^{(L)}\right|}.$$

Now let $\eta = \min_i \left(M\left(S^{(i)}\right) / \left|S^{(i)}\right|\right)$. We will construct a partition $\left(R^{(1)}, \ldots, R^{(H)}\right)$ of C such that $M\left(R^{(h)}\right) / \left|R^{(h)}\right| = \eta$ for all $h \in [H]$, which will imply a new tree T' with $|T'| \leq |T|$. Choose $j \in [L]$ such that $M\left(S^{(j)}\right) / \left|S^{(j)}\right| = \eta$, and suppose vertex $v^{(j)}$ of T expresses $\left|S^{(j)}\right\rangle$ as $|S_I\rangle \otimes |S_J\rangle$ for some nontrivial partition (I, J). Then

$$\eta = \frac{M\left(S^{(j)}\right)}{\left|S^{(j)}\right|} = \frac{M(S_I) + M(S_J)}{|S_I|\,|S_J|}$$

where $M\left(S^{(j)}\right) = M(S_I) + M(S_J)$ follows from the minimality of T. As in Lemma 93, there must be an h such that $S_I \subseteq C_I^{(h)}$ and $S_J \subseteq C_J^{(h)}$. But Lemma 93 then implies that

$M(C_I)/|C_I| \leq M(S_I)/|S_I|$ and that $M(C_J)/|C_J| \leq M(S_J)/|S_J|$. Combining these bounds with $|C_I| \geq |S_I|$ and $|C_J| \geq |S_J|$, we obtain by a harmonic mean inequality that

$$\frac{M(C_I \otimes C_J)}{|C_I \otimes C_J|} \leq \frac{M(C_I) + M(C_J)}{|C_I|\,|C_J|} \leq \frac{M(S_I^*) + M(S_J^*)}{|S_I^*|\,|S_J^*|} = \eta.$$

So setting $R^{(h)} := C_I^{(h)} \otimes C_J^{(h)}$ for all $h \in [H]$ yields a new tree T' no larger than T. Hence by the minimality of T,

$$M(C) = |T| = |T'| = H \cdot M(C_I \otimes C_J) = \frac{|C|}{|C_I|\,|C_J|} \cdot (M(C_I) + M(C_J)).$$

■

One can express Theorem 94 directly in terms of the matrix A as follows. Let $M(A) = M(C) = \mathrm{MOTS}(|C\rangle)$ where $C = \{x : Ax \equiv b\}$ (the vector b is irrelevant, so long as $Ax \equiv b$ is solvable). Then

$$M(A) = \min\left(2^{\mathrm{rank}(A_I)+\mathrm{rank}(A_J)-\mathrm{rank}(A)}\,(M(A_I) + M(A_J))\right) \qquad (*)$$

where the minimum is over all nontrivial partitions (A_I, A_J) of the columns of A. As a base case, if A has only one column, then $M(A) = 2$ if $A = 0$ and $M(A) = 1$ otherwise. This immediately implies the following.

Corollary 95 *There exists a deterministic $O(n3^n)$-time algorithm that computes $M(A)$, given A as input.*

Proof. First compute $\mathrm{rank}(A^*)$ for all 2^{n-1} matrices A^* that are formed by choosing a subset of the columns of A. This takes time $O\left(n^3 2^n\right)$. Then compute $M(A^*)$ for all A^* with one column, then for all A^* with two columns, and so on, applying the formula (*) recursively. This takes time

$$\sum_{t=1}^{n} \binom{n}{t} t 2^t = O(n3^n).$$

■

Another easy consequence of Theorem 94 is that the language $\{A : M(A) \leq s\}$ is in NP. I do not know whether this language is NP-complete but suspect it is.

As mentioned above, my characterization makes it possible to prove exponential lower bounds on the manifestly orthogonal tree size of coset states.

Theorem 96 *Suppose the entries of $A \in \mathbb{Z}_2^{k \times n}$ are drawn uniformly and independently at random, where $k \in \left[4\log_2 n, \frac{1}{2}\sqrt{n \ln 2}\right]$. Then $M(A) = \left(n/k^2\right)^{\Omega(k)}$ with probability $\Omega(1)$ over A.*

Proof. Let us upper-bound the probability that certain "bad events" occur when A is drawn. The first bad event is that A contains an all-zero column. This occurs with probability at most $2^{-k}n = o(1)$. The second bad event is that there exists a $k \times d$

submatrix of A with $d \geq 12k$ that has rank at most $2k/3$. This also occurs with probability $o(1)$. For we claim that, if A^* is drawn uniformly at random from $\mathbb{Z}_2^{k \times d}$, then

$$\Pr_{A_I}[\operatorname{rank}(A^*) \leq r] \leq \binom{d}{r}\left(\frac{2^r}{2^k}\right)^{d-r}.$$

To see this, imagine choosing the columns of A^* one by one. For $\operatorname{rank}(A^*)$ to be at most r, there must be at least $d-r$ columns that are linearly dependent on the previous columns. But each column is dependent on the previous ones with probability at most $2^r/2^k$. The claim then follows from the union bound. So the probability that *any* $k \times d$ submatrix of A has rank at most r is at most

$$\binom{n}{d}\binom{d}{r}\left(\frac{2^r}{2^k}\right)^{d-r} \leq n^d d^r \left(\frac{2^r}{2^k}\right)^{d-r}.$$

Set $r = 2k/3$ and $d = 12k$; then the above is at most

$$\exp\left\{12k\log n + \frac{2k}{3}\log(12k) - \left(12k - \frac{2k}{3}\right)\frac{k}{3}\right\} = o(1)$$

where we have used the fact that $k \geq 4\log n$.

Assume that neither bad event occurs, and let $\left(A_I^{(0)}, A_J^{(0)}\right)$ be a partition of the columns of A that minimizes the expression (*). Let $A^{(1)} = A_I^{(0)}$ if $\left|A_I^{(0)}\right| \geq \left|A_J^{(0)}\right|$ and $A^{(1)} = A_J^{(0)}$ otherwise, where $\left|A_I^{(0)}\right|$ and $\left|A_J^{(0)}\right|$ are the numbers of columns in $A_I^{(0)}$ and $A_J^{(0)}$ respectively (so that $\left|A_I^{(0)}\right| + \left|A_J^{(0)}\right| = n$). Likewise, let $\left(A_I^{(1)}, A_J^{(1)}\right)$ be an optimal partition of the columns of $A^{(1)}$, and let $A^{(2)} = A_I^{(1)}$ if $\left|A_I^{(1)}\right| \geq \left|A_J^{(1)}\right|$ and $A^{(2)} = A_J^{(1)}$ otherwise. Continue in this way until an $A^{(t)}$ is reached such that $\left|A^{(t)}\right| = 1$. Then an immediate consequence of (*) is that $M(A) \geq Z^{(0)} \cdot \cdots \cdot Z^{(t-1)}$ where

$$Z^{(\ell)} = 2^{\operatorname{rank}\left(A_I^{(\ell)}\right) + \operatorname{rank}\left(A_J^{(\ell)}\right) - \operatorname{rank}\left(A^{(\ell)}\right)}$$

and $A^{(0)} = A$.

Call ℓ a "balanced cut" if $\min\left\{\left|A_I^{(\ell)}\right|, \left|A_J^{(\ell)}\right|\right\} \geq 12k$, and an "unbalanced cut" otherwise. If ℓ is a balanced cut, then $\operatorname{rank}\left(A_I^{(\ell)}\right) \geq 2k/3$ and $\operatorname{rank}\left(A_J^{(\ell)}\right) \geq 2k/3$, so $Z^{(\ell)} \geq 2^{k/3}$. If ℓ is an unbalanced cut, then call ℓ a "freebie" if $\operatorname{rank}\left(A_I^{(\ell)}\right) + \operatorname{rank}\left(A_J^{(\ell)}\right) = \operatorname{rank}\left(A^{(\ell)}\right)$. There can be at most k freebies, since for each one, $\operatorname{rank}\left(A^{(\ell+1)}\right) < \operatorname{rank}\left(A^{(\ell)}\right)$ by the assumption that all columns of A are nonzero. For the other unbalanced cuts, $Z^{(\ell)} \geq 2$.

Assume $\left|A^{(\ell+1)}\right| = \left|A^{(\ell)}\right|/2$ for each balanced cut and $\left|A^{(\ell+1)}\right| = \left|A^{(\ell)}\right| - 12k$ for each unbalanced cut. Then if the goal is to minimize $Z^{(0)} \cdot \cdots \cdot Z^{(t-1)}$, clearly the best strategy is to perform balanced cuts first, then unbalanced cuts until $\left|A^{(\ell)}\right| = 12k^2$, at which point we can use the k freebies. Let B be the number of balanced cuts; then

$$Z^{(0)} \cdot \cdots \cdot Z^{(t-1)} = \left(2^{k/3}\right)^B 2^{\left(n/2^B - 12k^2\right)/12k}.$$

This is minimized by taking $B = \log_2\left(\frac{n \ln 2}{4k^2}\right)$, in which case $Z^{(0)} \cdot \cdots \cdot Z^{(t-1)} = \left(n/k^2\right)^{\Omega(k)}$. ■

A final application of my characterization is to separate orthogonal from manifestly orthogonal tree size.

Corollary 97 *There exist states with polynomially-bounded orthogonal tree size, but manifestly orthogonal tree size $n^{\Omega(\log n)}$. Thus* OTree $\neq$ MOTree.

Proof. Set $k = 4\log_2 n$, and let $C = \{x : Ax \equiv 0\}$ where A is drawn uniformly at random from $\mathbb{Z}_2^{k \times n}$. Then by Theorem 96,

$$\mathrm{MOTS}\left(|C\rangle\right) = \left(n/k^2\right)^{\Omega(k)} = n^{\Omega(\log n)}$$

with probability $\Omega\left(1\right)$ over A. On the other hand, if we view $|C\rangle$ in the Fourier basis (that is, apply a Hadamard to every qubit), then the resulting state has only $2^k = n^{16}$ basis states with nonzero amplitude, and hence has orthogonal tree size at most n^{17}. So by Proposition 71, part (i), $\mathrm{OTS}\left(|C\rangle\right) \leq 2n^{17}$ as well. ■

Indeed, the orthogonal tree states of Corollary 97 are superpositions over polynomially many separable states, so it also follows that $\Sigma_2 \not\subset$ MOTree.

13.7 Computing With Tree States

Suppose a quantum computer is restricted to being in a tree state at all times. (We can imagine that if the tree size ever exceeds some polynomial bound, the quantum computer explodes, destroying our laboratory.) Does the computer then have an efficient classical simulation? In other words, letting TreeBQP be the class of languages accepted by such a machine, does TreeBQP = BPP? A positive answer would make tree states more attractive as a Sure/Shor separator. For once we admit any states incompatible with the polynomial-time Church-Turing thesis, it seems like we might as well go all the way, and admit *all* states preparable by polynomial-size quantum circuits! The TreeBQP versus BPP problem is closely related to the problem of finding an efficient (classical) algorithm to *learn* multilinear formulas. In light of Raz's lower bound, and of the connection between lower bounds and learning noticed by Linial, Mansour, and Nisan [166], the latter problem might be less hopeless than it looks. In this section I show a weaker result: that TreeBQP is contained in $\Sigma_3^{\mathsf{P}} \cap \Pi_3^{\mathsf{P}}$, the third level of the polynomial hierarchy. Since BQP is not known to lie in PH, this result could be taken as weak evidence that TreeBQP $\neq$ BQP. (On the other hand, we do not yet have oracle evidence even for BQP $\not\subset$ AM, though not for lack of trying [5].)

Definition 98 TreeBQP *is the class of languages accepted by a* BQP *machine subject to the constraint that at every time step t, the machine's state $\left|\psi^{(t)}\right\rangle$ is exponentially close to a tree state. More formally, the initial state is $\left|\psi^{(0)}\right\rangle = |0\rangle^{\otimes(p(n)-n)} \otimes |x\rangle$ (for an input $x \in \{0,1\}^n$ and polynomial bound p), and a uniform classical polynomial-time algorithm generates a sequence of gates $g^{(1)}, \ldots, g^{(p(n))}$. Each $g^{(t)}$ can be either be selected from some finite universal basis of unitary gates (as will be shown in Theorem 99, part (i), the choice of gate set does not matter), or can be a 1-qubit measurement. When we perform a*

measurement, the state evolves to one of two possible pure states, with the usual probabilities, rather than to a mixed state. We require that the final gate $g^{(p(n))}$ is a measurement of the first qubit. If at least one intermediate state $\left|\psi^{(t)}\right\rangle$ had $\mathrm{TS}_{1/2^{\Omega(n)}}\left(\left|\psi^{(t)}\right\rangle\right) > p(n)$, then the outcome of the final measurement is chosen adversarially; otherwise it is given by the usual Born probabilities. The measurement must return 1 with probability at least 2/3 if the input is in the language, and with probability at most 1/3 otherwise.

Some comments on the definition: I allow $\left|\psi^{(t)}\right\rangle$ to deviate from a tree state by an exponentially small amount, in order to make the model independent of the choice of gate set. I allow intermediate measurements because otherwise it is unclear even how to simulate BPP.[8] The rule for measurements follows the "Copenhagen interpretation," in the sense that if a qubit is measured to be 1, then subsequent computation is not affected by what would have happened were the qubit measured to be 0. In particular, if measuring 0 would have led to states of tree size greater than $p(n)$, that does not invalidate the results of the path where 1 is measured.

The following theorem shows that TreeBQP has many of the properties one would want it to have.

Theorem 99

(i) The definition of TreeBQP *is invariant under the choice of gate set.*

(ii) The probabilities $(1/3, 2/3)$ can be replaced by any $(p, 1-p)$ with $2^{-2^{\sqrt{\log n}}} < p < 1/2$.

(iii) $\mathsf{BPP} \subseteq \mathsf{TreeBQP} \subseteq \mathsf{BQP}$.

Proof.

(i) The Solovay-Kitaev Theorem [153, 182] shows that given a universal gate set, one can approximate any k-qubit unitary to accuracy $1/\varepsilon$ using k qubits and a circuit of size $O(\mathrm{polylog}(1/\varepsilon))$. So let $\left|\psi^{(0)}\right\rangle, \ldots, \left|\psi^{(p(n))}\right\rangle \in \mathcal{H}_2^{\otimes p(n)}$ be a sequence of states, with $\left|\psi^{(t)}\right\rangle$ produced from $\left|\psi^{(t-1)}\right\rangle$ by applying a k-qubit unitary $g^{(t)}$ (where $k = O(1)$). Then using a polynomial-size circuit, one can approximate each $\left|\psi^{(t)}\right\rangle$ to accuracy $1/2^{\Omega(n)}$, as in the definition of TreeBQP. Furthermore, since the approximation circuit for $g^{(t)}$ acts only on k qubits, any intermediate state $|\varphi\rangle$ it produces satisfies $\mathrm{TS}_{1/2^{\Omega(n)}}(|\varphi\rangle) \leq k4^k \mathrm{TS}_{1/2^{\Omega(n)}}\left(\left|\psi^{(t-1)}\right\rangle\right)$ by Proposition 71.

(ii) To amplify to a constant probability, run k copies of the computation in tensor product, then output the majority answer. By part (i), outputting the majority can increase the tree size by a factor of at most 2^{k+1}. To amplify to $2^{-2^{\sqrt{\log n}}}$, observe that the Boolean majority function on k bits has a multilinear formula of size $k^{O(\log k)}$. For let $T_k^h(x_1, \ldots, x_k)$ equal 1 if $x_1 + \cdots + x_k \geq h$ and 0 otherwise; then

$$T_k^h(x_1, \ldots, x_k) = 1 - \prod_{i=0}^{h} \left(1 - T_{\lfloor k/2 \rfloor}^i\left(x_1, \ldots, x_{\lfloor k/2 \rfloor}\right) T_{\lceil k/2 \rceil}^{h-i}\left(x_{\lfloor k/2 \rfloor + 1}, \ldots, x_k\right)\right),$$

[8] If we try to simulate BPP in the standard way, we might produce complicated entanglement between the computation register and the register containing the random bits, and no longer have a tree state.

so $\mathrm{MFS}\left(T_k^h\right) \leq 2h \max_i \mathrm{MFS}\left(T_{\lceil k/2 \rceil}^h\right) + O(1)$, and solving this recurrence yields $\mathrm{MFS}\left(T_k^{k/2}\right) = k^{O(\log k)}$. Substituting $k = 2^{\sqrt{\log n}}$ into $k^{O(\log k)}$ yields $n^{O(1)}$, meaning the tree size increases by at most a polynomial factor.

(iii) To simulate BPP, just perform a classical reversible computation, applying a Hadamard followed by a measurement to some qubit whenever we need a random bit. Since the number of basis states with nonzero amplitude is at most 2, the simulation is clearly in $\mathsf{TreeBQP}$. The other containment is obvious.

■

Theorem 100 $\mathsf{TreeBQP} \subseteq \Sigma_3^{\mathsf{P}} \cap \Pi_3^{\mathsf{P}}$.

Proof. Since $\mathsf{TreeBQP}$ is closed under complement, it suffices to show that $\mathsf{TreeBQP} \subseteq \Pi_3^{\mathsf{P}}$. Our proof will combine approximate counting with a predicate to verify the correctness of a $\mathsf{TreeBQP}$ computation. Let C be a uniformly-generated quantum circuit, and let $M = \left(m^{(1)}, \ldots, m^{(p(n))}\right)$ be a sequence of binary measurement outcomes. We adopt the convention that after making a measurement, the state vector is *not* rescaled to have norm 1. That way the probabilities across all 'measurement branches' continue to sum to 1. Let $\left|\psi_{M,x}^{(0)}\right\rangle, \ldots, \left|\psi_{M,x}^{(p(n))}\right\rangle$ be the sequence of unnormalized pure states under measurement outcome sequence M and input x, where $\left|\psi_{M,x}^{(t)}\right\rangle = \sum_{y \in \{0,1\}^{p(n)}} \alpha_{y,M,x}^{(t)} |y\rangle$. Also, let $\Lambda(M,x)$ express that $\mathrm{TS}_{1/2^{\Omega(n)}}\left(\left|\psi_{M,x}^{(t)}\right\rangle\right) \leq p(n)$ for every t. Then C accepts if

$$W_x = \sum_{M\,:\,\Lambda(M,x)} \sum_{y \in \{0,1\}^{p(n)-1}} \left|\alpha_{1y,M,x}^{(p(n))}\right|^2 \geq \frac{2}{3},$$

while C rejects if $W_x \leq 1/3$. If we could compute each $\left|\alpha_{1y,M,x}^{(p(n))}\right|$ efficiently (as well as $\Lambda(M,x)$), we would then have a Π_2^{P} predicate expressing that $W_x \geq 2/3$. This follows since we can do approximate counting via hashing in $\mathsf{AM} \subseteq \Pi_2^{\mathsf{P}}$ [131], and thereby verify that an exponentially large sum of nonnegative terms is at least $2/3$, rather than at most $1/3$. The one further fact we need is that in our Π_2^{P} $(\forall\exists)$ predicate, we can take the existential quantifier to range over tuples of 'candidate solutions'—that is, (M, y) pairs together with lower bounds β on $\left|\alpha_{1y,M,x}^{(p(n))}\right|$.

It remains only to show how we verify that $\Lambda(M,x)$ holds and that $\left|\alpha_{1y,M,x}^{(p(n))}\right| = \beta$. First, we extend the existential quantifier so that it guesses not only M and y, but also a sequence of trees $T^{(0)}, \ldots, T^{(p(n))}$, representing $\left|\psi_{M,x}^{(0)}\right\rangle, \ldots, \left|\psi_{M,x}^{(p(n))}\right\rangle$ respectively. Second, using the last universal quantifier to range over $\widehat{y} \in \{0,1\}^{p(n)}$, we verify the following:

(1) $T^{(0)}$ is a fixed tree representing $|0\rangle^{\otimes(p(n)-n)} \otimes |x\rangle$.

(2) $\left|\alpha_{1y,M,x}^{(p(n))}\right|$ equals its claimed value to $\Omega(n)$ bits of precision.

(3) Let $g^{(1)},\ldots,g^{(p(n))}$ be the gates applied by C. Then for all t and $\widehat{y}$, if $g^{(t)}$ is unitary then $\alpha_{\widehat{y},M,x}^{(t)} = \langle\widehat{y}|\cdot g^{(t)}\left|\psi_{M,x}^{(t-1)}\right\rangle$ to $\Omega(n)$ bits of precision. Here the right-hand side is a sum of 2^k terms (k being the number of qubits acted on by $g^{(t)}$), each term efficiently computable given $T^{(t-1)}$. Similarly, if $g^{(t)}$ is a measurement of the i^{th} qubit, then $\alpha_{\widehat{y},M,x}^{(t)} = \alpha_{\widehat{y},M,x}^{(t-1)}$ if the i^{th} bit of $\widehat{y}$ equals $m^{(t)}$, while $\alpha_{\widehat{y},M,x}^{(t)} = 0$ otherwise.

■

In the proof of Theorem 100, the only fact about tree states I needed was that $\mathsf{Tree} \subseteq \mathsf{AmpP}$; that is, there is a polynomial-time classical algorithm that computes the amplitude α_x of any basis state $|x\rangle$. So if we define $\mathsf{AmpP\text{-}BQP}$ analogously to $\mathsf{TreeBQP}$ except that any states in AmpP are allowed, then $\mathsf{AmpP\text{-}BQP} \subseteq \mathsf{\Sigma_3^P} \cap \mathsf{\Pi_3^P}$ as well.

13.8 The Experimental Situation

The results of this chapter suggest an obvious challenge for experimenters: *prepare non-tree states in the lab.* For were this challenge met, it would rule out one way in which quantum mechanics could fail, just as the Bell inequality experiments of Aspect et al. [37] did twenty years ago. If they wished, quantum computing skeptics could then propose a new candidate Sure/Shor separator, and experimenters could try to rule out *that* one, and so on. The result would be to divide the question of whether quantum computing is possible into a series of smaller questions about which states can be prepared. In my view, this would aid progress in two ways: by helping experimenters set clear goals, and by forcing theorists to state clear conjectures.

However, my experimental challenge raises some immediate questions. In particular, what would it *mean* to prepare a non-tree state? How would we know if we succeeded? Also, have non-tree states already been prepared (or observed)? The purpose of this section is to set out my thoughts about these questions.

First of all, when discussing experiments, it goes without saying that we must convert asymptotic statements into statements about specific values of n. The central tenet of computational complexity theory is that this is possible. Thus, instead of asking whether n-qubit states with tree size $2^{\Omega(n)}$ can be prepared, we ask whether 200-qubit states with tree size at least (say) 2^{80} can be prepared. Even though the second question does not logically imply anything about the first, the second is closer to what we ultimately care about anyway. Admittedly, knowing that $\mathrm{TS}(|\psi_n\rangle) = n^{\Omega(\log n)}$ tells us little about $\mathrm{TS}(|\psi_{100}\rangle)$ or $\mathrm{TS}(|\psi_{200}\rangle)$, especially since in Raz's paper [195], the constant in the exponent $\Omega(\log n)$ is taken to be 10^{-6} (though this can certainly be improved). Thus, proving tight lower bounds for small n is one of the most important problems left open by this chapter. In Section 13.6 I show how to solve this problem for the case of manifestly orthogonal tree size.

A second objection is that my formalism applies only to pure states, but in reality all states are mixed. However, there are several natural ways to extend the formalism to mixed states. Given a mixed state ρ, we could minimize tree size over all purifications of ρ, or minimize the expected tree size $\sum_i |\alpha_i|^2 \mathrm{TS}(|\psi_i\rangle)$, or maximum $\max_i \mathrm{TS}(|\psi_i\rangle)$, over all decompositions $\rho = \sum_i \alpha_i |\psi_i\rangle\langle\psi_i|$.

A third objection is a real quantum state might be a "soup" of free-wandering fermions and bosons, with no localized subsystems corresponding to qubits. How can one determine the tree size of such a state? The answer is that one cannot. Any complexity measure for particle position and momentum states would have to be quite different from the measures considered in this chapter. On the other hand, the states of interest for quantum computing usually *do* involve localized qubits. Indeed, even if quantum information is stored in particle positions, one might force each particle into two sites (corresponding to $|0\rangle$ and $|1\rangle$), neither of which can be occupied by any other particle. In that case it again becomes meaningful to discuss tree size.

But how do we verify that a state with large tree size was prepared? Of course, if $|\psi\rangle$ is preparable by a polynomial-size quantum circuit, then *assuming quantum mechanics is valid* (and assuming our gates behave as specified), we can always test whether a given state $|\varphi\rangle$ is close to $|\psi\rangle$ or not. Let U map $|0\rangle^{\otimes n}$ to $|\psi\rangle$; then it suffices to test whether $U^{-1}|\varphi\rangle$ is close to $|0\rangle^{\otimes n}$. However, in the experiments under discussion, the validity of quantum mechanics is the very point in question. And once we allow Nature to behave in arbitrary ways, a skeptic could explain *any* experimental result without having to invoke states with large tree size.

The above fact has often been urged against me, but as it stands, it is no different from the fact that one could explain any astronomical observation without abandoning the Ptolemaic system. The issue here is not one of proof, but of accumulating observations that are consistent with the hypothesis of large tree size, and inconsistent with alternative hypotheses if we disallow special pleading. So for example, to test whether the subgroup state

$$|S\rangle = \frac{1}{\sqrt{|S|}} \sum_{x \in S} |x\rangle$$

was prepared, we might use CNOT gates to map $|x\rangle$ to $|x\rangle \left|v^T x\right\rangle$ for some vector $v \in \mathbb{Z}_2^n$. Based on our knowledge of S, we could then predict whether the qubit $\left|v^T x\right\rangle$ should be $|0\rangle$, $|1\rangle$, or an equal mixture of $|0\rangle$ and $|1\rangle$ when measured. Or we could apply Hadamard gates to all n qubits of $|S\rangle$, then perform the same test for the subgroup dual to S. In saying that a system is in state $|S\rangle$, it is not clear if we *mean* anything more than that it responds to all such tests in expected ways. Similar remarks apply to Shor states and cluster states.

In my view, tests of the sort described above are certainly *sufficient*, so the interesting question is whether they are *necessary*, or whether weaker and more indirect tests would also suffice. This question rears its head when we ask whether non-tree states have already been observed. For as pointed out to me by Anthony Leggett, there exist systems studied in condensed-matter physics that are strong candidates for having superpolynomial tree size. An example is the magnetic salt $\mathrm{LiHo}_x\mathrm{Y}_{1-x}\mathrm{F}_4$ studied by Ghosh et al. [124], which, like the cluster states of Briegel and Raussendorf [71], basically consists of a lattice of spins subject to pairwise nearest-neighbor Hamiltonians. The main differences are that the salt lattice is 3-D instead of 2-D, is tetragonal instead of cubic, and is irregular in that not every site is occupied by a spin. Also, there are weak interactions even between spins that are not nearest neighbors. But none of these differences seem likely to change a superpolynomial tree size into a polynomial one.

For me, the main issues are (1) how precisely can we characterize[9] the quantum state of the magnetic salt, and (2) how strong the evidence is that that *is* the state. What Ghosh et al. [124] did was to calculate bulk properties of the salt, such as its magnetic susceptibility and specific heat, with and without taking into account the quantum entanglement generated by the nearest-neighbor Hamiltonians. They found that including entanglement yielded a better fit to the experimentally measured values. However, this is clearly a far cry from preparing a system in a state of one's choosing by applying a known pulse sequence, and then applying any of a vast catalog of tests to verify that the state was prepared. So it would be valuable to have more direct evidence that states qualitatively like cluster states can exist in Nature.

In summary, the ideas of this chapter underscore the importance of current experimental work on large, persistently entangled quantum states; but they also suggest a new motivation and perspective for this work. They suggest that we reexamine known condensed-matter systems with a new goal in mind: understanding the complexity of their associated quantum states. They also suggest that 2-D cluster states and random subgroup states are interesting in a way that 1-D spin chains and Schrödinger cat states are not. Yet when experimenters try to prepare states of the former type, they often see it as merely a stepping stone towards demonstrating error-correction or another quantum computing benchmark. Thus, Knill et al. [158] prepared[10] the 5-qubit state

$$|\psi\rangle = \frac{1}{4}\begin{pmatrix} |00000\rangle + |10010\rangle + |01001\rangle + |10100\rangle \\ +|01010\rangle - |11011\rangle - |00110\rangle - |11000\rangle \\ -|11101\rangle - |00011\rangle - |11110\rangle - |01111\rangle \\ -|10001\rangle - |01100\rangle - |10111\rangle + |00101\rangle \end{pmatrix},$$

for which $\mathrm{MOTS}\left(|\psi\rangle\right) = 40$ from the decomposition

$$|\psi\rangle = \frac{1}{4}\begin{pmatrix} (|01\rangle + |10\rangle) \otimes (|010\rangle - |111\rangle) + (|01\rangle - |10\rangle) \otimes (|001\rangle - |100\rangle) \\ -(|00\rangle + |11\rangle) \otimes (|011\rangle + |110\rangle) + (|00\rangle - |11\rangle) \otimes (|000\rangle + |101\rangle) \end{pmatrix},$$

and for which I conjecture $\mathrm{TS}\left(|\psi\rangle\right) = 40$ as well. However, the sole motivation of the experiment was to demonstrate a 5-qubit quantum error-correcting code. In my opinion, whether states with large tree size can be prepared is a fundamental question in its own right. Were that question studied directly, perhaps we could address it for larger numbers of qubits.

Let me end by stressing that, in the perspective I am advocating, there is nothing sacrosanct about tree size as opposed to other complexity measures. This chapter concentrated on tree size because it is the subject of our main results, and because it is better to

[9] By "characterize," I mean give an explicit formula for the amplitudes at a particular time t, in some standard basis. If a state is characterized as the ground state of a Hamiltonian, then we first need to solve for the amplitudes before we can prove tree size lower bounds using Raz's method.

[10] Admittedly, what they really prepared is the 'pseudo-pure' state $\rho = \varepsilon\,|\psi\rangle\,\langle\psi| + (1-\varepsilon)\,I$, where I is the maximally mixed state and $\varepsilon \approx 10^{-5}$. Braunstein et al. [69] have shown that, if the number of qubits n is less than about 14, then such states cannot be entangled. That is, there exists a representation of ρ as a mixture of pure states, each of which is separable and therefore has tree size $O\left(n\right)$. This is a well-known limitation of the liquid NMR technology used by Knill et al. Thus, a key challenge is to replicate the successes of liquid NMR using colder qubits.

be specific than vague. On the other hand, Sections 13.3, 13.4, and 13.6 contain numerous results about orthogonal tree size, manifestly orthogonal tree size, Vidal's χ complexity, and other measures. Readers dissatisfied with *all* of these measures are urged to propose new ones, perhaps motivated directly by experiments. I see nothing wrong with having multiple ways to quantify the complexity of quantum states, and much wrong with having no ways.

13.9 Conclusion and Open Problems

A crucial step in quantum computing was to separate the question of whether quantum computers can be built from the question of what one could do with them. This separation allowed computer scientists to make great advances on the latter question, despite knowing nothing about the former. I have argued, however, that the tools of computational complexity theory are relevant to both questions. The claim that large-scale quantum computing is possible in principle is really a claim that certain *states* can exist—that quantum mechanics will not break down if we try to prepare those states. Furthermore, what distinguishes these states from states we have seen must be more than precision in amplitudes, or the number of qubits maintained coherently. The distinguishing property should instead be some sort of *complexity*. That is, Sure states should have succinct representations of a type that Shor states do not.

I have tried to show that, by adopting this viewpoint, we make the debate about whether quantum computing is possible less ideological and more scientific. By studying particular examples of Sure/Shor separators, quantum computing skeptics would strengthen their case—for they would then have a plausible research program aimed at identifying what, exactly, the barriers to quantum computation are. I hope, however, that the 'complexity theory of quantum states' initiated here will be taken up by quantum computing proponents as well. This theory offers a new perspective on the transition from classical to quantum computing, and a new connection between quantum computing and the powerful circuit lower bound techniques of classical complexity theory.

I end with some open problems.

(1) Can Raz's technique be improved to show exponential tree size lower bounds?

(2) Can we prove Conjecture 90, implying an $n^{\Omega(\log n)}$ tree size lower bound for Shor states?

(3) Let $|\varphi\rangle$ be a uniform superposition over all n-bit strings of Hamming weight $n/2$. It is easy to show by divide-and-conquer that $\mathrm{TS}(|\varphi\rangle) = n^{O(\log n)}$. Is this upper bound tight? More generally, can we show a superpolynomial tree size lower bound for any state with permutation symmetry?

(4) Is $\mathsf{Tree} = \mathsf{OTree}$? That is, are there tree states that are not orthogonal tree states?

(5) Is the tensor-sum hierarchy of Section 13.2 infinite? That is, do we have $\Sigma_k \neq \Sigma_{k+1}$ for all k?

(6) Is $\mathsf{TreeBQP} = \mathsf{BPP}$? That is, can a quantum computer that is always in a tree state be simulated classically? The key question seems to be whether the concept class of multilinear formulas is efficiently learnable.

(7) Is there a practical method to compute the tree size of, say, 10-qubit states? Such a method would have great value in interpreting experimental results.

Chapter 14

Quantum Search of Spatial Regions

This chapter represents joint work with Andris Ambainis.

The goal of Grover's quantum search algorithm [139] is to search an 'unsorted database' of size n in a number of queries proportional to $\sqrt{n}$. Classically, of course, order n queries are needed. It is sometimes asserted that, although the speedup of Grover's algorithm is only quadratic, this speedup is *provable*, in contrast to the exponential speedup of Shor's factoring algorithm [219]. But is that really true? Grover's algorithm is typically imagined as speeding up combinatorial search—and we do not know whether every problem in NP can be classically solved quadratically faster than the "obvious" way, any more than we know whether factoring is in BPP.

But could Grover's algorithm speed up search of a *physical region*? Here the basic problem, it seems to us, is the time needed for signals to travel across the region. For if we are interested in the fundamental limits imposed by physics, then we should acknowledge that the speed of light is finite, and that a bounded region of space can store only a finite amount of information, according to the holographic principle [65]. We discuss the latter constraint in detail in Section 14.3; for now, we say only that it suggests a model in which a 'quantum robot' occupies a superposition over finitely many locations, and moving the robot from one location to an adjacent one takes unit time. In such a model, the time needed to search a region could depend critically on its spatial layout. For example, if the n entries are arranged on a line, then even to move the robot from one end to the other takes $n-1$ steps. But what if the entries are arranged on, say, a 2-dimensional square grid (Figure 14.1)?

14.1 Summary of Results

This chapter gives the first systematic treatment of quantum search of spatial regions, with 'regions' modeled as connected graphs. Our main result is positive: we show that a quantum robot can search a d-dimensional hypercube with n vertices for a unique marked vertex in time $O\left(\sqrt{n}\log^{3/2} n\right)$ when $d=2$, or $O\left(\sqrt{n}\right)$ when $d \geq 3$. This matches (or in the case of 2 dimensions, nearly matches) the $\Omega\left(\sqrt{n}\right)$ lower bound for quantum search, and supports the view that Grover search of a physical region presents no problem of principle.

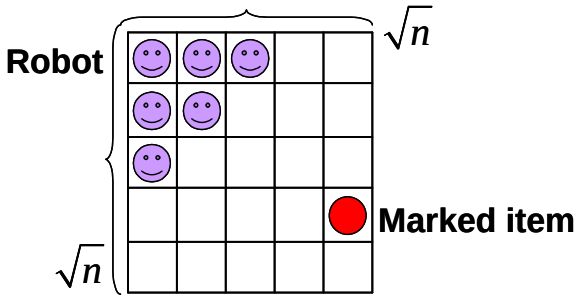

Figure 14.1: A quantum robot, in a superposition over locations, searching for a marked item on a 2D grid of size $\sqrt{n} \times \sqrt{n}$.

	$d = 2$	$d > 2$
Hypercube, 1 marked item	$O\left(\sqrt{n}\log^{3/2} n\right)$	$\Theta\left(\sqrt{n}\right)$
Hypercube, k or more marked items	$O\left(\sqrt{n}\log^{5/2} n\right)$	$\Theta\left(\frac{\sqrt{n}}{k^{1/2-1/d}}\right)$
Arbitrary graph, k or more marked items	$\sqrt{n}2^{O\left(\sqrt{\log n}\right)}$	$\widetilde{\Theta}\left(\frac{\sqrt{n}}{k^{1/2-1/d}}\right)$

Table 14.1: Upper and lower bounds for quantum search on a d-dimensional graph given in this chapter. The symbol $\widetilde{\Theta}$ means that the upper bound includes a polylogarithmic term. Note that, if $d = 2$, then $\Omega\left(\sqrt{n}\right)$ is always a lower bound, for any number of marked items.

Our basic technique is divide-and-conquer; indeed, once the idea is pointed out, an upper bound of $O\left(n^{1/2+\varepsilon}\right)$ follows readily. However, to obtain the tighter bounds is more difficult; for that we use the amplitude-amplification framework of Brassard et al. [67].

Section 14.6 presents the main results; Section 14.6.4 shows further that, when there are k or more marked vertices, the search time becomes $O\left(\sqrt{n}\log^{5/2} n\right)$ when $d = 2$, or $\Theta\left(\sqrt{n}/k^{1/2-1/d}\right)$ when $d \geq 3$. Also, Section 14.7 generalizes our algorithm to arbitrary graphs that have 'hypercube-like' expansion properties. Here the best bounds we can achieve are $\sqrt{n}2^{O\left(\sqrt{\log n}\right)}$ when $d = 2$, or $O\left(\sqrt{n}\,\mathrm{polylog}\, n\right)$ when $d > 2$ (note that d need not be an integer). Table 14.1 summarizes the results.

Section 14.8 shows, as an unexpected application of our search algorithm, that the quantum communication complexity of the well-known *disjointness problem* is $O\left(\sqrt{n}\right)$. This improves an $O\left(\sqrt{n}c^{\log^* n}\right)$ upper bound of Høyer and de Wolf [146], and matches the $\Omega\left(\sqrt{n}\right)$ lower bound of Razborov [199].

The rest of the chapter is about the formal model that underlies our results. Section 14.3 sets the stage for this model, by exploring the ultimate limits on information storage imposed by properties of space and time. This discussion serves only to motivate our results; thus, it can be safely skipped by readers unconcerned with the physical universe. In Section 16.7 we define *quantum query algorithms on graphs*, a model similar to quantum query algorithms as defined in Section 5.1, but with the added requirement that unitary operations be 'local' with respect to some graph. In Section 14.4.1 we address the difficult

question, which also arises in work on quantum random walks [19] and quantum cellular automata [238], of what 'local' means. Section 14.5 proves general facts about our model, including an upper bound of $O\left(\sqrt{n\delta}\right)$ for the time needed to search any graph with diameter δ, and a proof (using the hybrid argument of Bennett et al. [51]) that this upper bound is tight for certain graphs. We conclude in Section 14.9 with some open problems.

14.2 Related Work

In a paper on 'Space searches with a quantum robot,' Benioff [50] asked whether Grover's algorithm can speed up search of a physical region, as opposed to a combinatorial search space. His answer was discouraging: for a 2-D grid of size $\sqrt{n} \times \sqrt{n}$, Grover's algorithm is no faster than classical search. The reason is that, during each of the $\Theta\left(\sqrt{n}\right)$ Grover iterations, the algorithm must use order $\sqrt{n}$ steps just to travel across the grid and return to its starting point for the diffusion step. On the other hand, Benioff noted, Grover's algorithm does yield some speedup for grids of dimension 3 or higher, since those grids have diameter less than $\sqrt{n}$.

Our results show that Benioff's claim is mistaken: by using Grover's algorithm more carefully, one can search a 2-D grid for a single marked vertex in $O\left(\sqrt{n}\log^{3/2} n\right)$ time. To us this illustrates why one should not assume an algorithm is optimal on heuristic grounds. Painful experience—for example, the "obviously optimal" $O\left(n^3\right)$ matrix multiplication algorithm [226]—is what taught computer scientists to see the proving of lower bounds as more than a formality.

Our setting is related to that of quantum random walks on graphs [19, 83, 84, 216]. In an earlier version of this chapter, we asked whether quantum walks might yield an alternative spatial search algorithm, possibly even one that outperforms our divide-and-conquer algorithm. Motivated by this question, Childs and Goldstone [86] managed to show that in the continuous-time setting, a quantum walk can search a d-dimensional hypercube for a single marked vertex in time $O\left(\sqrt{n}\log n\right)$ when $d = 4$, or $O\left(\sqrt{n}\right)$ when $d \geq 5$. Our algorithm was still faster in 3 or fewer dimensions (see Table 14.2). Subsequently, however, Ambainis, Kempe, and Rivosh [31] gave an algorithm based on a discrete-time quantum walk, which was as fast as ours in 3 or more dimensions, and faster in 2 dimensions. In particular, when $d = 2$ their algorithm used only $O\left(\sqrt{n}\log n\right)$ time to find a unique marked vertex. Childs and Goldstone [85] then gave a continuous-time quantum walk algorithm with the same performance, and related this algorithm to properties of the Dirac equation. It is still open whether $O\left(\sqrt{n}\right)$ time is achievable in 2 dimensions.

Currently, the main drawback of the quantum walk approach is that all analyses have relied heavily on symmetries in the underlying graph. If even minor 'defects' are introduced, it is no longer known how to upper-bound the running time. By contrast, the analysis of our divide-and-conquer algorithm is elementary, and does not depend on eigenvalue bounds. We can therefore show that the algorithm works for any graphs with sufficiently good expansion properties.

Childs and Goldstone [86] argued that the quantum walk approach has the advantage of requiring fewer auxiliary qubits than the divide-and-conquer approach. However,

	$d=2$	$d=3$	$d=4$	$d \geq 5$
This chapter	$O\left(\sqrt{n}\log^{3/2} n\right)$	$O\left(\sqrt{n}\right)$	$O\left(\sqrt{n}\right)$	$O\left(\sqrt{n}\right)$
[86]	$O\left(n\right)$	$O\left(n^{5/6}\right)$	$O\left(\sqrt{n}\log n\right)$	$O\left(\sqrt{n}\right)$
[31, 85]	$O\left(\sqrt{n}\log n\right)$	$O\left(\sqrt{n}\right)$	$O\left(\sqrt{n}\right)$	$O\left(\sqrt{n}\right)$

Table 14.2: Time needed to find a unique marked item in a d-dimensional hypercube, using the divide-and-conquer algorithms of this chapter, the original quantum walk algorithm of Childs and Goldstone [86], and the improved walk algorithms of Ambainis, Kempe, and Rivosh [31] and Childs and Goldstone [85].

the need for many qubits was an artifact of how we implemented the algorithm in a previous version of the chapter. The current version uses only *one* qubit.

14.3 The Physics of Databases

Theoretical computer science generally deals with the limit as some resource (such as time or memory) increases to infinity. What is not always appreciated is that, as the resource bound increases, physical constraints may come into play that were negligible at 'sub-asymptotic' scales. We believe theoretical computer scientists ought to know something about such constraints, and to account for them when possible. For if the constraints are ignored on the ground that they "never matter in practice," then the obvious question arises: why use asymptotic analysis in the first place, rather than restricting attention to those instance sizes that occur in practice?

A constraint of particular interest for us is the *holographic principle* [65], which arose from black-hole thermodynamics. The principle states that the information content of any spatial region is upper-bounded by its *surface area* (not volume), at a rate of one bit per Planck area, or about 1.4×10^{69} bits per square meter. Intuitively, if one tried to build a spherical hard disk with mass density υ, one could not keep expanding it forever. For as soon as the radius reached the Schwarzschild bound of $r = \sqrt{3/\left(8\pi\upsilon\right)}$ (in Planck units, $c = G = \hbar = k = 1$), the hard disk would collapse to form a black hole, and thus its contents would be irretrievable.

Actually the situation is worse than that: even a *planar* hard disk of constant mass density would collapse to form a black hole once its radius became sufficiently large, $r = \Theta\left(1/\upsilon\right)$. (We assume here that the hard disk is disc-shaped. A linear or 1-D hard disk could expand indefinitely without collapse.) It is possible, though, that a hard disk's information content could asymptotically exceed its mass. For example, a black hole's mass is proportional to the radius of its event horizon, but the entropy is proportional to the *square* of the radius (that is, to the surface area). Admittedly, inherent difficulties with storage and retrieval make a black hole horizon less than ideal as a hard disk. However, even a weakly-gravitating system could store information at a rate asymptotically exceeding its mass-energy. For instance, Bousso [65] shows that an enclosed ball of radiation with radius r can store $n = \Theta\left(r^{3/2}\right)$ bits, even though its energy grows only as r. Our results in Section 14.7.1 will imply that a quantum robot could (in principle!) search such a 'radiation

disk' for a marked item in time $O\left(r^{5/4}\right) = O\left(n^{5/6}\right)$. This is some improvement over the trivial $O(n)$ upper bound for a 1-D hard disk, though it falls short of the desired $O(\sqrt{n})$.

In general, if $n = r^c$ bits are scattered throughout a 3-D ball of radius r (where $c \leq 3$ and the bits' locations are known), we will show in Theorem 130 that the time needed to search for a '1' bit grows as $n^{1/c+1/6} = r^{1+c/6}$ (omitting logarithmic factors). In particular, if $n = \Theta\left(r^2\right)$ (saturating the holographic bound), then the time grows as $n^{2/3}$ or $r^{4/3}$. To achieve a search time of $O(\sqrt{n}\operatorname{polylog} n)$, the bits would need to be concentrated on a 2-D surface.

Because of the holographic principle, we see that it is not only quantum mechanics that yields a $\Omega(\sqrt{n})$ lower bound on the number of steps needed for unordered search. If the items to be searched are laid out spatially, then general relativity in $3+1$ dimensions independently yields the same bound, $\Omega(\sqrt{n})$, up to a constant factor.[1] Interestingly, in $d+1$ dimensions the relativity bound would be $\Omega\left(n^{1/(d-1)}\right)$, which for $d > 3$ is weaker than the quantum mechanics bound. Given that our two fundamental theories yield the same lower bound, it is natural to ask whether that bound is tight. The answer seems to be that it is *not* tight, since (i) the entropy on a black hole horizon is not efficiently accessible[2], and (ii) weakly-gravitating systems are subject to the *Bekenstein bound* [48], an even stronger entropy constraint than the holographic bound.

Yet it is still of basic interest to know whether n bits in a radius-r ball can be searched in time $o(\min\{n, r\sqrt{n}\})$—that is, whether it is possible to do *anything* better than either brute-force quantum search (with the drawback pointed out by Benioff [50]), or classical search. Our results show that it is possible.

From a physical point of view, several questions naturally arise: (1) whether our complexity measure is realistic; (2) how to account for time dilation; and (3) whether given the number of bits we are imagining, cosmological bounds are also relevant. Let us address these questions in turn.

(1) One could argue that to maintain a 'quantum database' of size n requires n computing elements ([251], though see also [206]). So why not just exploit those elements to search the database in *parallel*? Then it becomes trivial to show that the search time is limited only by the radius of the database, so the algorithms of this chapter are unnecessary. Our response is that, while there might be n 'passive' computing elements (capable of storing data), there might be many fewer 'active' elements, which we consequently wish to place in a superposition over locations. This assumption seems physically unobjectionable. For a particle (and indeed any object) really does have an indeterminate location, not merely an indeterminate internal state (such as spin) *at* some location. We leave as an open problem, however, whether our assumption is valid for specific quantum computer architectures such as ion traps.

(2) So long as we invoke general relativity, should we not also consider the effects of time dilation? Those effects are indeed pronounced near a black hole horizon. Again, though, for our upper bounds we will have in mind systems far from the Schwarzschild

[1] Admittedly, the holographic principle is part of quantum gravity and not general relativity *per se*. All that matters for us, though, is that the principle seems logically independent of quantum-mechanical linearity, which is what produces the "other" $\Omega(\sqrt{n})$ bound.

[2] In the case of a black hole horizon, waiting for the bits to be emitted as Hawking radiation—as recent evidence suggests that they are [209]—takes time proportional to r^3, which is much too long.

limit, for which any time dilation is by at most a constant factor independent of n.

(3) How do cosmological considerations affect our analysis? Bousso [64] argues that, in a spacetime with positive cosmological constant $\Lambda > 0$, the total number of bits accessible to any one experiment is at most $3\pi/(\Lambda \ln 2)$, or roughly 10^{122} given current experimental bounds [208] on Λ.[3] Intuitively, even if the universe is spatially infinite, most of it recedes too quickly from any one observer to be harnessed as computer memory.

One response to this result is to assume an idealization in which Λ vanishes, although Planck's constant $\hbar$ does not vanish. As justification, one could argue that without the idealization $\Lambda = 0$, *all* asymptotic bounds in computer science are basically fictions. But perhaps a better response is to accept the $3\pi/(\Lambda \ln 2)$ bound, and then ask how close one can come to *saturating* it in different scenarios. Classically, the maximum number of bits that can be searched is, in a crude model[4], actually proportional to $1/\sqrt{\Lambda} \approx 10^{61}$ rather than $1/\Lambda$. The reason is that if a region had much more than $1/\sqrt{\Lambda}$ bits, then after $1/\sqrt{\Lambda}$ Planck times—that is, about 10^{10} years, or roughly the current age of the universe—most of the region would have receded beyond one's cosmological horizon. What our results suggest is that, using a quantum robot, one could come closer to saturating the cosmological bound—since, for example, a 2-D region of size $1/\Lambda$ can be searched in time $O\left(\frac{1}{\sqrt{\Lambda}} \operatorname{polylog} \frac{1}{\sqrt{\Lambda}}\right)$. How anyone could *prepare* (say) a database of size much greater than $1/\sqrt{\Lambda}$ remains unclear, but if such a database existed, it could be searched!

14.4 The Model

As discussed in Part I, much of what is known about the power of quantum computing comes from the *black-box* or *query* model—in which one counts only the number of queries to an oracle, not the number of computational steps. We will take this model as the starting point for a formal definition of quantum robots. Doing so will focus attention on our main concern: how much harder is it to evaluate a function when its inputs are spatially separated? As it turns out, all of our algorithms *will* be efficient as measured by the number of gates and auxiliary qubits needed to implement them.

For simplicity, we assume that a robot's goal is to evaluate a Boolean function $f : \{0,1\}^n \rightarrow \{0,1\}$, which could be partial or total. A 'region of space' is a connected undirected graph $G = (V, E)$ with vertices $V = \{v_1, \ldots, v_n\}$. Let $X = x_1 \ldots x_n \in \{0,1\}^n$ be an input to f; then each bit x_i is available only at vertex v_i. We assume the robot knows G and the vertex labels in advance, and so is ignorant only of the x_i bits. We thus sidestep a major difficulty for quantum walks [19], which is how to ensure that a process on an unknown graph is unitary.

[3] Also, Lloyd [170] argues that the total number of bits accessible *up till now* is at most the square of the number of Planck times elapsed so far, or about $\left(10^{61}\right)^2 = 10^{122}$. Lloyd's bound, unlike Bousso's, does not depend on Λ being positive. The numerical coincidence between the two bounds reflects the experimental finding [208, 207] that we live in a transitional era, when both Λ and "dust" contribute significantly to the universe's net energy balance ($\Omega_\Lambda \approx 0.7$, $\Omega_{\text{dust}} \approx 0.3$). In earlier times dust (and before that radiation) dominated, and Lloyd's bound was tighter. In later times Λ will dominate, and Bousso's bound will be tighter. *Why* we should live in such a transitional era is unknown.

[4] Specifically, neglecting gravity and other forces that could counteract the effect of Λ.

At any time, the robot's state has the form

$$\sum \alpha_{i,z} |v_i, z\rangle .$$

Here $v_i \in V$ is a vertex, representing the robot's location; and z is a bit string (which can be arbitrarily long), representing the robot's internal configuration. The state evolves via an alternating sequence of T algorithm steps and T oracle steps:

$$U^{(1)} \rightarrow O^{(1)} \rightarrow U^{(1)} \rightarrow \cdots \rightarrow U^{(T)} \rightarrow O^{(T)}.$$

An oracle step $O^{(t)}$ maps each basis state $|v_i, z\rangle$ to $|v_i, z \oplus x_i\rangle$, where x_i is exclusive-OR'ed into the first bit of z. An algorithm step $U^{(t)}$ can be any unitary matrix that (1) does not depend on X, and (2) acts 'locally' on G. How to make the second condition precise is the subject of Section 14.4.1.

The initial state of the algorithm is $|v_1, 0\rangle$. Let $\alpha_{i,z}^{(t)}(X)$ be the amplitude of $|v_i, z\rangle$ immediately after the t^{th} oracle step; then the algorithm succeeds with probability $1 - \varepsilon$ if

$$\sum_{|v_i,z\rangle \,:\, z_{OUT}=f(X)} \left|\alpha_{i,z}^{(T)}(X)\right|^2 \geq 1 - \varepsilon$$

for all inputs X, where z_{OUT} is a bit of z representing the output.

14.4.1 Locality Criteria

Classically, it is easy to decide whether a stochastic matrix acts *locally* with respect to a graph G: it does if it moves probability only along the edges of G. In the quantum case, however, interference makes the question much more subtle. In this section we propose three criteria for whether a unitary matrix U is local. Our algorithms can be implemented using the most restrictive of these criteria, whereas our lower bounds apply to all three of them.

The first criterion we call *Z-locality* (for zero): U is Z-local if, given any pair of non-neighboring vertices v_1, v_2 in G, U "sends no amplitude" from v_1 to v_2; that is, the corresponding entries in U are all 0. The second criterion, *C-locality* (for composability), says that this is not enough: not only must U send amplitude only between neighboring vertices, but it must be composed of a product of commuting unitaries, each of which acts on a single edge. The third criterion is perhaps the most natural one to a physicist: U is *H-local* (for Hamiltonian) if it can be obtained by applying a locally-acting, low-energy Hamiltonian for some fixed amount of time. More formally, let $U_{i,z \rightarrow i^*,z^*}$ be the entry in the $|v_i, z\rangle$ column and $|v_{i^*}, z^*\rangle$ row of U.

Definition 101 *U is Z-local if $U_{i,z \rightarrow i^*,z^*} = 0$ whenever $i \neq i^*$ and (v_i, v_{i^*}) is not an edge of G.*

Definition 102 *U is C-local if the basis states can be partitioned into subsets $P_1, \ldots, P_q$ such that*

(i) $U_{i,z \rightarrow i^*,z^*} = 0$ whenever $|v_i, z\rangle$ and $|v_{i^*}, z^*\rangle$ belong to distinct P_j's, and

(ii) for each j, all basis states in P_j are either from the same vertex or from two adjacent vertices.

Definition 103 *U is H-local if $U = e^{iH}$ for some Hermitian H with eigenvalues of absolute value at most π, such that $H_{i,z \to i^*,z^*} = 0$ whenever $i \neq i^*$ and (v_i, v_{i^*}) is not an edge in E.*

If a unitary matrix is C-local, then it is also Z-local and H-local. For the latter implication, note that any unitary U can be written as e^{iH} for some H with eigenvalues of absolute value at most π. So we can write the unitary U_j acting on each P_j as e^{iH_j}; then since the U_j's commute,

$$\prod U_j = e^{i \sum H_j}.$$

Beyond that, though, how are the locality criteria related? Are they approximately equivalent? If not, then does a problem's complexity in our model ever depend on which criterion is chosen? Let us emphasize that these questions are *not* answered by, for example, the Solovay-Kitaev theorem (see [182]), that an $n \times n$ unitary matrix can be approximated using a number of gates polynomial in n. For recall that the definition of C-locality requires the edgewise operations to commute—indeed, without that requirement, one could produce any unitary matrix at all. So the relevant question, which we leave open, is whether any Z-local or H-local unitary can be approximated by a product of, say, $O(\log n)$ C-local unitaries. (A product of $O(n)$ such unitaries trivially suffices, but that is far too many.) Again, the algorithms in this chapter will use C-local unitaries, whereas the lower bounds will apply even to Z-local and H-local unitaries.

14.5 General Bounds

Given a Boolean function $f : \{0,1\}^n \to \{0,1\}$, the quantum query complexity $Q(f)$ is the minimum T for which there exists a T-query quantum algorithm that evaluates f with probability at least $2/3$ on all inputs. (We will always be interested in the *two-sided, bounded-error* complexity, denoted $Q_2(f)$ elsewhere in this thesis.) Similarly, given a graph G with n vertices labeled $1, \ldots, n$, we let $Q(f, G)$ be the minimum T for which there exists a T-query quantum robot on G that evaluates f with probability $2/3$. Here the algorithm steps must be C-local; we use $Q^Z(f, G)$ and $Q^H(f, G)$ to denote the corresponding measure with Z-local and H-local steps respectively. Clearly $Q(f, G) \geq Q^Z(f, G)$ and $Q(f, G) \geq Q^H(f, G)$; we do not know whether all three measures are asymptotically equivalent.

Let δ_G be the diameter of G, and call f *nondegenerate* if it depends on all n input bits.

Proposition 104 *For all f, G,*

(i) $Q(f, G) \leq 2n - 3$.

(ii) $Q(f, G) \leq (2\delta_G + 1) Q(f)$.

(iii) $Q(f, G) \geq Q(f)$.

(iv) $Q(f, G) \geq \delta_G / 2$ *if f is nondegenerate.*

Proof.

(i) Starting from the root, a spanning tree for G can be traversed in $2(n-1)-1$ steps (there is no need to return to the root).

(ii) We can simulate a query in $2\delta_G$ steps, by fanning out from the start vertex v_1 and then returning. Applying a unitary at v_1 takes 1 step.

(iii) Obvious.

(iv) There exists a vertex v_i whose distance to v_1 is at least $\delta_G/2$, and f could depend on x_i.

■

We now show that the model is robust.

Proposition 105 *For nondegenerate f, the following change $Q(f,G)$ by at most a constant factor.*

(i) Replacing the initial state $|v_1,0\rangle$ by an arbitrary (known) $|\psi\rangle$.

(ii) Requiring the final state to be localized at some vertex v_i with probability at least $1-\varepsilon$, for a constant $\varepsilon > 0$.

(iii) Allowing multiple algorithm steps between each oracle step (and measuring the complexity by the number of algorithm steps).

Proof.

(i) We can transform $|v_1,0\rangle$ to $|\psi\rangle$ (and hence $|\psi\rangle$ to $|v_1,0\rangle$) in $\delta_G = O(Q(f,G))$ steps, by fanning out from v_1 along the edges of a minimum-height spanning tree.

(ii) Assume without loss of generality that z_{OUT} is accessed only once, to write the output. Then after z_{OUT} is accessed, uncompute (that is, run the algorithm backwards) to localize the final state at v_1. The state can then be localized at any v_i in $\delta_G = O(Q(f,G))$ steps. We can succeed with any constant probability by repeating this procedure a constant number of times.

(iii) The oracle step O is its own inverse, so we can implement a sequence $U_1, U_2, \ldots$ of algorithm steps as follows (where I is the identity):

$$U_1 \rightarrow O \rightarrow I \rightarrow O \rightarrow U_2 \rightarrow \cdots$$

■

A function of particular interest is $f = \mathrm{OR}(x_1,\ldots,x_n)$, which outputs 1 if and only if $x_i = 1$ for some i. We first give a general upper bound on $Q(\mathrm{OR},G)$ in terms of the diameter of G. (Throughout the chapter, we sometimes omit floor and ceiling signs if they clearly have no effect on the asymptotics.)

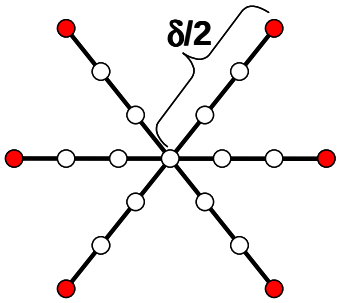

Figure 14.2: The 'starfish' graph G. The marked item is at one of the tip vertices.

Proposition 106

$$Q(\mathrm{OR}, G) = O\left(\sqrt{n\delta_G}\right).$$

Proof. Let τ be a minimum-height spanning tree for G, rooted at v_1. A depth-first search on τ uses $2n-2$ steps. Let S_1 be the set of vertices visited by depth-first search in steps 1 to δ_G, S_2 be those visited in steps δ_G+1 to $2\delta_G$, and so on. Then

$$S_1 \cup \cdots \cup S_{2n/\delta_G} = V.$$

Furthermore, for each S_j there is a classical algorithm A_j, using at most $3\delta_G$ steps, that starts at v_1, ends at v_1, and outputs '1' if and only if $x_i = 1$ for some $v_i \in S_j$. Then we simply perform Grover search at v_1 over all A_j; since each iteration takes $O(\delta_G)$ steps and there are $O\left(\sqrt{2n/\delta_G}\right)$ iterations, the number of steps is $O\left(\sqrt{n\delta_G}\right)$. ■

The bound of Proposition 106 is tight:

Theorem 107 *For all δ, there exists a graph G with diameter $\delta_G = \delta$ such that*

$$Q(\mathrm{OR}, G) = \Omega\left(\sqrt{n\delta}\right).$$

Indeed, $Q^Z(f, G)$ and $Q^H(f, G)$ are also $\Omega\left(\sqrt{n\delta}\right)$.

Proof. For simplicity, we first consider the C-local and Z-local cases, and then discuss what changes in the H-local case. Let G be a 'starfish' with central vertex v_1 and $M = 2(n-1)/\delta$ legs $L_1, \ldots, L_M$, each of length $\delta/2$ (see Figure 14.2). We use the hybrid argument of Bennett et al. [51]. Suppose we run the algorithm on the all-zero input X_0. Then define the *query magnitude* $\Gamma_j^{(t)}$ to be the probability of finding the robot in leg L_j immediately after the t^{th} query:

$$\Gamma_j^{(t)} = \sum_{v_i \in L_j} \sum_z \left|\alpha_{i,z}^{(t)}(X_0)\right|^2.$$

Let T be the total number of queries, and let $w = T/(c\delta)$ for some constant $0 < c < 1/2$. Clearly

$$\sum_{q=0}^{w-1} \sum_{j=1}^{M} \Gamma_j^{(T-qc\delta)} \leq \sum_{q=0}^{w-1} 1 = w.$$

Hence there must exist a leg L_{j^*} such that

$$\sum_{q=0}^{w-1} \Gamma_{j^*}^{(T-qc\delta)} \leq \frac{w}{M} = \frac{w\delta}{2(n-1)}.$$

Let v_{i^*} be the tip vertex of L_{j^*}, and let Y be the input which is 1 at v_{i^*} and 0 elsewhere. Then let X_q be a hybrid input, which is X_0 during queries 1 to $T - qc\delta$, but Y during queries $T - qc\delta + 1$ to T. Also, let

$$\left|\psi^{(t)}(X_q)\right\rangle = \sum_{i,z} \alpha_{i,z}^{(t)}(X_q) |v_i, z\rangle$$

be the algorithm's state after t queries when run on X_q, and let

$$\begin{aligned} D(q,r) &= \left\| \left|\psi^{(T)}(X_q)\right\rangle - \left|\psi^{(T)}(X_r)\right\rangle \right\|_2^2 \\ &= \sum_{v_i \in G} \sum_{z} \left| \alpha_{i,z}^{(T)}(X_q) - \alpha_{i,z}^{(T)}(X_r) \right|^2. \end{aligned}$$

Then for all $q \geq 1$, we claim that $D(q-1,q) \leq 4\Gamma_{j^*}^{(T-qc\delta)}$. For by unitarity, the Euclidean distance between $\left|\psi^{(t)}(X_{q-1})\right\rangle$ and $\left|\psi^{(t)}(X_q)\right\rangle$ can only increase as a result of queries $T - qc\delta + 1$ through $T - (q-1)c\delta$. But no amplitude from outside L_{j^*} can reach v_{i^*} during that interval, since the distance is $\delta/2$ and there are only $c\delta < \delta/2$ time steps. Therefore, switching from X_{q-1} to X_q can only affect amplitude that is in L_{j^*} immediately after query $T - qc\delta$:

$$\begin{aligned} D(q-1,q) &\leq \sum_{v_i \in L_{j^*}} \sum_{z} \left| \alpha_{i,z}^{(T-qc\delta)}(X_q) - \left(-\alpha_{i,z}^{(T-qc\delta)}(X_q)\right) \right|^2 \\ &= 4 \sum_{v_i \in L_{j^*}} \sum_{z} \left| \alpha_{i,z}^{(T-qc\delta)}(X_0) \right|^2 = 4\Gamma_{j^*}^{(T-qc\delta)}. \end{aligned}$$

It follows that

$$\sqrt{D(0,w)} \leq \sum_{q=1}^{w} \sqrt{D(q-1,q)} \leq 2 \sum_{q=1}^{w} \sqrt{\Gamma_{j^*}^{(T-qc\delta)}} \leq 2w \sqrt{\frac{\delta}{2(n-1)}} = \frac{T}{c} \sqrt{\frac{2}{\delta(n-1)}}.$$

Here the first inequality uses the triangle inequality, and the third uses the Cauchy-Schwarz inequality. Now assuming the algorithm is correct we need $D(0,w) = \Omega(1)$, which implies that $T = \Omega\left(\sqrt{n\delta}\right)$.

In the H-local case, it is no longer true that no amplitude from outside L_{j^*} can reach v_{i^*} in $c\delta$ time steps. But if c is a small enough constant, then the amount of amplitude that can reach v_{i^*} decreases exponentially in δ. To see this, assume without loss of generality that all amplitude not in L_{j^*} starts in the state $|v_0, \psi\rangle$, where $|\psi\rangle$ is some superposition over auxiliary qubits. Let H be the local Hamiltonian that acts between the t^{th} and $(t+1)^{st}$ queries, all of whose eigenvalues have absolute value at most π. Since H is Hermitian, we can decompose it as $V\Lambda V^{-1}$ where V is unitary and Λ is diagonal. So by Taylor series expansion,

$$e^{iH} = \sum_{j \geq 0} \frac{i^j}{j!} V \Lambda^j V^{-1}.$$

Now let S be the set of basis states $|v_b, z_b\rangle$ such that the distance from v_0 to v_b is ℓ, for some $\ell > 4\pi$. Notice that for all $j < \ell$ and $|v_b, z_b\rangle \in S$, we have

$$\langle v_b, z_b | H^j | v_0, \psi \rangle = \langle v_b, z_b | V \Lambda^j V^{-1} | v_0, \psi \rangle = 0$$

by the locality of H. Therefore

$$\begin{aligned}
\sum_{|v_b, z_b\rangle \in S} \left| \langle v_b, z_b | e^{iH} | v_0, \psi \rangle \right|^2 &= \sum_{|v_b, z_b\rangle \in S} \left| \sum_{j \geq \ell} \frac{i^j}{j!} \langle v_b, z_b | V \Lambda^j V^{-1} | v_0, \psi \rangle \right|^2 \\
&\leq \left(\sum_{j \geq \ell} \sqrt{\sum_{|v_b, z_b\rangle \in S} \left| \frac{i^j}{j!} \langle v_b, z_b | V \Lambda^j V^{-1} | v_0, \psi \rangle \right|^2} \right)^2 \\
&\leq \left(\sum_{j \geq \ell} \sqrt{\frac{\pi^j}{j!}} \right)^2 \\
&\leq \frac{4\pi^\ell}{\ell!}.
\end{aligned}$$

Here the second line uses the triangle inequality, the third line uses the fact that $V\Lambda^j V^{-1}$ has maximum eigenvalue at most π^j (and therefore $\left(i^j/j!\right) V\Lambda^j V^{-1}$ has maximum eigenvalue at most $\pi^j/j!$), and the fourth line uses the fact that $\ell > 4\pi$. Intuitively, the probability that H sends the robot a distance ℓ from v_0 is at most $4\pi^\ell/\ell!$, which decreases exponentially in ℓ. One can now use a Chernoff-Hoeffding bound to upper-bound the probability that $c\delta$ local Hamiltonians, applied in succession, ever move the robot a distance $\delta/2$ from v_0. It is clear that the resulting upper bound is $2^{-\Omega(\delta)}$ for small enough c. Therefore

$$D\left(q-1, q\right) \leq 4\Gamma_{j^*}^{(T-qc\delta)} + 2^{-\Omega(\delta)}$$

and the remainder of the proof goes through as before. ■

14.6 Search on Grids

Let $\mathcal{L}_d\left(n\right)$ be a d-dimensional grid graph of size $n^{1/d} \times \cdots \times n^{1/d}$. That is, each vertex is specified by d coordinates $i_1, \ldots, i_d \in \left\{1, \ldots, n^{1/d}\right\}$, and is connected to the at most $2d$

vertices obtainable by adding or subtracting 1 from a single coordinate (boundary vertices have fewer than $2d$ neighbors). We write simply $\mathcal{L}_d$ when n is clear from context. In this section we present our main positive results: that $Q(\mathrm{OR}, \mathcal{L}_d) = \Theta(\sqrt{n})$ for $d \geq 3$, and $Q(\mathrm{OR}, \mathcal{L}_2) = O(\sqrt{n} \operatorname{polylog} n)$ for $d = 2$.

Before proving these claims, let us develop some intuition by showing weaker bounds, taking the case $d = 2$ for illustration. Clearly $Q(\mathrm{OR}, \mathcal{L}_2) = O\left(n^{3/4}\right)$: we simply partition $\mathcal{L}_2(n)$ into $\sqrt{n}$ subsquares, each a copy of $\mathcal{L}_2(\sqrt{n})$. In $5\sqrt{n}$ steps, the robot can travel from the start vertex to any subsquare C, search C classically for a marked vertex, and then return to the start vertex. Thus, by searching all $\sqrt{n}$ of the C's in superposition and applying Grover's algorithm, the robot can search the grid in time $O\left(n^{1/4}\right) \times 5\sqrt{n} = O\left(n^{3/4}\right)$.

Once we know that, we might as well partition $\mathcal{L}_2(n)$ into $n^{1/3}$ subsquares, each a copy of $\mathcal{L}_2\left(n^{2/3}\right)$. Searching any one of these subsquares by the previous algorithm takes time $O\left(\left(n^{2/3}\right)^{3/4}\right) = O(\sqrt{n})$, an amount of time that also suffices to travel to the subsquare and back from the start vertex. So using Grover's algorithm, the robot can search $\mathcal{L}_2(n)$ in time $O\left(\sqrt{n^{1/3}} \cdot \sqrt{n}\right) = O\left(n^{2/3}\right)$. We can continue recursively in this manner to make the running time approach $O(\sqrt{n})$. The trouble is that, with each additional layer of recursion, the robot needs to repeat the search more often to upper-bound the error probability. Using this approach, the best bounds we could obtain are roughly $O(\sqrt{n} \operatorname{polylog} n)$ for $d \geq 3$, or $\sqrt{n} 2^{O\left(\sqrt{\log n}\right)}$ for $d = 2$. In what follows, we use the amplitude amplification approach of Brassard et al. [67] to improve these bounds, in the case of a single marked vertex, to $O(\sqrt{n})$ for $d \geq 3$ (Section 14.6.2) and $O\left(\sqrt{n} \log^{3/2} n\right)$ for $d = 2$ (Section 14.6.3). Section 14.6.4 generalizes these results to the case of multiple marked vertices.

Intuitively, the reason the case $d = 2$ is special is that there, the diameter of the grid is $\Theta(\sqrt{n})$, which matches exactly the time needed for Grover search. For $d \geq 3$, by contrast, the robot can travel across the grid in much less time than is needed to search it.

14.6.1 Amplitude Amplification

We start by describing amplitude amplification [67], a generalization of Grover search. Let A be a quantum algorithm that, with probability ϵ, outputs a correct answer together with a witness that proves the answer correct. (For example, in the case of search, the algorithm outputs a vertex label i such that $x_i = 1$.) Amplification generates a new algorithm that calls A order $1/\sqrt{\epsilon}$ times, and that produces both a correct answer and a witness with probability $\Omega(1)$. In particular, assume A starts in basis state $|s\rangle$, and let m be a positive integer. Then the amplification procedure works as follows:

(1) Set $|\psi_0\rangle = A|s\rangle$.

(2) For $i = 1$ to m set $|\psi_{i+1}\rangle = ASA^{-1}W|\psi_i\rangle$, where

- W flips the phase of basis state $|y\rangle$ if and only if $|y\rangle$ contains a description of a correct witness, and
- S flips the phase of basis state $|y\rangle$ if and only if $|y\rangle = |s\rangle$.

We can decompose $|\psi_0\rangle$ as $\sin\alpha\,|\Psi_{\text{succ}}\rangle + \cos\alpha\,|\Psi_{\text{fail}}\rangle$, where $|\Psi_{\text{succ}}\rangle$ is a superposition over basis states containing a correct witness and $|\Psi_{\text{fail}}\rangle$ is a superposition over all other basis states. Brassard et al. [67] showed the following:

Lemma 108 ([67]) $|\psi_i\rangle = \sin\left[(2i+1)\,\alpha\right]|\Psi_{\text{succ}}\rangle + \cos\left[(2i+1)\,\alpha\right]|\Psi_{\text{fail}}\rangle$.

If measuring $|\psi_0\rangle$ gives a correct witness with probability ϵ, then $|\sin\alpha|^2 = \epsilon$ and $|\alpha| \geq 1/\sqrt{\epsilon}$. So taking $m = O(1/\sqrt{\epsilon})$ yields $\sin\left[(2m+1)\,\alpha\right] \approx 1$. For our algorithms, though, the multiplicative constant under the big-O also matters. To upper-bound this constant, we prove the following lemma.

Lemma 109 *Suppose a quantum algorithm A outputs a correct answer and witness with probability exactly ϵ. Then by using $2m+1$ calls to A or A^{-1}, where*

$$m \leq \frac{\pi}{4\arcsin\sqrt{\epsilon}} - \frac{1}{2},$$

we can output a correct answer and witness with probability at least

$$\left(1 - \frac{(2m+1)^2}{3}\epsilon\right)(2m+1)^2\,\epsilon.$$

Proof. We perform m steps of amplitude amplification, which requires $2m+1$ calls A or A^{-1}. By Lemma 108, this yields the final state

$$\sin\left[(2m+1)\,\alpha\right]|\Psi_{\text{succ}}\rangle + \cos\left[(2m+1)\,\alpha\right]|\Psi_{\text{fail}}\rangle\,.$$

where $\alpha = \arcsin\sqrt{\epsilon}$. Therefore the success probability is

$$\begin{aligned}
\sin^2\left[(2m+1)\arcsin\sqrt{\epsilon}\right] &\geq \sin^2\left[(2m+1)\sqrt{\epsilon}\right] \\
&\geq \left((2m+1)\sqrt{\epsilon} - \frac{(2m+1)^3}{6}\epsilon^{3/2}\right)^2 \\
&\geq (2m+1)^2\,\epsilon - \frac{(2m+1)^4}{3}\epsilon^2.
\end{aligned}$$

Here the first line uses the monotonicity of $\sin^2 x$ in the interval $[0, \pi/2]$, and the second line uses the fact that $\sin x \geq x - x^3/6$ for all $x \geq 0$ by Taylor series expansion. ■

Note that there is no need to uncompute any garbage left by A, beyond the uncomputation that happens "automatically" within the amplification procedure.

14.6.2 Dimension At Least 3

Our goal is the following:

Theorem 110 *If $d \geq 3$, then $Q\left(\text{OR}, \mathcal{L}_d\right) = \Theta\left(\sqrt{n}\right)$.*

In this section, we prove Theorem 110 for the special case of a unique marked vertex; then, in Sections 14.6.4 and 14.6.5, we will generalize to multiple marked vertices. Let $\mathrm{OR}^{(k)}$ be the problem of deciding whether there are no marked vertices or exactly k of them, given that one of these is true. Then:

Theorem 111 *If $d \geq 3$, then $Q\left(\mathrm{OR}^{(1)}, \mathcal{L}_d\right) = \Theta\left(\sqrt{n}\right)$.*

Choose constants $\beta \in (2/3, 1)$ and $\mu \in (1/3, 1/2)$ such that $\beta\mu > 1/3$ (for example, $\beta = 4/5$ and $\mu = 5/11$ will work). Let ℓ_0 be a large positive integer; then for all positive integers R, let $\ell_R = \ell_{R-1} \left\lceil \ell_{R-1}^{1/\beta - 1} \right\rceil$. Also let $n_R = \ell_R^d$. Assume for simplicity that $n = n_R$ for some R; in other words, that the hypercube $\mathcal{L}_d(n_R)$ to be searched has sides of length ℓ_R. Later we will remove this assumption.

Consider the following recursive algorithm $\mathcal{A}$. If $n = n_0$, then search $\mathcal{L}_d(n_0)$ classically, returning 1 if a marked vertex is found and 0 otherwise. Otherwise partition $\mathcal{L}_d(n_R)$ into n_R/n_{R-1} subcubes, each one a copy of $\mathcal{L}_d(n_{R-1})$. Take the algorithm that consists of picking a subcube C uniformly at random, and then running $\mathcal{A}$ recursively on C. Amplify this algorithm $(n_R/n_{R-1})^{\mu}$ times.

The intuition behind the exponents is that $n_{R-1} \approx n_R^{\beta}$, so searching $\mathcal{L}_d(n_{R-1})$ should take about $n_R^{\beta/2}$ steps, which dominates the $n_R^{1/d}$ steps needed to travel across the hypercube when $d \geq 3$. Also, at level R we want to amplify a number of times that is less than $(n_R/n_{R-1})^{1/2}$ by some polynomial amount, since full amplification would be inefficient. The reason for the constraint $\beta\mu > 1/3$ will appear in the analysis.

We now provide a more explicit description of $\mathcal{A}$, which shows that $\mathcal{A}$ can be implemented using C-local unitaries and only a single bit of workspace. At any time, the quantum robot's state will have the form $\sum_{i,z} \alpha_{i,z} |v_i, z\rangle$, where v_i is a vertex of $\mathcal{L}_d(n_R)$ and z is a single bit that records whether or not a marked vertex has been found. Given a subcube C, let $v(C)$ be the "corner" vertex of C; that is, the vertex that is minimal in all d coordinates. Then the initial state when searching C will be $|v(C), 0\rangle$. Beware, however, that "initial state" in this context just means the state $|s\rangle$ from Section 14.6.1. Because of the way amplitude amplification works, $\mathcal{A}$ will often be invoked on C with other initial states, and even run in reverse.

Below we give pseudocode for $\mathcal{A}$. Our procedure calls the three unitaries A, W, and S from Section 14.6.1 as subroutines. For convenience, we write $\mathcal{A}_R, A_R, W_R, S_R$ to denote the level of recursion that is currently active.

Algorithm 112 ($\mathcal{A}_R$) *Searches a subcube C of size n_R for the marked vertex, and amplifies the result to have larger probability. Default initial state: $|v(C), 0\rangle$.*

If** $R = 0$ **then:

(1) Use classical C-local operations to visit all n_0 vertices of C in any order. At each $v_i \in C$, use a query transformation to map the state $|v_i, z\rangle$ to $|v_i, z \oplus x_i\rangle$.

(2) Return to $v(C)$.

If** $R \geq 1$ **then:

(1) Let m_R be the smallest integer such that $2m_R + 1 \geq (n_R/n_{R-1})^{\mu}$.

(2) Call A_R.

(3) For $i = 1$ to m_R, call W_R, then A_R^{-1}, then S_R, then A_R.

Suppose $\mathcal{A}_R$ is run on the initial state $|v(C), 0\rangle$, and let $C_1, \ldots, C_{n_R/n_0}$ be the *minimal subcubes* in C—meaning those of size n_0. Then the final state after $\mathcal{A}_R$ terminates should be

$$\frac{1}{\sqrt{n_R/n_0}} \sum_{i=1}^{n_R/n_0} |v(C_i), 0\rangle$$

if C does not contain the marked vertex. Otherwise the final state should have non-negligible overlap with $|v(C_{i^*}), 1\rangle$, where C_{i^*} is the minimal subcube in C that contains the marked vertex. In particular, if $R = 0$, then the final state should be $|v(C), 1\rangle$ if C contains the marked vertex, and $|v(C), 0\rangle$ otherwise.

The two phase-flip subroutines, W_R and S_R, are both trivial to implement. To apply W_R, map each basis state $|v_i, z\rangle$ to $(-1)^z |v_i, z\rangle$. To apply S_R, map each basis state $|v_i, z\rangle$ to $-|v_i, z\rangle$ if $v_i = v(C)$ for some subcube C of size n_R, and to $|v_i, z\rangle$ otherwise. Below we give pseudocode for A_R.

Algorithm 113 (A_R) *Searches a subcube C of size n_R for the marked vertex. Default initial state: $|v(C), 0\rangle$.*

(1) Partition C into n_R/n_{R-1} smaller subcubes $C_1, \ldots, C_{n_R/n_{R-1}}$, each of size n_{R-1}.

(2) For all $j \in \{1, \ldots, d\}$, let V_j be the set of corner vertices $v(C_i)$ that differ from $v(C)$ only in the first j coordinates. Thus $V_0 = \{v(C)\}$, and in general $|V_j| = \ell_R^j$. For $j = 1$ to d, let $|V_j\rangle$ be the state

$$|V_j\rangle = \frac{1}{\ell_R^{j/2}} \sum_{v(C_i) \in V_j} |v(C_i), 0\rangle$$

Apply a sequence of transformations U_1, U_2, ..., U_d where U_j is a unitary that maps $|V_{j-1}\rangle$ to $|V_j\rangle$ by applying C-local unitaries that move amplitude only along the j^{th} coordinate.

(3) Call $\mathcal{A}_{R-1}$ recursively, to search $C_1, \ldots, C_{n_R/n_{R-1}}$ in superposition and amplify the results.

If A_R is run on the initial state $|v(C), 0\rangle$, then the final state should be

$$\frac{1}{\sqrt{n_R/n_{R-1}}} \sum_{i=1}^{n_R/n_0} |\phi_i\rangle,$$

where $|\phi_i\rangle$ is the correct final state when $\mathcal{A}_{R-1}$ is run on subcube C_i with initial state $|v(C_i), 0\rangle$. A key point is that there is no need for A_R to call $\mathcal{A}_{R-1}$ twice, once to compute and once to uncompute—for the uncomputation is already built in to $\mathcal{A}$. This is what will enable us to prove an upper bound of $O(\sqrt{n})$ instead of $O(\sqrt{n}2^R) = O(\sqrt{n}\operatorname{polylog} n)$.

We now analyze the running time of $\mathcal{A}$.

Lemma 114 *$\mathcal{A}_R$ uses $O\left(n_R^\mu\right)$ steps.*

Proof. Let $T_{\mathcal{A}}(R)$ and $T_A(R)$ be the total numbers of steps used by $\mathcal{A}_R$ and A_R respectively in searching $\mathcal{L}_d(n_R)$. Then we have $T_{\mathcal{A}}(0) = O(1)$, and

$$\begin{aligned} T_{\mathcal{A}}(R) &\leq (2m_R + 1)\, T_A(R) + 2m_R, \\ T_A(R) &\leq d n_R^{1/d} + T_{\mathcal{A}}(R-1) \end{aligned}$$

for all $R \geq 1$. For W_R and S_R can both be implemented in a single step, while A_R uses $d\ell_R = d n_R^{1/d}$ steps to move the robot across the hypercube. Combining,

$$\begin{aligned} T_{\mathcal{A}}(R) &\leq (2m_R + 1)\left(d n_R^{1/d} + T_{\mathcal{A}}(R-1)\right) + 2m_R \\ &\leq \left((n_R/n_{R-1})^\mu + 2\right)\left(d n_R^{1/d} + T_{\mathcal{A}}(R-1)\right) + (n_R/n_{R-1})^\mu + 1 \\ &= O\left((n_R/n_{R-1})^\mu\, n_R^{1/d}\right) + \left((n_R/n_{R-1})^\mu + 2\right) T_{\mathcal{A}}(R-1) \\ &= O\left((n_R/n_{R-1})^\mu\, n_R^{1/d}\right) + (n_R/n_{R-1})^\mu\, T_{\mathcal{A}}(R-1) \\ &= O\left((n_R/n_{R-1})^\mu\, n_R^{1/d} + (n_R/n_{R-2})^\mu\, n_{R-1}^{1/d} + \cdots + (n_R/n_0)^\mu\, n_1^{1/d}\right) \\ &= n_R^\mu \cdot O\left(\frac{n_R^{1/d}}{n_{R-1}^\mu} + \frac{n_{R-1}^{1/d}}{n_{R-2}^\mu} + \cdots + \frac{n_1^{1/d}}{n_0^\mu}\right) \\ &= n_R^\mu \cdot O\left(n_R^{1/d-\beta\mu} + \cdots + n_2^{1/d-\beta\mu} + n_1^{1/d-\beta\mu}\right) \\ &= n_R^\mu \cdot O\left(n_R^{1/d-\beta\mu} + \left(n_R^{1/d-\beta\mu}\right)^{1/\beta} + \cdots + \left(n_R^{1/d-\beta\mu}\right)^{1/\beta^{R-1}}\right) \\ &= O\left(n_R^\mu\right). \end{aligned}$$

Here the second line follows because $2m_R + 1 \leq (n_R/n_{R-1})^\mu + 2$, the fourth because the $(n_R/n_{R-1})^\mu$ terms increase doubly exponentially, so adding 2 to each will not affect the asymptotics; the seventh because $n_i^\mu = \Omega\left(\left(n_{i+1}^\mu\right)^\beta\right)$, the eighth because $n_{R-1} \leq n_R^\beta$; and the last because $\beta\mu > 1/3 \geq 1/d$, hence $n_1^{1/d-\beta\mu} < 1$. ■

Next we need to lower-bound the success probability. Say that $\mathcal{A}$ or A "succeeds" if a measurement in the standard basis yields the result $|v(C_{i^*}), 1\rangle$, where C_{i^*} is the minimal subcube that contains the marked vertex. Of course, the marked vertex itself can then be found in $n_0 = O(1)$ steps.

Lemma 115 *Assuming there is a unique marked vertex, $\mathcal{A}_R$ succeeds with probability $\Omega\left(1/n_R^{1-2\mu}\right)$.*

Proof. Let $P_{\mathcal{A}}(R)$ and $P_A(R)$ be the success probabilities of $\mathcal{A}_R$ and A_R respectively when searching $\mathcal{L}_d(n_R)$. Then clearly $P_{\mathcal{A}}(0) = 1$, and $P_A(R) = (n_{R-1}/n_R)\, P_{\mathcal{A}}(R-1)$

for all $R \geq 1$. So by Lemma 109,

$$\begin{aligned}
P_{\mathcal{A}}(R) &\geq \left(1 - \frac{1}{3}(2m_R+1)^2 P_A(R)\right)(2m_R+1)^2 P_A(R) \\
&= \left(1 - \frac{1}{3}(2m_R+1)^2 \frac{n_{R-1}}{n_R} P_{\mathcal{A}}(R-1)\right)(2m_R+1)^2 \frac{n_{R-1}}{n_R} P_{\mathcal{A}}(R-1) \\
&\geq \left(1 - \frac{1}{3}(n_R/n_{R-1})^{2\mu} \frac{n_{R-1}}{n_R} P_{\mathcal{A}}(R-1)\right)(n_R/n_{R-1})^{2\mu} \frac{n_{R-1}}{n_R} P_{\mathcal{A}}(R-1) \\
&\geq \left(1 - \frac{1}{3}(n_{R-1}/n_R)^{1-2\mu}\right)(n_{R-1}/n_R)^{1-2\mu} P_{\mathcal{A}}(R-1) \\
&\geq (n_0/n_R)^{1-2\mu} \prod_{r=1}^{R} \left(1 - \frac{1}{3}(n_{R-1}/n_R)^{1-2\mu}\right) \\
&\geq (n_0/n_R)^{1-2\mu} \prod_{r=1}^{R} \left(1 - \frac{1}{3n_R^{(1-\beta)(1-2\mu)}}\right) \\
&\geq (n_0/n_R)^{1-2\mu} \left(1 - \sum_{r=1}^{R} \frac{1}{3N_R^{(1-\beta)(1-2\mu)}}\right) \\
&= \Omega\left(1/n_R^{1-2\mu}\right).
\end{aligned}$$

Here the third line follows because $2m_R + 1 \geq n_{R-1}/n_R$ and the function $x - \frac{1}{3}x^2$ is nondecreasing in the interval $[0,1]$; the fourth because $P_{\mathcal{A}}(R-1) \leq 1$; the sixth because $n_{R-1} \leq n_R^{\beta}$; and the last because $\beta < 1$ and $\mu < 1/2$, the n_R's increase doubly exponentially, and n_0 is sufficiently large. ■

Finally, take $\mathcal{A}_R$ itself and amplify it to success probability $\Omega(1)$ by running it $O(n_R^{1/2-\mu})$ times. This yields an algorithm for searching $\mathcal{L}_d(n_R)$ with overall running time $O\left(n_R^{1/2}\right)$, which implies that $Q\left(\mathrm{OR}^{(1)}, \mathcal{L}_d(n_R)\right) = O\left(n_R^{1/2}\right)$.

All that remains is to handle values of n that do not equal n_R for any R. The solution is simple: first find the largest R such that $n_R < n$. Then set $n' = n_R \left\lceil n^{1/d}/\ell_R \right\rceil^d$, and embed $\mathcal{L}_d(n)$ into the larger hypercube $\mathcal{L}_d(n')$. Clearly $Q\left(\mathrm{OR}^{(1)}, \mathcal{L}_d(n)\right) \leq Q\left(\mathrm{OR}^{(1)}, \mathcal{L}_d(n')\right)$. Also notice that $n' = O(n)$ and that $n' = O\left(n_R^{1/\beta}\right) = O\left(n_R^{3/2}\right)$. Next partition $\mathcal{L}_d(n')$ into n'/n_R subcubes, each a copy of $\mathcal{L}_d(n_R)$. The algorithm will now have one additional level of recursion, which chooses a subcube of $\mathcal{L}_d(n')$ uniformly at random, runs $\mathcal{A}_R$ on that subcube, and then amplifies the resulting procedure $\Theta\left(\sqrt{n'/n_R}\right)$ times. The total time is now

$$O\left(\sqrt{\frac{n'}{n_R}}\left(\left(n'\right)^{1/d} + n_R^{1/2}\right)\right) = O\left(\sqrt{\frac{n'}{n_R}} n_R^{1/2}\right) = O\left(\sqrt{n}\right),$$

while the success probability is $\Omega(1)$. This completes Theorem 111.

14.6.3 Dimension 2

In the $d = 2$ case, the best we can achieve is the following:

Theorem 116 $Q(\mathrm{OR}, \mathcal{L}_2) = O\left(\sqrt{n}\log^{5/2} n\right)$.

Again, we start with the single marked vertex case and postpone the general case to Sections 14.6.4 and 14.6.5.

Theorem 117 $Q\left(\mathrm{OR}^{(1)}, \mathcal{L}_2\right) = O\left(\sqrt{n}\log^{3/2} n\right)$.

For $d \geq 3$, we performed amplification on large (greater than $O\left(1/n^{1-2\mu}\right)$) probabilities only once, at the end. For $d = 2$, on the other hand, any algorithm that we construct with any nonzero success probability will have running time $\Omega(\sqrt{n})$, simply because that is the diameter of the grid. If we want to keep the running time $O(\sqrt{n})$, then we can only perform $O(1)$ amplification steps at the end. Therefore we need to keep the success probability relatively high throughout the recursion, meaning that we suffer an increase in the running time, since amplification to high probabilities is less efficient.

The procedures $\mathcal{A}_R$, A_R, W_R, and S_R are identical to those in Section 14.6.2; all that changes are the parameter settings. For all integers $R \geq 0$, we now let $n_R = \ell_0^{2R}$, for some odd integer $\ell_0 \geq 3$ to be set later. Thus, $\mathcal{A}_R$ and A_R search the square grid $\mathcal{L}_2(n_R)$ of size $\ell_0^R \times \ell_0^R$. Also, let $m = (\ell_0 - 1)/2$; then $\mathcal{A}_R$ applies m steps of amplitude amplification to A_R.

We now prove the counterparts of Lemmas 114 and 115 for the two-dimensional case.

Lemma 118 *$\mathcal{A}_R$ uses $O\left(R\ell_0^{R+1}\right)$ steps.*

Proof. Let $T_{\mathcal{A}}(R)$ and $T_A(R)$ be the time used by $\mathcal{A}_R$ and A_R respectively in searching $\mathcal{L}_2(n_R)$. Then $T_{\mathcal{A}}(0) = 1$, and for all $R \geq 1$,

$$\begin{aligned} T_{\mathcal{A}}(R) &\leq (2m+1)\,T_A(R) + 2m, \\ T_A(R) &\leq 2n_R^{1/2} + T_{\mathcal{A}}(R-1). \end{aligned}$$

Combining,

$$\begin{aligned} T_{\mathcal{A}}(R) &\leq (2m+1)\left(2n_R^{1/2} + T_{\mathcal{A}}(R-1)\right) + 2m \\ &= \ell_0\left(2\ell_0^R + T_{\mathcal{A}}(R-1)\right) + \ell_0 - 1 \\ &= O\left(\ell_0^{R+1} + \ell_0 T_{\mathcal{A}}(R-1)\right) \\ &= O\left(R\ell_0^{R+1}\right). \end{aligned}$$

■

Lemma 119 *$\mathcal{A}_R$ succeeds with probability $\Omega(1/R)$.*

Proof. Let $P_{\mathcal{A}}(R)$ and $P_A(R)$ be the success probabilities of $\mathcal{A}_R$ and A_R respectively when searching $\mathcal{L}_2(n_R)$. Then $P_A(R) = P_{\mathcal{A}}(R-1)/\ell_0^2$ for all $R \geq 1$. So by Lemma 109, and using the fact that $2m+1 = \ell_0$,

$$\begin{aligned} P_{\mathcal{A}}(R) &\geq \left(1 - \frac{(2m+1)^2}{3} P_A(R)\right)(2m+1)^2 P_A(R) \\ &= \left(1 - \frac{\ell_0^2}{3}\frac{P_{\mathcal{A}}(R-1)}{\ell_0^2}\right)\ell_0^2 \frac{P_{\mathcal{A}}(R-1)}{\ell_0^2} \\ &= P_{\mathcal{A}}(R-1) - \frac{1}{3}P_{\mathcal{A}}^2(R-1) \\ &= \Omega(1/R). \end{aligned}$$

This is because $\Omega(R)$ iterations of the map $x_R := x_{R-1} - \frac{1}{3}x_{R-1}^2$ are needed to drop from (say) $2/R$ to $1/R$, and $x_0 = P_{\mathcal{A}}(0) = 1$ is greater than $2/R$. ■

We can amplify $\mathcal{A}_R$ to success probability $\Omega(1)$ by repeating it $O\left(\sqrt{R}\right)$ times. This yields an algorithm for searching $\mathcal{L}_2(n_R)$ that uses $O\left(R^{3/2}\ell_0^{R+1}\right) = O\left(\sqrt{n_R}R^{3/2}\ell_0\right)$ steps in total. We can minimize this expression subject to $\ell_0^{2R} = n_R$ by taking ℓ_0 to be constant and R to be $\Theta(\log n_R)$, which yields $Q\left(\mathrm{OR}^{(1)}, \mathcal{L}_2(n_R)\right) = O\left(\sqrt{n_R}\log n_R^{3/2}\right)$. If n is not of the form ℓ_0^{2R}, then we simply find the smallest integer R such that $n < \ell_0^{2R}$, and embed $\mathcal{L}_2(n)$ in the larger grid $\mathcal{L}_2\left(\ell_0^{2R}\right)$. Since ℓ_0 is a constant, this increases the running time by at most a constant factor. We have now proved Theorem 117.

14.6.4 Multiple Marked Items

What about the case in which there are multiple i's with $x_i = 1$? If there are k marked items (where k need not be known in advance), then Grover's algorithm can find a marked item with high probability in $O\left(\sqrt{n/k}\right)$ queries, as shown by Boyer et al. [66]. In our setting, however, this is too much to hope for—since even if there are many marked vertices, they might all be in a faraway part of the hypercube. Then $\Omega\left(n^{1/d}\right)$ steps are needed, even if $\sqrt{n/k} < n^{1/d}$. Indeed, we can show a stronger lower bound. Recall that $\mathrm{OR}^{(k)}$ is the problem of deciding whether there are no marked vertices or exactly k of them.

Theorem 120 *For all constants $d \geq 2$,*

$$Q\left(\mathrm{OR}^{(k)}, \mathcal{L}_d\right) = \Omega\left(\frac{\sqrt{n}}{k^{1/2-1/d}}\right).$$

Proof. For simplicity, we assume that both $k^{1/d}$ and $\left(n/3^d k\right)^{1/d}$ are integers. (In the general case, we can just replace k by $\left\lceil k^{1/d}\right\rceil^d$ and n by the largest number of the form $\left(3m\left\lceil k^{1/d}\right\rceil\right)^d$ which is less than n. This only changes the lower bound by a lower order term.)

We use a hybrid argument almost identical to that of Theorem 107. Divide $\mathcal{L}_d$ into n/k subcubes, each having k vertices and side length $k^{1/d}$. Let S be a regularly-spaced

set of $M = n/\left(3^d k\right)$ of these subcubes, so that any two subcubes in S have distance at least $2k^{1/d}$ from one another. Then choose a subcube $C_j \in S$ uniformly at random and mark all k vertices in C_j. This enables us to consider each $C_j \in S$ itself as a *single* vertex (out of M in total), having distance at least $2k^{1/d}$ to every other vertex.

More formally, given a subcube $C_j \in S$, let $\widetilde{C}_j$ be the set of vertices consisting of C_j and the $3^d - 1$ subcubes surrounding it. (Thus, $\widetilde{C}_j$ is a subcube of side length $3k^{1/d}$.) Then the query magnitude of $\widetilde{C}_j$ after the t^{th} query is

$$\Gamma_j^{(t)} = \sum_{v_i \in \widetilde{C}_j} \sum_z \left|\alpha_{i,z}^{(t)}\left(X_0\right)\right|^2,$$

where X_0 is the all-zero input. Let T be the number of queries, and let $w = T/\left(ck^{1/d}\right)$ for some constant $c > 0$. Then as in Theorem 107, there must exist a subcube $\widetilde{C}_{j^*}$ such that

$$\sum_{q=0}^{w-1} \Gamma_{j^*}^{\left(T-qck^{1/d}\right)} \leq \frac{w}{M} = \frac{3^d k w}{n}.$$

Let Y be the input which is 1 in C_{j^*} and 0 elsewhere; then let X_q be a hybrid input which is X_0 during queries 1 to $T - qck^{1/d}$, but Y during queries $T - qck^{1/d} + 1$ to T. Next let

$$D\left(q, r\right) = \sum_{v_i \in G} \sum_z \left|\alpha_{i,z}^{(T)}\left(X_q\right) - \alpha_{i,z}^{(T)}\left(X_r\right)\right|^2.$$

Then as in Theorem 107, for all $c < 1$ we have $D\left(q-1, q\right) \leq 4\Gamma_{j^*}^{\left(T-qck^{1/d}\right)}$. For in the $ck^{1/d}$ queries from $T - qck^{1/d} + 1$ through $T - \left(q-1\right)ck^{1/d}$, no amplitude originating outside $\widetilde{C}_{j^*}$ can travel a distance $k^{1/d}$ and thereby reach C_{j^*}. Therefore switching from X_{q-1} to X_q can only affect amplitude that is in $\widetilde{C}_{j^*}$ immediately after query $T - qck^{1/d}$. It follows that

$$\sqrt{D\left(0, w\right)} \leq \sum_{q=1}^{w} \sqrt{D\left(q-1, q\right)} \leq 2\sum_{q=1}^{w} \sqrt{\Gamma_{j^*}^{\left(T-qck^{1/d}\right)}} \leq 2w\sqrt{\frac{3^d k}{n}} = \frac{2\sqrt{3^d}k^{1/2-1/d}T}{c\sqrt{n}}.$$

Hence $T = \Omega\left(\sqrt{n}/k^{1/2-1/d}\right)$ for constant d, since assuming the algorithm is correct we need $D\left(0, w\right) = \Omega\left(1\right)$. ■

Notice that if $k \approx n$, then the bound of Theorem 120 becomes $\Omega\left(n^{1/d}\right)$ which is just the diameter of $\mathcal{L}_d$. Also, if $d = 2$, then $1/2 - 1/d = 0$ and the bound is simply $\Omega\left(\sqrt{n}\right)$ independent of k. The bound of Theorem 120 can be achieved (up to a constant factor that depends on d) for $d \geq 3$, and nearly achieved for $d = 2$. We first construct an algorithm for the case when k is known.

Theorem 121

(i) For $d \geq 3$,

$$Q\left(\mathrm{OR}^{(k)}, \mathcal{L}_d\right) = O\left(\frac{\sqrt{n}}{k^{1/2-1/d}}\right).$$

(ii) For $d = 2$,

$$Q\left(\mathrm{OR}^{(k)}, \mathcal{L}_2\right) = O\left(\sqrt{n}\log^{3/2} n\right).$$

To prove Theorem 121, we first divide $\mathcal{L}_d(n)$ into n/γ subcubes, each of size $\gamma^{1/d} \times \cdots \times \gamma^{1/d}$ (where γ will be fixed later). Then in each subcube, we choose one vertex uniformly at random.

Lemma 122 *If $\gamma \geq k$, then the probability that exactly one marked vertex is chosen is at least $k/\gamma - (k/\gamma)^2$.*

Proof. Let x be a marked vertex. The probability that x is chosen is $1/\gamma$. Given that x is chosen, the probability that one of the other marked vertices, y, is chosen is 0 if x and y belong to the same subcube, or $1/\gamma$ if they belong to different subcubes. Therefore, the probability that x alone is chosen is at least

$$\frac{1}{\gamma}\left(1 - \frac{k-1}{\gamma}\right) \geq \frac{1}{\gamma}\left(1 - \frac{k}{\gamma}\right).$$

Since the events "x alone is chosen" are mutually disjoint, we conclude that the probability that exactly one marked vertex is chosen is at least $k/\gamma - (k/\gamma)^2$. ■

In particular, fix γ so that $\gamma/3 < k < 2\gamma/3$; then Lemma 122 implies that the probability of choosing exactly one marked vertex is at least $2/9$. The algorithm is now as follows. As in the lemma, subdivide $\mathcal{L}_d(n)$ into n/γ subcubes and choose one location at random from each. Then run the algorithm for the unique-solution case (Theorem 111 or 117) on the chosen locations only, as if they were vertices of $\mathcal{L}_d(n/\gamma)$.

The running time in the unique case was $O\left(\sqrt{n/\gamma}\right)$ for $d \geq 3$ or

$$O\left(\sqrt{\frac{n}{\gamma}}\log^{3/2}(n/\gamma)\right) = O\left(\sqrt{\frac{n}{\gamma}}\log^{3/2} n\right)$$

for $d = 2$. However, each local unitary in the original algorithm now becomes a unitary affecting two vertices v and w in neighboring subcubes C_v and C_w. When placed side by side, C_v and C_w form a rectangular box of size $2\gamma^{1/d} \times \gamma^{1/d} \times \cdots \times \gamma^{1/d}$. Therefore the distance between v and w is at most $(d+1)\gamma^{1/d}$. It follows that each local unitary in the original algorithm takes $O\left(d\gamma^{1/d}\right)$ time in the new algorithm. For $d \geq 3$, this results in an overall running time of

$$O\left(\sqrt{\frac{n}{\gamma}}d\gamma^{1/d}\right) = O\left(d\frac{\sqrt{n}}{\gamma^{1/2-1/d}}\right) = O\left(\frac{\sqrt{n}}{k^{1/2-1/d}}\right).$$

For $d = 2$ we obtain

$$O\left(\sqrt{\frac{n}{\gamma}}\gamma^{1/2}\log^{3/2} n\right) = O\left(\sqrt{n}\log^{3/2} n\right).$$

14.6.5 Unknown Number of Marked Items

We now show how to deal with an unknown k. Let $\mathrm{OR}^{(\geq k)}$ be the problem of deciding whether there are no marked vertices or *at least* k of them, given that one of these is true.

Theorem 123

(i) For $d \geq 3$,

$$Q\left(\mathrm{OR}^{(\geq k)}, \mathcal{L}_d\right) = O\left(\frac{\sqrt{n}}{k^{1/2-1/d}}\right).$$

(ii) For $d = 2$,

$$Q\left(\mathrm{OR}^{(\geq k)}, \mathcal{L}_2\right) = O\left(\sqrt{n}\log^{5/2} n\right).$$

Proof. We use the straightforward 'doubling' approach of Boyer et al. [66]:

(1) For $j = 0$ to $\log_2(n/k)$

- Run the algorithm of Theorem 121 with subcubes of size $\gamma_j = 2^j k$.
- If a marked vertex is found, then output 1 and halt.

(2) Query a random vertex v, and output 1 if v is a marked vertex and 0 otherwise.

Let $k^* \geq k$ be the number of marked vertices. If $k^* \leq n/3$, then there exists a $j \leq \log_2(n/k)$ such that $\gamma_j/3 \leq k^* \leq 2\gamma_j/3$. So Lemma 122 implies that the j^{th} iteration of step (1) finds a marked vertex with probability at least $2/9$. On the other hand, if $k^* \geq n/3$, then step (2) finds a marked vertex with probability at least $1/3$. For $d \geq 3$, the time used in step (1) is at most

$$\sum_{j=0}^{\log_2(n/k)} \frac{\sqrt{n}}{\gamma_j^{1/2-1/d}} = \frac{\sqrt{n}}{k^{1/2-1/d}} \left[\sum_{j=0}^{\log_2(n/k)} \frac{1}{2^{j(1/2-1/d)}}\right] = O\left(\frac{\sqrt{n}}{k^{1/2-1/d}}\right),$$

the sum in brackets being a decreasing geometric series. For $d = 2$, the time is $O\left(\sqrt{n}\log^{5/2} n\right)$, since each iteration takes $O\left(\sqrt{n}\log^{3/2} n\right)$ time and there are at most $\log n$ iterations. In neither case does step (2) affect the bound, since $k \leq n$ implies that $n^{1/d} \leq \sqrt{n}/k^{1/2-1/d}$. ■

Taking $k = 1$ gives algorithms for unconstrained OR with running times $O(\sqrt{n})$ for $d \geq 3$ and $O(\sqrt{n}\log^{5/2} n)$ for $d = 2$, thereby establishing Theorems 110 and 116.

14.7 Search on Irregular Graphs

In Section 14.2, we claimed that our divide-and-conquer approach has the advantage of being *robust*: it works not only for highly symmetric graphs such as hypercubes, but for any graphs having comparable expansion properties. Let us now substantiate this claim.

Say a family of connected graphs $\{G_n = (V_n, E_n)\}$ is *d-dimensional* if there exists a $\kappa > 0$ such that for all n, ℓ and $v \in V_n$,

$$|B(v,\ell)| \geq \min\left(\kappa \ell^d, n\right),$$

where $B(v,\ell)$ is the set of vertices having distance at most ℓ from v in G_n. Intuitively, G_n is d-dimensional (for $d \geq 2$ an integer) if its expansion properties are at least as good as those of the hypercube $\mathcal{L}_d(n)$.[5] It is immediate that the diameter of G_n is at most $(n/\kappa)^{1/d}$. Note, though, that G_n might not be an expander graph in the usual sense, since we have not required that every sufficiently small *set* of vertices has many neighbors.

Our goal is to show the following.

Theorem 124 *If G is d-dimensional, then*

(i) For a constant $d > 2$,

$$Q(\mathrm{OR}, G) = O\left(\sqrt{n}\,\mathrm{polylog}\, n\right).$$

(ii) For $d = 2$,

$$Q(\mathrm{OR}, G) = \sqrt{n} 2^{O\left(\sqrt{\log n}\right)}.$$

In proving part (i), the intuition is simple: we want to decompose G recursively into subgraphs (called *clusters*), which will serve the same role as subcubes did in the hypercube case. The procedure is as follows. For some constant $n_1 > 1$, first choose $\lceil n/n_1 \rceil$ vertices uniformly at random to be designated as 1-*pegs*. Then form 1-*clusters* by assigning each vertex in G to its closest 1-peg, as in a Voronoi diagram. (Ties are broken randomly.) Let $v(C)$ be the peg of cluster C. Next, split up any 1-cluster C with more than n_1 vertices into $\lceil |C| / n_1 \rceil$ arbitrarily-chosen 1-clusters, each with size at most n_1 and with $v(C)$ as its 1-peg. Observe that

$$\sum_{i=1}^{\lceil n/n_1 \rceil} \left\lceil \frac{|C_i|}{n_1} \right\rceil \leq 2 \left\lceil \frac{n}{n_1} \right\rceil,$$

where $n = |C_1| + \cdots + \left|C_{\lceil n/n_1 \rceil}\right|$. Therefore, the splitting-up step can at most double the number of clusters.

In the next iteration, set $n_2 = n_1^{1/\beta}$, for some constant $\beta \in (2/d, 1)$. Choose $2\lceil n/n_2 \rceil$ vertices uniformly at random as 2-pegs. Then form 2-clusters by assigning each 1-cluster C to the 2-peg that is closest to the 1-peg $v(C)$. Given a 2-cluster C', let $|C'|$ be the number of 1-clusters in C'. Then as before, split up any C' with $|C'| > n_2/n_1$ into $\lceil |C'| / (n_2/n_1) \rceil$ arbitrarily-chosen 2-clusters, each with size at most n_2/n_1 and with

[5] In general, it makes sense to consider non-integer d as well.

$v(C')$ as its 2-peg. Continue recursively in this manner, setting $n_R = n_{R-1}^{1/\beta}$ and choosing $2^{R-1}\lceil n/n_R \rceil$ vertices as R-pegs for each R. Stop at the maximum R such that $n_R \leq n$. For technical convenience, set $n_0 = 1$, and consider each vertex v to be the 0-peg of the 0-cluster $\{v\}$.

At the end we have a tree of clusters, which can be searched recursively just as in the hypercube case. In more detail, basis states now have the form $|v, z, C\rangle$, where v is a vertex, z is an answer bit, and C is the (label of the) cluster currently being searched. (Unfortunately, because multiple R-clusters can have the same peg, a single auxiliary qubit no longer suffices.) Also, let $K'(C)$ be the number of $(R-1)$-clusters in R-cluster C; then $K'(C) \leq K(R)$ where $K(R) = 2\lceil n_R/n_{R-1} \rceil$. If $K'(C) < K(R)$, then place $K(R) - K'(C)$ "dummy" $(R-1)$-clusters in C, each of which has $(R-1)$-peg $v(C)$.

The algorithm $\mathcal{A}_R$ from Section 14.6.2 now does the following, when invoked on the initial state $|v(C), 0, C\rangle$, where C is an R-cluster. If $R = 0$, then $\mathcal{A}_R$ uses a query transformation to prepare the state $|v(C), 1, C\rangle$ if $v(C)$ is the marked vertex and $|v(C), 0, C\rangle$ otherwise. If $R \geq 1$ and C is not a dummy cluster, then $\mathcal{A}_R$ performs m_R steps of amplitude amplification on A_R, where m_R is the largest integer such that $2m_R + 1 \leq \sqrt{n_R/n_{R-1}}$.[6] If C is a dummy cluster, then $\mathcal{A}_R$ does nothing for an appropriate number of steps, and then returns that no marked item was found.

We now describe the subroutine A_R, for $R \geq 1$. When invoked with $|v(C), 0, C\rangle$ as its initial state, A_R first prepares a uniform superposition

$$\frac{1}{\sqrt{K(R)}} \sum_{i=1}^{K(R)} |v(C_i), 0, C_i\rangle .$$

It then calls $\mathcal{A}_{R-1}$ recursively, to search $C_1, \ldots, C_{K(R)}$ in superposition and amplify the results.

For $R \geq 1$, define the *radius* of an R-cluster C to be the maximum, over all $(R-1)$-clusters C' in C, of the distance from $v(C)$ to $v(C')$. Also, call an R-cluster *good* if it has radius at most ℓ_R, where $\ell_R = \left(\frac{2}{\kappa} n_R \ln n\right)^{1/d}$.

Lemma 125 *With probability $1 - o(1)$ over the choice of clusters, all clusters are good.*

Proof. Let v be the $(R-1)$-peg of an $(R-1)$-cluster. Then $|B(v, \ell)| \geq \kappa \ell^d$, where $B(v, \ell)$ is the ball of radius ℓ about v. So the probability that v has distance greater than ℓ_R to the nearest R-peg is at most

$$\left(1 - \frac{\kappa \ell_R^d}{n}\right)^{\lceil n/n_R \rceil} \leq \left(1 - \frac{2 \ln n}{n/n_R}\right)^{n/n_R} < \frac{1}{n^2}.$$

Furthermore, the total number of pegs is easily seen to be $O(n)$. It follows by the union bound that *every* $(R-1)$-peg for *every* R has distance at most ℓ_R to the nearest R-peg, with probability $1 - O(1/n) = 1 - o(1)$ over the choice of clusters. ■

We now analyze the running time and success probability of $\mathcal{A}_R$.

[6] In the hypercube case, we performed fewer amplifications in order to lower the running time from $\sqrt{n} \operatorname{polylog} n$ to $\sqrt{n}$. Here, though, the splitting-up step produces a $\operatorname{polylog} n$ factor anyway.

Lemma 126 *$\mathcal{A}_R$ uses $O\left(\sqrt{n_R}\log^{1/d} n\right)$ steps, assuming that all clusters are good.*

Proof. Let $T_{\mathcal{A}}(R)$ and $T_A(R)$ be the time used by $\mathcal{A}_R$ and A_R respectively in searching an R-cluster. Then we have

$$\begin{aligned} T_{\mathcal{A}}(R) &\leq \sqrt{n_R/n_{R-1}}T_A(R), \\ T_A(R) &\leq \ell_R + T_{\mathcal{A}}(R-1) \end{aligned}$$

with the base case $T_{\mathcal{A}}(0) = 1$. Combining,

$$\begin{aligned} T_{\mathcal{A}}(R) &\leq \sqrt{n_R/n_{R-1}}\left(\ell_R + T_{\mathcal{A}}(R-1)\right) \\ &\leq \sqrt{n_R/n_{R-1}}\ell_R + \sqrt{n_R/n_{R-2}}\ell_{R-1} + \cdots + \sqrt{n_R/n_0}\ell_1 \\ &= \sqrt{n_R}\cdot O\left(\frac{(n_R\ln n)^{1/d}}{\sqrt{n_{R-1}}} + \cdots + \frac{(n_1\ln n)^{1/d}}{\sqrt{n_0}}\right) \\ &= \sqrt{n_R}\left(\ln^{1/d} n\right)\cdot O\left(n_R^{1/d-\beta/2} + \cdots + n_1^{1/d-\beta/2}\right) \\ &= \sqrt{n_R}\left(\ln^{1/d} n\right)\cdot O\left(n_1^{1/d-\beta/2} + \left(n_1^{1/d-\beta/2}\right)^{1/\beta} + \cdots + \left(n_1^{1/d-\beta/2}\right)^{(1/\beta)^{R-1}}\right) \\ &= O\left(\sqrt{n_R}\log^{1/d} n\right), \end{aligned}$$

where the last line holds because $\beta > 2/d$ and therefore $n_1^{1/d-\beta/2} < 1$. ■

Lemma 127 *$\mathcal{A}_R$ succeeds with probability $\Omega(1/\operatorname{polylog} n_R)$ in searching a graph of size $n = n_R$, assuming there is a unique marked vertex.*

Proof. For all $R \geq 0$, let C_R be the R-cluster that contains the marked vertex, and let $P_{\mathcal{A}}(R)$ and $P_A(R)$ be the success probabilities of $\mathcal{A}_R$ and A_R respectively when searching C_R. Then for all $R \geq 1$, we have $P_A(R) = P_{\mathcal{A}}(R-1)/(2K(R))$, and therefore

$$\begin{aligned} P_{\mathcal{A}}(R) &\geq \left(1 - \frac{(2m_R+1)^2}{3}P_A(R)\right)(2m_R+1)^2 P_A(R) \\ &= \left(1 - \frac{(2m_R+1)^2}{3}\cdot\frac{P_{\mathcal{A}}(R-1)}{2K(R)}\right)(2m_R+1)^2\frac{P_{\mathcal{A}}(R-1)}{2K(R)} \\ &= \Omega\left(P_{\mathcal{A}}(R-1)\right) \\ &= \Omega\left(1/\operatorname{polylog} n_R\right). \end{aligned}$$

Here the third line holds because $(2m_R+1)^2 \approx n_R/n_{R-1} \approx K(R)/2$, and the last line because $R = \Theta(\log\log n_R)$. ■

Finally, we repeat $\mathcal{A}_R$ itself $O(\operatorname{polylog} n_R)$ times, to achieve success probability $\Omega(1)$ using $O\left(\sqrt{n_R}\operatorname{polylog} n_R\right)$ steps in total. Again, if n is not equal to n_R for any R, then we simply find the largest R such that $n_R < n$, and then add one more level of recursion that searches a random R-cluster and amplifies the result $\Theta\left(\sqrt{n/n_R}\right)$ times. The

resulting algorithm uses $O(\sqrt{n}\operatorname{polylog} n)$ steps, thereby establishing part (i) of Theorem 124 for the case of a unique marked vertex. The generalization to multiple marked vertices is straightforward.

Corollary 128 *If G is d-dimensional for a constant $d > 2$, then*

$$Q\left(\mathrm{OR}^{(\geq k)}, G\right) = O\left(\frac{\sqrt{n}\operatorname{polylog}\frac{n}{k}}{k^{1/2-1/d}}\right).$$

Proof. Assume without loss of generality that $k = o(n)$, since otherwise a marked item is trivially found in $O\left(n^{1/d}\right)$ steps. As in Theorem 123, we give an algorithm $\mathcal{B}$ consisting of $\log_2(n/k) + 1$ iterations. In iteration $j = 0$, choose $\lceil n/k \rceil$ vertices $w_1, \ldots, w_{\lceil n/k \rceil}$ uniformly at random. Then run the algorithm for the unique marked vertex case, but instead of taking all vertices in G as 0-pegs, take only $w_1, \ldots, w_{\lceil n/k \rceil}$. On the other hand, still choose the 1-pegs, 2-pegs, and so on uniformly at random from among all vertices in G. For all R, the number of R-pegs should be $\lceil (n/k)/n_R \rceil$. In general, in iteration j of $\mathcal{B}$, choose $\left\lceil n/\left(2^j k\right)\right\rceil$ vertices $w_1, \ldots, w_{\lceil n/(2^j k) \rceil}$ uniformly at random, and then run the algorithm for a unique marked vertex as if $w_1, \ldots, w_{\lceil n/(2^j k) \rceil}$ were the only vertices in the graph.

It is easy to see that, assuming there are k or more marked vertices, with probability $\Omega(1)$ there exists an iteration j such that exactly one of $w_1, \ldots, w_{\lceil n/(2^j k) \rceil}$ is marked. Hence $\mathcal{B}$ succeeds with probability $\Omega(1)$. It remains only to upper-bound $\mathcal{B}$'s running time.

In iteration j, notice that Lemma 125 goes through if we use $\ell_R^{(j)} := \left(\frac{2}{\kappa} 2^j k n_R \ln \frac{n}{k}\right)^{1/d}$ instead of ℓ_R. That is, with probability $1 - O(k/n) = 1 - o(1)$ over the choice of clusters, every R-cluster has radius at most $\ell_R^{(j)}$. So letting $T_{\mathcal{A}}(R)$ be the running time of $\mathcal{A}_R$ on an R-cluster, the recurrence in Lemma 126 becomes

$$T_{\mathcal{A}}(R) \leq \sqrt{n_R/n_{R-1}}\left(\ell_R^{(j)} + T_{\mathcal{A}}(R-1)\right) = O\left(\sqrt{n_R}\left(2^j k \log(n/k)\right)^{1/d}\right),$$

which is

$$O\left(\frac{\sqrt{n}\log^{1/d}\frac{n}{k}}{\left(2^j k\right)^{1/2-1/d}}\right)$$

if $n_R = \Theta\left(n/\left(2^j k\right)\right)$. As usual, the case where there is no R such that $n_R = \Theta\left(n/\left(2^j k\right)\right)$ is trivially handled by adding one more level of recursion. If we factor in the $O(1/\operatorname{polylog} n_R)$ repetitions of $\mathcal{A}_R$ needed to boost the success probability to $\Omega(1)$, then the total running time of iteration j is

$$O\left(\frac{\sqrt{n}\operatorname{polylog}\frac{n}{k}}{\left(2^j k\right)^{1/2-1/d}}\right).$$

Therefore $\mathcal{B}$'s running time is

$$O\left(\sum_{j=0}^{\log_2(n/k)} \frac{\sqrt{n}\operatorname{polylog} n}{\left(2^j k\right)^{1/2-1/d}}\right) = O\left(\frac{\sqrt{n}\operatorname{polylog} n}{k^{1/2-1/d}}\right).$$

■

For the $d = 2$ case, the best upper bound we can show is $\sqrt{n}2^{O(\sqrt{\log n})}$. This is obtained by simply modifying $\mathcal{A}_R$ to have a deeper recursion tree. Instead of taking $n_R = n_{R-1}^{1/\mu}$ for some μ, we take $n_R = 2^{\sqrt{\log n}} n_{R-1} = 2^{R\sqrt{\log n}}$, so that the total number of levels is $\lceil \sqrt{\log n} \rceil$. Lemma 125 goes through without modification, while the recurrence for the running time becomes

$$\begin{aligned} T_{\mathcal{A}}(R) &\leq \sqrt{n_R/n_{R-1}}\,(\ell_R + T_{\mathcal{A}}(R-1)) \\ &\leq \sqrt{n_R/n_{R-1}}\ell_R + \sqrt{n_R/n_{R-2}}\ell_{R-1} + \cdots + \sqrt{n_R/n_0}\ell_1 \\ &= O\left(2^{\sqrt{\log n}(R/2)}\sqrt{\ln n} + \cdots + 2^{\sqrt{\log n}(R/2)}\sqrt{\ln n}\right) \\ &= \sqrt{n}2^{O(\sqrt{\log n})}. \end{aligned}$$

Also, since the success probability decreases by at most a constant factor at each level, we have that $P_{\mathcal{A}}(R) = 2^{-O(\sqrt{\log n})}$, and hence $2^{O(\sqrt{\log n})}$ amplification steps suffice to boost the success probability to $\Omega(1)$. Handling multiple marked items adds an additional factor of $\log n$, which is absorbed into $2^{O(\sqrt{\log n})}$. This completes Theorem 124.

14.7.1 Bits Scattered on a Graph

In Section 14.3, we discussed several ways to pack a given amount of entropy into a spatial region of given dimensions. However, we said nothing about how the entropy is *distributed* within the region. It might be uniform, or concentrated on the boundary, or distributed in some other way. So we need to answer the following: suppose that in some graph, h out of the n vertices *might* be marked, and we know which h those are. Then how much time is needed to determine whether any of the h *is* marked? If the graph is the hypercube $\mathcal{L}_d$ for $d \geq 2$ or is d-dimensional for $d > 2$, then the results of the previous sections imply that $O(\sqrt{n}\operatorname{polylog} n)$ steps suffice. However, we wish to use fewer steps, taking advantage of the fact that h might be much smaller than n. Formally, suppose we are given a graph G with n vertices, of which h are potentially marked. Let $\mathrm{OR}^{(h,\geq k)}$ be the problem of deciding whether G has no marked vertices or at least k of them, given that one of these is the case.

Proposition 129 *For all integer constants $d \geq 2$, there exists a d-dimensional graph G such that*

$$Q\left(\mathrm{OR}^{(h,\geq k)}, G\right) = \Omega\left(n^{1/d}\left(\frac{h}{k}\right)^{1/2-1/d}\right).$$

Proof. Let G be the d-dimensional hypercube $\mathcal{L}_d(n)$. Create h/k subcubes of potentially marked vertices, each having k vertices and side length $k^{1/d}$. Space these subcubes out in $\mathcal{L}_d(n)$ so that the distance between any pair of them is $\Omega\left((nk/h)^{1/d}\right)$. Then choose a subcube C uniformly at random and mark all k vertices in C. This enables us to consider each subcube as a single vertex, having distance $\Omega\left((nk/h)^{1/d}\right)$ to every other vertex. The lower bound now follows by a hybrid argument essentially identical to that of Theorem 120. ■

In particular, if $d = 2$ then $\Omega(\sqrt{n})$ time is always needed, since the potentially marked vertices might all be far from the start vertex. The lower bound of Proposition 129 can be achieved up to a polylogarithmic factor.

Proposition 130 *If G is d-dimensional for a constant $d > 2$, then*

$$Q\left(\mathrm{OR}^{(h,\geq k)}, G\right) = O\left(n^{1/d}\left(\frac{h}{k}\right)^{1/2-1/d} \mathrm{polylog}\,\frac{h}{k}\right).$$

Proof. Assume without loss of generality that $k = o(h)$, since otherwise a marked item is trivially found. Use algorithm $\mathcal{B}$ from Corollary 128, with the following simple change. In iteration j, choose $\lceil h/\left(2^j k\right)\rceil$ potentially marked vertices $w_1, \ldots, w_{\lceil h/(2^j k)\rceil}$ uniformly at random, and then run the algorithm for a unique marked vertex as if $w_1, \ldots, w_{\lceil h/(2^j k)\rceil}$ were the only vertices in the graph. That is, take $w_1, \ldots, w_{\lceil h/(2^j k)\rceil}$ as 0-pegs; then for all $R \geq 1$, choose $\lceil h/\left(2^j k n_R\right)\rceil$ vertices of G uniformly at random as R-pegs. Lemma 125 goes through if we use $\widehat{\ell}_R^{(j)} := \left(\frac{2}{\kappa}\frac{n}{h}2^j k n_R \ln\frac{h}{k}\right)^{1/d}$ instead of ℓ_R. So following Corollary 128, the running time of iteration j is now

$$O\left(\sqrt{n_R}\left(\frac{n}{h}2^j k\right)^{1/d} \mathrm{polylog}\,\frac{h}{k}\right) = O\left(n^{1/d}\left(\frac{h}{2^j k}\right)^{1/2-1/d} \mathrm{polylog}\,\frac{h}{k}\right)$$

if $n_R = \Theta\left(h/\left(2^j k\right)\right)$. Therefore the total running time is

$$O\left(\sum_{j=0}^{\log_2(h/k)} n^{1/d}\left(\frac{h}{2^j k}\right)^{1/2-1/d} \mathrm{polylog}\,\frac{h}{k}\right) = O\left(n^{1/d}\left(\frac{h}{k}\right)^{1/2-1/d} \mathrm{polylog}\,\frac{h}{k}\right).$$

■

Intuitively, Proposition 130 says that the worst case for search occurs when the h potential marked vertices are scattered evenly throughout the graph.

14.8 Application to Disjointness

In this section we show how our results can be used to strengthen a seemingly unrelated result in quantum computing. Suppose Alice has a string $X = x_1 \ldots x_n \in \{0,1\}^n$, and Bob has a string $Y = y_1 \ldots y_n \in \{0,1\}^n$. In the *disjointness problem*, Alice and Bob must decide with high probability whether there exists an i such that $x_i = y_i = 1$, using as few bits of communication as possible. Buhrman, Cleve, and Wigderson [76] observed that in the quantum setting, Alice and Bob can solve this problem using only $O(\sqrt{n}\log n)$ qubits of communication. This was subsequently improved by Høyer and de Wolf [146] to $O\left(\sqrt{n}c^{\log^* n}\right)$, where c is a constant and $\log^* n$ is the iterated logarithm function. Using the search algorithm of Theorem 110, we can improve this to $O(\sqrt{n})$, which matches the celebrated $\Omega(\sqrt{n})$ lower bound of Razborov [199].

Theorem 131 *The bounded-error quantum communication complexity of the disjointness problem is $O(\sqrt{n})$.*

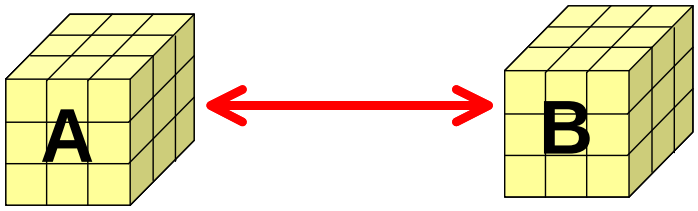

Figure 14.3: Alice and Bob 'synchronize' locations on their respective cubes.

Proof. The protocol is as follows. Alice and Bob both store their inputs in a 3-D cube $\mathcal{L}_3(n)$ (Figure 14.3); that is, they let $x_{jkl} = x_i$ and $y_{jkl} = y_i$, where $i = n^{2/3}j + n^{1/3}k + l + 1$ and $j, k, l \in \{0, \ldots, n^{1/3} - 1\}$. Throughout, they maintain a joint state of the form

$$\sum \alpha_{j,k,l,z_A,z_B,c} |v_{jkl}, z_A\rangle \otimes |c\rangle \otimes |v_{jkl}, z_B\rangle, \tag{14.1}$$

where c is used for communication between the players, and z_A and z_B store the answers to queries. Thus, whenever Alice is at location (j, k, l) of her cube, Bob is at location (j, k, l) of his cube. To decide whether there exists a (j, k, l) with $x_{jkl} = y_{jkl} = 1$, Alice simply runs our search algorithm for an unknown number of marked items, but with two changes. First, after each query, Alice inverts her phase if and only if $x_{jkl} = y_{jkl} = 1$; this requires 2 qubits of communication from Bob, to send y_{jkl} to Alice and then to erase it. Second, before each movement step, Alice tells Bob in which of the six possible directions she is going to move. That way, Bob can synchronize his location with Alice's, and thereby maintain the state in the form (14.1). This requires 6 qubits of communication from Alice, to send the direction to Bob and then to erase it. Notice that no further communication is necessary, since there are no auxiliary registers in our algorithm that need to be communicated. Since the algorithm uses $O(\sqrt{n})$ steps, the number of qubits communicated in the disjointness protocol is therefore also $O(\sqrt{n})$. ■

14.9 Open Problems

As discussed in Section 14.4.1, a salient open problem raised by this work is to prove relationships among Z-local, C-local, and H-local unitary matrices. In particular, can any Z-local or H-local unitary be approximated by a product of a small number of C-local unitaries? Also, is it true that $Q(f, G) = \Theta\left(Q^Z(f, G)\right) = \Theta\left(Q^H(f, G)\right)$ for all f, G?

A second problem is to obtain interesting lower bounds in our model. For example, let G be a $\sqrt{n} \times \sqrt{n}$ grid, and suppose $f(X) = 1$ if and only if every row of G contains a vertex v_i with $x_i = 1$. Clearly $Q(f, G) = O\left(n^{3/4}\right)$, and we conjecture that this is optimal. However, we were unable to show any lower bound better than $\Omega(\sqrt{n})$.

Finally, what is the complexity of finding a unique marked vertex on a 2-D square grid? As mentioned in Section 14.2, Ambainis, Kempe, and Rivosh [31] showed that $Q\left(\mathrm{OR}^{(1)}, \mathcal{L}_2\right) = O(\sqrt{n} \log n)$. Can the remaining factor of $\log n$ be removed?

Chapter 15

Quantum Computing and Postselection

> "Gill, in his seminal paper on probabilistic complexity classes, defined the class PP and asked whether the class was closed under intersection. In 1990, Fenner and Kurtz and later myself, decided to try a new approach to the question: Consider a class defined like PP but with additional restrictions, show that this class is closed under intersection and then show the class was really the same as PP."
>
> —Lance Fortnow, *My Computational Complexity Web Log* [113]
>
> (the approach didn't succeed, though as this chapter will show, all it was missing was quantum mechanics)

Postselection is the power of discarding all runs of a computation in which a given event does not occur. Clearly, such an ability would let us solve NP-complete problems in polynomial time, since we could guess a random solution, and then postselect on its being correct. But would postselection let us do *more* than NP? Using a classical computer, the class of problems we could efficiently solve coincides with a class called $\mathsf{BPP}_{\mathsf{path}}$, which was defined by Han, Hemaspaandra, and Thierauf [142] and which sits somewhere between MA and PP.

This chapter studies the power of postselection when combined with quantum computing. In Section 15.1, I define a new complexity class called PostBQP, which is similar to BQP except that we can measure a qubit that has some nonzero probability of being $|1\rangle$, and *assume* the outcome will be $|1\rangle$. The main result is that PostBQP equals the classical complexity class PP.

My original motivation, which I explain in Section 15.2, was to analyze the computational power of "fantasy" versions of quantum mechanics, and thereby gain insight into why quantum mechanics is the way it is. For example, I show in Section 15.2 that if we changed the measurement probability rule from $|\psi|^2$ to $|\psi|^p$ for some $p \neq 2$, or allowed linear but nonunitary gates, then we could simulate postselection, and hence solve PP-complete problems in polynomial time. I was also motivated by a concept that I call *anthropic computing*: arranging things so that you're more likely to exist if a computer produces a desired

output than if it doesn't. As a simple example, under the many-worlds interpretation of quantum mechanics, you might kill yourself in all universes where a computer's output is incorrect. My result implies that, using this "technique," the class of problems that you could efficiently solve is exactly PP.

However, it recently dawned on me that the PostBQP = PP result is also interesting for purely classical reasons. In particular, it yields an almost-trivial, quantum computing based proof that PP is closed under intersection. This proof does not use rational approximations, threshold polynomials, or any of the other techniques pioneered by Beigel, Reingold, and Spielman [47] in their justly-celebrated original proof. Another immediate corollary of my new characterization of PP is a result originally due to Fortnow and Reingold [115]: that PP is closed under polynomial-time truth-table reductions. Indeed, I can show that PP is closed under BQP truth-table reductions, which is a new result as far as I know. I conclude in Section 15.3 with some open problems.

15.1 The Class PostBQP

I hereby define a complexity class:

Definition 132 PostBQP *(Postselected Bounded-Error Quantum Polynomial-Time) is the class of languages L for which there exists a uniform family of polynomial-size quantum circuits such that for all inputs x,*

(i) At the end of the computation, the first qubit has a nonzero probability of being measured to be $|1\rangle$.

(ii) If $x \in L$, then conditioned on the first qubit being $|1\rangle$, the second qubit is $|1\rangle$ with probability at least $2/3$.

(iii) If $x \notin L$, then conditioned on the first qubit being $|1\rangle$, the second qubit is $|1\rangle$ with probability at most $1/3$.

We can think of PostBQP as the "nondeterministic" version of BQP. Admittedly, there are already three other contenders for that title: QMA, defined by Watrous [239]; QCMA, defined by Aharonov and Naveh [21]; and NQP, defined by Adleman, DeMarrais, and Huang [16] (which turns out to equal coC$_=$P [107]). As we will see, PostBQP contains all of these as subclasses.

It is immediate that NP $\subseteq$ PostBQP $\subseteq$ PP. For the latter inclusion, we can use the same techniques used by Adleman, DeMarrais, and Huang [16] to show that BQP $\subseteq$ PP, but sum only over paths where the first qubit is $|1\rangle$ at the end. This is made explicit in Theorem 57 of Chapter 10.

How robust is PostBQP? Just as Bernstein and Vazirani [55] showed that intermediate measurements don't increase the power of ordinary quantum computers, so it's easy to show that intermediate postselection steps don't increase the power of PostBQP. Whenever we want to postselect on a qubit being $|1\rangle$, we simply CNOT that qubit into a fresh ancilla qubit that is initialized to $|0\rangle$ and that will never be written to again. Then, at the end, we compute the AND of all the ancilla qubits, and swap the result into the

first qubit. It follows that we can repeat a PostBQP computation a polynomial number of times, and thereby amplify the probability gap from $(1/3, 2/3)$ to $\left(2^{-p(n)}, 1-2^{-p(n)}\right)$ for any polynomial p.

A corollary of the above observations is that PostBQP has strong closure properties.

Proposition 133 PostBQP *is closed under union, intersection, and complement. Indeed, it is closed under* BQP *truth table reductions, meaning that* $\mathsf{PostBQP} = \mathsf{BQP}^{\mathsf{PostBQP}}_{\|,\,\text{classical}}$*, where* $\mathsf{BQP}^{\mathsf{PostBQP}}_{\|,\,\text{classical}}$ *is the class of problems solvable by a* BQP *machine that can make a polynomial number of nonadaptive classical queries to a* PostBQP *oracle.*

Proof. Clearly PostBQP is closed under complement. To show closure under intersection, let $L_1, L_2 \in \mathsf{PostBQP}$. Then to decide whether $x \in L_1 \cap L_2$, run amplified computations (with error probability at most $1/6$) to decide if $x \in L_1$ and if $x \in L_2$, postselect on both computations succeeding, and accept if and only if both accept. It follows that PostBQP is closed under union as well.

In general, suppose a $\mathsf{BQP}^{\mathsf{PostBQP}}_{\|,\,\text{classical}}$ machine M submits queries $q_1, \ldots, q_{p(n)}$ to the PostBQP oracle. Then run amplified computations (with error probability at most, say, $\frac{1}{10p(n)}$) to decide the answers to these queries, and postselect on all $p(n)$ of them succeeding. By the union bound, if M had error probability ε with a perfect PostBQP oracle, then its new error probability is at most $\varepsilon + 1/10$, which can easily be reduced through amplification. ■

One might wonder why Proposition 133 doesn't go through with *adaptive* queries. The reason is subtle: suppose we have two PostBQP computations, the second of which relies on the output of the first. Then even if the first computation is amplified a polynomial number of times, it still has an exponentially small probability of error. But since the second computation uses postselection, *any* nonzero error probability could be magnified arbitrarily, and is therefore too large.

I now prove the main result.

Theorem 134 $\mathsf{PostBQP} = \mathsf{PP}$.

Proof. We have already observed that $\mathsf{PostBQP} \subseteq \mathsf{PP}$. For the other direction, let $f : \{0,1\}^n \rightarrow \{0,1\}$ be an efficiently computable Boolean function and let $s = |\{x : f(x) = 1\}|$. Then we need to decide in PostBQP whether $s < 2^{n-1}$ or $s \geq 2^{n-1}$. (As a technicality, we can guarantee using padding that $s > 0$.)

The algorithm is as follows: first prepare the state $2^{-n/2} \sum_{x \in \{0,1\}^n} |x\rangle |f(x)\rangle$. Then following Abrams and Lloyd [15], apply Hadamard gates to all n qubits in the first register and postselect[1] on that register being $|0\rangle^{\otimes n}$, to obtain $|0\rangle^{\otimes n} |\psi\rangle$ where

$$|\psi\rangle = \frac{(2^n - s)|0\rangle + s|1\rangle}{\sqrt{(2^n - s)^2 + s^2}}.$$

[1] Postselection is actually overkill here, since the first register has at least $1/4$ probability of being $|0\rangle^{\otimes n}$.

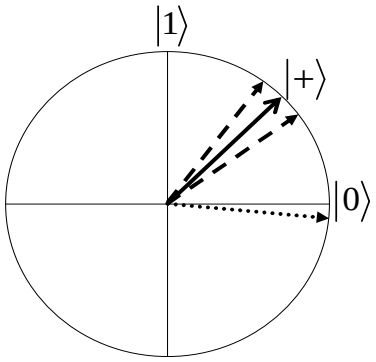

Figure 15.1: If s and $2^n - 2s$ are both positive, then as we vary the ratio of β to α, we eventually get close to $|+\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$ (dashed lines). On the other hand, if $2^n - 2s$ is not positive (dotted line), then we never even get into the first quadrant.

Next, for some positive real numbers α, β to be specified later, prepare $\alpha |0\rangle |\psi\rangle + \beta |1\rangle H |\psi\rangle$ where

$$H |\psi\rangle = \frac{\sqrt{1/2}\,(2^n)\,|0\rangle + \sqrt{1/2}\,(2^n - 2s)\,|1\rangle}{\sqrt{(2^n - s)^2 + s^2}}$$

is the result of applying a Hadamard gate to $|\psi\rangle$. Then postselect on the second qubit being $|1\rangle$. This yields the reduced state

$$|\varphi_{\beta/\alpha}\rangle = \frac{\alpha s\,|0\rangle + \beta\sqrt{1/2}\,(2^n - 2s)\,|1\rangle}{\sqrt{\alpha^2 s^2 + (\beta^2/2)\,(2^n - 2s)^2}}$$

in the first qubit.

Suppose $s < 2^{n-1}$, so that s and $\sqrt{1/2}\,(2^n - 2s)$ are both at least 1. Then we claim there exists an integer $i \in [-n, n]$ such that, if we set $\beta/\alpha = 2^i$, then $|\varphi_{2^i}\rangle$ is close to the state $|+\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$:

$$|\langle +|\varphi_{2^i}\rangle| \geq \frac{1 + \sqrt{2}}{\sqrt{6}} > 0.985.$$

For since $\sqrt{1/2}\,(2^n - 2s)/s$ lies between 2^{-n} and 2^n, there must be an integer $i \in [-n, n-1]$ such that $|\varphi_{2^i}\rangle$ and $|\varphi_{2^{i+1}}\rangle$ fall on opposite sides of $|+\rangle$ in the first quadrant (see Figure 15.1). So the worst case is that $\langle +|\varphi_{2^i}\rangle = \langle +|\varphi_{2^{i+1}}\rangle$, which occurs when $|\varphi_{2^i}\rangle = \sqrt{2/3}\,|0\rangle + \sqrt{1/3}\,|0\rangle$ and $|\varphi_{2^{i+1}}\rangle = \sqrt{1/3}\,|0\rangle + \sqrt{2/3}\,|0\rangle$. On the other hand, suppose $s \geq 2^{n-1}$, so that $\sqrt{1/2}\,(2^n - 2s) \leq 0$. Then $|\varphi_{2^i}\rangle$ never lies in the first or third quadrants, and therefore $|\langle +|\varphi_{2^i}\rangle| \leq 1/\sqrt{2} < 0.985$.

It follows that, by repeating the whole algorithm $n(2n+1)$ times (as in Proposition 133), with n invocations for each integer $i \in [-n, n]$, we can learn whether $s < 2^{n-1}$ or $s \geq 2^{n-1}$ with exponentially small probability of error. ■

Combining Proposition 133 with Theorem 134 immediately yields that PP is closed under intersection, as well as under BQP truth-table reductions.

15.2 Fantasy Quantum Mechanics

> "It is striking that it has so far not been possible to find a logically consistent theory that is close to quantum mechanics, other than quantum mechanics itself."
>
> —Steven Weinberg, *Dreams of a Final Theory* [241]

Is quantum mechanics an island in theoryspace? By "theoryspace," I mean the space of logically conceivable physical theories, with two theories close to each other if they differ in few respects. An "island" in theoryspace is then a natural and interesting theory, whose neighbors are all somehow perverse or degenerate. The Standard Model is not an island, because we do not know of any compelling (non-anthropic) reason why the masses and coupling constants should have the values they do. Likewise, general relativity is probably not an island, because of alternatives such as the Brans-Dicke theory.

To many physicists, however, quantum mechanics *does* seem like an island: change any one aspect, and the whole theory becomes inconsistent or nonsensical. There are many mathematical results supporting this opinion: for example, Gleason's Theorem [127] and other "derivations" of the $|\psi|^2$ probability rule [93, 252]; arguments for why amplitudes have to be complex numbers, rather than real numbers or quaternions [81, 143]; and "absurd" consequences of allowing nonlinear transformations between states [15, 126, 191]. The point of these results is to provide some sort of explanation for why quantum mechanics has the properties it does.

In 1998, Abrams and Lloyd [15] suggested that computational complexity could also be pressed into such an explanatory role. In particular, they showed that under almost any nonlinear variant of quantum mechanics, one could build a "nonlinear quantum computer" able to solve NP-complete and even #P-complete problems in polynomial time.[2] One interpretation of their result is that we should look very hard for nonlinearities in experiments! But a different interpretation, the one I prefer, is that their result provides independent evidence that quantum mechanics is linear.[3]

[2] A caveat is that it remains an open problem whether this can be done fault-tolerantly. The answer might depend on the allowed types of nonlinear gate. On the other hand, if arbitrary 1-qubit nonlinear gates can be implemented without error, then even PSPACE-complete problems can be solved in polynomial time. This is tight, since nonlinear quantum computers can also be simulated in PSPACE. I will give more details in a forthcoming survey paper [12].

[3] Note that I would *not* advocate this interpretation if it was merely (say) graph isomorphism that was efficiently solvable in nonlinear quantum mechanics, just as I do not take Shor's factoring algorithm as evidence for the falsehood of ordinary quantum mechanics. I will explain in [12] why I think this distinction, between NP-complete problems and specific NP-intermediate problems, is a justified one.

In this section I build on Theorem 134 to offer similar "evidence" that quantum mechanics is unitary, and that the measurement rule is $|\psi|^2$.

Let $\mathsf{BQP}_{\mathsf{nu}}$ be the class of problems solvable by a uniform family of polynomial-size, bounded-error quantum circuits, where the circuits can consist of arbitrary 1- and 2-qubit *invertible* linear transformations, rather than just unitary transformations. Immediately before a measurement, the amplitude α_x of each basis state $|x\rangle$ is divided by $\sqrt{\sum_y |\alpha_y|^2}$ to normalize it.

Proposition 135 $\mathsf{BQP}_{nu} = \mathsf{PP}$.

Proof. The inclusion $\mathsf{BQP}_{\mathsf{nu}} \subseteq \mathsf{PP}$ follows easily from Adleman, DeMarrais, and Huang's proof that $\mathsf{BQP} \subseteq \mathsf{PP}$ [16], which does not depend on unitarity. For the other direction, by Theorem 134 it suffices to show that $\mathsf{PostBQP} \subseteq \mathsf{BQP}_{\mathsf{nu}}$. To postselect on a qubit being $|1\rangle$, we simply apply the 1-qubit nonunitary operation

$$\begin{pmatrix} 2^{-q(n)} & 0 \\ 0 & 1 \end{pmatrix}$$

for some sufficiently large polynomial q. ■

Next, for any nonnegative real number p, define BQP_p similarly to BQP, except that when we measure, the probability of obtaining a basis state $|x\rangle$ equals $|\alpha_x|^p / \sum_y |\alpha_y|^p$ rather than $|\alpha_x|^2$. Thus $\mathsf{BQP}_2 = \mathsf{BQP}$. Assume that all gates are unitary and that there are no intermediate measurements, just a single standard-basis measurement at the end.

Theorem 136 $\mathsf{PP} \subseteq \mathsf{BQP}_p \subseteq \mathsf{PSPACE}$ *for all constants* $p \neq 2$, *with* $\mathsf{BQP}_p = \mathsf{PP}$ *when* $p \in \{4, 6, 8, \ldots\}$.

Proof. To simulate PP in BQP_p, run the algorithm of Theorem 134, having initialized $O\left(\frac{1}{|p-2|} \operatorname{poly}(n)\right)$ ancilla qubits to $|0\rangle$. Suppose the algorithm's state at some point is $\sum_x \alpha_x |x\rangle$, and we want to postselect on the event $|x\rangle \in S$, where S is a subset of basis states. Here is how: if $p < 2$, then for some sufficiently large polynomial q, apply Hadamard gates to $c = 2q(n)/(2-p)$ fresh ancilla qubits conditioned on $|x\rangle \in S$. The result is to increase the "probability mass" of each $|x\rangle \in S$ from $|\alpha_x|^p$ to

$$2^c \cdot \left|2^{-c/2}\alpha_x\right|^p = 2^{(2-p)c/2} |\alpha_x|^p = 2^{q(n)} |\alpha_x|^p,$$

while the probability mass of each $|x\rangle \notin S$ remains unchanged. Similarly, if $p > 2$, then apply Hadamard gates to $c = 2q(n)/(p-2)$ fresh ancilla qubits conditioned on $|x\rangle \notin S$. This decreases the probability mass of each $|x\rangle \notin S$ from $|\alpha_x|^p$ to $2^c \cdot \left|2^{-c/2}\alpha_x\right|^p = 2^{-q(n)} |\alpha_x|^p$, while the probability mass of each $|x\rangle \in S$ remains unchanged. The final observation is that Theorem 134 still goes through if $p \neq 2$. For it suffices to distinguish the case $|\langle +|\varphi_{2^i}\rangle| > 0.985$ from $|\langle +|\varphi_{2^i}\rangle| \leq 1/\sqrt{2}$ with exponentially small probability of error, using polynomially many copies of the state $|\varphi_{2^i}\rangle$. But we can do this for any p, since all $|\psi|^p$ rules behave well under tensor products (in the sense that $|\alpha\beta|^p = |\alpha|^p |\beta|^p$).

The inclusion $\mathsf{BQP}_p \subseteq \mathsf{PSPACE}$ follows easily from the techniques used by Bernstein and Vazirani [55] to show $\mathsf{BQP} \subseteq \mathsf{PSPACE}$. Let S be the set of accepting states; then simply compute $\sum_{x \in S} |\alpha_x|^p$ and $\sum_{x \notin S} |\alpha_x|^p$ and see which is greater.

To simulate BQP_p in PP when $p \in \{4, 6, 8, \ldots\}$, we generalize the technique of Adleman, DeMarrais, and Huang [16], which handled the case $p = 2$. As in Theorem 57 in Chapter 10, assume that all gates are Hadamard or Toffoli gates; then we can write each amplitude α_x as a sum of exponentially many contributions, $a_{x,1} + \cdots + a_{x,N}$, where each $a_{x,i}$ is a rational real number computable in classical polynomial time. Then letting S be the set of accepting states, it suffices to test whether

$$\begin{aligned}
\sum_{x \in S} |\alpha_x|^p &= \sum_{x \in S} \alpha_x^p \\
&= \sum_{x \in S} \left(\sum_{i \in \{1,\ldots,N\}} a_{x,i} \right)^p \\
&= \sum_{x \in S} \sum_{B \subseteq \{1,\ldots,N\}, |B| = p} \prod_{i \in B} a_{x,i}
\end{aligned}$$

is greater than $\sum_{x \notin S} |\alpha_x|^p$, which we can do in PP. ■

15.3 Open Problems

The new proof that PP is closed under intersection came as a total surprise to me. But on reflection, it goes a long way toward convincing me of a thesis expressed in Chapter 1: that quantum computing offers a new perspective from which to revisit the central questions of classical complexity theory. What other classical complexity classes can we characterize in quantum terms, and what other questions can we answer by that means?

A first step might be to prove even stronger closure properties for PP. Recall from Proposition 133 that $\mathsf{PostBQP}$ is closed under polynomial-time truth-table reductions. Presumably this can't be generalized to closure under Turing reductions, since if it could then we would have $\mathsf{PP} = \mathsf{P}^{\mathsf{PP}}$, which is considered unlikely.[4] But can we show closure under nonadaptive *quantum* reductions? More formally, let $\mathsf{BQP}_{\|}^{\mathsf{PostBQP}}$ be the class of problems solvable by a BQP machine that can make a single quantum query, which consists of a list of polynomially many questions for a $\mathsf{PostBQP}$ oracle. Then does $\mathsf{BQP}_{\|}^{\mathsf{PostBQP}}$ equal $\mathsf{PostBQP}$? The difficulty in showing this seems to be uncomputing garbage after the $\mathsf{PostBQP}$ oracle is simulated.

As for fantasy quantum mechanics, an interesting open question is whether $\mathsf{BQP}_p = \mathsf{PP}$ for all nonnegative real numbers $p \neq 2$. An obvious idea for simulating BQP_p in PP would be to use a Taylor series expansion for the probability masses $|\alpha_x|^p$. Unfortunately, I have no idea how to get fast enough convergence.

[4] Indeed, Beigel [46] gave an oracle relative to which $\mathsf{P}^{\mathsf{NP}} \not\subset \mathsf{PP}$.

Chapter 16

The Power of History

Quantum mechanics lets us calculate the probability that (say) an electron will be found in an excited state if measured at a particular time. But it is silent about *multiple-time* or *transition* probabilities: that is, what is the probability that the electron will be in an excited state at time t_1, given that it was in its ground state at an earlier time t_0? The usual response is that this question is meaningless, unless of course the electron was *measured* (or otherwise known with probability 1) to be in its ground state at t_0. A different response—pursued by Schrödinger [213], Bohm [59], Bell [49], Nelson [181], Dieks [97], and others—treats the question as provisionally meaningful, and then investigates how one might answer it mathematically. Specific attempts at answers are called "hidden-variable theories."

The appeal of hidden-variable theories is that they provide one possible solution to the measurement problem. For they allow us to apply unitary quantum mechanics to the entire universe (including ourselves), yet still discuss the probability of a future observation conditioned on our current observations. Furthermore, they let us do so without making any assumptions about decoherence or the nature of observers. For example, even if an observer were placed in coherent superposition, that observer would still have a sequence of definite experiences, and the probability of any such sequence could be calculated.

This chapter initiates the study of hidden variables from a quantum computing perspective. I restrict attention to the simplest possible setting: that of discrete time, a finite-dimensional Hilbert space, and a fixed orthogonal basis. Within this setting, I reformulate known hidden-variable theories due to Dieks [97] and Schrödinger [213], and also introduce a new theory based on network flows. However, a more important contribution is the *axiomatic approach* that I use. I propose five axioms for hidden-variable theories, and then compare theories against each other based on which of the axioms they satisfy. A central question in this approach is which subsets of axioms can be satisfied simultaneously.

In a second part of the chapter, I make the connection to quantum computing explicit, by studying the computational complexity of simulating hidden-variable theories. Below I describe the computational results.

16.1 The Complexity of Sampling Histories

It is often stressed that hidden-variable theories yield exactly the same predictions as ordinary quantum mechanics. On the other hand, these theories describe a different picture of physical reality, with an additional layer of dynamics beyond that of a state vector evolving unitarily. I address a question that, to my knowledge, had never been raised before: *what is the computational complexity of simulating that additional dynamics?* In other words, if we could examine a hidden variable's entire history, then could we solve problems in polynomial time that are intractable even for quantum computers?

I present strong evidence that the answer is yes. The Graph Isomorphism problem asks whether two graphs G and H are isomorphic; while given a basis for a lattice $\mathcal{L} \in \mathbb{R}^n$, the Approximate Shortest Vector problem asks for a nonzero vector in $\mathcal{L}$ within a $\sqrt{n}$ factor of the shortest one. I show that both problems are efficiently solvable by sampling a hidden variable's history, provided the hidden-variable theory satisfies a reasonable axiom. By contrast, despite a decade of effort, neither problem is known to lie in BQP. Thus, if we let DQP (Dynamical Quantum Polynomial-Time) be the class of problems solvable in the new model, then this already provides circumstantial evidence that BQP is strictly contained in DQP.

However, the evidence is stronger than this. For I actually show that DQP contains the entire class Statistical Zero Knowledge, or SZK. Furthermore, Chapter 6 showed that relative to an oracle, SZK is not contained in BQP. Combining the result that $\mathsf{SZK} \subseteq \mathsf{DQP}$ with the oracle separation of Chapter 6, one obtains that $\mathsf{BQP} \neq \mathsf{DQP}$ relative to an oracle as well.

Besides solving SZK problems, I also show that by sampling histories, one could search an unordered database of N items for a single "marked item" using only $O\left(N^{1/3}\right)$ database queries. By comparison, Grover's quantum search algorithm [139] requires $\Theta\left(N^{1/2}\right)$ queries, while classical algorithms require $\Theta\left(N\right)$ queries. On the other hand, I also show that the $N^{1/3}$ upper bound is the best possible—so even in the histories model, one cannot search an N-item database in $(\log N)^c$ steps for some fixed power c. This implies that $\mathsf{NP} \not\subset \mathsf{DQP}$ relative to an oracle, which in turn suggests that DQP is *still* not powerful enough to solve NP-complete problems in polynomial time.

At this point I should address a concern that many readers will have. Once we extend quantum mechanics by positing the "unphysical" ability to sample histories, isn't it completely unsurprising if we can then solve problems that were previously intractable? I believe the answer is no, for three reasons.

First, almost every change that makes the quantum computing model more powerful, seems to make it *so much* more powerful that NP-complete and even harder problems become solvable efficiently. To give some examples, NP-complete problems can be solved in polynomial time using a nonlinear Schrödinger equation, as shown by Abrams and Lloyd [15]; using closed timelike curves, as shown by Brun [72] and Bacon [40] (and conjectured by Deutsch [91]); or using a measurement rule of the form $|\psi|^p$ for any $p \neq 2$, as shown in Chapter 15. It is also easy to see that we could solve NP-complete problems if, given a quantum state $|\psi\rangle$, we could request a classical description of $|\psi\rangle$, such as a list of ampli-

tudes or a preparation procedure.[1] By contrast, the DQP model is the first independently motivated model I know of that seems more powerful than quantum computing, but only *slightly* so.[2] Moreover, the striking fact that unordered search takes about $N^{1/3}$ steps in the DQP model, as compared to N steps classically and $N^{1/2}$ quantum-mechanically, suggests that DQP somehow "continues a sequence" that begins with P and BQP. It would be interesting to find a model in which search takes $N^{1/4}$ or $N^{1/5}$ steps.

The second reason the results are surprising is that, given a hidden variable, the distribution over its possible values at any *single* time is governed by standard quantum mechanics, and is therefore efficiently samplable on a quantum computer. So if examining the variable's history confers any extra computational power, then it can only be because of *correlations* between the variable's values at different times.

The third reason is the criterion for success. I am not saying merely that one can solve Graph Isomorphism under *some* hidden-variable theory; or even that, under any theory satisfying the indifference axiom, there exists an algorithm to solve it; but rather that there exists a *single* algorithm that solves Graph Isomorphism under any theory satisfying indifference. Thus, we must consider even theories that are specifically designed to thwart such an algorithm.

But what is the motivation for these results? The first motivation is that, within the community of physicists who study hidden-variable theories such as Bohmian mechanics, there is great interest in actually *calculating* the hidden-variable trajectories for specific physical systems [190, 140]. My results show that, when many interacting particles are involved, this task might be fundamentally intractable, even if a quantum computer were available. The second motivation is that, in classical computer science, studying "unrealistic" models of computation has often led to new insights into realistic ones; and likewise I expect that the DQP model could lead to new results about standard quantum computation. Indeed, in a sense this has already happened—for the collision lower bound of Chapter 6 grew out of work on the BQP versus DQP question.

16.2 Outline of Chapter

Sections 16.3 through 16.6.2 develop the axiomatic approach to hidden variables; then Sections 16.7 through 16.10 study the computational complexity of sampling hidden-variable histories.

Section 16.3 formally defines hidden-variable theories in my sense; then Section 16.3.1 contrasts these theories with related ideas such as Bohmian mechanics and modal interpretations. Section 16.3.2 addresses the most common objections to my approach: for example, that the implicit dependence on a fixed basis is unacceptable.

In Section 16.4, I introduce five possible axioms for hidden-variable theories. These are indifference to the identity operation; robustness to small perturbations; commutativity

[1] For as Abrams and Lloyd [15] observed, we can so arrange things that $|\psi\rangle = |0\rangle$ if an NP-complete instance of interest to us has no solution, but $|\psi\rangle = \sqrt{1-\varepsilon}\,|0\rangle + \sqrt{\varepsilon}\,|1\rangle$ for some tiny ε if it has a solution.

[2] One can define other, less motivated, models with the same property by allowing "non-collapsing measurements" of quantum states, but these models are very closely related to DQP. Indeed, a key ingredient in the results of this chapter will be to show that certain kinds of non-collapsing measurements can be *simulated* using histories.

with respect to spacelike-separated unitaries; commutativity for the special case of product states; and invariance under decomposition of mixed states into pure states. Ideally, a theory would satisfy all of these axioms. However, I show in Section 16.5 that no theory satisfies both indifference and commutativity; no theory satisfies both indifference and a stronger version of robustness; no theory satisfies indifference, robustness, and decomposition invariance; and no theory satisfies a stronger version of decomposition invariance.

In Section 16.6 I shift from negative to positive results. Section 16.6.1 presents a hidden-variable theory called the *flow theory* or $\mathcal{FT}$, which is based on the Max-Flow-Min-Cut theorem from combinatorial optimization. The idea is to define a network of "pipes" from basis states at an initial time to basis states at a final time, and then route as much probability mass as possible through these pipes. The capacity of each pipe depends on the corresponding entry of the unitary acting from the initial to final time. To find the probability of transitioning from basis state $|i\rangle$ to basis state $|j\rangle$, we then determine how much of the flow originating at $|i\rangle$ is routed along the pipe to $|j\rangle$. The main results are that $\mathcal{FT}$ is well-defined and that it is robust to small perturbations. Since $\mathcal{FT}$ trivially satisfies the indifference axiom, this implies that the indifference and robustness axioms can be satisfied simultaneously, which was not at all obvious *a priori*.

Section 16.6.2 presents a second theory that I call the *Schrödinger theory* or $\mathcal{ST}$, since it is based on a pair of integral equations introduced in a 1931 paper of Schrödinger [213]. Schrödinger conjectured, but was unable to prove, the existence and uniqueness of a solution to these equations; the problem was not settled until the work of Nagasawa [179] in the 1980's. In the discrete setting the problem is simpler, and I give a self-contained proof of existence using a matrix scaling technique due to Sinkhorn [221]. The idea is as follows: we want to convert a unitary matrix that maps one quantum state to another, into a nonnegative matrix whose i^{th} column sums to the initial probability of basis state $|i\rangle$, and whose j^{th} row sums to the final probability of basis state $|j\rangle$. To do so, we first replace each entry of the unitary matrix by its absolute value, then normalize each column to sum to the desired initial probability, then normalize each row to sum to the desired final probability. But then the columns are no longer normalized correctly, so we normalize them *again*, then normalize the rows again, and so on. I show that this iterative process converges, from which it follows that $\mathcal{ST}$ is well-defined. I also observe that $\mathcal{ST}$ satisfies the indifference and product commutativity axioms, and violates the decomposition invariance axiom. I conjecture that $\mathcal{ST}$ satisfies the robustness axiom; proving that conjecture is one of the main open problems of the chapter.

In Section 16.7 I shift attention to the complexity of sampling histories. I formally define DQP as the class of problems solvable by a classical polynomial-time algorithm with access to a "history oracle." Given a sequence of quantum circuits as input, this oracle returns a sample from a corresponding distribution over histories of a hidden variable, according to some hidden-variable theory $\mathcal{T}$. The oracle can choose $\mathcal{T}$ "adversarially," subject to the constraint that $\mathcal{T}$ satisfies the indifference and robustness axioms. Thus, a key result from Section 16.7 that I rely on is that there *exists* a hidden-variable theory satisfying indifference and robustness.

Section 16.7.1 establishes the most basic facts about DQP: for example, that $\mathsf{BQP} \subseteq \mathsf{DQP}$, and that DQP is independent of the choice of gate set. Then Section 16.8

presents the "juggle subroutine," a crucial ingredient in both of the main hidden-variable algorithms. Given a state of the form $(|a\rangle + |b\rangle)/\sqrt{2}$ or $(|a\rangle - |b\rangle)/\sqrt{2}$, the goal of this subroutine is to "juggle" a hidden variable between $|a\rangle$ and $|b\rangle$, so that when we inspect the hidden variable's history, both $|a\rangle$ and $|b\rangle$ are observed with high probability. The difficulty is that this needs to work under *any* indifferent hidden-variable theory.

Next, Section 16.9 combines the juggle subroutine with a technique of Valiant and Vazirani [231] to prove that $\mathsf{SZK} \subseteq \mathsf{DQP}$, from which it follows in particular that Graph Isomorphism and Approximate Shortest Vector are in DQP. Then Section 16.10 applies the juggle subroutine to search an N-item database in $O\left(N^{1/3}\right)$ queries, and also proves that this $N^{1/3}$ bound is optimal.

I conclude in Section 16.11 with some directions for further research.

16.3 Hidden-Variable Theories

Suppose we have an $N \times N$ unitary matrix U, acting on a state

$$|\psi\rangle = \alpha_1 |1\rangle + \cdots + \alpha_N |N\rangle,$$

where $|1\rangle, \ldots, |N\rangle$ is a standard orthogonal basis. Let

$$U|\psi\rangle = \beta_1 |1\rangle + \cdots + \beta_N |N\rangle.$$

Then can we construct a stochastic matrix S, which maps the vector of probabilities

$$\overrightarrow{p} = \begin{bmatrix} |\alpha_1|^2 \\ \vdots \\ |\alpha_N|^2 \end{bmatrix}$$

induced by measuring $|\psi\rangle$, to the vector

$$\overrightarrow{q} = \begin{bmatrix} |\beta_1|^2 \\ \vdots \\ |\beta_N|^2 \end{bmatrix}$$

induced by measuring $U|\psi\rangle$? Trivially yes. The following matrix maps *any* vector of probabilities to $\overrightarrow{q}$, ignoring the input vector $\overrightarrow{p}$ entirely:

$$S_{\mathcal{PT}} = \begin{bmatrix} |\beta_1|^2 & \cdots & |\beta_1|^2 \\ \vdots & & \vdots \\ |\beta_N|^2 & \cdots & |\beta_N|^2 \end{bmatrix}.$$

Here $\mathcal{PT}$ stands for *product theory*. The product theory corresponds to a strange picture of physical reality, in which memories and records are completely unreliable, there being no causal connection between states of affairs at earlier and later times.

So we would like S to depend on U itself somehow, not just on $|\psi\rangle$ and $U|\psi\rangle$. Indeed, ideally S would be a function *only* of U, and not of $|\psi\rangle$. But this is impossible, as

the following example shows. Let U be a $\pi/4$ rotation, and let $|+\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$ and $|-\rangle = (|0\rangle - |1\rangle)/\sqrt{2}$. Then $U|+\rangle = |1\rangle$ implies that

$$S(|+\rangle, U) = \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix},$$

whereas $U|-\rangle = |0\rangle$ implies that

$$S(|-\rangle, U) = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}.$$

On the other hand, it is easy to see that, if S can depend on $|\psi\rangle$ as well as U, then there are infinitely many choices for the function $S(|\psi\rangle, U)$. Every choice reproduces the predictions of quantum mechanics perfectly when restricted to single-time probabilities. So how can we possibly choose among them? My approach in Sections 16.4 and 16.6 will be to write down axioms that we would like S to satisfy, and then investigate which of the axioms can be satisfied simultaneously.

Formally, a *hidden-variable theory* is a family of functions $\{S_N\}_{N \geq 1}$, where each S_N maps an N-dimensional mixed state ρ and an $N \times N$ unitary matrix U onto a singly stochastic matrix $S_N(\rho, U)$. I will often suppress the dependence on N, ρ, and U, and occasionally use subscripts such as $\mathcal{PT}$ or $\mathcal{FT}$ to indicate the theory in question. Also, if $\rho = |\psi\rangle\langle\psi|$ is a pure state I may write $S(|\psi\rangle, U)$ instead of $S(|\psi\rangle\langle\psi|, U)$.

Let $(M)_{ij}$ denote the entry in the i^{th} column and j^{th} row of matrix M. Then $(S)_{ij}$ is the probability that the hidden variable takes value $|j\rangle$ after U is applied, conditioned on it taking value $|i\rangle$ before U is applied. At a minimum, any theory must satisfy the following marginalization axiom: for all $j \in \{1, \ldots, N\}$,

$$\sum_i (S)_{ij} (\rho)_{ii} = \left(U \rho U^{-1}\right)_{jj}.$$

This says that after U is applied, the hidden variable takes value $|j\rangle$ with probability $\left(U \rho U^{-1}\right)_{jj}$, which is the usual Born probability.

Often it will be convenient to refer, not to S itself, but to the matrix $P(\rho, U)$ of joint probabilities whose (i, j) entry is $(P)_{ij} = (S)_{ij} (\rho)_{ii}$. The i^{th} column of P must sum to $(\rho)_{ii}$, and the j^{th} row must sum to $\left(U \rho U^{-1}\right)_{jj}$. Indeed, I will define the theories $\mathcal{FT}$ and $\mathcal{ST}$ by first specifying the matrix P, and then setting $(S)_{ij} := (P)_{ij} / (\rho)_{ii}$. This approach has the drawback that if $(\rho)_{ii} = 0$, then the i^{th} column of S is undefined. To get around this, I adopt the convention that

$$S(\rho, U) := \lim_{\varepsilon \to 0^+} S(\rho_\varepsilon, U)$$

where $\rho_\varepsilon = (1 - \varepsilon)\rho + \varepsilon I$ and I is the $N \times N$ maximally mixed state. Technically, the limits

$$\lim_{\varepsilon \to 0^+} \frac{(P(\rho_\varepsilon, U))_{ij}}{(\rho_\varepsilon)_{ii}}$$

might not exist, but in the cases of interest it will be obvious that they do.

16.3.1 Comparison with Previous Work

Before going further, I should contrast my approach with previous approaches to hidden variables, the most famous of which is Bohmian mechanics [59]. My main difficulty with Bohmian mechanics is that it commits itself to a Hilbert space of particle positions and momenta. Furthermore, it is crucial that the positions and momenta be *continuous*, in order for particles to evolve deterministically. To see this, let $|L\rangle$ and $|R\rangle$ be discrete positions, and suppose a particle is in state $|L\rangle$ at time t_0, and state $\left(|L\rangle + |R\rangle\right)/\sqrt{2}$ at a later time t_1. Then a hidden variable representing the position would have entropy 0 at t_1, since it is always $|L\rangle$ then; but entropy 1 at t_1, since it is $|L\rangle$ or $|R\rangle$ both with $1/2$ probability. Therefore the earlier value cannot determine the later one.[3] It follows that Bohmian mechanics is incompatible with the belief that all physical observables are discrete. But in my view, there are strong reasons to hold that belief, which include black hole entropy bounds; the existence of a natural minimum length scale (10^{-33} cm); results on area quantization in quantum gravity [205]; the fact that many physical quantities once thought to be continuous have turned out to be discrete; the infinities of quantum field theory; the implausibility of analog "hypercomputers"; and conceptual problems raised by the independence of the continuum hypothesis.

Of course there exist stochastic analogues of Bohmian mechanics, among them Nelsonian mechanics [181] and Bohm and Hiley's "stochastic interpretation" [60]. But it is not obvious why we should prefer these to other stochastic hidden-variable theories. From a quantum-information perspective, it is much more natural to take an abstract approach—one that allows arbitrary finite-dimensional Hilbert spaces, and that does not rule out any transition rule *a priori*.

Stochastic hidden variables have also been considered in the context of modal interpretations; see Dickson [96], Bacciagaluppi and Dickson [39], and Dieks [97] for example. However, the central assumptions in that work are extremely different from mine. In modal interpretations, a pure state evolving unitarily poses no problems at all: one simply rotates the hidden-variable basis along with the state, so that the state always represents a "possessed property" of the system in the current basis. Difficulties arise only for mixed states; and there, the goal is to track a whole set of possessed properties. By contrast, my approach is to fix an orthogonal basis, then track a single hidden variable that is an element of that basis. The issues raised by pure states and mixed states are essentially the same.

Finally I should mention the consistent-histories interpretation of Griffiths [137] and Gell-Mann and Hartle [122]. This interpretation assigns probabilities to various histories through a quantum system, so long as the "interference" between those histories is negligible. Loosely speaking, then, the situations where consistent histories make sense are precisely the ones where the question of transition probabilities can be avoided.

[3] Put differently, Bohm's conservation of probability result breaks down because the "wavefunctions" at t_0 and t_1 are degenerate, with all amplitude concentrated on finitely many points. But in a discrete Hilbert space, *every* wavefunction is degenerate in this sense!

16.3.2 Objections

Hidden-variable theories, as I define them, are open to several technical objections. For example, I required transition probabilities for only one orthogonal observable. What about other observables? The problem is that, according to the Kochen-Specker theorem, we cannot assign consistent values to all observables at any *single* time, let alone give transition probabilities for those values. This is an issue in any setting, not just mine. The solution I prefer is to postulate a fixed orthogonal basis of "distinguishable experiences," and to interpret a measurement in any other basis as a unitary followed by a measurement in the fixed basis. As mentioned in Section 16.3.1, modal interpretations opt for a different solution, which involves sets of bases that change over time with the state itself.

Another objection is that the probability of transitioning from basis state $|i\rangle$ at time t_1 to basis state $|j\rangle$ at time t_2 might depend on how finely we divide the time interval between t_1 and t_2. In other words, for some state $|\psi\rangle$ and unitaries V, W, we might have

$$S\left(|\psi\rangle, WV\right) \neq S\left(V|\psi\rangle, W\right) S\left(|\psi\rangle, V\right)$$

(a similar point was made by Gillespie [125]). Indeed, this is true for any hidden-variable theory other than the product theory $\mathcal{PT}$. To see this, observe that for all unitaries U and states $|\psi\rangle$, there exist unitaries V, W such that $U = WV$ and $V|\psi\rangle = |1\rangle$. Then applying V destroys all information in the hidden variable (that is, decreases its entropy to 0); so if we then apply W, then the variable's final value must be uncorrelated with the initial value. In other words, $S\left(V|\psi\rangle, W\right) S\left(|\psi\rangle, V\right)$ must equal $S_{\mathcal{PT}}\left(|\psi\rangle, U\right)$. It follows that to any hidden-variable theory we must associate a time scale, or some other rule for deciding when the transitions take place.

In response, it should be noted that exactly the same problem arises in *continuous*-time stochastic hidden-variable theories. For if a state $|\psi\rangle$ is governed by the Schrödinger equation $d|\psi\rangle/dt = iH_t|\psi\rangle$, and a hidden variable's probability distribution $\overrightarrow{p}$ is governed by the stochastic equation $d\overrightarrow{p}/d\tau = A_\tau \overrightarrow{p}$, then there is still an arbitrary parameter $d\tau/dt$ on which the dynamics depend.

Finally, it will be objected that I have ignored special relativity. In Section 16.4 I will define a *commutativity axiom*, which informally requires that the stochastic matrix S not depend on the temporal order of spacelike separated events. Unfortunately, we will see that when entangled states are involved, commutativity is irreconcilable with another axiom that seems even more basic. The resulting nonlocality has the same character as the nonlocality of Bohmian mechanics—that is, one cannot use it to send superluminal signals in the usual sense, but it is unsettling nonetheless.

16.4 Axioms for Hidden-Variable Theories

I now state five axioms that we might like hidden-variable theories to satisfy.

Indifference. The indifference axiom says that if U is block-diagonal, then S should also be block-diagonal with the same block structure or some refinement thereof. Formally, let a *block* be a subset $B \subseteq \{1, \ldots, N\}$ such that $(U)_{ij} = 0$ for all $i \in B, j \notin B$ and $i \notin B, j \in B$. Then for all blocks B, we should have $(S)_{ij} = 0$ for all $i \in B, j \notin B$ and

$i \notin B, j \in B$. In particular, indifference implies that given any state ρ in a tensor product space $\mathcal{H}_A \otimes \mathcal{H}_B$, and any unitary U that acts only on $\mathcal{H}_A$ (that is, never maps a basis state $|i_A\rangle \otimes |i_B\rangle$ to $|j_A\rangle \otimes |j_B\rangle$ where $i_B \neq j_B$), the stochastic matrix $S(\rho, U)$ acts only on $\mathcal{H}_A$ as well.

Robustness. A theory is robust if it is insensitive to small errors in a state or unitary (which, in particular, implies continuity). Suppose we obtain $\widetilde{\rho}$ and $\widetilde{U}$ by perturbing ρ and U respectively. Then for all polynomials p, there should exist a polynomial q such that for all N,

$$\left\| P\left(\widetilde{\rho}, \widetilde{U}\right) - P(\rho, U) \right\|_\infty \leq \frac{1}{p(N)}$$

where $\|M\|_\infty = \max_{ij} \left|(M)_{ij}\right|$, whenever $\|\widetilde{\rho} - \rho\|_\infty \leq 1/q(N)$ and $\left\|\widetilde{U} - U\right\|_\infty \leq 1/q(N)$. Robustness has an important advantage for quantum computing: if a hidden-variable theory is robust then the set of gates used to define the unitaries $U_1, \ldots, U_T$ is irrelevant, since by the Solovay-Kitaev Theorem (see [153, 182]), any universal quantum gate set can simulate any other to a precision ε with $O(\log^c 1/\varepsilon)$ overhead.

Commutativity. Let ρ_{AB} be a bipartite state, and let U_A and U_B act only on subsystems A and B respectively. Then commutativity means that the order in which U_A and U_B are applied is irrelevant:

$$S\left(U_A \rho_{AB} U_A^{-1}, U_B\right) S(\rho_{AB}, U_A) = S\left(U_B \rho_{AB} U_B^{-1}, U_A\right) S(\rho_{AB}, U_B).$$

Product Commutativity. A theory is product commutative if it satisfies commutativity for all separable pure states $|\psi\rangle = |\psi_A\rangle \otimes |\psi_B\rangle$.

Decomposition Invariance. A theory is decomposition invariant if

$$S(\rho, U) = \sum_{i=1}^N p_i S(|\psi_i\rangle \langle\psi_i|, U)$$

for every decomposition

$$\rho = \sum_{i=1}^N p_i |\psi_i\rangle \langle\psi_i|$$

of ρ into pure states. Theorem 138, part (ii) will show that the analogous axiom for $P(\rho, U)$ is unsatisfiable.

16.4.1 Comparing Theories

To fix ideas, let us compare some hidden-variable theories with respect to the above axioms. We have already seen the product theory $\mathcal{PT}$ in Section 16.3. It is easy to show that $\mathcal{PT}$ satisfies robustness, commutativity, and decomposition invariance. However, I consider $\mathcal{PT}$ unsatisfactory because it violates indifference: even if a unitary U acts only on the first of two qubits, $S_{\mathcal{PT}}(\rho, U)$ will readily produce transitions involving the second qubit.

Recognizing this problem, Dieks [97] proposed an alternative theory that amounts to the following.[4] First partition the set of basis states into minimal blocks $B_1, \ldots, B_m$

[4] Dieks (personal communication) says he would no longer defend this theory.

	$\mathcal{PT}$ (Product)	$\mathcal{DT}$ (Dieks)	$\mathcal{FT}$ (Flow)	$\mathcal{ST}$ (Schrödinger)
Indifference	No	Yes	Yes	Yes
Robustness	Yes	No	Yes	?
Commutativity	Yes	No	No	No
Product Commutativity	Yes	Yes	No	Yes
Decomposition Invariance	Yes	Yes	No	No

Table 16.1: Four hidden-variable theories and the axioms they satisfy

between which U never sends amplitude. Then apply the product theory separately to each block; that is, if i and j belong to the same block B_k then set

$$(S)_{ij} = \frac{\left(U\rho U^{-1}\right)_{jj}}{\sum_{\widehat{j}\in B_k}\left(U\rho U^{-1}\right)_{\widehat{j}\widehat{j}}},$$

and otherwise set $(S)_{ij} = 0$. The resulting *Dieks theory*, $\mathcal{DT}$, satisfies indifference by construction. However, it does not satisfy robustness (or even continuity), since the set of blocks can change if we replace '0' entries in U by arbitrarily small nonzero entries.

In Section 16.6 I will introduce two other hidden-variable theories, the flow theory $\mathcal{FT}$ and the Schrödinger theory $\mathcal{ST}$. Table 16.1 lists which axioms the four theories satisfy.

If we could prove that $\mathcal{ST}$ satisfies robustness, then Table 1 together with the impossibility results of Section 16.5 would completely characterize which of the axioms can be satisfied simultaneously.

16.5 Impossibility Results

This section shows that certain sets of axioms cannot be satisfied by any hidden-variable theory. I first show that the failure of $\mathcal{DT}$, $\mathcal{FT}$, and $\mathcal{ST}$ to satisfy commutativity is inherent, and not a fixable technical problem.

Theorem 137 *No hidden-variable theory satisfies both indifference and commutativity.*

Proof. Assume indifference holds, and let our initial state be $|\psi\rangle = \frac{|00\rangle+|11\rangle}{\sqrt{2}}$. Suppose U_A applies a $\pi/8$ rotation to the first qubit, and U_B applies a $-\pi/8$ rotation to the second qubit. Then

$$U_A|\psi\rangle = U_B|\psi\rangle = \frac{1}{\sqrt{2}}\left(\cos\frac{\pi}{8}|00\rangle - \sin\frac{\pi}{8}|01\rangle + \sin\frac{\pi}{8}|10\rangle + \cos\frac{\pi}{8}|11\rangle\right),$$

$$U_AU_B|\psi\rangle = U_BU_A|\psi\rangle = \frac{1}{2}\left(|00\rangle - |01\rangle + |10\rangle + |11\rangle\right).$$

Let v_t be the value of the hidden variable after t unitaries have been applied. Let E be the event that $v_0 = |00\rangle$ initially, and $v_2 = |10\rangle$ at the end. If U_A is applied before U_B, then the unique 'path' from v_0 to v_2 consistent with indifference sets $v_1 = |10\rangle$. So

$$\Pr[E] \leq \Pr[v_1 = |10\rangle] = \frac{1}{2}\sin^2\frac{\pi}{8}.$$

But if U_B is applied before U_A, then the probability that $v_0 = |11\rangle$ and $v_2 = |10\rangle$ is at most $\frac{1}{2}\sin^2\frac{\pi}{8}$, by the same reasoning. Thus, since v_2 must equal $|10\rangle$ with probability 1/4, and since the only possibilities for v_0 are $|00\rangle$ and $|11\rangle$,

$$\Pr[E] \geq \frac{1}{4} - \frac{1}{2}\sin^2\frac{\pi}{8} > \frac{1}{2}\sin^2\frac{\pi}{8}.$$

We conclude that commutativity is violated. ■

Let me remark on the relationship between Theorem 137 and Bell's Theorem. Any hidden-variable theory that is "local" in Bell's sense would immediately satisfy both indifference and commutativity. However, the converse is not obvious, since there might be nonlocal information in the states $U_A|\psi\rangle$ or $U_B|\psi\rangle$, which an indifferent commutative theory could exploit but a local one could not. Theorem 137 rules out this possibility, and in that sense is a strengthening of Bell's Theorem.

The next result places limits on decomposition invariance.

Theorem 138

(i) No theory satisfies indifference, robustness, and decomposition invariance.

(ii) No theory has the property that

$$P(\rho, U) = \sum_{i=1}^{N} p_i P(|\psi_i\rangle\langle\psi_i|, U)$$

for every decomposition $\sum_{i=1}^{N} p_i |\psi_i\rangle\langle\psi_i|$ *of* ρ.

Proof.

(i) Suppose the contrary. Let

$$R_\theta = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix},$$
$$|\varphi_\theta\rangle = \cos\theta|0\rangle + \sin\theta|1\rangle.$$

Then for every θ not a multiple of $\pi/2$, we must have

$$S(|\varphi_{-\theta}\rangle, R_\theta) = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix},$$
$$S(|\varphi_{\pi/2-\theta}\rangle, R_\theta) = \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix}.$$

So by decomposition invariance, letting $I = (|0\rangle\langle 0| + |1\rangle\langle 1|)/2$ denote the maximally mixed state,

$$S(I, R_\theta) = S\left(\frac{|\varphi_{-\theta}\rangle\langle\varphi_{-\theta}| + |\varphi_{\pi/2-\theta}\rangle\langle\varphi_{\pi/2-\theta}|}{2}, R_\theta\right) = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}$$

and therefore

$$P(I, R_\theta) = \begin{bmatrix} \frac{(\rho)_{00}}{2} & \frac{(\rho)_{11}}{2} \\ \frac{(\rho)_{00}}{2} & \frac{(\rho)_{11}}{2} \end{bmatrix} = \begin{bmatrix} \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} \end{bmatrix}.$$

By robustness, this holds for $\theta = 0$ as well. But this is a contradiction, since by indifference $P(I, R_0)$ must be half the identity.

(ii) Suppose the contrary; then

$$P\left(I, R_{\pi/8}\right) = \frac{P\left(|0\rangle, R_{\pi/8}\right) + P\left(|1\rangle, R_{\pi/8}\right)}{2}.$$

So considering transitions from $|0\rangle$ to $|1\rangle$,

$$\left(P\left(I, R_{\pi/8}\right)\right)_{01} = \frac{\left(P\left(|0\rangle, R_{\pi/8}\right)\right)_{11} + 0}{2} = \frac{1}{2}\sin^2\frac{\pi}{8}.$$

But

$$P\left(I, R_{\pi/8}\right) = \frac{P\left(\left|\varphi_{\pi/8}\right\rangle, R_{\pi/8}\right) + P\left(\left|\varphi_{5\pi/8}\right\rangle, R_{\pi/8}\right)}{2}$$

also. Since $R_{\pi/8}\left|\varphi_{\pi/8}\right\rangle = \left|\varphi_{\pi/4}\right\rangle$, we have

$$\begin{aligned} \left(P\left(I, R_{\pi/8}\right)\right)_{01} &\geq \frac{1}{2}\left(P\left(\left|\varphi_{\pi/8}\right\rangle, R_{\pi/8}\right)\right)_{01} \\ &\geq \frac{1}{2}\left(\frac{1}{2} - \left(P\left(\left|\varphi_{\pi/8}\right\rangle, R_{\pi/8}\right)\right)_{11}\right) \\ &\geq \frac{1}{2}\left(\frac{1}{2} - \sin^2\frac{\pi}{8}\right) \\ &> \frac{1}{2}\sin^2\frac{\pi}{8} \end{aligned}$$

which is a contradiction.

■

Notice that all three conditions in Theorem 138, part (i) were essential—for $\mathcal{PT}$ satisfies robustness and decomposition invariance, $\mathcal{DT}$ satisfies indifference and decomposition invariance, and $\mathcal{FT}$ satisfies indifference and robustness.

The last impossibility result says that no hidden-variable theory satisfies both indifference and "strong continuity," in the sense that for all $\varepsilon > 0$ there exists $\delta > 0$ such that $\|\widetilde{\rho} - \rho\| \leq \delta$ implies $\|S(\widetilde{\rho}, U) - S(\rho, U)\| \leq \varepsilon$. To see this, let

$$\begin{aligned} U &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}, \\ \rho &= \sqrt{1 - 2\delta^2}\,|0\rangle + \delta\,|1\rangle + \delta\,|2\rangle, \\ \widetilde{\rho} &= \sqrt{1 - 2\delta^2}\,|0\rangle + \delta\,|1\rangle - \delta\,|2\rangle. \end{aligned}$$

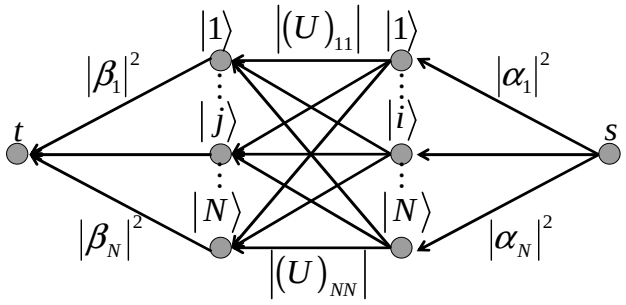

Figure 16.1: A network (weighted directed graph with source and sink) corresponding to the unitary U and state $|\psi\rangle$

Then by indifference,

$$S(\rho, U) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}, \qquad S(\widetilde{\rho}, U) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}.$$

This is the reason why I defined robustness in terms of the joint probabilities matrix P rather than the stochastic matrix S. On the other hand, note that by giving up indifference, one *can* satisfy strong continuity, as is shown by $\mathcal{PT}$.

16.6 Specific Theories

This section presents two nontrivial examples of hidden-variable theories: the flow theory in Section 16.6.1, and the Schrödinger theory in Section 16.6.2.

16.6.1 Flow Theory

The idea of the flow theory is to convert a unitary matrix into a weighted directed graph, and then route probability mass through that graph like oil through pipes. Given a unitary U, let

$$\begin{bmatrix} \beta_1 \\ \vdots \\ \beta_N \end{bmatrix} = \begin{bmatrix} (U)_{11} & \cdots & (U)_{N1} \\ \vdots & & \vdots \\ (U)_{1N} & \cdots & (U)_{NN} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_N \end{bmatrix},$$

where for the time being

$$|\psi\rangle = \alpha_1 |1\rangle + \cdots + \alpha_N |N\rangle,$$
$$U|\psi\rangle = \beta_1 |1\rangle + \cdots + \beta_N |N\rangle$$

are pure states. Then consider the network G shown in Figure 16.1. We have a source vertex s, a sink vertex t, and N input and N output vertices labeled by basis states $|1\rangle, \ldots, |N\rangle$.

Each edge of the form $(s, |i\rangle)$ has capacity $|\alpha_i|^2$, each edge $(|i\rangle, |j\rangle)$ has capacity $\left|(U)_{ij}\right|$, and each edge $(|j\rangle, t)$ has capacity $|\beta_j|^2$. A natural question is how much probability mass can flow from s to t without violating the capacity constraints. Rather surprisingly, I will show that one unit of mass (that is, all of it) can. Interestingly, this result would be false if edge $(|i\rangle, |j\rangle)$ had capacity $\left|(U)_{ij}\right|^2$ (or even $\left|(U)_{ij}\right|^{1+\varepsilon}$) instead of $\left|(U)_{ij}\right|$. I will also show that there exists a mapping from networks to maximal flows in those networks, that is *robust* in the sense that a small change in edge capacities produces only a small change in the amount of flow through any edge.

The proofs of these theorems use classical results from the theory of network flows (see [88] for an introduction). In particular, let a *cut* be a set of edges that separates s from t; the *value* of a cut is the sum of the capacities of its edges. Then a fundamental result called the *Max-Flow-Min-Cut Theorem* [111] says that the maximum possible amount of flow from s to t equals the minimum value of any cut. Using that result I can show the following.

Theorem 139 *One unit of flow can be routed from s to t in G.*

Proof. By the above, it suffices to show that any cut C in G has value at least 1. Let A be the set of $i \in \{1, \ldots, N\}$ such that $(s, |i\rangle) \notin C$, and let B be the set of j such that $(|j\rangle, t) \notin C$. Then C must contain every edge $(|i\rangle, |j\rangle)$ such that $i \in A$ and $j \in B$, and we can assume without loss of generality that C contains no other edges. So the value of C is

$$\sum_{i \notin A} |\alpha_i|^2 + \sum_{j \notin B} |\beta_j|^2 + \sum_{i \in A,\ j \in B} \left|(U)_{ij}\right|.$$

Therefore we need to prove the matrix inequality

$$\left(1 - \sum_{i \in A} |\alpha_i|^2\right) + \left(1 - \sum_{j \in B} |\beta_j|^2\right) + \sum_{i \in A,\ j \in B} \left|(U)_{ij}\right| \geq 1,$$

or

$$1 + \sum_{i \in A,\ j \in B} \left|(U)_{ij}\right| \geq \sum_{i \in A} |\alpha_i|^2 + \sum_{j \in B} |\beta_j|^2. \tag{16.1}$$

Let U be fixed, and consider the maximum of the right-hand side of equation (16.1) over all $|\psi\rangle$. Since

$$\beta_j = \sum_i (U)_{ij} \alpha_i,$$

this maximum is equal to the largest eigenvalue λ of the positive semidefinite matrix

$$\sum_{i \in A} |i\rangle \langle i| + \sum_{j \in B} |u_j\rangle \langle u_j|$$

where for each j,

$$|u_j\rangle = (U)_{1j} |1\rangle + \cdots + (U)_{Nj} |N\rangle.$$

Let H_A be the subspace of states spanned by $\{|i\rangle : i \in A\}$, and let H_B be the subspace spanned by $\{|u_j\rangle : j \in B\}$. Also, let $L_A(|\psi\rangle)$ be the length of the projection of $|\psi\rangle$ onto H_A, and let $L_B(|\psi\rangle)$ be the length of the projection of $|\psi\rangle$ onto H_B. Then since the $|i\rangle$'s and $|u_j\rangle$'s form orthogonal bases for H_A and H_B respectively, we have

$$\begin{aligned}\lambda &= \max_{|\psi\rangle} \left(\sum_{i \in A} |\langle i|\psi\rangle|^2 + \sum_{j \in B} |\langle u_j|\psi\rangle|^2 \right) \\ &= \max_{|\psi\rangle} \left(L_A(|\psi\rangle)^2 + L_B(|\psi\rangle)^2 \right).\end{aligned}$$

So letting θ be the angle between H_A and H_B,

$$\begin{aligned}\lambda &= 2\cos^2\frac{\theta}{2} \\ &= 1 + \cos\theta \\ &\leq 1 + \max_{|a\rangle \in H_A,\ |b\rangle \in H_B} |\langle a|b\rangle| \\ &= 1 + \max_{\substack{|\gamma_1|^2+\cdots+|\gamma_N|^2=1 \\ |\delta_1|^2+\cdots+|\delta_N|^2=1}} \left| \left(\sum_{i \in A} \gamma_i \langle i| \right) \left(\sum_{j \in B} \delta_j |u_j\rangle \right) \right| \\ &\leq 1 + \sum_{i \in A,\ j \in B} \left| (U)_{ij} \right|\end{aligned}$$

which completes the theorem. ■

Observe that Theorem 139 still holds if U acts on a mixed state ρ, since we can write ρ as a convex combination of pure states $|\psi\rangle\langle\psi|$, construct a flow for each $|\psi\rangle$ separately, and then take a convex combination of the flows.

Using Theorem 139, I now define the flow theory $\mathcal{FT}$. Let $F(\rho, U)$ be the set of maximal flows for ρ, U—representable by $N \times N$ arrays of real numbers f_{ij} such that $0 \leq f_{ij} \leq \left| (U)_{ij} \right|$ for all i, j, and also

$$\sum_j f_{ij} = (\rho)_{ii}, \quad \sum_i f_{ij} = \left(U\rho U^{-1} \right)_{jj}.$$

Clearly $F(\rho, U)$ is a convex polytope, which Theorem 139 asserts is nonempty. Form a maximal flow $f^*(\rho, U) \in F(\rho, U)$ as follows: first let f^*_{11} be the maximum of f_{11} over all $f \in F(\rho, U)$. Then let f^*_{12} be the maximum of f_{12} over all $f \in F(\rho, U)$ such that $f_{11} = f^*_{11}$. Continue to loop through all i, j pairs in lexicographic order, setting each f^*_{ij} to its maximum possible value consistent with the $(i-1)N + j - 1$ previous values. Finally, let $(P)_{ij} = f^*_{ij}$ for all i, j. As discussed in Section 16.3, given P we can easily obtain the stochastic matrix S by dividing the i^{th} column by $(\rho)_{ii}$, or taking a limit in case $(\rho)_{ii} = 0$.

It is easy to check that $\mathcal{FT}$ so defined satisfies the indifference axiom. Showing that $\mathcal{FT}$ satisfies robustness is harder. Our proof is based on the Ford-Fulkerson algorithm [111], a classic algorithm for computing maximal flows that works by finding a sequence of "augmenting paths," each of which increases the flow from s to t by some positive amount.

Theorem 140 *$\mathcal{FT}$ satisfies robustness.*

Proof. Let G be an arbitrary flow network with source s, sink t, and directed edges $e_1, \ldots, e_m$, where each e_i has capacity c_i and leads from v_i to w_i. It will be convenient to introduce a fictitious edge e_0 from t to s with unlimited capacity; then maximizing the flow through G is equivalent to maximizing the flow through e_0. Suppose we produce a new network $\widetilde{G}$ by increasing a single capacity c_{i^*} by some $\varepsilon > 0$. Let f^* be the optimal flow for G, obtained by first maximizing the flow f_0 through e_0, then maximizing the flow f_1 through e_1 holding f_0 fixed, and so on up to f_m. Let $\widetilde{f}^*$ be the maximal flow for $\widetilde{G}$ produced in the same way. We claim that for all $i \in \{0, \ldots, m\}$,

$$\left|\widetilde{f}_i^* - f_i^*\right| \leq \varepsilon.$$

To see that the theorem follows from this claim: first, if f^* is robust under adding ε to c_{i^*}, then it must also be robust under subtracting ε from c_{i^*}. Second, if we change ρ, U to $\widetilde{\rho}, \widetilde{U}$ such that $\|\widetilde{\rho} - \rho\|_\infty \leq 1/q\,(N)$ and $\left\|\widetilde{U} - U\right\|_\infty \leq 1/q\,(N)$, then we can imagine the $N^2 + 2N$ edge capacities are changed one by one, so that

$$\begin{aligned}\left\|f^*\left(\widetilde{\rho}, \widetilde{U}\right) - f^*\left(\rho, U\right)\right\|_\infty &\leq \sum_{ij} \left|\left|\left(\widetilde{U}\right)_{ij}\right| - \left|\left(U\right)_{ij}\right|\right| + \sum_i \left|\left(\widetilde{\rho}\right)_{ii} - \left(\rho\right)_{ii}\right| \\ &\quad + \sum_j \left|\left(\widetilde{U}\widetilde{\rho}\widetilde{U}^{-1}\right)_{jj} - \left(U\rho U^{-1}\right)_{jj}\right| \\ &\leq \frac{4N^2}{q\,(N)}.\end{aligned}$$

(Here we have made no attempt to optimize the bound.)

We now prove the claim. To do so we describe an iterative algorithm for computing f^*. First maximize the flow f_0 through e_0, by using the Ford-Fulkerson algorithm to find a maximal flow from s to t. Let $f^{(0)}$ be the resulting flow, and let $G^{(1)}$ be the residual network that corresponds to $f^{(0)}$. For each i, that is, $G^{(1)}$ has an edge $e_i = (v_i, w_i)$ of capacity $c_i^{(1)} = c_i - f_i^{(0)}$, and an edge $\overline{e}_i = (w_i, v_i)$ of capacity $\overline{c}_i^{(1)} = f_i^{(0)}$. Next maximize f_1 subject to f_0 by using the Ford-Fulkerson algorithm to find "augmenting cycles" from w_1 to v_1 and back to w_1 in $G^{(1)} \setminus \{e_0, \overline{e}_0\}$. Continue in this manner until each of $f_1, \ldots, f_m$ has been maximized subject to the previous f_i's. Finally set $f^* = f^{(m)}$.

Now, one way to compute $\widetilde{f}^*$ is to start with f^*, then repeatedly "correct" it by applying the same iterative algorithm to maximize $\widetilde{f}_0$, then $\widetilde{f}_1$, and so on. Let $\varepsilon_i = \left|\widetilde{f}_i^* - f_i^*\right|$; then we need to show that $\varepsilon_i \leq \varepsilon$ for all $i \in \{0, \ldots, m\}$. The proof is by induction on i. Clearly $\varepsilon_0 \leq \varepsilon$, since increasing c_{i^*} by ε can increase the value of the minimum cut from s to t by at most ε. Likewise, after we maximize $\widetilde{f}_0$, the value of the minimum cut from w_1 to v_1 can increase by at most $\varepsilon - \varepsilon_0 + \varepsilon_0 = \varepsilon$. For of the at most ε new units of flow from w_1 to v_1 that increasing c_{i^*} made available, ε_0 of them were "taken up" in maximizing $\widetilde{f}_0$, but the process of maximizing $\widetilde{f}_0$ could have again increased the minimum cut from w_1 to v_1 by up to ε_0. Continuing in this way,

$$\varepsilon_2 \leq \varepsilon - \varepsilon_0 + \varepsilon_0 - \varepsilon_1 + \varepsilon_1 = \varepsilon,$$

and so on up to ε_m. This completes the proof. ■

That $\mathcal{FT}$ violates decomposition invariance now follows from Theorem 138, part (i). One can also show that $\mathcal{FT}$ violates product commutativity, by considering the following example: let $|\psi\rangle = |\varphi_{\pi/4}\rangle \otimes |\varphi_{-\pi/8}\rangle$ be a 2-qubit initial state, and let $R^A_{\pi/4}$ and $R^B_{\pi/4}$ be $\pi/4$ rotations applied to the first and second qubits respectively. Then

$$S\left(R^A_{\pi/4}|\psi\rangle, R^B_{\pi/4}\right) S\left(|\psi\rangle, R^A_{\pi/4}\right) \neq S\left(R^B_{\pi/4}|\psi\rangle, R^A_{\pi/4}\right) S\left(|\psi\rangle, R^B_{\pi/4}\right).$$

We omit a proof for brevity.

16.6.2 Schrödinger Theory

The final hidden-variable theory, which I call the *Schrödinger theory* or $\mathcal{ST}$, is the most interesting one mathematically. The idea—to make a matrix into a stochastic matrix via row and column rescaling—is natural enough that we came upon it independently, only later learning that it originated in a 1931 paper of Schrödinger [213]. The idea was subsequently developed by Fortet [112], Beurling [58], Nagasawa [179], and others. My goal is to give what (to my knowledge) is the first self-contained, reasonably accessible presentation of the main result in this area; and to interpret that result in what I think is the correct way: as providing one example of a hidden-variable theory, whose strengths and weaknesses should be directly compared to those of other theories.

Most of the technical difficulties in [58, 112, 179, 213] arise because the stochastic process being constructed involves continuous time and particle positions. Here I eliminate those difficulties by restricting attention to discrete time and finite-dimensional Hilbert spaces. I thereby obtain a generalized version[5] of a problem that computer scientists know as (r,c)-*scaling of matrices* [221, 118, 167].

As in the case of the flow theory, given a unitary U acting on a state ρ, the first step is to replace each entry of U by its absolute value, obtaining a nonnegative matrix $U^{(0)}$ defined by $\left(U^{(0)}\right)_{ij} := \left|(U)_{ij}\right|$. We then wish to find nonnegative column multipliers $\alpha_1, \ldots, \alpha_N$ and row multipliers $\beta_1, \ldots, \beta_N$ such that for all i, j,

$$\alpha_i \beta_1 \left(U^{(0)}\right)_{i1} + \cdots + \alpha_i \beta_N \left(U^{(0)}\right)_{iN} = (\rho)_{ii}, \tag{16.2}$$

$$\alpha_1 \beta_j \left(U^{(0)}\right)_{1j} + \cdots + \alpha_N \beta_j \left(U^{(0)}\right)_{Nj} = \left(U\rho U^{-1}\right)_{jj}. \tag{16.3}$$

If we like, we can interpret the α_i's and β_j's as dynamical variables that reach equilibrium precisely when equations (16.2) and (16.3) are satisfied. Admittedly, it might be thought physically implausible that such a complicated dynamical process should take place at every instant of time. On the other hand, it is hard to imagine a more "benign" way to convert $U^{(0)}$ into a joint probabilities matrix, than by simply rescaling its rows and columns.

I will show that multipliers satisfying (16.2) and (16.3) always exist. The intuition of a dynamical process reaching equilibrium turns out to be key to the proof. For all $t \geq 0$,

[5] In (r,c)-scaling, we are given an invertible real matrix, and the goal is to rescale all rows and columns to sum to 1. The generalized version is to rescale the rows and columns to given values (not necessarily 1).

let

$$\left(U^{(2t+1)}\right)_{ij} = \frac{(\rho)_{ii}}{\sum_k \left(U^{(2t)}\right)_{ik}} \left(U^{(2t)}\right)_{ij},$$
$$\left(U^{(2t+2)}\right)_{ij} = \frac{\left(U\rho U^{-1}\right)_{jj}}{\sum_k \left(U^{(2t+1)}\right)_{kj}} \left(U^{(2t+1)}\right)_{ij}.$$

In words, we obtain $U^{(2t+1)}$ by normalizing each column i of $U^{(2t)}$ to sum to $(\rho)_{ii}$; likewise we obtain $U^{(2t+2)}$ by normalizing each row j of $U^{(2t+1)}$ to sum to $\left(U\rho U^{-1}\right)_{jj}$. The crucial fact is that the above process always converges to some $P(\rho, U) = \lim_{t\to\infty} U^{(t)}$. We can therefore take

$$\alpha_i = \prod_{t=0}^{\infty} \frac{(\rho)_{ii}}{\sum_k \left(U^{(2t)}\right)_{ik}},$$
$$\beta_j = \prod_{t=0}^{\infty} \frac{\left(U\rho U^{-1}\right)_{jj}}{\sum_k \left(U^{(2t+1)}\right)_{kj}}$$

for all i, j. Although I will not prove it here, it turns out that this yields the *unique* solution to equations (16.2) and (16.3), up to a global rescaling of the form $\alpha_i \to \alpha_i c$ for all i and $\beta_j \to \beta_j / c$ for all j [179].

The convergence proof will reuse a result about network flows from Section 16.6.1, in order to define a nondecreasing "progress measure" based on Kullback-Leibler distance.

Theorem 141 *The limit $P(\rho, U) = \lim_{t\to\infty} U^{(t)}$ exists.*

Proof. A consequence of Theorem 139 is that for every ρ, U, there exists an $N \times N$ array of nonnegative real numbers f_{ij} such that

(1) $f_{ij} = 0$ whenever $\left|(U)_{ij}\right| = 0$,

(2) $f_{i1} + \cdots + f_{iN} = (\rho)_{ii}$ for all i, and

(3) $f_{1j} + \cdots + f_{Nj} = \left(U\rho U^{-1}\right)_{jj}$ for all j.

Given any such array, define a progress measure

$$Z^{(t)} = \prod_{ij} \left(U^{(t)}\right)_{ij}^{f_{ij}},$$

where we adopt the convention $0^0 = 1$. We claim that $Z^{(t+1)} \geq Z^{(t)}$ for all $t \geq 1$. To see this, assume without loss of generality that we are on an odd step $2t + 1$, and let

$C_i^{(2t)} = \sum_j \left(U^{(2t)}\right)_{ij}$ be the i^{th} column sum before we normalize it. Then

$$\begin{aligned} Z^{(2t+1)} &= \prod_{ij} \left(U^{(2t+1)}\right)_{ij}^{f_{ij}} \\ &= \prod_{ij} \left(\frac{(\rho)_{ii}}{C_i^{(2t)}} \left(U^{(2t)}\right)_{ij}\right)^{f_{ij}} \\ &= \left(\prod_{ij} \left(U^{(2t)}\right)_{ij}^{f_{ij}}\right) \left(\prod_i \left(\frac{(\rho)_{ii}}{C_i^{(2t)}}\right)^{f_{i1}+\cdots+f_{iN}}\right) \\ &= Z^{(2t)} \cdot \prod_i \left(\frac{(\rho)_{ii}}{C_i^{(2t)}}\right)^{(\rho)_{ii}}. \end{aligned}$$

As a result of the $2t^{th}$ normalization step, we had $\sum_i C_i^{(2t)} = 1$. Subject to that constraint, the maximum of

$$\prod_i \left(C_i^{(2t)}\right)^{(\rho)_{ii}}$$

over the $C_i^{(2t)}$'s occurs when $C_i^{(2t)} = (\rho)_{ii}$ for all i—a simple calculus fact that follows from the nonnegativity of Kullback-Leibler distance. This implies that $Z^{(2t+1)} \geq Z^{(2t)}$. Similarly, normalizing rows leads to $Z^{(2t+2)} \geq Z^{(2t+1)}$.

It follows that the limit $P(\rho, U) = \lim_{t\to\infty} U^{(t)}$ exists. For suppose not; then some $C_i^{(t)}$ is bounded away from $(\rho)_{ii}$, so there exists an $\varepsilon > 0$ such that $Z^{(t+1)} \geq (1+\varepsilon) Z^{(t)}$ for all even t. But this is a contradiction, since $Z^{(0)} > 0$ and $Z^{(t)} \leq 1$ for all t. ■

Besides showing that $P(\rho, U)$ is well-defined, Theorem 141 also yields a procedure to *calculate* $P(\rho, U)$ (as well as the α_i's and β_j's). It can be shown that this procedure converges to within entrywise error ε after a number steps polynomial in N and $1/\varepsilon$. Also, once we have $P(\rho, U)$, the stochastic matrix $S(\rho, U)$ is readily obtained by normalizing each column of $P(\rho, U)$ to sum to 1. This completes the definition of the Schrödinger theory $\mathcal{ST}$.

It is immediate that $\mathcal{ST}$ satisfies indifference. Also:

Proposition 142 *$\mathcal{ST}$ satisfies product commutativity.*

Proof. Given a state $|\psi\rangle = |\psi_A\rangle \otimes |\psi_B\rangle$, let $U_A \otimes I$ act only on $|\psi_A\rangle$ and let $I \otimes U_B$ act only on $|\psi_B\rangle$. Then we claim that

$$S(|\psi\rangle, U_A \otimes I) = S(|\psi_A\rangle, U_A) \otimes I.$$

The reason is simply that multiplying all amplitudes in $|\psi_A\rangle$ and $U_A|\psi_A\rangle$ by a constant factor α_x, as we do for each basis state $|x\rangle$ of $|\psi_B\rangle$, has no effect on the scaling procedure that produces $S(|\psi_A\rangle, U_A)$. Similarly

$$S(|\psi\rangle, I \otimes U_B) = I \otimes S(|\psi_B\rangle, U_B).$$

It follows that

$$\begin{aligned} S\left(|\psi_A\rangle, U_A\right) \otimes S\left(|\psi_B\rangle, U_B\right) &= S\left(U_A|\psi_A\rangle \otimes |\psi_B\rangle, I \otimes U_B\right) S\left(|\psi\rangle, U_A \otimes I\right) \\ &= S\left(|\psi_A\rangle \otimes U_B|\psi_B\rangle, U_A \otimes I\right) S\left(|\psi\rangle, I \otimes U_B\right). \end{aligned}$$

■

On the other hand, numerical simulations readily show that $\mathcal{ST}$ violates decomposition invariance, even when $N = 2$ (I omit a concrete example for brevity).

16.7 The Computational Model

I now explain the histories model of computation, building up to the complexity class DQP. From now on, the states ρ that we consider will always be pure states of $\ell = \log_2 N$ qubits. That is, $\rho = |\psi\rangle\langle\psi|$ where

$$|\psi\rangle = \sum_{x \in \{0,1\}^\ell} \alpha_x |x\rangle.$$

The algorithms of this chapter will work under *any* hidden-variable theory that satisfies the indifference axiom. On the other hand, if we take into account that even in theory (let alone in practice), a generic unitary cannot be represented exactly with a finite universal gate set, only approximated arbitrarily well, then we also need the robustness axiom. Thus, it is reassuring that there *exists* a hidden-variable theory (namely $\mathcal{FT}$) that satisfies both indifference and robustness.

Let a quantum computer have the initial state $|0\rangle^{\otimes\ell}$, and suppose we apply a sequence $\mathcal{U} = (U_1, \ldots, U_T)$ of unitary operations, each of which is implemented by a polynomial-size quantum circuit. Then a *history* of a hidden variable through the computation is a sequence $H = (v_0, \ldots, v_T)$ of basis states, where v_t is the variable's value immediately after U_t is applied (thus $v_0 = |0\rangle^{\otimes\ell}$). Given any hidden-variable theory $\mathcal{T}$, we can obtain a probability distribution $\Omega(\mathcal{U}, \mathcal{T})$ over histories by just applying $\mathcal{T}$ repeatedly, once for each U_t, to obtain the stochastic matrices

$$S\left(|0\rangle^{\otimes\ell}, U_1\right), \quad S\left(U_1|0\rangle^{\otimes\ell}, U_2\right), \quad \ldots \quad S\left(U_{T-1}\cdots U_1|0\rangle^{\otimes\ell}, U_T\right).$$

Note that $\Omega(\mathcal{U}, \mathcal{T})$ is a Markov distribution; that is, each v_t is independent of the other v_i's conditioned on v_{t-1} and v_{t+1}. Admittedly, $\Omega(\mathcal{U}, \mathcal{T})$ could depend on the precise way in which the combined circuit $U_T \cdots U_1$ is "sliced" into component circuits $U_1, \ldots, U_T$. But as noted in Section 16.3.2, such dependence on the granularity of unitaries is unavoidable in any hidden-variable theory other than $\mathcal{PT}$.

Given a hidden-variable theory $\mathcal{T}$, let $\mathcal{O}(\mathcal{T})$ be an oracle that takes as input a positive integer ℓ, and a sequence of quantum circuits $\mathcal{U} = (U_1, \ldots, U_T)$ that act on ℓ qubits. Here each U_t is specified by a sequence $\left(g_{t,1}, \ldots, g_{t,m(t)}\right)$ of gates chosen from some finite universal gate set $\mathcal{G}$. The oracle $\mathcal{O}(\mathcal{T})$ returns as output a sample $(v_0, \ldots, v_T)$ from the history distribution $\Omega(\mathcal{U}, \mathcal{T})$ defined previously. Now let A be a deterministic classical Turing machine that is given oracle access to $\mathcal{O}(\mathcal{T})$. The machine A receives an input x, makes a single oracle query to $\mathcal{O}(\mathcal{T})$, then produces an output based on the response. We

say a set of strings L is in DQP if there exists an A such that for all sufficiently large n and inputs $x \in \{0,1\}^n$, and all theories $\mathcal{T}$ satisfying the indifference and robustness axioms, A correctly decides whether $x \in L$ with probability at least $2/3$, in time polynomial in n.

Let me make some remarks about the above definition. There is no real significance in the requirement that A be deterministic and classical, and that it be allowed only one query to $\mathcal{O}(\mathcal{T})$. I made this choice only because it suffices for the upper bounds; it might be interesting to consider the effects of other choices. However, other aspects of the definition are not arbitrary. The order of quantifiers matters; we want a single A that works for *any* hidden-variable theory satisfying indifference and robustness. Also, we require A to succeed only for sufficiently large n since by choosing a large enough polynomial $q(N)$ in the statement of the robustness axiom, an adversary might easily make A incorrect on a finite number of instances.

16.7.1 Basic Results

Having defined the complexity class DQP, in this subsection I establish its most basic properties. First of all, it is immediate that $\mathsf{BQP} \subseteq \mathsf{DQP}$; that is, sampling histories is at least as powerful as standard quantum computation. For v_1, the first hidden-variable value returned by $\mathcal{O}(\mathcal{T})$, can be seen as simply the result of applying a polynomial-size quantum circuit U_1 to the initial state $|0\rangle^{\otimes \ell}$ and then measuring in the standard basis. A key further observation is the following.

Theorem 143 *Any universal gate set yields the same complexity class* DQP*. By universal, we mean that any unitary matrix (real or complex) can be approximated, without the need for ancilla qubits.*

Proof. Let $\mathcal{G}$ and $\mathcal{G}'$ be universal gate sets. Also, let $\mathcal{U} = (U_1, \ldots, U_T)$ be a sequence of ℓ-qubit unitaries, each specified by a polynomial-size quantum circuit over $\mathcal{G}$. We have $T, \ell = O(\text{poly}(n))$ where n is the input length. We can also assume without loss of generality that $\ell \geq n$, since otherwise we simply insert $n - \ell$ dummy qubits that are never acted on (by the indifference axiom, this will not affect the results). We want to approximate $\mathcal{U}$ by another sequence of ℓ-qubit unitaries, $\mathcal{U}' = (U'_1, \ldots, U'_T)$, where each U'_t is specified by a quantum circuit over $\mathcal{G}'$. In particular, for all t we want $\|U'_t - U_t\|_\infty \leq 2^{-\ell^2 T}$. By the Solovay-Kitaev Theorem [153, 182], we can achieve this using $\text{poly}\left(n, \ell^2 T\right) = \text{poly}(n)$ gates from $\mathcal{G}'$; moreover, the circuit for U'_t can be constructed in polynomial time given the circuit for U_t.

Let $|\psi_t\rangle = U_t \cdots U_1 |0\rangle^{\otimes \ell}$ and $|\psi'_t\rangle = U'_t \cdots U'_1 |0\rangle^{\otimes \ell}$. Notice that for all $t \in \{1, \ldots, T\}$,

$$\begin{aligned} \left\| |\psi'_t\rangle - |\psi_t\rangle \right\|_\infty &\leq 2^\ell \left(\left\| |\psi'_{t-1}\rangle - |\psi_{t-1}\rangle \right\|_\infty + 2^{-\ell^2 T} \right) \\ &\leq T 2^{\ell T} \left(2^{-\ell^2 T} \right) = T 2^{-\ell(\ell-1)T}, \end{aligned}$$

since $\left\| |\psi'_0\rangle - |\psi_0\rangle \right\|_\infty = 0$. Here $\| \|_\infty$ denotes the maximum entrywise difference between two vectors in $\mathbb{C}^{2^\ell}$. Also, given a theory $\mathcal{T}$, let P_t and P'_t be the joint probabilities matrices

corresponding to U_t and U'_t respectively. Then by the robustness axiom, there exists a polynomial q such that if $\|U'_t - U_t\|_\infty \leq 1/q\left(2^\ell\right)$ and $\left\||\psi'_{t-1}\rangle - |\psi_{t-1}\rangle\right\|_\infty \leq 1/q\left(2^\ell\right)$, then $\|P_t - P'_t\|_\infty \leq 2^{-3\ell}$. For all such polynomials q, we have $2^{-\ell^2 T} \leq 1/q\left(2^\ell\right)$ and $T2^{-\ell(\ell-1)T} \leq 1/q\left(2^\ell\right)$ for sufficiently large $n \leq \ell$. Therefore $\|P_t - P'_t\|_\infty \leq 2^{-3\ell}$ for all t and sufficiently large n.

Now assume n is sufficiently large, and consider the distributions $\Omega\left(\mathcal{U}, \mathcal{T}\right)$ and $\Omega\left(\mathcal{U}', \mathcal{T}\right)$ over classical histories $H = (v_0, \ldots, v_T)$. For all $t \in \{1, \ldots, T\}$ and $x \in \{0,1\}^\ell$, we have

$$\left| \Pr_{\Omega(\mathcal{U},\mathcal{T})}\left[v_t = |x\rangle\right] - \Pr_{\Omega(\mathcal{U}',\mathcal{T})}\left[v_t = |x\rangle\right] \right| \leq 2^\ell \left(2^{-3\ell}\right) = 2^{-2\ell}.$$

It follows by the union bound that the variation distance $\|\Omega\left(\mathcal{U}', \mathcal{T}\right) - \Omega\left(\mathcal{U}, \mathcal{T}\right)\|$ is at most

$$T2^\ell \left(2^{-2\ell}\right) = \frac{T}{2^\ell} \leq \frac{T}{2^n}.$$

In other words, $\Omega\left(\mathcal{U}', \mathcal{T}\right)$ can be distinguished from $\Omega\left(\mathcal{U}, \mathcal{T}\right)$ with bias at most $T/2^n$, which is exponentially small. So any classical postprocessing algorithm that succeeds with high probability given $H \in \Omega\left(\mathcal{U}, \mathcal{T}\right)$, also succeeds with high probability given $H \in \Omega\left(\mathcal{U}', \mathcal{T}\right)$. This completes the theorem. ■

Unfortunately, the best upper bound on DQP I have been able to show is DQP $\subseteq$ EXP; that is, any problem in DQP is solvable in deterministic exponential time. The proof is trivial: let $\mathcal{T}$ be the flow theory $\mathcal{FT}$. Then by using the Ford-Fulkerson algorithm, we can clearly construct the requisite maximum flows in time polynomial in 2^ℓ (hence exponential in n), and thereby calculate the probability of each possible history $(v_1, \ldots, v_T)$ to suitable precision.

16.8 The Juggle Subroutine

This section presents a crucial subroutine that will be used in both main algorithms: the algorithm for simulating statistical zero knowledge in Section 16.9, and the algorithm for search in $N^{1/3}$ queries in Section 16.10. Given an ℓ-qubit state $\left(|a\rangle + |b\rangle\right)/\sqrt{2}$, where $|a\rangle$ and $|b\rangle$ are unknown basis states, the goal of the juggle subroutine is to learn both a and b. The name arises because the strategy will be to "juggle" a hidden variable, so that if it starts out at $|a\rangle$ then with non-negligible probability it transitions to $|b\rangle$, and vice versa. Inspecting the entire history of the hidden variable will then reveal both a and b, as desired.

To produce this behavior, we will exploit a basic feature of quantum mechanics: that observable information in one basis can become unobservable phase information in a different basis. We will apply a sequence of unitaries that hide all information about a and b in phases, thereby forcing the hidden variable to "forget" whether it started at $|a\rangle$ or $|b\rangle$. We will then invert those unitaries to return the state to $\left(|a\rangle + |b\rangle\right)/\sqrt{2}$, at which point the hidden variable, having "forgotten" its initial value, must be unequal to that value with probability 1/2.

I now give the subroutine. Let $|\psi\rangle = \left(|a\rangle + |b\rangle\right)/\sqrt{2}$ be the initial state. The first unitary, U_1, consists of Hadamard gates on $\ell - 1$ qubits chosen uniformly at random, and the identity operation on the remaining qubit, i. Next U_2 consists of a Hadamard gate

on qubit i. Finally U_3 consists of Hadamard gates on all ℓ qubits. Let $a = a_1 \ldots a_\ell$ and $b = b_1 \ldots b_\ell$. Then since $a \neq b$, we have $a_i \neq b_i$ with probability at least $1/\ell$. Assuming that occurs, the state

$$U_1 |\psi\rangle = \frac{1}{2^{\ell/2}} \left(\sum_{z \in \{0,1\}^\ell \,:\, z_i = a_i} (-1)^{a \cdot z - a_i z_i} |z\rangle + \sum_{z \in \{0,1\}^\ell \,:\, z_i = b_i} (-1)^{b \cdot z - b_i z_i} |z\rangle \right)$$

assigns nonzero amplitude to all 2^ℓ basis states. Then $U_2 U_1 |\psi\rangle$ assigns nonzero amplitude to $2^{\ell-1}$ basis states $|z\rangle$, namely those for which $a \cdot z \equiv b \cdot z \pmod 2$. Finally $U_3 U_2 U_1 |\psi\rangle = |\psi\rangle$.

Let v_t be the value of the hidden variable after U_t is applied. Then assuming $a_i \neq b_i$, I claim that v_3 is independent of v_0. So in particular, if $v_0 = |a\rangle$ then $v_3 = |b\rangle$ with $1/2$ probability, and if $v_0 = |b\rangle$ then $v_3 = |a\rangle$ with $1/2$ probability. To see this, observe that when U_1 is applied, there is no interference between basis states $|z\rangle$ such that $z_i = a_i$, and those such that $z_i = b_i$. So by the indifference axiom, the probability mass at $|a\rangle$ must spread out evenly among all $2^{\ell-1}$ basis states that agree with a on the i^{th} bit, and similarly for the probability mass at $|b\rangle$. Then after U_2 is applied, v_2 can differ from v_1 only on the i^{th} bit, again by the indifference axiom. So each basis state of $U_2 U_1 |\psi\rangle$ must receive an equal contribution from probability mass originating at $|a\rangle$, and probability mass originating at $|b\rangle$. Therefore v_2 is independent of v_0, from which it follows that v_3 is independent of v_0 as well.

Unfortunately, the juggle subroutine only works with probability $1/(2\ell)$—for it requires that $a_i \neq b_i$, and even then, inspecting the history $(v_0, v_1, \ldots)$ only reveals both $|a\rangle$ and $|b\rangle$ with probability $1/2$. Furthermore, the definition of DQP does not allow more than one call to the history oracle. However, all we need to do is pack multiple subroutine calls into a single oracle call. That is, choose U_4 similarly to U_1 (except with a different value of i), and set $U_5 = U_2$ and $U_6 = U_3$. Do the same with U_7, U_8, and U_9, and so on. Since $U_3, U_6, U_9, \ldots$ all return the quantum state to $|\psi\rangle$, the effect is that of multiple independent juggle attempts. With $2\ell^2$ attempts, we can make the failure probability at most $(1 - 1/(2\ell))^{2\ell^2} < e^{-\ell}$.

As a final remark, it is easy to see that the juggle subroutine works equally well with states of the form $|\psi\rangle = (|a\rangle - |b\rangle)/\sqrt{2}$. This will prove useful in Section 16.10.

16.9 Simulating SZK

This section shows that SZK $\subseteq$ DQP. Here SZK, or Statistical Zero Knowledge, was originally defined as the class of problems that possess a certain kind of "zero-knowledge proof protocol"—that is, a protocol between an omniscient prover and a verifier, by which the verifier becomes convinced of the answer to a problem, yet without learning anything else about the problem. However, for present purposes this cryptographic definition of SZK is irrelevant. For Sahai and Vadhan [209] have given an alternate and much simpler characterization: a problem is in SZK if and only if it can be reduced to a problem called Statistical Difference, which involves deciding whether two probability distributions are close or far.

More formally, let P_0 and P_1 be functions that map n-bit strings to $q(n)$-bit strings for some polynomial q, and that are specified by classical polynomial-time algorithms. Let Λ_0 and Λ_1 be the probability distributions over $P_0(x)$ and $P_1(x)$ respectively, if $x \in \{0,1\}^n$ is chosen uniformly at random. Then the problem is to decide whether $\|\Lambda_0 - \Lambda_1\|$ is less than $1/3$ or greater than $2/3$, given that one of these is the case. Here

$$\|\Lambda_0 - \Lambda_1\| = \frac{1}{2} \sum_{y \in \{0,1\}^{q(n)}} \left| \Pr_{x \in \{0,1\}^n} [P_0(x) = y] - \Pr_{x \in \{0,1\}^n} [P_1(x) = y] \right|$$

is the variation distance between Λ_0 and Λ_1.

To illustrate, let us see why Graph Isomorphism is in SZK. Given two graphs G_0 and G_1, take Λ_0 to be the uniform distribution over all permutations of G_0, and Λ_1 to be uniform over all permutations of G_1. This way, if G_0 and G_1 are isomorphic, then Λ_0 and Λ_1 will be identical, so $\|\Lambda_0 - \Lambda_1\| = 0$. On the other hand, if G_0 and G_1 are non-isomorphic, then Λ_0 and Λ_1 will be perfectly distinguishable, so $\|\Lambda_0 - \Lambda_1\| = 1$. Since Λ_0 and Λ_1 are clearly samplable by polynomial-time algorithms, it follows that any instance of Graph Isomorphism can be expressed as an instance of Statistical Difference. For a proof that Approximate Shortest Vector is in SZK, the reader is referred to Goldreich and Goldwasser [129] (see also Aharonov and Ta-Shma [23]).

The proof will use the following "amplification lemma" from [209]:[6]

Lemma 144 (Sahai and Vadhan) *Given efficiently-samplable distributions Λ_0 and Λ_1, we can construct new efficiently-samplable distributions Λ_0' and Λ_1', such that if $\|\Lambda_0 - \Lambda_1\| \leq 1/3$ then $\|\Lambda_0' - \Lambda_1'\| \leq 2^{-n}$, while if $\|\Lambda_0 - \Lambda_1\| \geq 2/3$ then $\|\Lambda_0' - \Lambda_1'\| \geq 1 - 2^{-n}$.*

In particular, Lemma 144 means we can assume without loss of generality that either $\|\Lambda_0 - \Lambda_1\| \leq 2^{-n^c}$ or $\|\Lambda_0 - \Lambda_1\| \geq 1 - 2^{-n^c}$ for some constant $c > 0$.

Having covered the necessary facts about SZK, we can now proceed to the main result.

Theorem 145 $\mathsf{SZK} \subseteq \mathsf{DQP}$.

Proof. We show how to solve Statistical Difference by using a history oracle. For simplicity, we start with the special case where P_0 and P_1 are both one-to-one functions. In this case, the circuit sequence $\mathcal{U}$ given to the history oracle does the following: it first prepares the state

$$\frac{1}{2^{(n+1)/2}} \sum_{b \in \{0,1\}, x \in \{0,1\}^n} |b\rangle |x\rangle |P_b(x)\rangle .$$

It then applies the juggle subroutine to the joint state of the $|b\rangle$ and $|x\rangle$ registers, taking $\ell = n + 1$. Notice that by the indifference axiom, the hidden variable will never transition from one value of $P_b(x)$ to another—exactly as if we had *measured* the third register in the standard basis. All that matters is the reduced state $|\psi\rangle$ of the first two registers, which

[6] Note that in this lemma, the constants $1/3$ and $2/3$ are not arbitrary; it is important for technical reasons that $(2/3)^2 > 1/3$.

has the form $(|0\rangle |x_0\rangle + |1\rangle |x_1\rangle)/\sqrt{2}$ for some x_0, x_1 if $\|\Lambda_0 - \Lambda_1\| = 0$, and $|b\rangle |x\rangle$ for some b, x if $\|\Lambda_0 - \Lambda_1\| = 1$. We have already seen that the juggle subroutine can distinguish these two cases: when the hidden-variable history is inspected, it will contain two values of the $|b\rangle$ register in the former case, and only one value in the latter case. Also, clearly the case $\|\Lambda_0 - \Lambda_1\| \leq 2^{-n^c}$ is statistically indistinguishable from $\|\Lambda_0 - \Lambda_1\| = 0$ with respect to the subroutine, and likewise $\|\Lambda_0 - \Lambda_1\| \geq 1 - 2^{-n^c}$ is indistinguishable from $\|\Lambda_0 - \Lambda_1\| = 1$.

We now consider the general case, where P_0 and P_1 need not be one-to-one. Our strategy is to reduce to the one-to-one case, by using a well-known hashing technique of Valiant and Vazirani [231]. Let $\mathcal{D}_{n,k}$ be the uniform distribution over all affine functions mapping $\{0,1\}^n$ to $\{0,1\}^k$, where we identify those sets with the finite fields $\mathbb{F}_2^n$ and $\mathbb{F}_2^k$ respectively. What Valiant and Vazirani showed is that, for all subsets $A \subseteq \{0,1\}^n$ such that $2^{k-2} \leq |A| \leq 2^{k-1}$, and all $s \in \{0,1\}^k$,

$$\Pr_{h \in \mathcal{D}_{n,k}} \left[\left| A \cap h^{-1}(s) \right| = 1 \right] \geq \frac{1}{8}.$$

As a corollary, the expectation over $h \in \mathcal{D}_{n,k}$ of

$$\left| \left\{ s \in \{0,1\}^k : \left| A \cap h^{-1}(s) \right| = 1 \right\} \right|$$

is at least $2^k/8$. It follows that, if x is drawn uniformly at random from A, then

$$\Pr_{h,x} \left[\left| A \cap h^{-1}(h(x)) \right| = 1 \right] \geq \frac{2^k/8}{|A|} \geq \frac{1}{4}.$$

This immediately suggests the following algorithm for the many-to-one case. Draw k uniformly at random from $\{2, \ldots, n+1\}$; then draw $h_0, h_1 \in \mathcal{D}_{n,k}$. Have $\mathcal{U}$ prepare the state

$$\frac{1}{2^{(n+1)/2}} \sum_{b \in \{0,1\}, x \in \{0,1\}^n} |b\rangle |x\rangle |P_b(x)\rangle |h_b(x)\rangle,$$

and then apply the juggle subroutine to the joint state of the $|b\rangle$ and $|x\rangle$ registers, ignoring the $|P_b(x)\rangle$ and $|h_b(x)\rangle$ registers as before.

Suppose $\|\Lambda_0 - \Lambda_1\| = 0$. Also, given $x \in \{0,1\}^n$ and $i \in \{0,1\}$, let $A_i = P_i^{-1}(P_i(x))$ and $H_i = h_i^{-1}(h_i(x))$, and suppose $2^{k-2} \leq |A_0| = |A_1| \leq 2^{k-1}$. Then

$$\Pr_{s,h_0,h_1} \left[|A_0 \cap H_0| = 1 \wedge |A_1 \cap H_1| = 1 \right] \geq \left(\frac{1}{4} \right)^2,$$

since the events $|A_0 \cap H_0| = 1$ and $|A_1 \cap H_1| = 1$ are independent of each other conditioned on x. Assuming both events occur, as before the juggle subroutine will reveal both $|0\rangle |x_0\rangle$ and $|1\rangle |x_1\rangle$ with high probability, where x_0 and x_1 are the unique elements of $A_0 \cap H_0$ and $A_1 \cap H_1$ respectively. By contrast, if $\|\Lambda_0 - \Lambda_1\| = 1$ then only one value of the $|b\rangle$ register will ever be observed. Again, replacing $\|\Lambda_0 - \Lambda_1\| = 0$ by $\|\Lambda_0 - \Lambda_1\| \leq 2^{-n^c}$, and $\|\Lambda_0 - \Lambda_1\| = 1$ by $\|\Lambda_0 - \Lambda_1\| \geq 1 - 2^{-n^c}$, can have only a negligible effect on the history distribution.

Of course, the probability that the correct value of k is chosen, and that $A_0 \cap H_0$ and $A_1 \cap H_1$ both have a unique element, could be as low as $1/(16n)$. To deal with this, we simply increase the number of calls to the juggle subroutine by an $O(n)$ factor, drawing new values of k, h_0, h_1 for each call. We pack multiple subroutine calls into a single oracle call as described in Section 16.8, except that now we uncompute the entire state (returning it to $|0\cdots 0\rangle$) and then recompute it between subroutine calls. A final remark: since the algorithm that calls the history oracle is deterministic, we "draw" new values of k, h_0, h_1 by having $\mathcal{U}$ prepare a uniform superposition over all possible values. The indifference axiom justifies this procedure, by guaranteeing that within each call to the juggle subroutine, the hidden-variable values of k, h_0, and h_1 remain constant. ■

Recall from Chapter 6 that there exists an oracle A relative to which $\mathsf{SZK}^A \not\subseteq \mathsf{BQP}^A$. By contrast, since Theorem 145 is easily seen to relativize, we have $\mathsf{SZK}^A \in \mathsf{DQP}^A$ for all oracles A. It follows that there exists an oracle A relative to which $\mathsf{BQP}^A \neq \mathsf{DQP}^A$.

16.10 Search in $N^{1/3}$ Queries

Given a Boolean function $f : \{0,1\}^n \rightarrow \{0,1\}$, the database search problem is simply to find a string x such that $f(x) = 1$. We can assume without loss of generality that this "marked item" x is unique.[7] We want to find it using as few queries to f as possible, where a query returns $f(y)$ given y.

Let $N = 2^n$. Then classically, of course, $\Theta(N)$ queries are necessary and sufficient. By querying f in superposition, Grover's algorithm [139] finds x using $O\left(N^{1/2}\right)$ queries, together with $\widetilde{O}\left(N^{1/2}\right)$ auxiliary computation steps (here the $\widetilde{O}$ hides a factor of the form $(\log N)^c$). Bennett et al. [51] showed that any quantum algorithm needs $\Omega\left(N^{1/2}\right)$ queries.

In this section, I show how to find the marked item by sampling histories, using only $O\left(N^{1/3}\right)$ queries and $\widetilde{O}\left(N^{1/3}\right)$ computation steps. Formally, the model is as follows. Each of the quantum circuits $U_1, \ldots, U_T$ that algorithm A gives to the history oracle $\mathcal{O}(\mathcal{T})$ is now able to query f. Suppose U_t makes q_t queries to f; then the total number of queries made by A is defined to be $Q = q_1 + \cdots + q_T$. The total number of *computation* steps is at least the number of steps required to write down $U_1, \ldots, U_T$, but could be greater.

Theorem 146 *In the* DQP *model, we can search a database of N items for a unique marked item using $O\left(N^{1/3}\right)$ queries and $\widetilde{O}\left(N^{1/3}\right)$ computation steps.*

Proof. Assume without loss of generality that $N = 2^n$ with $n|3$, and that each database item is labeled by an n-bit string. Let $x \in \{0,1\}^n$ be the label of the unique marked item. Then the sequence of quantum circuits $\mathcal{U}$ does the following: it first runs $O\left(2^{n/3}\right)$ iterations of Grover's algorithm, in order to produce the n-qubit state $\alpha|x\rangle + \beta\sum_{y\in\{0,1\}^n}|y\rangle$, where

$$\alpha = \sqrt{\frac{1}{2^{n/3} + 2^{-n/3+1} + 1}},$$
$$\beta = 2^{-n/3}\alpha$$

[7] For if there are multiple marked items, then we can reduce to the unique marked item case by using the Valiant-Vazirani hashing technique described in Theorem 145.

(one can check that this state is normalized). Next $\mathcal{U}$ applies Hadamard gates to the first $n/3$ qubits. This yields the state

$$2^{-n/6}\alpha \sum_{y\in\{0,1\}^{n/3}} (-1)^{x_A\cdot y} |y\rangle |x_B\rangle + 2^{n/6}\beta \sum_{z\in\{0,1\}^{2n/3}} |0\rangle^{\otimes n/3} |z\rangle ,$$

where x_A consists of the first $n/3$ bits of x, and x_B consists of the remaining $2n/3$ bits. Let Y be the set of $2^{n/3}$ basis states of the form $|y\rangle |x_B\rangle$, and Z be the set of $2^{2n/3}$ basis states of the form $|0\rangle^{\otimes n/3} |z\rangle$.

Notice that $2^{-n/6}\alpha = 2^{n/6}\beta$. So with the sole exception of $|0\rangle^{\otimes n/3} |x_B\rangle$ (which belongs to both Y and Z), the "marked" basis states in Y have the same amplitude as the "unmarked" basis states in Z. This is what we wanted. Notice also that, if we manage to find any $|y\rangle |x_B\rangle \in Y$, then we can find x itself using $2^{n/3}$ further classical queries: simply test all possible strings that end in x_B. Thus, the goal of our algorithm will be to cause the hidden variable to visit an element of Y, so that inspecting the variable's history reveals that element.

As in Theorem 145, the tools that we need are the juggle subroutine, and a way of reducing many basis states to two. Let s be drawn uniformly at random from $\{0,1\}^{n/3}$. Then $\mathcal{U}$ appends a third register to the state, and sets it equal to $|z\rangle$ if the first two registers have the form $|0\rangle^{\otimes n/3} |z\rangle$, or to $|s,y\rangle$ if they have the form $|y\rangle |x_B\rangle$. Disregarding the basis state $|0\rangle^{\otimes n/3} |x_B\rangle$ for convenience, the result is

$$2^{-n/6}\alpha \left(\sum_{y\in\{0,1\}^{n/3}} (-1)^{x_A\cdot y} |y\rangle |x_B\rangle |s,y\rangle + \sum_{z\in\{0,1\}^{2n/3}} |0\rangle^{\otimes n/3} |z\rangle |z\rangle \right).$$

Next $\mathcal{U}$ applies the juggle subroutine to the joint state of the first two registers. Suppose the hidden-variable value has the form $|0\rangle^{\otimes n/3} |z\rangle |z\rangle$ (that is, lies outside Y). Then with probability $2^{-n/3}$ over s, the first $n/3$ bits of z are equal to s. Suppose this event occurs. Then conditioned on the third register being $|z\rangle$, the reduced state of the first two registers is

$$\frac{(-1)^{x_A\cdot z_B} |z_B\rangle |x_B\rangle + |0\rangle^{\otimes n/3} |z\rangle}{\sqrt{2}},$$

where z_B consists of the last $n/3$ bits of z. So it follows from Section 16.8 that with probability $\Omega(1/n)$, the juggle subroutine will cause the hidden variable to transition from $|0\rangle^{\otimes n/3} |z\rangle$ to $|z_B\rangle |x_B\rangle$, and hence from Z to Y.

The algorithm calls the juggle subroutine $\Theta\left(2^{n/3}n\right) = \Theta\left(N^{1/3}\log N\right)$ times, drawing a new value of s and recomputing the third register after each call. Each call moves the hidden variable from Z to Y with independent probability $\Omega\left(2^{-n/3}/n\right)$; therefore with high probability *some* call does so. Note that this juggling phase does not involve any database queries. Also, as in Theorem 145, "drawing" s really means preparing a uniform superposition over all possible s. Finally, the probability that the hidden variable ever visits the basis state $|0\rangle^{\otimes n/3} |x_B\rangle$ is exponentially small (by the union bound), which justifies our having disregarded it. ■

A curious feature of Theorem 146 is the tradeoff between queries and computation steps. Suppose we had run Q iterations of Grover's algorithm, or in other words made Q queries to f. Then provided $Q \leq \sqrt{N}$, the marked state $|x\rangle$ would have occurred with probability $\Omega\left(Q^2/N\right)$, meaning that $\widetilde{O}\left(N/Q^2\right)$ calls to the juggle subroutine would have been sufficient to find x. Of course, the choice of Q that minimizes $\max\left\{Q, N/Q^2\right\}$ is $Q = N^{1/3}$. On the other hand, had we been willing to spend $\widetilde{O}\left(N\right)$ computation steps, we could have found x with only a *single* query![8] Thus, one might wonder whether some other algorithm could push the number of queries below $N^{1/3}$, without simultaneously increasing the number of computation steps. The following theorem rules out that possibility.

Theorem 147 *In the* DQP *model, $\Omega\left(N^{1/3}\right)$ computation steps are needed to search an N-item database for a unique marked item. As a consequence, there exists an oracle relative to which* NP $\not\subset$ DQP*; that is,* NP*-complete problems are not efficiently solvable by sampling histories.*

Proof. Let $N = 2^n$ and $f : \{0,1\}^n \rightarrow \{0,1\}$. Given a sequence of quantum circuits $\mathcal{U} = (U_1, \ldots, U_T)$ that query f, and assuming that $x \in \{0,1\}^n$ is the unique string such that $f(x) = 1$, let $|\psi_t(x)\rangle$ be the quantum state after U_t is applied but before U_{t+1} is. Then the "hybrid argument" of Bennett et al. [51] implies that, by simply changing the location of the marked item from x to x^*, we can ensure that

$$\| |\psi_t(x)\rangle - |\psi_t(x^*)\rangle \| = O\left(\frac{Q_t^2}{N}\right)$$

where $\|\ \|$ represents trace distance, and Q_t is the total number of queries made to f by $U_1, \ldots, U_t$. Therefore $O\left(Q_t^2/N\right)$ provides an upper bound on the probability of noticing the $x \rightarrow x^*$ change by monitoring v_t, the value of the hidden variable after U_t is applied. So by the union bound, the probability of noticing the change by monitoring the entire history $(v_1, \ldots, v_T)$ is at most of order

$$\sum_{t=1}^{T} \frac{Q_t^2}{N} \leq \frac{TQ_T^2}{N}.$$

This cannot be $\Omega(1)$ unless $T = \Omega\left(N^{1/3}\right)$ or $Q_T = \Omega\left(N^{1/3}\right)$, either of which implies an $\Omega\left(N^{1/3}\right)$ lower bound on the total number of steps.

To obtain an oracle relative to which NP $\not\subset$ DQP, we can now use a standard and well-known "diagonalization method" due to Baker, Gill, and Solovay [41] to construct an infinite sequence of exponentially hard search problems, such that any DQP machine fails on at least one of the problems, whereas there exists an NP machine that succeeds on all of them. Details are omitted. ■

16.11 Conclusions and Open Problems

The idea that certain observables in quantum mechanics might have trajectories governed by dynamical laws has reappeared many times: in Schrödinger's 1931 stochastic approach

[8] One should not make too much of this fact; one way to interpret it is simply that the "number of queries" should be redefined as $Q + T$ rather than Q.

[213], Bohmian mechanics [59], modal interpretations [39, 96, 97], and elsewhere. Yet because all of these proposals yield the same predictions for single-time probabilities, if we are to decide between them it must be on the basis of internal mathematical considerations. One message of this chapter has been that such considerations can actually get us quite far.

To focus attention on the core issues, I restricted attention to the simplest possible setting: discrete time, a finite-dimensional Hilbert space, and a single orthogonal basis. Within this setting, I proposed what seem like reasonable axioms that any hidden-variable theory should satisfy: for example, indifference to the identity operation, robustness to small perturbations, and independence of the temporal order of spacelike-separated events. I then showed that not all of these axioms can be satisfied simultaneously. But perhaps more surprisingly, I also showed that certain subsets of axioms *can* be satisfied for quite nontrivial reasons. In showing that the indifference and robustness axioms can be simultaneously satisfied, Section 16.6 revealed an unexpected connection between unitary matrices and the classical theory of network flows.

As mentioned previously, an important open problem is to show that the Schrödinger theory satisfies robustness. Currently, I can only show that the matrix $P_{\mathcal{ST}}(\rho, U)$ is robust to *exponentially* small perturbations, not polynomially small ones. The problem is that if any row or column sum in the $U^{(t)}$ matrix is extremely small, then the (r, c)-scaling process will magnify tiny errors in the entries. Intuitively, though, this effect should be washed out by later scaling steps.

A second open problem is whether there exists a theory that satisfies indifference, as well as commutativity for all separable *mixed* states (not just separable pure states). A third problem is to investigate other notions of robustness—for example, robustness to small *multiplicative* rather than additive errors.

On the complexity side, perhaps the most interesting problem left open by this chapter is the computational complexity of simulating Bohmian mechanics. I strongly conjecture that this problem, like the hidden-variable problems we have seen, is strictly harder than simulating an ordinary quantum computer. The trouble is that Bohmian mechanics does not quite fit in the framework of this chapter: as discussed in Section 16.3.2, we cannot have deterministic hidden-variable trajectories for discrete degrees of freedom such as qubits. Even worse, Bohmian mechanics violates the continuous analogue of the indifference axiom. On the other hand, this means that by trying to implement (say) the juggle subroutine with Bohmian trajectories, one might learn not only about Bohmian mechanics and its relation to quantum computation, but also about how essential the indifference axiom really is for our implementation.

Another key open problem is to show better upper bounds on DQP. Recall that I was only able to show $\mathsf{DQP} \subseteq \mathsf{EXP}$, by giving a classical exponential-time algorithm to simulate the flow theory $\mathcal{FT}$. Can we improve this to (say) $\mathsf{DQP} \subseteq \mathsf{PSPACE}$? Clearly it would suffice to give a PSPACE algorithm that computes the transition probabilities for some theory $\mathcal{T}$ satisfying the indifference and robustness axioms. On the other hand, this might not be *necessary*—that is, there might be an indirect simulation method that does not work by computing (or even sampling from) the distribution over histories. It would also be nice to pin down the complexities of simulating specific hidden-variable theories, such as $\mathcal{FT}$ and $\mathcal{ST}$.

Chapter 17

Summary of Part II

Recall our hypothetical visitor from Conway's Game of Life, on a complexity safari of the physical universe. Based on the results in Part II, the following are some intuitions about efficient computation that I would advise our visitor to toss in the garbage.

- We can be fairly confident that the class of functions efficiently computable in the physical world coincides with P (or BPP, which is presumably equal).
- Although there are models of efficient computation more powerful than P, involving the manipulation of arbitrary real or complex numbers, these models will inevitably blow up small errors in the numbers nonlinearly, and must be therefore be unphysical.
- A robot, moving at unit speed, would need order n steps to search a spatial region of size n for a marked item.
- The ability to see one's entire "history" in a single time step cannot yield any complexity-theoretic advantage, since one could always just record the history as one went along, at the cost of a polynomial increase in memory.

On the other hand, just as in Part I, we have seen that many of the intuitions in our visitor's suitcase are good to go. For example:

- If the items in a database have distance d from one another, then the time needed to search the database is about d times what it would be if the items had unit distance from one another.
- It is possible to choose a probability distribution over histories, in such a way that state i is never followed in a history by state j if the corresponding transition probability is zero, and such that a small change to the transition matrices produces only a small change in the history distribution.
- If, at the moment of your death, your entire life's history flashed before you in an instant, then you could probably still not solve NP-complete problems in polynomial time.

Bibliography

[1] S. Aaronson. Book review on A New Kind of Science. *Quantum Information and Computation*, 2(5):410–423, 2002. quant-ph/0206089.

[2] S. Aaronson. Quantum lower bound for the collision problem. In *Proc. ACM STOC*, pages 635–642, 2002. quant-ph/0111102.

[3] S. Aaronson. Algorithms for Boolean function query properties. *SIAM J. Comput.*, 32(5):1140–1157, 2003.

[4] S. Aaronson. Quantum certificate complexity. In *Proc. IEEE Conference on Computational Complexity*, pages 171–178, 2003. ECCC TR03-005, quant-ph/0210020.

[5] S. Aaronson. Quantum lower bound for recursive Fourier sampling. *Quantum Information and Computation*, 3(2):165–174, 2003. ECCC TR02-072, quant-ph/0209060.

[6] S. Aaronson. The complexity of agreement. ECCC TR04-061, 2004.

[7] S. Aaronson. Is quantum mechanics an island in theoryspace? In A. Khrennikov, editor, *Proceedings of the Växjö Conference "Quantum Theory: Reconsideration of Foundations"*, 2004. quant-ph/0401062.

[8] S. Aaronson. Limitations of quantum advice and one-way communication. *Theory of Computing*, 2004. To appear. Conference version in *Proc. IEEE Complexity 2004*, pp. 320-332. quant-ph/0402095.

[9] S. Aaronson. Lower bounds for local search by quantum arguments. In *Proc. ACM STOC*, pages 465–474, 2004. ECCC TR03-057, quant-ph/0307149.

[10] S. Aaronson. Multilinear formulas and skepticism of quantum computing. In *Proc. ACM STOC*, pages 118–127, 2004. quant-ph/0311039.

[11] S. Aaronson. Quantum computing and hidden variables. Accepted to Phys. Rev. A. quant-ph/0408035 and quant-ph/0408119, 2004.

[12] S. Aaronson. NP-complete problems and physical reality: a survey. In preparation; invited for SIGACT News, 2005.

[13] S. Aaronson and A. Ambainis. Quantum search of spatial regions. *Theory of Computing*, 2004. To appear. Conference version in *Proc. IEEE FOCS 2003*, pp. 200-209. quant-ph/0303041.

[14] S. Aaronson and D. Gottesman. Improved simulation of stabilizer circuits. *Phys. Rev. Lett.*, 70(052328), 2004. quant-ph/0406196.

[15] D. S. Abrams and S. Lloyd. Nonlinear quantum mechanics implies polynomial-time solution for NP-complete and #P problems. *Phys. Rev. Lett.*, 81:3992–3995, 1998. quant-ph/9801041.

[16] L. Adleman, J. DeMarrais, and M.-D. Huang. Quantum computability. *SIAM J. Comput.*, 26(5):1524–1540, 1997.

[17] M. Agrawal, N. Kayal, and N. Saxena. PRIMES is in P. www.cse.iitk.ac.in/users/manindra/primality.ps, 2002.

[18] D. Aharonov. Quantum computation - a review. In Dietrich Stauffer, editor, *Annual Review of Computational Physics*, volume VI. 1998. quant-ph/9812037.

[19] D. Aharonov, A. Ambainis, J. Kempe, and U. Vazirani. Quantum walks on graphs. In *Proc. ACM STOC*, pages 50–59, 2001. quant-ph/0012090.

[20] D. Aharonov and M. Ben-Or. Fault-tolerant quantum computation with constant error. In *Proc. ACM STOC*, pages 176–188, 1997. quant-ph/9906129.

[21] D. Aharonov and T. Naveh. Quantum NP - a survey. quant-ph/0210077, 2002.

[22] D. Aharonov and O. Regev. Lattice problems in NP intersect coNP. In *Proc. IEEE FOCS*, pages 362–371, 2004.

[23] D. Aharonov and A. Ta-Shma. Adiabatic quantum state generation and statistical zero knowledge. In *Proc. ACM STOC*, pages 20–29, 2003. quant-ph/0301023.

[24] D. Aldous. Minimization algorithms and random walk on the d-cube. *Annals of Probability*, 11(2):403–413, 1983.

[25] A. Ambainis. In preparation.

[26] A. Ambainis. A note on quantum black-box complexity of almost all Boolean functions. *Inform. Proc. Lett.*, 71:5–7, 1999. quant-ph/9811080.

[27] A. Ambainis. Quantum lower bounds by quantum arguments. *J. Comput. Sys. Sci.*, 64:750–767, 2002. Earlier version in ACM STOC 2000. quant-ph/0002066.

[28] A. Ambainis. Polynomial degree vs. quantum query complexity. In *Proc. IEEE FOCS*, pages 230–239, 2003. quant-ph/0305028.

[29] A. Ambainis. Quantum lower bounds for collision and element distinctness with small range. quant-ph/0305179, 2003.

[30] A. Ambainis. Quantum walk algorithm for element distinctness. In *Proc. IEEE FOCS*, 2004. quant-ph/0311001.

[31] A. Ambainis, J. Kempe, and A. Rivosh. Coins make quantum walks faster. In *Proc. ACM-SIAM Symp. on Discrete Algorithms (SODA)*, 2005. To appear. quant-ph/0402107.

[32] A. Ambainis, A. Nayak, A. Ta-Shma, and U. V. Vazirani. Quantum dense coding and quantum finite automata. *J. ACM*, 49:496–511, 2002. Earlier version in ACM STOC 1999. quant-ph/9804043.

[33] A. Ambainis, L. J. Schulman, A. Ta-Shma, U. V. Vazirani, and A. Wigderson. The quantum communication complexity of sampling. *SIAM J. Comput.*, 32:1570–1585, 2003.

[34] A. Ambainis, L. J. Schulman, and U. V. Vazirani. Computing with highly mixed states (extended abstract). In *Proc. ACM STOC*, pages 697–704, 2000. quant-ph/0003136.

[35] M. Arndt, O. Nairz, J. Vos-Andreae, C. Keller, G. van der Zouw, and A. Zeilinger. Wave-particle duality of C_{60} molecules. *Nature*, 401:680–682, 1999.

[36] S. Arora, R. Impagliazzo, and U. Vazirani. Relativizing versus nonrelativizing techniques: the role of local checkability. Manuscript, 1992.

[37] A. Aspect, P. Grangier, and G. Roger. Experimental realization of Einstein-Podolsky-Rosen-Bohm gedankenexperiment: a new violation of Bell's inequalities. *Phys. Rev. Lett.*, 49:91–94, 1982.

[38] L. Babai. Bounded round interactive proofs in finite groups. *SIAM J. Discrete Math*, 5(1):88–111, 1992.

[39] G. Bacciagaluppi and M. Dickson. Dynamics for modal interpretations of quantum theory. *Found. Phys.*, 29:1165–1201, 1999. quant-ph/9711048.

[40] D. Bacon. Quantum computational complexity in the presence of closed timelike curves. quant-ph/0309189, 2003.

[41] T. Baker, J. Gill, and R. Solovay. Relativizations of the P=?NP question. *SIAM J. Comput.*, 4:431–442, 1975.

[42] S. Bakhtiari, R. Safavi-Naini, and J. Pieprzyk. Cryptographic hash functions: a survey. Technical Report 95-09, Department of Computer Science, University of Wollongong, July 1995.

[43] Z. Bar-Yossef, T. S. Jayram, and I. Kerenidis. Exponential separation of quantum and classical one-way communication complexity. In *Proc. ACM STOC*, pages 128–137, 2004. ECCC TR04-036.

[44] H. Barnum, M. Saks, and M. Szegedy. Quantum query complexity and semi-definite programming. In *Proc. IEEE Conference on Computational Complexity*, pages 179–193, 2003.

[45] R. Beals, H. Buhrman, R. Cleve, M. Mosca, and R. de Wolf. Quantum lower bounds by polynomials. *J. ACM*, 48(4):778–797, 2001. Earlier version in IEEE FOCS 1998. quant-ph/9802049.

[46] R. Beigel. Perceptrons, PP, and the polynomial hierarchy. *Computational Complexity*, 4:339–349, 1994.

[47] R. Beigel, N. Reingold, and D. Spielman. PP is closed under intersection. *J. Comput. Sys. Sci.*, 50(2):191–202, 1995.

[48] J. D. Bekenstein. A universal upper bound on the entropy to energy ratio for bounded systems. *Phys. Rev. D*, 23(2):287–298, 1981.

[49] J. S. Bell. *Speakable and Unspeakable in Quantum Mechanics*. Cambridge, 1987.

[50] P. Benioff. Space searches with a quantum robot. In S. J. Lomonaco and H. E. Brandt, editors, *Quantum Computation and Information*, Contemporary Mathematics Series. AMS, 2002. quant-ph/0003006.

[51] C. Bennett, E. Bernstein, G. Brassard, and U. Vazirani. Strengths and weaknesses of quantum computing. *SIAM J. Comput.*, 26(5):1510–1523, 1997. quant-ph/9701001.

[52] C. H. Bennett. Logical reversibility of computation. *IBM Journal of Research and Development*, 17:525–532, 1973.

[53] C. H. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres, and W. Wootters. Teleporting an unknown quantum state by dual classical and EPR channels. *Phys. Rev. Lett.*, 70:1895–1898, 1993.

[54] C. H. Bennett and J. Gill. Relative to a random oracle A, $P^A \neq NP^A \neq coNP^A$ with probability 1. *SIAM J. Comput.*, 10(1):96–113, 1981.

[55] E. Bernstein and U. Vazirani. Quantum complexity theory. *SIAM J. Comput.*, 26(5):1411–1473, 1997. First appeared in ACM STOC 1993.

[56] S. N. Bernstein. Sur l'ordre de la meilleure approximation des fonctions continues par les polynômes de degré donné. *Mem. Cl. Sci. Acad. Roy. Belg.*, 4:1–103, 1912. French.

[57] A. Berthiaume and G. Brassard. Oracle quantum computing. In *Proc. Workshop on Physics of Computation: PhysComp'92*, pages 195–199. IEEE, 1992.

[58] A. Beurling. An automorphism of direct product measures. *Ann. Math.*, 72:189–200, 1960.

[59] D. Bohm. A suggested interpretation of the quantum theory in terms of "hidden" variables. *Phys. Rev.*, 85:166–193, 1952.

[60] D. Bohm and B. Hiley. *The Undivided Universe*. Routledge, 1993.

[61] D. Boneh and R. Lipton. Algorithms for black box fields and their application to cryptography. In *Proceedings of CRYPTO*, volume 109, pages 283–297. Lecture Notes in Computer Science, 1996.

[62] M. L. Bonet and S. R. Buss. Size-depth tradeoff for Boolean formulae. *Inform. Proc. Lett.*, 11:151–155, 1994.

[63] R. B. Boppana, J. Håstad, and S. Zachos. Does co-NP have short interactive proofs? *Inform. Proc. Lett.*, 25:127–132, 1987.

[64] R. Bousso. Positive vacuum energy and the N-bound. *J. High Energy Physics*, 0011(038), 2000. hep-th/0010252.

[65] R. Bousso. The holographic principle. *Reviews of Modern Physics*, 74(3), 2002. hep-th/0203101.

[66] M. Boyer, G. Brassard, P. Høyer, and A. Tapp. Tight bounds on quantum searching. *Fortschritte Der Physik*, 46(4-5):493–505, 1998. quant-ph/9605034.

[67] G. Brassard, P. Høyer, M. Mosca, and A. Tapp. Quantum amplitude amplification and estimation. In S. J. Lomonaco and H. E. Brandt, editors, *Quantum Computation and Information*, Contemporary Mathematics Series. AMS, 2002. quant-ph/0005055.

[68] G. Brassard, P. Høyer, and A. Tapp. Quantum algorithm for the collision problem. *ACM SIGACT News*, 28:14–19, 1997. quant-ph/9705002.

[69] S. L. Braunstein, C. M. Caves, N. Linden, S. Popescu, and R. Schack. Separability of very noisy mixed states and implications for NMR quantum computing. *Phys. Rev. Lett.*, 83:1054–1057, 1999. quant-ph/9811018.

[70] R. P. Brent. The parallel evaluation of general arithmetic expressions. *J. ACM*, 21:201–206, 1974.

[71] H. J. Briegel and R. Raussendorf. Persistent entanglement in arrays of interacting particles. *Phys. Rev. Lett.*, 86:910–913, 2001. quant-ph/0004051.

[72] T. Brun. Computers with closed timelike curves can solve hard problems. *Foundations of Physics Letters*, 16:245–253, 2003. gr-qc/0209061.

[73] N. H. Bshouty, R. Cleve, and W. Eberly. Size-depth tradeoffs for algebraic formulae. *SIAM J. Comput.*, 24(4):682–705, 1995.

[74] S. Bublitz, U. Schürfeld, B. Voigt, and I. Wegener. Properties of complexity measures for PRAMs and WRAMs. *Theoretical Comput. Sci.*, 48:53–73, 1986.

[75] H. Buhrman, R. Cleve, J. Watrous, and R. de Wolf. Quantum fingerprinting. *Phys. Rev. Lett.*, 87(16), 2001. quant-ph/0102001.

[76] H. Buhrman, R. Cleve, and A. Wigderson. Quantum vs. classical communication and computation. In *Proc. ACM STOC*, pages 63–68, 1998. quant-ph/9702040.

[77] H. Buhrman, C. Dürr, M. Heiligman, P. Høyer, F. Magniez, M. Santha, and R. de Wolf. Quantum algorithms for element distinctness. In *Proc. IEEE Conference on Computational Complexity*, pages 131–137, 2001. quant-ph/0007016.

[78] H. Buhrman and R. de Wolf. Complexity measures and decision tree complexity: a survey. *Theoretical Comput. Sci.*, 288:21–43, 2002.

[79] P. Bürgisser, M. Clausen, and M. A. Shokrollahi. *Algebraic Complexity Theory.* Springer-Verlag, 1997.

[80] A. R. Calderbank and P. W. Shor. Good quantum error-correcting codes exist. *Phys. Rev. A*, 54:1098–1105, 1996. quant-ph/9512032.

[81] C. M. Caves, C. A. Fuchs, and R. Schack. Unknown quantum states: the quantum de Finetti representation. *J. Math. Phys.*, 45(9):4537–4559, 2002. quant-ph/0104088.

[82] E. W. Cheney. *Introduction to Approximation Theory.* McGraw-Hill, 1966.

[83] A. M. Childs, R. Cleve, E. Deotto, E. Farhi, S. Gutmann, and D. A. Spielman. Exponential algorithmic speedup by quantum walk. In *Proc. ACM STOC*, pages 59–68, 2003. quant-ph/0209131.

[84] A. M. Childs, E. Farhi, and S. Gutmann. An example of the difference between quantum and classical random walks. *Quantum Information and Computation*, 1(1-2):35–43, 2002. quant-ph/0103020.

[85] A. M. Childs and J. Goldstone. Spatial search and the Dirac equation. *Phys. Rev. A*, 70(042312), 2004. quant-ph/0405120.

[86] A. M. Childs and J. Goldstone. Spatial search by quantum walk. *Phys. Rev. A*, 70(022314), 2004. quant-ph/0306054.

[87] R. Cleve, A. Ekert, C. Macchiavello, and M. Mosca. Quantum algorithms revisited. *Proc. Roy. Soc. London*, A454:339–354, 1998. quant-ph/9708016.

[88] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms (2nd edition).* MIT Press, 2001.

[89] J. Cronin. CP symmetry violation - the search for its origin. Nobel Lecture, December 8, 1980.

[90] D. Deutsch. Quantum theory, the Church-Turing principle and the universal quantum computer. *Proc. Roy. Soc. London*, A400:97–117, 1985.

[91] D. Deutsch. Quantum mechanics near closed timelike lines. *Phys. Rev. D*, 44:3197–3217, 1991.

[92] D. Deutsch. *The Fabric of Reality.* Penguin, 1998.

[93] D. Deutsch. Quantum theory of probability and decisions. *Proc. Roy. Soc. London*, A455:3129–3137, 1999. quant-ph/9906015.

[94] D. Deutsch, A. Barenco, and A. Ekert. Universality in quantum computation. *Proc. Roy. Soc. London*, A449:669–677, 1995. quant-ph/9505018.

[95] D. Deutsch and R. Jozsa. Rapid solution of problems by quantum computation. *Proc. Roy. Soc. London*, A439:553–558, 1992.

[96] M. Dickson. Modal interpretations of quantum mechanics. In *Stanford Encyclopedia of Philosophy*. Stanford University, 2002. At http://plato.stanford.edu/entries/qm-modal/.

[97] D. Dieks. Modal interpretation of quantum mechanics, measurements, and macroscopic behaviour. *Phys. Rev. A*, 49:2290–2300, 1994.

[98] R. Diestel. *Graph Theory (2nd edition)*. Springer-Verlag, 2000.

[99] S. Droste, T. Jansen, and I. Wegener. Upper and lower bounds for randomized search heuristics in black-box optimization. ECCC TR03-048, 2003.

[100] W. Dür and H. J. Briegel. Stability of macroscopic entanglement under decoherence. *Phys. Rev. Lett.*, 92, 2004. quant-ph/0307180.

[101] P. Duriš, J. Hromkovič, J. D. P. Rolim, and G. Schnitger. Las Vegas versus determinism for one-way communication complexity, finite automata, and polynomial-time computations. In *Proc. Intl. Symp. on Theoretical Aspects of Computer Science (STACS)*, pages 117–128, 1997.

[102] C. Dürr and P. Høyer. A quantum algorithm for finding the minimum. quant-ph/9607014, 1996.

[103] G. Egan. *Quarantine: A Novel of Quantum Catastrophe*. Eos, 1995. First printing 1992.

[104] H. Ehlich and K. Zeller. Schwankung von Polynomen zwischen Gitterpunkten. *Mathematische Zeitschrift*, 86:41–44, 1964.

[105] M. Ettinger and P. Høyer. On quantum algorithms for noncommutative hidden subgroups. *Advances in Applied Mathematics*, 25(3):239–251, 2000. quant-ph/9807029.

[106] E. Farhi, J. Goldstone, S. Gutmann, J. Lapan, A. Lundgren, and D. Preda. A quantum adiabatic evolution algorithm applied to random instances of an NP-complete problem. *Science*, 292:472–476, 2001. quant-ph/0104129.

[107] S. Fenner, F. Green, S. Homer, and R. Pruim. Determining acceptance possibility for a quantum computation is hard for the polynomial hierarchy. *Proc. Roy. Soc. London*, A455:3953–3966, 1999. quant-ph/9812056.

[108] R. P. Feynman. Simulating physics with computers. *Int. J. Theoretical Physics*, 21(6-7):467–488, 1982.

[109] R. P. Feynman. *The Character of Physical Law*. MIT Press, 1998. Originally published 1965.

[110] V. Fitch. The discovery of charge-conjugation parity asymmetry. Nobel Lecture, December 8, 1980.

[111] L. R. Ford and D. R. Fulkerson. *Flows in Networks*. Princeton, 1962.

[112] R. Fortet. Résolution d'un système d'équations de M. Schrödinger. *J. Math Pures et. Appl.*, 9:83–105, 1940.

[113] L. Fortnow. My Computational Complexity Web Log. Wednesday, October 30, 2002 entry. fortnow.com/lance/complog.

[114] L. Fortnow. One complexity theorist's view of quantum computing. *Theoretical Comput. Sci.*, 292(3):597–610, 2003.

[115] L. Fortnow and N. Reingold. PP is closed under truth-table reductions. *Information and Computation*, 124(1):1–6, 1996.

[116] L. Fortnow and J. Rogers. Complexity limitations on quantum computation. *J. Comput. Sys. Sci.*, 59(2):240–252, 1999. cs.CC/9811023.

[117] L. Fortnow and M. Sipser. Are there interactive protocols for co-NP languages? *Inform. Proc. Lett.*, 28:249–251, 1988.

[118] J. Franklin and J. Lorenz. On the scaling of multidimensional matrices. *Linear Algebra Appl.*, 114/115:717–735, 1989.

[119] J. R. Friedman, V. Patel, W. Chen, S. K. Tolpygo, and J. E. Lukens. Quantum superposition of distinct macroscopic states. *Nature*, 406:43–46, 2000.

[120] M. Furst, J. B. Saxe, and M. Sipser. Parity, circuits, and the polynomial time hierarchy. *Math. Systems Theory*, 17:13–27, 1984.

[121] S. B. Gashkov. The complexity of the realization of Boolean functions by networks of functional elements and by formulas in bases whose elements realize continuous functions. *Prob. Kibernetiki*, 37:52–118, 1980.

[122] M. Gell-Mann and J. Hartle. Quantum mechanics in the light of quantum cosmology. In W. H. Zurek, editor, *Complexity, Entropy, and the Physics of Information*. Addison-Wesley, 1990.

[123] G. C. Ghirardi, A. Rimini, and T. Weber. Unified dynamics for microscopic and macroscopic systems. *Phys. Rev. D*, 34:470–491, 1986.

[124] S. Ghosh, T. F. Rosenbaum, G. Aeppli, and S. N. Coppersmith. Entangled quantum state of magnetic dipoles. *Nature*, 425:48–51, 2003. cond-mat/0402456.

[125] D. T. Gillespie. Why quantum mechanics cannot be formulated as a Markov process. *Phys. Rev. A*, 49:1607, 1994.

[126] N. Gisin. Weinberg's non-linear quantum mechanics and superluminal communications. *Phys. Lett. A*, 143:1–2, 1990.

[127] A. M. Gleason. Measures on the closed subspaces of a Hilbert space. *J. Math. Mech.*, 6:885–893, 1957.

[128] O. Goldreich. On quantum computing. www.wisdom.weizmann.ac.il/~oded/on-qc.html, 2004.

[129] O. Goldreich and S. Goldwasser. On the limits of non-approximability of lattice problems. In *Proc. ACM STOC*, pages 1–9, 1998.

[130] O. Goldreich, S. Micali, and A. Wigderson. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. *J. ACM*, 38(1):691–729, 1991.

[131] S. Goldwasser and M. Sipser. Private coins versus public coins in interactive proof systems. In *Randomness and Computation*, volume 5 of *Advances in Computing Research*. JAI Press, 1989.

[132] D. Gottesman. Class of quantum error-correcting codes saturating the quantum Hamming bound. *Phys. Rev. A*, 54:1862–1868, 1996. quant-ph/9604038.

[133] D. Gottesman. The Heisenberg representation of quantum computers. Talk at Int. Conf. on Group Theoretic Methods in Physics. quant-ph/9807006, 1998.

[134] F. Green, S. Homer, C. Moore, and C. Pollett. Counting, fanout, and the complexity of quantum ACC. *Quantum Information and Computation*, 2(1):35–65, 2002. quant-ph/0106017.

[135] F. Green and R. Pruim. Relativized separation of EQP from P^{NP}. *Inform. Proc. Lett.*, 80(5):257–260, 2001.

[136] D. M. Greenberger, M. A. Horne, and A. Zeilinger. Bell's theorem without inequalities. In A. I. Miller, editor, *Sixty-Two Years of Uncertainty: Historical, Philosophical, and Physical Inquiries into the Foundations of Quantum Mechanics*. Plenum, 1990.

[137] R. B. Griffiths. Choice of consistent family, and quantum incompatibility. *Phys. Rev. A*, 57:1604, 1998. quant-ph/9708028.

[138] M. Grigni, L. Schulman, M. Vazirani, and U. Vazirani. Quantum mechanical algorithms for the nonabelian hidden subgroup problem. In *Proc. ACM STOC*, pages 68–74, 2001.

[139] L. K. Grover. A fast quantum mechanical algorithm for database search. In *Proc. ACM STOC*, pages 212–219, 1996. quant-ph/9605043.

[140] E. Guay and L. Marchildon. Two-particle interference in standard and Bohmian quantum mechanics. *J. Phys. A.: Math. Gen.*, 36:5617–5624, 2003. quant-ph/0302085.

[141] S. Hallgren. Polynomial-time quantum algorithms for Pell's equation and the principal ideal problem. In *Proc. ACM STOC*, pages 653–658, 2002.

[142] Y. Han, L. Hemaspaandra, and T. Thierauf. Threshold computation and cryptographic security. *SIAM J. Comput.*, 26(1):59–78, 1997.

[143] L. Hardy. Quantum theory from five reasonable axioms. quant-ph/0101012, 2003.

[144] A. J. Hoffman and H. W. Wielandt. The variation of the spectrum of a normal matrix. *Duke J. Math*, 20:37–39, 1953.

[145] A. S. Holevo. Some estimates of the information transmitted by quantum communication channels. *Problems of Information Transmission*, 9:177–183, 1973. English translation.

[146] P. Høyer and R. de Wolf. Improved quantum communication complexity bounds for disjointness and equality. In *Proc. Intl. Symp. on Theoretical Aspects of Computer Science (STACS)*, pages 299–310, 2002. quant-ph/0109068.

[147] R. Impagliazzo and A. Wigderson. P=BPP unless E has subexponential circuits: derandomizing the XOR Lemma. In *Proc. ACM STOC*, pages 220–229, 1997.

[148] D. Janzing, P. Wocjan, and T. Beth. Cooling and low energy state preparation for 3-local Hamiltonians are FQMA-complete. quant-ph/0303186, 2003.

[149] D. S. Johnson, C. H. Papadimitriou, and M. Yannakakis. How easy is local search? *J. Comput. Sys. Sci.*, 37:79–100, 1988.

[150] E. Kashefi, A. Kent, V. Vedral, and K. Banaszek. A comparison of quantum oracles. *Phys. Rev. A*, 65, 2002. quant-ph/0109104.

[151] I. Kerenidis and R. de Wolf. Exponential lower bound for 2-query locally decodable codes via a quantum argument. In *Proc. ACM STOC*, pages 106–115, 2003. quant-ph/0208062.

[152] A. Kitaev. Quantum measurements and the abelian stabilizer problem. ECCC TR96-003, quant-ph/9511026, 1996.

[153] A. Kitaev. Quantum computation: algorithms and error correction. *Russian Math. Surveys*, 52(6):1191–1249, 1997.

[154] H. Klauck. Quantum communication complexity. In *Proc. Intl. Colloquium on Automata, Languages, and Programming (ICALP)*, pages 241–252, 2000. quant-ph/0005032.

[155] H. Klauck. Quantum time-space tradeoffs for sorting. In *Proc. ACM STOC*, pages 69–76, 2003. quant-ph/0211174.

[156] H. Klauck, R. Špalek, and R. de Wolf. Quantum and classical strong direct product theorems and optimal time-space tradeoffs. In *Proc. IEEE FOCS*, 2004. quant-ph/0402123.

[157] A. Klivans and D. van Melkebeek. Graph nonisomorphism has subexponential size proofs unless the polynomial-time hierarchy collapses. *SIAM J. Comput.*, 31:1501–1526, 2002. Earlier version in ACM STOC 1999.

[158] E. Knill, R. Laflamme, R. Martinez, and C. Negrevergne. Implementation of the five qubit error correction benchmark. *Phys. Rev. Lett.*, 86:5811–5814, 2001. quant-ph/0101034.

[159] E. Knill, R. Laflamme, and W. Zurek. Resilient quantum computation. *Science*, 279:342–345, 1998. quant-ph/9702058.

[160] E. Kushilevitz and N. Nisan. *Communication Complexity.* Cambridge, 1997.

[161] S. Kutin. A quantum lower bound for the collision problem. quant-ph/0304162, 2003.

[162] R. E. Ladner. On the structure of polynomial time reducibility. *J. ACM*, 22:155–171, 1975.

[163] C. Lautemann. BPP and the polynomial hierarchy. *Inform. Proc. Lett.*, 17:215–217, 1983.

[164] A. J. Leggett. Testing the limits of quantum mechanics: motivation, state of play, prospects. *J. Phys. Condensed Matter*, 14:R415–451, 2002.

[165] L. Levin. Polynomial time and extravagant models, in The tale of one-way functions. *Problems of Information Transmission*, 39(1):92–103, 2003. cs.CR/0012023.

[166] N. Linial, Y. Mansour, and N. Nisan. Constant depth circuits, Fourier transform, and learnability. *J. ACM*, 40(3):607–620, 1993.

[167] N. Linial, A. Samorodnitsky, and A. Wigderson. A deterministic strongly polynomial algorithm for matrix scaling and approximate permanents. *Combinatorica*, 20(4):545–568, 2000.

[168] D. C. Llewellyn and C. Tovey. Dividing and conquering the square. *Discrete Appl. Math*, 43:131–153, 1993.

[169] D. C. Llewellyn, C. Tovey, and M. Trick. Local optimization on graphs. *Discrete Appl. Math*, 23:157–178, 1989. Erratum: 46:93–94, 1993.

[170] S. Lloyd. Computational capacity of the universe. *Phys. Rev. Lett.*, 88, 2002. quant-ph/0110141.

[171] C. Lund, L. Fortnow, H. Karloff, and N. Nisan. Algebraic methods for interactive proof systems. *J. ACM*, 39:859–868, 1992.

[172] A. A. Markov. On a question by D. I. Mendeleev. *Zapiski Imperatorskoi Akademii Nauk*, SP6(62):1–24, 1890. Russian. English translation at www.math.technion.ac.il/hat/fpapers/markov4.pdf.

[173] V. A. Markov. Über Polynome, die in einem gegebenen Intervalle möglichst wenig von Null abweichen. *Math. Ann.*, 77:213–258, 1916. German. Originally written in 1892.

[174] N. Megiddo and C. H. Papadimitriou. On total functions, existence theorems, and computational complexity. *Theoretical Comput. Sci.*, 81:317–324, 1991.

[175] N. D. Mermin. From cbits to qbits: teaching computer scientists quantum mechanics. *American J. Phys.*, 71(1):23–30, 2003. quant-ph/0207118.

[176] G. Midrijanis. A polynomial quantum query lower bound for the set equality problem. In *Proc. Intl. Colloquium on Automata, Languages, and Programming (ICALP)*, pages 996–1005, 2004. quant-ph/0401073.

[177] G. L. Miller. Riemann's hypothesis and tests for primality. *J. Comput. Sys. Sci.*, 13:300–317, 1976.

[178] M. Minsky and S. Papert. *Perceptrons (2nd edition)*. MIT Press, 1988. First appeared in 1968.

[179] M. Nagasawa. Transformations of diffusions and Schrödinger processes. *Prob. Theory and Related Fields*, 82:109–136, 1989.

[180] A. Nayak. Optimal lower bounds for quantum automata and random access codes. In *Proc. IEEE FOCS*, pages 369–377, 1999. quant-ph/9904093.

[181] E. Nelson. *Quantum Fluctuations*. Princeton, 1985.

[182] M. Nielsen and I. Chuang. *Quantum Computation and Quantum Information*. Cambridge, 2000.

[183] N. Nisan. CREW PRAMs and decision trees. *SIAM J. Comput.*, 20(6):999–1007, 1991.

[184] N. Nisan and M. Szegedy. On the degree of Boolean functions as real polynomials. *Computational Complexity*, 4(4):301–313, 1994.

[185] N. Nisan and A. Wigderson. Hardness vs. randomness. *J. Comput. Sys. Sci.*, 49(2):149–167, 1994.

[186] H. Nishimura and T. Yamakami. Polynomial time quantum computation with advice. *Inform. Proc. Lett.*, 90:195–204, 2003. ECCC TR03-059, quant-ph/0305100.

[187] C. H. Papadimitriou. Talk at UC Berkeley, February 6, 2003.

[188] C. H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

[189] R. Penrose. *The Emperor's New Mind*. Oxford, 1989.

[190] C. Philippidis, C. Dewdney, and B. J. Hiley. Quantum interference and the quantum potential. *Nuovo Cimento*, 52B:15–28, 1979.

[191] J. Polchinski. Weinberg's nonlinear quantum mechanics and the Einstein-Podolsky-Rosen paradox. *Phys. Rev. Lett.*, 66:397–400, 1991.

[192] M. Rabin and A. C-C. Yao. Manuscript, 1979.

[193] E. Rains. Talk given at AT&T, Murray Hill, New Jersey, on March 12, 1997.

[194] R. Raussendorf, D. E. Browne, and H. J. Briegel. Measurement-based quantum computation on cluster states. *Phys. Rev. A*, 68, 2003. quant-ph/0301052.

[195] R. Raz. Multi-linear formulas for permanent and determinant are of super-polynomial size. In *Proc. ACM STOC*, pages 633–641, 2004. ECCC TR03-067.

[196] R. Raz. Multilinear-$NC_1 \neq$ multilinear-NC_2. In *Proc. IEEE FOCS*, pages 344–351, 2004. ECCC TR04-042.

[197] R. Raz, G. Tardos, O. Verbitsky, and N. Vereshchagin. Arthur-Merlin games in Boolean decision trees. *J. Comput. Sys. Sci.*, 59(2):346–372, 1999.

[198] A. A. Razborov. Lower bounds for the size of circuits of bounded depth with basis $\{\&, \oplus\}$. *Mathematicheskie Zametki*, 41(4):598–607, 1987. English translation in *Math. Notes. Acad. Sci. USSR* 41(4):333–338, 1987.

[199] A. A. Razborov. Quantum communication complexity of symmetric predicates. *Izvestiya Math. (English version)*, 67(1):145–159, 2003. quant-ph/0204025.

[200] A. A. Razborov and S. Rudich. Natural proofs. *J. Comput. Sys. Sci.*, 55(1):24–35, 1997.

[201] I. B. Damgård. Collision free hash functions and public key signature schemes. In *Proceedings of Eurocrypt'87*, volume 304 of *Lecture Notes in Computer Science*. Springer-Verlag, 1988.

[202] O. Reingold. Undirected ST-connectivity in log-space. 2004.

[203] T. J. Rivlin. *Chebyshev Polynomials: From Approximation Theory to Algebra and Number Theory*. Wiley, 1990.

[204] T. J. Rivlin and E. W. Cheney. A comparison of uniform approximations on an interval and a finite subset thereof. *SIAM J. Numerical Analysis*, 3(2):311–320, 1966.

[205] C. Rovelli and L. Smolin. Discreteness of area and volume in quantum gravity. *Nuclear Physics*, B442:593–622, 1995. Erratum in Vol. B456, p. 753. gr-qc/9411005.

[206] T. Rudolph and L. Grover. Quantum searching a classical database (or how we learned to stop worrying and love the bomb). quant-ph/0206066, 2002.

[207] B. S. Ryden. *Introduction to Cosmology*. Addison-Wesley, 2002.

[208] S. Perlmutter and 32 others (Supernova Cosmology Project). Measurements of Ω and Λ from 42 high-redshift supernovae. *Astrophysical Journal*, 517(2):565–586, 1999. astro-ph/9812133.

[209] A. Sahai and S. Vadhan. A complete promise problem for statistical zero-knowledge. *J. ACM*, 50(2):196–249, 2003. ECCC TR00-084. Earlier version in IEEE FOCS 1997.

[210] M. Santha. On the Monte-Carlo decision tree complexity of read-once formulae. *Random Structures and Algorithms*, 6(1):75–87, 1995.

[211] M. Santha and M. Szegedy. Quantum and classical query complexities of local search are polynomially related. In *Proc. ACM STOC*, pages 494–501, 2004.

[212] N. Sauer. On the density of families of sets. *J. Combinatorial Theory Series A*, 13:145–147, 1972.

[213] E. Schrödinger. Über die Umkehrung der Naturgesetze. *Sitzungsber. Preuss. Akad. Wissen. Phys. Math. Kl.*, pages 144–153, 1931.

[214] L. J. Schulman and U. V. Vazirani. Molecular scale heat engines and scalable quantum computation. In *Proc. ACM STOC*, pages 322–329, 1999.

[215] A. Shamir. IP=PSPACE. *J. ACM*, 39(4):869–877, 1992.

[216] N. Shenvi, J. Kempe, and K. B. Whaley. A quantum random walk search algorithm. *Phys. Rev. A*, 67(5), 2003. quant-ph/0210064.

[217] Y. Shi. Both Toffoli and controlled-NOT need little help to do universal quantum computation. *Quantum Information and Computation*, 3(1):84–92, 2002. quant-ph/0205115.

[218] Y. Shi. Quantum lower bounds for the collision and the element distinctness problems. In *Proc. IEEE FOCS*, pages 513–519, 2002. quant-ph/0112086.

[219] P. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5):1484–1509, 1997. Earlier version in IEEE FOCS 1994. quant-ph/9508027.

[220] D. Simon. On the power of quantum computation. In *Proc. IEEE FOCS*, pages 116–123, 1994.

[221] R. Sinkhorn. A relationship between arbitrary positive matrices and doubly stochastic matrices. *Ann. Math. Statist.*, 35:876–879, 1964.

[222] M. Sipser. A complexity theoretic approach to randomness. In *Proc. ACM STOC*, pages 330–335, 1983.

[223] R. Smolensky. Algebraic methods in the theory of lower bounds for Boolean circuit complexity. In *Proc. ACM STOC*, pages 77–82, 1987.

[224] J. Håstad. Some optimal inapproximability results. *J. ACM*, 48:798–859, 2001.

[225] A. Steane. Multiple particle interference and quantum error correction. *Proc. Roy. Soc. London*, A452:2551–2577, 1996. quant-ph/9601029.

[226] V. Strassen. Gaussian elimination is not optimal. *Numerische Mathematik*, 14(13):354–356, 1969.

[227] G. 't Hooft. Quantum gravity as a dissipative deterministic system. *Classical and Quantum Gravity*, 16:3263–3279, 1999. gr-qc/9903084.

[228] S. Toda. PP is as hard as the polynomial-time hierarchy. *SIAM J. Comput.*, 20(5):865–877, 1991.

[229] B. Tsirelson. Quantum information processing lecture notes, 1997. www.math.tau.ac.il/~tsirel/Courses/QuantInf/lect7.ps.

[230] G. Turán and F. Vatan. On the computation of Boolean functions by analog circuits of bounded fan-in (extended abstract). In *Proc. IEEE FOCS*, pages 553–564, 1994.

[231] L. G. Valiant and V. V. Vazirani. NP is as easy as detecting unique solutions. *Theoretical Comput. Sci.*, 47(3):85–93, 1986.

[232] W. van Dam, S. Hallgren, and L. Ip. Algorithms for some hidden shift problems. In *Proc. ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 489–498, 2003. quant-ph/0211140.

[233] W. van Dam, M. Mosca, and U. Vazirani. How powerful is adiabatic quantum computation? In *Proc. IEEE FOCS*, pages 279–287, 2001. quant-ph/0206003.

[234] L. Vandersypen, M. Steffen, G. Breyta, C. S. Yannoni, M. H. Sherwood, and I. L. Chuang. Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance. *Nature*, 414:883–887, 2001. quant-ph/0112176.

[235] U. Vazirani. UC Berkeley Quantum computation course lecture notes, 2004. At www.cs.berkeley.edu/~vazirani/quantum.html.

[236] G. Vidal. Efficient classical simulation of slightly entangled quantum computations. *Phys. Rev. Lett.*, 91, 2003. quant-ph/0301063.

[237] H. E. Warren. Lower bounds for approximation by non-linear manifolds. *Trans. Amer. Math. Soc.*, 133:167–178, 1968.

[238] J. Watrous. On one-dimensional quantum cellular automata. In *Proc. IEEE FOCS*, pages 528–537, 1995.

[239] J. Watrous. Succinct quantum proofs for properties of finite groups. In *Proc. IEEE FOCS*, pages 537–546, 2000. cs.CC/0009002.

[240] I. Wegener and L. Zádori. A note on the relations between critical and sensitive complexity. *EIK: Journal of Information Processing and Cybernetics*, 25:417–421, 1989.

[241] S. Weinberg. *Dreams of a Final Theory*. Vintage, 1994.

[242] E. Wigner. The unreasonable effectiveness of mathematics in the natural sciences. *Communications in Pure and Applied Mathematics*, 13(1), 1960.

[243] A. Winter. Quantum and classical message identification via quantum channels. In *A. S. Holevo Festschrift*. Rinton, 2004. To appear. quant-ph/0401060.

[244] R. de Wolf. *Quantum Computing and Communication Complexity*. PhD thesis, University of Amsterdam, 2001.

[245] R. de Wolf. Characeterization of non-deterministic quantum query and quantum communication complexity. *SIAM J. Comput.*, 32(3):681–699, 2003. Earlier version in Proc. IEEE Complexity 2000. cs.CC/0001014.

[246] S. Wolfram. *A New Kind of Science*. Wolfram Media, 2002.

[247] A. C-C. Yao. Some complexity questions related to distributive computing. In *Proc. ACM STOC*, pages 209–213, 1979.

[248] A. C-C. Yao. Quantum circuit complexity. In *Proc. IEEE FOCS*, pages 352–361, 1993.

[249] A. C-C. Yao. Princeton University course assignment, 2001. At www.cs.princeton.edu/courses/archive/spr01/cs598a/assignments/hw3.ps.

[250] A. C-C. Yao. On the power of quantum fingerprinting. In *Proc. ACM STOC*, pages 77–81, 2003.

[251] Ch. Zalka. Could Grover's algorithm help in searching an actual database? quant-ph/9901068, 1999.

[252] W. H. Zurek. Environment-assisted invariance, causality, and probabilities in quantum physics. *Phys. Rev. Lett.*, 90, 2003. quant-ph/0211037.