QUANTUM COMPUTING SINCE DEMOCRITUS

SCOTT AARONSON

Quantum Computing since Democritus

SCOTT AARONSON

Massachusetts Institute of Technology

CAMBRIDGE
UNIVERSITY PRESS

CAMBRIDGE UNIVERSITY PRESS
Cambridge, New York, Melbourne, Madrid, Cape Town,
Singapore, São Paulo, Delhi, Mexico City

Cambridge University Press
The Edinburgh Building, Cambridge CB2 8RU, UK

Published in the United States of America by Cambridge University Press,
New York

www.cambridge.org
Information on this title: www.cambridge.org/9780521199568

First published 2013

Printed and bound in the United Kingdom by the MPG Books Group

A catalog record for this publication is available from the British Library

Library of Congress Cataloging in Publication data
Aaronson, Scott.
Quantum computing since Democritus / Scott Aaronson.
pages cm
Includes bibliographical references and index.
ISBN 978-0-521-19956-8 (pbk.)
1. Quantum theory – Mathematics. 2. Quantum computers. I. Title.
QC174.17.M35A27 2013
621.39′1 – dc23 2012036798

ISBN 978-0-521-19956-8 Paperback

Contents

Preface

A CRITICAL REVIEW OF SCOTT AARONSON’S
QUANTUM COMPUTING SINCE DEMOCRITUS
by Scott Aaronson.
Quantum Computing since Democritus is a candidate for the weirdest book ever to be published by Cambridge University Press. The strangeness starts with the title, which conspicuously fails to explain what this book is *about*. Is this another textbook on quantum computing – the fashionable field at the intersection of physics, math, and computer science that’s been promising the world a new kind of computer for two decades, but has yet to build an actual device that can do anything more impressive than factor 21 into 3 × 7 (with high probability)? If so, then what does *this* book add to the dozens of others that have already mapped out the fundamentals of quantum computing theory? Is the book, instead, a quixotic attempt to connect quantum computing to ancient history? But what does Democritus, the Greek atomist philosopher, really have to do with the book’s content, at least half of which would have been new to scientists of the 1970s, let alone of 300 BC?

Having now read the book, I confess that I’ve had my mind blown, my worldview reshaped, by the author’s truly brilliant, original perspectives on everything from quantum computing (as promised in the title) to Gödel’s and Turing’s theorems to the **P** versus **NP** question to the interpretation of quantum mechanics to artificial intelligence to Newcomb’s Paradox to the black-hole information loss problem. So, if anyone were perusing this book at a bookstore, or with Amazon’s “Look Inside” feature, I would *certainly* tell that person to buy a copy immediately. I’d also add that the author is extremely handsome.

Yet it's hard to avoid the suspicion that *Quantum Computing since Democritus* is basically a "brain dump": a collection of thoughts about theoretical computer science, physics, math, and philosophy that were on the author's mind around the fall of 2006, when he gave a series of lectures at the University of Waterloo that eventually turned into this book. The material is tied together by the author's nerdy humor, his "Socratic" approach to every question, and his obsession with the theory of computation and how it relates to the physical world. But if there's some overarching "thesis" that I'm supposed to take away, I can't for the life of me articulate what it is.

More pointedly, one wonders who the *audience* for this book is supposed to be. On the one hand, it has *way* too much depth for a popular book. Like Roger Penrose's *The Road to Reality* – whose preface promises an accessible adventure even for readers who struggled with fractions in elementary school, but whose first few chapters then delve into holomorphic functions and fiber bundles – *Quantum Computing since Democritus* is not for math-phobes. A curious layperson could *certainly* learn a lot from this book, but he or she would have to be willing to skip over some dense passages, possibly to return to them later. So if you're someone who can stomach "science writing" only after it's been carefully cleansed of the science, look elsewhere.

On the other hand, the book is *also* too wide-ranging, breezy, and idiosyncratic to be used much as a textbook or reference work. Sure, it has theorems, proofs, and exercises, and it covers the basics of an astonishing number of fields: logic, set theory, computability, complexity, cryptography, quantum information, and computational learning theory, among others. It seems likely that students in any of those fields, from the undergraduate level on up, could gain valuable insights from this book, or could use it as an entertaining self-study or refresher course. Besides these basics, the book also has significant material on quantum complexity theory – for example, on the power of quantum proofs and advice – that (to this reviewer's knowledge) hasn't appeared anywhere else in book form. But still, the book flits from topic to topic too hastily to be a definitive text on anything.

So, is the book aimed at non-scientists who won't *actually* make it past the first chapter, but want something to put on their coffee table to impress party guests? The only other possibility I can think of is that there's an underserved audience for science books that are neither "popular" nor "professional": books that describe a piece of the intellectual landscape from one researcher's heavily biased vantage point, using the same sort of language you might hear in a hallway conversation with a colleague from a different field. Maybe, besides those colleagues, this hypothetical "underserved audience" would include precocious high-school students, or programmers and engineers who enjoyed their theoretical courses back in college and want to find out what's new. Maybe this is the same audience that frequents these "science blogs" I've heard about: online venues where anyone in the world can apparently watch real scientists, people at the forefront of human knowledge, engage in petty spats, name-calling, and every other juvenile behavior, and can even egg the scientists on to embarrass themselves further. (The book's author, it should be noted, writes a particularly crass and infamous such blog.) *If* such an audience actually exists, then perhaps the author knew exactly what he was doing in aiming at it. My sense, though, is that he was having too much fun to be guided by any such conscious plan.

NOW FOR THE ACTUAL PREFACE

While I appreciate the reviewer's kind words about my book (and even my appearance!) in the preceding pages, I also take issue, in the strongest possible terms, with his ignorant claim that *Quantum Computing since Democritus* has no overarching thesis. It *does* have a thesis – even though, strangely, I wasn't the one who figured out what it was. For identifying the central message of this book, I need to thank Love Communications, an advertising agency based in

Sydney, Australia, which put the message into the mouths of fashion models for the purpose of selling printers.

Let me explain – the story is worth it.

In 2006, I taught a course entitled "Quantum Computing since Democritus" at the University of Waterloo. Over the next year, I posted rough notes from the course on my blog, *Shtetl-Optimized*[1] – notes that were eventually to become this book. I was heartened by the enthusiastic response from readers of my blog; indeed, that response is what convinced me to publish this book in the first place. But there was one response neither I nor anyone else could have predicted.

On October 1, 2007, I received an email from one Warren Smith in Australia, who said he had seen a television commercial for Ricoh printers. The commercial, he went on, featured two female fashion models in a makeup room, having the following conversation:

> Model 1: But if quantum mechanics isn't physics in the usual sense – if it's not about matter, or energy, or waves – then what *is* it about?
>
> Model 2: Well, from my perspective, it's about information, probabilities, and observables, and how they relate to each other.
>
> Model 1: That's interesting!

The commercial then flashed the tagline "A more intelligent model," followed by a picture of a Ricoh printer.

Smith said he was curious where the unusual text had come from, so he googled it. Doing so brought him to Chapter 9 of my "Quantum Computing since Democritus" notes (p. 110), where he found the following passage:

> But if quantum mechanics isn't physics in the usual sense – if it's not about matter, or energy, or waves, or particles – then what *is* it

[1] www.scottaaronson.com/blog

The idea that quantum mechanics is "about" information, probabilities, and observables, rather than waves and particles, certainly isn't an original one. The physicist John Archibald Wheeler said similar things in the 1970s; and today an entire field, that of quantum computing and information, is built around the idea. Indeed, in the discussion on my blog that followed the Australian models episode, one the commonest (and to me, funniest) arguments was that I had no right to complain, because the appropriated passage *wasn't special in any way*: it was an obvious thought that could be found in any physics book!

How I wish it were so. Even in 2013, the view of quantum mechanics as a theory of information and probabilities remains very much a minority one. Pick up almost any physics book – whether popular or technical – and you'll learn that (a) modern physics says all sorts of paradoxical-seeming things, like that waves are particles and particles are waves, (b) at a deep level, no one really understands these things, (c) even translating them into math requires years of intensive study, but (d) they make the atomic spectra come out right, and that's what matters in the end.

One eloquent statement of this "conventional view" was provided by Carl Sagan, in *The Demon-Haunted World*:

> Imagine you seriously want to understand what quantum mechanics is about. There is a mathematical underpinning that you must first acquire, mastery of each mathematical subdiscipline leading you to the threshold of the next. In turn you must learn arithmetic, Euclidean geometry, high school algebra, differential and integral calculus, ordinary and partial differential equations, vector calculus, certain special functions of mathematical physics, matrix algebra, and group theory . . . The job of the popularizer of science, trying to get across some idea of quantum mechanics to a general audience that has not gone through these initiation rites, is daunting. Indeed, there are no successful popularizations of quantum mechanics in my opinion – partly for this reason. These

> mathematical complexities are compounded by the fact that quantum theory is so resolutely counterintuitive. Common sense is almost useless in approaching it. It's no good, Richard Feynman once said, asking why it *is* that way. No one knows why it is that way. That's just the way it is (p. 249).

It's understandable why physicists talk this way: because physics is an experimental science. In physics you're *allowed* to say, "these are the rules, not because they make sense, but because we ran the experiment and got such-and-such a result." You can even say it proudly, gleefully – *defying* the skeptics to put their preconceived notions up against Nature's verdict.

Personally, I simply *believe* the experimentalists, when they say the world works in a completely different way than I thought it did. It's not a matter of convincing me. Nor do I presume to predict what the experimentalists will discover next. All I want to know is: *What went wrong with my intuition? How should I fix it, to put it more in line with what the experiments found? How could I have reasoned, such that the actual behavior of the world* ***wouldn't*** *have surprised me so much?*

With several previous scientific revolutions – Newtonian physics, Darwinian evolution, special relativity – I feel like I more-or-less know the answers to the above questions. If my intuition isn't yet fully adjusted even to those theories, then at least I know how it *needs* to be adjusted. And thus, for example, if I were creating a new universe, I might or might not decide to make it Lorentz invariant, but I'd certainly *consider* the option, and I'd understand why Lorentz-invariance was the inevitable consequence of a couple of other properties I might want.

But quantum mechanics is different. Here, the physicists assure us, *no one knows* how we should adjust our intuition so that the behavior of subatomic particles would no longer seem so crazy. Indeed, maybe there *is* no way; maybe subatomic behavior will always remain an arbitrary brute fact, with nothing to say about it beyond "such-and-such formulas give you the right answer." My response is radical: if that's true, then *I don't much care* how subatomic particles behave. No doubt other people *need* to know – the people designing lasers or transistors, for example – so let them learn. As for me, I'll simply study another subject that makes more sense to me – like, say, theoretical computer science. Telling me that my physical intuition was wrong, without giving me any path to *correct* that intuition, is like flunking me on an exam without providing any hint about how I could've done better. As soon as I'm free to do so, I'll simply gravitate to other courses where I get As, where my intuition *does* work.

Fortunately, I think that, as the result of decades of work in quantum computation and quantum foundations, we *can* do a lot better today than simply calling quantum mechanics a mysterious brute fact. To spill the beans, here's the perspective of this book:

> *Quantum mechanics is a beautiful generalization of the laws of probability: a generalization based on the 2-norm rather than the 1-norm, and on complex numbers rather than nonnegative real numbers. It can be studied completely separately from its applications to physics (and indeed, doing so provides a good starting point for learning the physical applications later). This generalized probability theory leads naturally to a new model of computation – the quantum computing model – that challenges ideas about computation once considered a priori, and that theoretical computer scientists might have been driven to invent for their own purposes, even if there were no relation to physics. In short, while quantum mechanics was invented a century ago to solve technical problems in physics, today it can be fruitfully explained from an extremely different perspective: as part of the history of ideas, in math, logic, computation, and philosophy, about the limits of the knowable.*

In this book I try to make good on the above claims, taking a leisurely and winding route to do so. I start, in Chapter 1, as near to the "beginning" as I possibly can: with Democritus, the ancient Greek philosopher. Democritus's surviving fragments – which speculate, among other things, that all natural phenomena arise from complicated interactions between a few kinds of tiny "atoms," whizzing around in mostly empty space – get closer to a modern scientific worldview than anything else in antiquity (and certainly closer than any of Plato's or Aristotle's ideas). Yet no sooner had Democritus formulated the atomist hypothesis, than he noticed uneasily its tendency to "swallow whole" the very sense-experiences that he was presumably trying to explain in the first place. How could *those* be reduced to the motions of atoms? Democritus expressed the dilemma in the form of a dialogue between the Intellect and the Senses:

> Intellect: By convention there is sweetness, by convention bitterness, by convention color, in reality only atoms and the void.
>
> Senses: Foolish intellect! Do you seek to overthrow us, while it is from us that you take your evidence?

This two-line dialogue will serve as a sort of touchstone for the entire book. One of my themes will be how quantum mechanics seems to give both the Intellect *and* the Senses unexpected new weapons in their 2300-year-old argument – while still (I think) not producing a clear victory for either.

In Chapters 2 and 3, I move on to discuss the deepest knowledge we have that intentionally *doesn't* depend on "brute facts" about the physical world: namely, mathematics. Even there, something inside me (and, I suspect, inside many other computer scientists!) is suspicious of those *parts* of mathematics that bear the obvious imprint of physics, such as partial differential equations, differential geometry, Lie groups, or anything else that's "too continuous." So instead, I start with some of the most "physics-free" parts of math yet discovered: set theory, logic, and computability. I discuss the great discoveries of Cantor, Frege, Gödel, Turing, Church, and Cohen, which helped to map the contours of mathematical reasoning itself – and which, in the course of showing why all of mathematics can't be reduced to a fixed "mechanical process," also demonstrated just how much of it *could* be, and clarified what we mean by "mechanical process" in the first place. Since I can't resist, in Chapter 4 I then wade into the hoary debate about whether the human mind, too, is governed by "fixed mechanical processes." I set out the various positions as fairly as I can (but no doubt reveal my biases).

Chapter 5 introduces computability theory's modern cousin, *computational complexity theory*, which plays a central role in the rest of the book. I try to illustrate, in particular, how computational complexity lets us systematically take "deep philosophical mysteries" about the limits of knowledge, and convert them into "merely" insanely difficult unsolved mathematical problems, which arguably capture most of what we want to know! There's no better example of such a conversion than the **P** versus **NP** problem, which I discuss in Chapter 6. Then, as warmups to quantum computing, Chapter 7 examines the many uses of *classical* randomness, both in computational complexity and in other parts of life; and Chapter 8 explains how computational complexity ideas were applied to revolutionize the theory and practice of *cryptography* beginning in the 1970s.

All of that is just to set the stage for the most notorious part of the book: Chapter 9, which presents my view of quantum mechanics as a "generalized probability theory." Then Chapter 10 explains the basics of my own field, the *quantum theory of computation*, which can be briefly defined as the merger of quantum mechanics with computational complexity theory. As a "reward" for persevering through all this technical material, Chapter 11 offers a critical examination of the ideas of Sir Roger Penrose, who famously holds that the brain is not merely a quantum computer but quantum *gravitational* computer, able to solve Turing-uncomputable problems – and that this, or something like it, can be shown by an "super-quantum" correlations; derandomization of randomized algorithms; science, religion, and the nature of rationality; and why computer science is not a branch of physics departments.

A final remark. One thing you *won't* find in this book is much discussion of the "practicalities" of quantum computing: either physical implementation, or error correction, or the details of Shor's, Grover's, or other basic quantum algorithms. One reason for this neglect is incidental: the book is based on lectures I gave at the University of Waterloo's Institute for Quantum Computing, and the students were already learning all about those aspects in their other classes. A second reason is that those aspects are covered in *dozens* of other books[7] and online lecture notes (including some of my own), and I saw no need to reinvent the wheel. But a third reason is, frankly, that the technological prospect of building a new kind of computer, exciting as it is, is not why I went into quantum computing in the first place. (*Shhh*, please don't tell any funding agency directors I said that.)

To be clear, I think it's entirely possible that I'll see practical quantum computers in my lifetime (and also possible, of course, that I *won't* see them). And if we *do* get scalable, universal quantum computers, then they'll almost certainly find real applications (not even counting codebreaking): mostly, I think, for specialized tasks like quantum simulation, but to a lesser extent for solving combinatorial optimization problems. If that ever happens, I expect I'll be as excited about it as anyone on earth – and, of course, tickled if any of the work I've done finds applications in that new world. On the other hand, if someone gave me a practical quantum computer tomorrow, then I confess that I can't think of anything that I, personally, would want to use it for: only things that *other people* could use it for!

[7] The "standard reference" for the field remains *Quantum Computation and Quantum Information*, by Michael Nielsen and Isaac Chuang.

Partly for that reason, if scalable quantum computing were proved to be *im*possible, that would excite me a thousand times more than if it were proved to be possible. For such a failure would imply something wrong or incomplete with our understanding of quantum mechanics itself: a revolution in physics! As a congenital pessimist, though, my *guess* is that Nature won't be so kind to us, and that scalable quantum computing will turn out to be possible after all.

In summary, you could say that I'm in this field less because of what you could do with a quantum computer, than because of what the *possibility* of quantum computers *already* does to our conception of the world. *Either* practical quantum computers can be built, and the limits of the knowable are not what we thought they are; *or* they can't be built, and the principles of quantum mechanics themselves need revision; *or* there's a yet-undreamt method to simulate quantum mechanics efficiently using a conventional computer. All three of these possibilities sound like crackpot speculations, but at least one of them is right! So whichever the outcome, what can one say but – to reverse-plagiarize a certain TV commercial – "that's interesting?"

WHAT'S NEW

In revising this manuscript for publication, the biggest surprise for me was how much *happened* in the fields discussed by the book between when I originally gave the lectures (2006) and "now" (2013). This book is supposed to be about deep questions that are as old as science and philosophy, or at the least, as old as the birth of quantum mechanics and of computer science almost a century ago. And at least on a day-to-day basis, it can *feel* like nothing ever changes in the discussion of these questions. And thus, having to update my lectures extensively, after the passage of a mere six years, was an indescribably pleasant burden for me.

Just to show you how things are evolving, let me give a partial list of the developments that are covered in this book, but that *couldn't* have been covered in my original 2006 lectures, for the simple reason that they hadn't happened yet. IBM's Watson computer defeated the *Jeopardy!* world champion Ken Jennings, forcing me to update my discussion of AI with a new example (see p. 37), very different in character from previous examples like ELIZA and Deep Blue. Virginia Vassilevska Williams, building on work of Andrew Stothers, discovered how to multiply two $n \times n$ matrices using only $O(n^{2.373})$ steps, *slightly* beating Coppersmith and Winograd's previous record of $O(n^{2.376})$, which had held for so long that "2.376" had come to feel like a constant of nature (see p. 49).

There were major advances in the area of *lattice-based cryptography*, which provides the leading candidates for public-key encryption systems secure even against quantum computers (see pp. 105–107). Most notably, solving a 30-year-old open problem, Craig Gentry used lattices to propose the first *fully homomorphic cryptosystems*. These systems let a client delegate an arbitrary computation to an untrusted server – feeding the server encrypted inputs and getting back an encrypted output – in such a way that only the client can decrypt (and verify) the output; the server never has any clue what computation it was hired to perform.

In the foundations of quantum mechanics, Chiribella *et al.* (see p. 131) gave a novel argument for "why" quantum mechanics should involve the specific rules it does. Namely, they proved that those rules are the only ones compatible with certain general axioms of probability theory, *together with* the slightly mysterious axiom that "all mixed states can be purified": that is, whenever you don't know everything there is to know about a physical system A, your ignorance must be fully explainable by positing correlations between A and some faraway system B, such that you *would* know everything there is to know about the combined system AB.

In quantum computing theory, Bernstein and Vazirani's "Recursive Fourier Sampling" (RFS) problem – on which I spent a fair bit of time in my 2006 lectures – has been superseded by my "Fourier Checking" problem (see p. 145). RFS retains its place in history, as the first black-box problem ever proposed that a quantum computer can provably solve superpolynomially faster than a classical probabilistic computer – and, as such, an important forerunner to Simon's and Shor's breakthroughs. Today, though, if we want a candidate for a problem in **BQP**$\setminus$**PH** – in other words, something that a quantum computer can easily do, but which is not even in the classical "polynomial-time hierarchy" – then Fourier Checking seems superior to RFS in every way.

Happily, several things discussed as "open problems" in my 2006 lectures have since lost that status. For example, Andrew Drucker and I showed that **BQP/qpoly** is contained in **QMA/poly** (and, moreover, the proof relativizes), falsifying my conjecture that there should be an oracle separation between those classes (see p. 214). Also, in a justly celebrated breakthrough in quantum computing theory, Jain *et al.* proved that **QIP** = **PSPACE** (see p. 263), meaning that quantum interactive proof systems are no more powerful than classical ones. In that case, at least, I conjectured the right answer! (There was actually *another* breakthrough in the study of quantum interactive proof systems, which I *don't* discuss in the book. My postdoc Thomas Vidick, together with Tsuyoshi Ito,[8] recently showed that **NEXP** $\subseteq$ **MIP***, which means that any *multiple*-prover interactive proof system can be "immunized" against the possibility that the provers secretly coordinate their responses using quantum entanglement.)

Chapter 20 of this book discusses David Deutsch's model for quantum mechanics in the presence of closed timelike curves, as well as my (then-)new result, with John Watrous, that Deutsch's model provides exactly the computational power of **PSPACE**. (So that, in particular, quantum time-travel computers would be no more powerful than *classical* time-travel computers, in case you

[8] T. Ito and T. Vidick, A Multi-prover Interactive Proof for NEXP Sound against Entangled Provers. In *Proceedings of IEEE Symposium on Foundations of Computer Science* (2012), pp. 243–252.

were wondering.) Since 2006, however, there have been important papers questioning the assumptions behind Deutsch's model, and proposing alternative models, which generally lead to computational power *less* than **PSPACE**. For example, one model, proposed by Lloyd *et al.*, would "merely" let the time traveler solve all problems in **PP**! I discuss these developments on pp. 319–322.

What about circuit lower bounds – which is theoretical computer scientists' codeword for "trying to prove $\mathbf{P} \neq \mathbf{NP}$," in much the same way that "closed timelike curves" is the physicists' codeword for "time travel?" I'm pleased to report that there have been interesting developments since 2006, certainly more than I would have expected back then. As one example, Rahul Santhanam used interactive proof techniques to prove the non-relativizing result that the class **PromiseMA** doesn't have circuits of any fixed polynomial size (see p. 257). Santhanam's result was part of what spurred Avi Wigderson and myself, in 2007, to formulate the *algebrization barrier* (see p. 258), a generalization of Baker, Gill, and Solovay's relativization barrier from the 1970s (see pp. 245–246). Algebrization explained why the interactive proof techniques can take us only so far and no further in our quest to prove $\mathbf{P} \neq \mathbf{NP}$: as one example, why those techniques led to superlinear circuit lower bounds for **PromiseMA**, but not for the class **NP** just "slightly below it." The challenge we raised was to find new circuit lower bound techniques that convincingly *evade* the algebrization barrier. That challenge was met in 2010, by Ryan Williams' breakthrough proof that $\mathbf{NEXP} \not\subset \mathbf{ACC^0}$ (discussed on pp. 260–261).

Of course, even Williams' result, exciting as it was, is a helluva long way from a proof of $\mathbf{P} \neq \mathbf{NP}$. But the past six years have also witnessed a flowering of interest in, and development of, Ketan Mulmuley's Geometric Complexity Theory (GCT) program (see pp. 261–262), which is to proving $\mathbf{P} \neq \mathbf{NP}$ almost exactly as string theory is to the goal of a unified theory of physics. That is, in terms of concrete results, the GCT program hasn't yet come anywhere close to fulfilling its initial hopes, and even the program's most

Acknowledgments

As my summer student in 2008, Chris Granade enthusiastically took charge of converting the scattered notes and audio recordings from my course into coherent drafts that I could post on my website, the first step on their long journey into book form. More recently, Alex Arkhipov, my phenomenal PhD student at MIT, went through the drafts with a fine-tooth comb, flagging passages that were wrong, unclear, or no longer relevant. I'm deeply grateful to both of them: this book is also *their* book; it wouldn't exist without their help.

It also wouldn't exist without Simon Capelin, my editor at Cambridge University Press, who approached me with the idea. Simon understood what I needed: he prodded me every few months to see if I'd made progress, but never in an accusatory way, always relying on my own internal guilt to see the project through. (And I *did* see it through – eventually.) Simon also assured me that, even though *Quantum Computing since Democritus* was . . . a bit *different* from CUP's normal fare, he would make every effort to preserve what he called the book's "quirky charm." I also thank all the others at CUP and Aptara Corp. who helped to make the book a reality: Sarah Hamilton, Emma Walker, and Disha Malhotra.

I thank the students and faculty who sat in on my "Quantum Computing since Democritus" course at the University of Waterloo in Fall 2006. Their questions and arguments made the course what it was (as you can still see in this book, especially in the last chapters). On top of that, the students also took care of the audio recordings and preliminary written transcripts. More broadly, I remember my two years as a postdoc at Waterloo's Institute for Quantum Computing as one of the happiest times of my life. I thank everyone there, and especially IQC's director Ray Laflamme, for not only

letting me teach such a nutty course but *encouraging* it, and even (in Ray's and several other cases) sitting in on the course themselves and contributing many insights.

I thank MIT's Computer Science and Artificial Intelligence Laboratory and its Electrical Engineering and Computer Science Department, as well as the US National Science Foundation, the Defense Advanced Research Projects Agency, the Sloan Foundation, and TIBCO, Inc., for all the support they've given me over the last six years.

I thank the readers of my blog, *Shtetl-Optimized* (http://www.scottaaronson.com/blog), for their many comments on the draft chapters that I posted there, and for catching numerous errors. I especially thank those readers who encouraged me to turn *Quantum Computing since Democritus* into a book – some even promised they'd buy it when it came out.

I thank the people who advised me from my high school to my postdoc years: Chris Lynch, Bart Selman, Lov Grover, Umesh Vazirani, and Avi Wigderson. John Preskill was never "formally" an advisor, but I still think of him as one. I owe all of them more than I can say. I also thank everyone else in (and beyond) the quantum information and theoretical computer science communities whose discussions and arguments with me over the years left their imprints on this book. I can't possibly produce a full list of such people, so here's a partial one: Dorit Aharonov, Andris Ambainis, Dave Bacon, Michael Ben-Or, Raphael Bousso, Harry Buhrman, Sean Carroll, Greg Chaitin, Richard Cleve, David Deutsch, Andy Drucker, Ed Farhi, Chris Fuchs, Daniel Gottesman, Alex Halderman, Robin Hanson, Richard Karp, Elham Kashefi, Julia Kempe, Greg Kuperberg, Seth Lloyd, Michele Mosca, Michael Nielsen, Christos Papadimitriou, Len Schulman, Lenny Susskind, Oded Regev, Barbara Terhal, Michael Vassar, John Watrous, Ronald de Wolf. I apologize for the inevitable omissions (or to those who don't want their names in this book, you're welcome!).

I thank the following alert readers for catching errors in the first printing of this book: Evan Berkowitz, Ernest Davis, Bob Galesloot, Andrew Marks, Cris Moore, and Tyler Singer-Clark.

Lastly, I thank my mom and dad, my brother David, and of course my wife Dana, who will now finally be able to know me while I'm *not* putting off finishing the damn book.

1 Atoms and the void

I would rather discover a single cause than become king of the Persians.

– Democritus

So why Democritus? First of all, who *was* Democritus? He was this Ancient Greek dude. He was born around 450 BC in this podunk Greek town called Abdera, where people from Athens said that even the air causes stupidity. He was a disciple of Leucippus, according to my source, which is Wikipedia. He's called a "pre-Socratic," even though actually he was a contemporary of Socrates. That gives you a sense of how important he's considered: "Yeah, the *pre*-Socratics – maybe stick 'em in somewhere in the first week of class." Incidentally, there's a story that Democritus journeyed to Athens to meet Socrates, but then was too shy to introduce himself.

Almost none of Democritus's writings survive. Some survived into the Middle Ages, but they're lost now. What we know about him is mostly due to other philosophers, like Aristotle, bringing him up in order to criticize him.

So, what did they criticize? Democritus thought the whole universe is composed of atoms in a void, constantly moving around according to determinate, understandable laws. These atoms can hit each other and bounce off, or they can stick together to make bigger things. They can have different sizes, weights, and shapes – maybe some are spheres, some are cylinders, whatever. On the other hand, Democritus says that properties like color and taste are *not* intrinsic to atoms, but instead emerge out of the interactions of many atoms.

You might wonder how such a crazy theory could be *useful* to physicists, even at the crassest level. How could it even make *predictions*, if it essentially says that everything that could happen does? Well, the thing I didn't tell you is that there's a separate rule for what happens when you make a measurement: a rule that's "tacked on" (so to speak), external to the equations themselves. That rule says, essentially, that the act of looking at a particle *forces it to make up its mind* about where it wants to be, and that the particle makes its choice *probabilistically*. And the rule tells you exactly how to calculate the probabilities. And of course it's been spectacularly well confirmed.

But here's the problem: as the universe is chugging along, doing its thing, how are we supposed to know when to apply this measurement rule, and when not to? What counts as a "measurement," anyway? The laws of physics aren't supposed to say things like "such-and-such happens *until someone looks*, and then a completely different thing happens!" Physical laws are supposed to be *universal*. They're supposed to describe human beings the same way they describe supernovas and quasars: all just examples of vast, complicated clumps of particles interacting according to simple rules.

So from a physics perspective, things would be so much cleaner if we could dispense with this "measurement" business entirely! Then we could say, in a more sophisticated update of Democritus: there's nothing but atoms and the void, evolving in quantum superposition.

But wait: if we're not here making nosy measurements, wrecking the pristine beauty of quantum mechanics, then how did "we" (whatever that means) ever get the evidence in the first place that quantum mechanics is true? How did we ever come to believe in this theory that seems so uncomfortable with the fact of our own existence?

So, that's the modern version of the Democritus dilemma, and physicists and philosophers have been arguing about it for almost a hundred years, and in this book we're not going to solve it.

The other thing I'm not going to do in this book is try to sell you on some favorite "interpretation" of quantum mechanics. You're free to believe whatever interpretation your conscience dictates. (What's my own view? Well, I agree with *every* interpretation to the extent it says there's a problem, and disagree with every interpretation to the extent it claims to have solved the problem!)

See, just like we can classify religions as monotheistic and polytheistic, we can classify interpretations of quantum mechanics by where they come down on the "putting-yourself-in-coherent-superposition" issue. On the one side, we've got the interpretations that enthusiastically sweep the issue under the rug: Copenhagen and its Bayesian and epistemic grandchildren. In these interpretations, you've got your quantum system, you've got your measuring device, and there's a line between them. Sure, the line can shift from one experiment to the next, but for any given experiment, it's gotta be somewhere. In principle, you can even imagine putting other people on the quantum side, but you *yourself* are always on the classical side. Why? Because a quantum state is just a representation of your knowledge – and you, by definition, are a classical being.

But what if you want to apply quantum mechanics to the whole universe, *including* yourself? The answer, in the epistemic-type interpretations, is simply that you don't ask that sort of question! Incidentally, that was Bohr's all-time favorite philosophical move, his WWF piledriver: "You're not allowed to ask such a question!"

On the other side, we've got the interpretations that *do* try in different ways to make sense of putting yourself in superposition: many-worlds, Bohmian mechanics, etc.

Now, to hardheaded problem-solvers like ourselves, this might seem like a big dispute over words – why bother? I actually agree with that: if it were just a dispute over words, then we *shouldn't* bother! But as David Deutsch pointed out in the late 1970s, we *can* conceive of experiments that would differentiate the first type of interpretation from the second type. The simplest experiment would just be to put yourself in coherent superposition and see what happens! Or if that's too dangerous, put someone *else* in coherent superposition. The point being that, if human beings were regularly put into superposition, then the whole business of drawing a line between "classical observers" and the rest of the universe would become untenable.

But alright – human brains are wet, goopy, sloppy things, and maybe we won't be able to maintain them in coherent superposition for 500 million years. So what's the next best thing? Well, we could try to put a *computer* in superposition. The more sophisticated the computer was – the more it resembled something like a brain, like ourselves – the further up we would have pushed the "line" between quantum and classical. You can see how it's only a minuscule step from here to the idea of quantum computing.

I'd like to draw a more general lesson here. What's the point of talking about philosophical questions? Because we're going to be doing a fair bit of it here – I mean, of philosophical bullshitting. Well, there's a standard answer, and it's that philosophy is an intellectual clean-up job – the janitors who come in after the scientists have made a mess, to try and pick up the pieces. So in this view, philosophers sit in their armchairs waiting for something surprising to happen in science – like quantum mechanics, like the Bell inequality, like Gödel's Theorem – and then (to switch metaphors) swoop in like vultures and say, ah, this is what it *really* meant.

Well, on its face, that seems sort of boring. But as you get more accustomed to this sort of work, I think what you'll find is . . . it's *still* boring!

Personally, I'm interested in results – in finding solutions to nontrivial, well-defined open problems. So, what's the role of philosophy in that? I want to suggest a more exalted role than intellectual janitor: philosophy can be a *scout*. It can be an explorer – mapping out intellectual terrain for science to *later* move in on, and build condominiums on or whatever. Not every branch of science was scouted out ahead of time by philosophy, but some were. And in recent history, I think quantum computing is really the poster child here. It's fine to tell people to "Shut up and calculate," but the question is, *what* should they calculate? At least in quantum computing, which is my field, the sorts of things that we like to calculate – capacities of quantum channels, error probabilities of quantum algorithms – are things people would never have *thought* to calculate if not for philosophy.

2 Sets

Here, we're gonna talk about sets. What will these sets contain? Other sets! Like a bunch of cardboard boxes that you open only to find *more* cardboard boxes, and so on all the way down.

You might ask "how is this relevant to a book on quantum computing?"

Well, hopefully we'll see a few answers later. For now, suffice it to say that math is the foundation of all human thought, and set theory – countable, uncountable, etc. – that's the foundation of math. So regardless of what a book is about, it seems like a fine place to start.

I probably should tell you explicitly that I'm compressing a whole math course into this chapter. On the one hand, that means I don't really expect you to understand everything. On the other hand, to the extent you do understand – hey! You got a whole math course in one chapter! You're welcome.

So let's start with the empty set and see how far we get.

THE EMPTY SET.

Any questions so far?

Actually, before we talk about sets, we need a language for talking about sets. The language that Frege, Russell, and others developed is called *first-order logic*. It includes Boolean connectives (and, or, not), the equals sign, parentheses, variables, predicates, quantifiers ("there exists" and "for all") – and that's about it. I'm told that the physicists have trouble with these. Hey, I'm just ribbin' ya. If you haven't seen this way of thinking before, then you haven't seen it. But maybe, for the benefit of the physicists, let's go over the basic rules of logic.

AXIOMS OF SET THEORY

The axioms all involve a universe of objects called "sets," and a relationship between sets that's called "membership" or "containment" and written using the symbol $\in$. Every operation on sets will ultimately be defined in terms of the containment relationship.

- **Empty set:** There exists an empty set: that is, a set x for which there is no y such that $y \in x$.
- **Extensionality:** If two sets contain the same members, then the sets are equal. That is, for all x and y, if ($z \in x$ if and only if $z \in y$ for all z), then $x = y$.
- **Pairing:** For all sets x and y, there exists a set $z = \{x, y\}$: that is, a set z such that, for all w, $w \in z$ if and only if ($w = x$ or $w = y$).
- **Union:** For all sets x, there exists a set equal to the union of all sets in x.
- **Existence of infinite sets:** There exists a set x that contains the empty set and that contains $\{y\}$ for every $y \in x$. (Why must this x have infinitely many elements?)
- **Power set:** For all sets x, there exists a set consisting of the subsets of x.
- **Replacement (actually an infinity of axioms, one for every function A mapping sets to sets):** For all sets x, there exists a set $z = \{A(y) \mid y \in x\}$, which results from applying A to all the elements of x. (Technically, one also has to define what one means by a "function mapping sets to sets," something that can be done although I won't do it here.)
- **Foundation:** All nonempty sets x have a member y such that for all z, either $z \notin x$ or $z \notin y$. (This is a technical axiom, whose point is to rule out sets like $\{\{\{\{\ldots\}\}\}\}$.)

These axioms – called the Zermelo–Fraenkel axioms – are the foundation for basically all of math. So I thought you should see them at least once in your life.

Alright, one of the most basic questions we can ask about a set is: how big is it? What's its size, its cardinality? Meaning, how many elements does it have? You might say, just count the elements. But what if there are infinitely many? Are there more integers than odd integers? This brings us to Georg Cantor (1845–1918), and the first of his several enormous contributions to human knowledge. He says two sets have the same cardinality if and only if their elements can be put in one-to-one correspondence. Period. And if, no matter how you try to pair off the elements, one set always has elements left over, the set with the elements left over is the bigger set.

What possible cardinalities are there? Of course, there are finite ones, one for each natural number. Then there's the first infinite cardinality, the cardinality of the integers, which Cantor called $\aleph_0$ ("aleph-zero"). The rational numbers have the same cardinality $\aleph_0$, a fact that's also expressed by saying that the rational numbers are *countable*, meaning that they can be placed in one-to-one correspondence with the integers. In other words, we can make an infinite list of them so that each rational number appears eventually in the list.

What's the proof that the rational numbers are countable? You haven't seen it before? Oh, alright. First, list 0 and then all the rational numbers where the sum of absolute values of the numerator and denominator is 2. Then, list all the rational numbers where the sum of absolute values of the numerator and denominator is 3. And so on. It's clear that every rational number will eventually appear in this list. Hence, there's only a countable infinity of them. QED.

But Cantor's biggest contribution was to show that not *every* infinity is countable – so, for example, the infinity of real numbers is greater than the infinity of integers. More generally, just as there are infinitely many numbers, there are also infinitely many infinities.

You haven't seen the proof of that either? Alright, alright. Let's say you have an infinite set A. We'll show how to produce another infinite set, B, which is even bigger than A. This B will simply be the set of all *subsets* of A, which is guaranteed to exist by the power set axiom. How do we know B is bigger than A? Well, suppose we could pair off every element $a \in A$ with an element $f(a) \in B$, in such a way that no elements of B were left over. Then, we could define a new subset $S \subseteq A$, consisting of *every a that's not contained in f(a)*. Then S is also an element of B. But notice that S can't have been paired off with any $a \in A$ – since otherwise, a would be contained in $f(a)$ if and only if it *wasn't* contained in $f(a)$, contradiction. Therefore, B is larger than A, and we've ended up with a bigger infinity than the one we started with.

This is certainly one of the four or five greatest proofs in all of math – again, good to see at least once in your life.

Besides cardinal numbers, it's also useful to discuss *ordinal* numbers. Rather than defining these, it's easier to just illustrate them. We start with the natural numbers:

$$0, 1, 2, 3, \ldots$$

Then, we say, let's *define* something that's greater than every natural number:

$$\omega$$

What comes after ω?

$$\omega + 1, \omega + 2, \ldots$$

Now, what comes after all of these?

$$2\omega$$

Alright, we get the idea:

$$3\omega, 4\omega, \ldots$$

Alright, we get the idea:

$$\omega^2,\ \omega^3,\ \ldots$$

Alright, we get the idea:

$$\omega^\omega,\ \omega^{\omega^\omega},\ \ldots$$

We could go on for quite a while! Basically, for any set of ordinal numbers (finite or infinite), we stipulate that there's a first ordinal number that comes after everything in that set.

The set of ordinal numbers has the important property of being *well ordered*, which means that every subset has a minimum element. This is unlike the integers or the positive real numbers, where any element has another that comes before it.

Now, here's something interesting. All of the ordinal numbers I've listed have a special property, which is that they have at most countably many predecessors (i.e., at most $\aleph_0$ of them). What if we consider the set of *all* ordinals with at most countably many predecessors? Well, that set also has a successor, call it α. But does α itself have $\aleph_0$ predecessors? Certainly not, since otherwise α wouldn't be the successor to the set; it would be *in* the set! The set of predecessors of α has the next possible cardinality, which is called $\aleph_1$.

What this sort of argument proves is that the set of cardinalities is *itself* well ordered. After the infinity of the integers, there's a "next bigger infinity," and a "next bigger infinity after that," and so on. You never see an infinite decreasing sequence of infinities, as you do with the real numbers.

So, starting from $\aleph_0$ (the cardinality of the integers), we've seen two different ways to produce "bigger infinities than infinity." One of these ways yields the cardinality of sets of integers (or, equivalently, the cardinality of real numbers), which we denote $2^{\aleph_0}$. The other way yields $\aleph_1$. Is $2^{\aleph_0}$ *equal* to $\aleph_1$? Or to put it another way: is there any infinity of *intermediate* size between the infinity of the integers and the infinity of the reals?

Well, this question was David Hilbert's first problem in his famous 1900 address. It stood as one of the great math problems for over half a century, until it was finally "solved" (in a somewhat disappointing way, as you'll see).

Cantor himself believed there were no intermediate infinities, and called this conjecture the Continuum Hypothesis. Cantor was extremely frustrated with himself for not being able to prove it.

Besides the Continuum Hypothesis, there's another statement about these infinite sets that no one could prove or disprove from the Zermelo–Fraenkel axioms. This statement is the infamous Axiom of Choice. It says that, if you have a (possibly infinite) set of sets, then it's possible to form a new set by choosing one item from each set. Sound reasonable? Well, if you accept it, you also have to accept that there's a way to cut a solid sphere into a finite number of pieces, and then rearrange those pieces into another solid sphere a thousand times its size. (That's the "Banach–Tarski paradox." Admittedly, the "pieces" are a bit hard to cut out with a knife . . .)

Why does the Axiom of Choice have such dramatic consequences? Basically, because it asserts that certain sets exist, but without giving any rule for *forming* those sets. As Bertrand Russell put it: "To choose one sock from each of infinitely many pairs of socks requires the Axiom of Choice, but for shoes the Axiom is not needed." (What's the difference?)

The Axiom of Choice turns out to be equivalent to the statement that every set can be well ordered: in other words, the elements of any set can be paired off with the ordinals $0, 1, 2, \ldots, \omega, \omega + 1, \ldots, 2\omega, 3\omega, \ldots$ up to some ordinal. If you think, for example, about the set of real numbers, this seems far from obvious.

It's easy to see that well-ordering implies the Axiom of Choice: just well-order the whole infinity of socks, then choose the sock from each pair that comes first in the ordering.

Do you want to see the other direction? Why the Axiom of Choice implies that every set can be well ordered? Yes?

OK! We have a set A that we want to well-order. For every proper subset $B \subset A$, we'll use the Axiom of Choice to pick an element $f(B) \in A - B$ (where $A - B$ means the set of all elements of A that aren't also elements of B). Now we can start well-ordering A, as follows: first let $s_0 = f(\{\})$, then let $s_1 = f(\{s_0\})$, $s_2 = f(\{s_0, s_1\})$, and so on.

Can this process go on forever? No, it can't. For if it did, then by a process of "transfinite induction," we could stuff arbitrarily large infinite cardinalities into A. And while admittedly A is infinite, it has at most a *fixed* infinite size! So the process has to stop somewhere. But where? At a proper subset B of A? No, it can't do that either – since if it did, then we'd just continue the process by adding $f(B)$. So the only place it can stop is A itself. Therefore, A can be well ordered.

3 Gödel, Turing, and friends

In the last chapter, we talked about the rules for first-order logic. There's an amazing result called Gödel's Completeness Theorem that says that these rules are all you ever need. In other words: if, starting from some set of axioms, you can't derive a contradiction using these rules, then the axioms must have a model (i.e., they must be consistent). Conversely, if the axioms are inconsistent, then the inconsistency can be proved using these rules alone.

Think about what that means. It means that Fermat's Last Theorem, the Poincaré Conjecture, or any other mathematical achievement you care to name can be proved by starting from the axioms for set theory, and then applying these piddling little rules over and over again. Probably 300 million times, but still . . .

How does Gödel prove the Completeness Theorem? The proof has been described as "extracting semantics from syntax." We simply cook up objects to order as the axioms request them! And if we ever run into an inconsistency, that can only be because there was an inconsistency in the original axioms.

One immediate consequence of the Completeness Theorem is the *Löwenheim–Skolem Theorem*: every consistent set of axioms has a model of at most countable cardinality. (Note: One of the best predictors of success in mathematical logic is having an umlaut in your name.) Why? Because the process of cooking up objects to order as the axioms request them can only go on for a countably infinite number of steps!

It's a shame that, after proving his Completeness Theorem, Gödel never really did anything else of note. (Pause for comic effect.) Well, alright, I guess a year later he proved the *Incompleteness* Theorem.

The Incompleteness Theorem says that, given any consistent, computable set of axioms, there's a true statement about the integers that can never be proved from those axioms. Here, *consistent* means that you can't derive a contradiction, while *computable* means that either there are finitely many axioms, or else if there are infinitely many, at least there's an algorithm to generate all the axioms.

(If we didn't have the computability requirement, then we could simply take our "axioms" to consist of all true statements about the integers! In practice, that isn't a very useful set of axioms.)

But wait! Doesn't the Incompleteness Theorem contradict the Completeness Theorem, which says that any statement that's entailed by the axioms can be proved from the axioms? Hold that question; we're gonna clear it up later.

First, though, let's see how the Incompleteness Theorem is proved. People always say "the proof of the Incompleteness Theorem was a technical tour de force, it took 30 pages, it requires an elaborate construction involving prime numbers," etc. Unbelievably, 80 years after Gödel, that's still how the proof is presented in math classes!

Alright, should I let you in on a secret? The proof of the Incompleteness Theorem is about *two lines*. It's almost a triviality. The caveat is that, to give the two-line proof, you first need the concept of a computer.

When I was in junior high school, I had a friend who was really good at math, but maybe not so good at programming. He wanted to write a program using arrays, but he didn't know what an array was. So what did he do? He associated each element of the array with a unique prime number, then he multiplied them all together; then, whenever he wanted to read something out of the array, he *factored* the product. (If he was programming a quantum computer, maybe that wouldn't be quite so bad!) Anyway, what my friend did, that's basically what Gödel did. He made up an elaborate hack in order to program without programming.

TURING MACHINES

OK, time to bring Mr. T. on the scene.

In 1936, the word "computer" meant a person (usually a woman) whose job was to compute with pencil and paper. Turing wanted to show that, in principle, such a "computer" could be simulated by a machine. What would the machine look like? Well, it would have to able to write down its calculations somewhere. Since we don't really care about handwriting, font size, etc., it's easiest to imagine that the calculations are written on a sheet of paper divided into squares, with one symbol per square, and a finite number of possible symbols. Traditionally, paper has two dimensions, but without loss of generality we can imagine a long, one-dimensional paper tape. How long? For the time being, we'll assume as long as we need.

What can the machine do? Well, clearly it has to be able to read symbols off the tape and modify them based on what it reads. We'll assume for simplicity that the machine reads only one symbol at a time. But in that case, it had better be able to move back and forth on the tape. It would also be nice if, once it's computed an answer, the machine can halt! But at any time, how does the machine decide which things to do? According to Turing, this decision should depend only on two pieces of information: (1) the symbol currently being read, and (2) the machine's current "internal configuration" or "state." Based on its internal state and the symbol currently being read, the machine should (1) write a new symbol in the current square, overwriting whatever symbol is there, (2) move backward or forward one square, and (3) switch to a new state or halt.

Finally, since we want this machine to be physically realizable, the number of possible internal states should be finite. These are the only requirements.

Turing's first result is the existence of a "universal" machine: a machine whose job is to simulate any other machine described via symbols on the tape. In other words, *universal programmable computers can exist*. You don't have to build one machine for email, another for playing DVDs, another for Tomb Raider, and so on: you can build a single machine that simulates any of the other machines, by running different programs stored in memory. But this result is not even the main result of the paper.

So what's the main result? It's that there's a basic problem, called the halting problem, that no program can ever solve. The halting problem is this: we're given a program, and we want to decide if it ever halts. Of course, we can run the program for a while, but what if the program hasn't halted after a million years? At what point should we give up?

One piece of evidence that this problem might be hard is that, if we *could* solve it, then we could also solve many famous unsolved math problems. For example, Goldbach's Conjecture says that every even number 4 or greater can be written as a sum of two primes. Now, we can easily write a program that tests 4, 6, 8, and so on, halting only if it finds a number that can't be written as a sum of two primes. Then deciding whether that program ever halts is equivalent to deciding the truth of Goldbach's Conjecture.

But can we *prove* there's no program to solve the halting problem? This is what Turing does. His key idea is not even to *try* to analyze the internal dynamics of such a program, supposing it existed. Instead, he simply says, suppose by way of contradiction that such a program P exists. Then, we can modify P to produce a new program P′ that does the following. Given another program Q as input, P′

(1) runs forever if Q halts given its own code as input, or
(2) halts if Q runs forever given its own code as input.

Now, we just feed P′ its own code as input. By the conditions above, P′ will run forever if it halts, or halt if it runs forever. Therefore, P′ – and by implication P – can't have existed in the first place.

As I said, once you have Turing's results, Gödel's results fall out for free as a bonus. Why? Well, suppose the Incompleteness Theorem was false – that is, there existed a consistent, computable proof system F from which any statement about integers could be either proved or disproved. Then given a computer program, we could simply search through every possible proof in F, until we found either a proof that the program halts or a proof that it doesn't halt. This is possible because the statement that a particular computer program halts is ultimately just a statement about integers. But this would give us an algorithm to solve the halting problem, which we already know is impossible. Therefore, F can't exist.

By thinking more carefully, we can actually squeeze out a stronger result. Let P be a program that, given as input another program Q, tries to decide whether Q halts by the strategy above (i.e., searching through every possible proof and disproof that Q halts in some formal system F). Then, as in Turing's proof, suppose we modify P to produce a new program P′ that

(1) runs forever if Q given its own code as input is proved to halt, or
(2) halts if Q given its own code as input is proved to run forever.

Now suppose we feed P′ its own code as input. Then we know that P′ will run forever, without ever discovering a proof or disproof that it halts. For if P′ finds a proof that it halts, then it will run forever, and if it finds a proof that it runs forever, then it will halt, which is a contradiction.

But there's an obvious paradox: why isn't the above argument, *itself*, a proof that P′ will run forever given its own code as input? And why won't P′ discover this proof that it runs forever – and therefore halt, and therefore run forever, and therefore halt, etc.?

The answer is that, in "proving" that P′ runs forever, we made a hidden assumption: namely, that the proof system F is consistent. If F were inconsistent, then there could perfectly well be a proof that P′ halts, even if the reality were that P′ ran forever.

But this means that, if F could *prove* that F was consistent, then F could also prove that P′ ran forever – thereby bringing back the above contradiction. The only possible conclusion is that *if F*

"Greater." (The axioms aren't stupid: they know that if they said "smaller," then you could simply try every smaller number and verify that none of them encode a proof of PA's inconsistency.)

"Alright then, what's $X + 1$?"
"Y."

And so on. The axioms will keep cooking up fictitious numbers to satisfy your requests, and assuming that PA itself is consistent, you'll never be able to trap them in an inconsistency. The point of the Completeness Theorem is that the whole infinite set of fictitious numbers the axioms cook up will constitute a *model* for PA – just not the usual model (i.e., the ordinary positive integers)! If we insist on talking about the usual model, then we switch from the domain of the Completeness Theorem to the domain of the Incompleteness Theorem.

Do you remember the puzzle from Chapter 2? The puzzle was whether there's any theorem that can only be proved by assuming as an axiom that it *can* be proved. In other words, does "just believing in yourself" make any formal difference in mathematics? We're now in a position to answer that question.

Let's suppose, for concreteness, that the theorem we want to prove is the Riemann Hypothesis (RH), and the formal system we want to prove it in is Zermelo–Fraenkel set theory (ZF). Suppose we can prove in ZF that, if ZF proves RH, then RH is true. Then taking the contrapositive, we can also prove in ZF that if RH is false, then ZF does *not* prove RH. In other words, we can prove in ZF + not(RH) that not(RH) is perfectly consistent with ZF. But this means that the theory ZF + not(RH) proves its own consistency – and this, by Gödel, means that ZF + not(RH) is inconsistent. But saying that ZF + not(RH) is inconsistent is equivalent to saying that RH is a theorem of ZF. Therefore, we've proved RH. In general, we find that, if a statement can be proved by assuming as an axiom that it's provable, then it can also be proved *without* assuming that axiom. This result is known as Löb's Theorem (again with the umlauts), though personally I think that a better name would be the "You-Had-the-Mojo-All-Along Theorem."

Oh, you remember earlier we talked about the Axiom of Choice and the Continuum Hypothesis? These are natural statements about the continuum that, since the continuum is such a well-defined mathematical entity, must certainly be either true or false. So, how did those things ever get decided? Well, Gödel proved in 1939 that assuming the Axiom of Choice (AC) or the Continuum Hypothesis (CH) can never lead to an inconsistency. In other words, if the theories ZF + AC or ZF + CH were inconsistent, that could only be because ZF itself was inconsistent.

This raised an obvious question: can we also consistently assume that AC and CH are *false*? Gödel worked on this problem but wasn't able to answer it. Finally, Paul Cohen gave an affirmative answer in 1963, by inventing a new technique called "forcing." (For that, he won the only Fields Medal that's ever been given for set theory and the foundations of math.)

So, we now know that the usual axioms of mathematics don't decide the Axiom of Choice and the Continuum Hypothesis one way or another. You're free to believe both, neither, or one and not the other without fear of contradiction. And sure enough, opinion among mathematicians about AC and CH remains divided to this day, with many interesting arguments for and against (which we unfortunately don't have time to explore the details of).

Let me end with a possibly surprising observation: the independence of AC and CH from ZF set theory *is itself a theorem of Peano Arithmetic*. For, ultimately, Gödel and Cohen's consistency theorems boil down to combinatorial assertions about manipulations of first-order sentences – which can in principle be proved directly, without ever thinking about the transfinite sets that those sentences purport to describe. (In practice, translating these results into combinatorics would be horrendously complicated, and Cohen has said that *trying* to think about these problems in finite combinatorial terms led him nowhere. But we know that in theory it could be done.) This provides a nice illustration of what, to me, is the central philosophical question underlying this whole business: do we ever *really* talk about the continuum, or do we only ever talk about finite sequences of symbols that talk about the continuum?

BONUS ADDENDUM

What does any of this have to do with quantum mechanics? I will now attempt the heroic task of making a connection. What I've tried to impress on you is that there are profound difficulties if we want to assume the world is continuous. Take a pen, for example: how many different positions can I put it in on the surface of a table? $\aleph_1$? More than $\aleph_1$? Less than $\aleph_1$? We don't want the answers to "physics" questions to depend on the axioms of set theory!

Ah, but you say my question is physically meaningless, since the pen's position could never actually be measured to infinite precision? Sure – but the point is, you need a physical theory to *tell* you that!

Of course, quantum mechanics gets its very name from the fact that a lot of the observables in the theory, like energy levels, are discrete – "quantized." This seems paradoxical, since one of the criticisms that computer scientists level against quantum computing is that, as they see it, it's a *continuous* model of computation!

My own view is that quantum mechanics, like classical probability theory, should be seen as somehow "intermediate" between a continuous and discrete theory. (Here, I'm assuming that the Hilbert space[1] or probability space is finite dimensional.) What I mean is that,

[1] Please don't be alarmed by the term "Hilbert space," which I'll use occasionally in this book. All it means is "the space of all possible quantum states of some system." With *infinite*-dimensional systems, the definition of Hilbert space is a bit subtle – but in this book, we'll only care about finite-dimensional systems. And as we'll see in Chapter 9, the Hilbert space of a finite-dimensional system is nothing other than $\mathbb{C}^N$: an N-dimensional complex vector space.

while there *are* continuous parameters (the probabilities or amplitudes, respectively), those parameters are not directly observable, and that has the effect of "shielding" us from the bizarro universe of the Axiom of Choice and the Continuum Hypothesis. We don't need a detailed physical theory to tell us that whether amplitudes are rational or irrational, whether there are more or less than $\aleph_1$ possible amplitudes, etc., are physically meaningless questions. This follows directly from the fact that, if we wanted to learn an amplitude exactly, then (even assuming no error!) we would need to measure the appropriate state infinitely many times.

EXERCISE

Let BB(n), or the "nth Busy Beaver number," be the maximum number of steps that an n-state Turing machine can make on an initially blank tape before halting. (Here, the maximum is over all n-state Turing machines that eventually halt.)

1. Prove that BB(n) grows faster than any computable function.
2. Let $S = 1/\mathrm{BB}(1) + 1/\mathrm{BB}(2) + 1/\mathrm{BB}(3) + \cdots$

   Is S a computable real number? In other words, is there an algorithm that, given as input a positive integer k, outputs a rational number S' such that $|S - S'| < 1/k$?

FURTHER READING

An excellent resource for the material in this chapter is *Gödel's Theorem: An Incomplete Guide to its Use and Abuse*, by Torkel Franzén (A. K. Peters Ltd, 2005).

4 Minds and machines

Now we're going to launch into something I know you've all been waiting for: a philosophical food fight about minds, machines, and intelligence!

First, though, let's finish talking about computability. One concept we'll need again and again in this chapter is that of an oracle. The idea is a pretty obvious one: we *assume* we have a "black box," or "oracle," that immediately solves some hard computational problem, and then see what the consequences are! (When I was a freshman, I once started talking to my professor about the consequences of a hypothetical "NP-completeness fairy": a being that would instantly tell you whether a given Boolean formula was satisfiable or not. The professor had to correct me: they're not called "fairies"; they're called "oracles." *Much* more professional!)

Oracles were apparently first studied by Turing, in his 1938 PhD thesis. Obviously, anyone who could write a whole thesis about these fictitious entities would have to be an *extremely* pure theorist, someone who wouldn't be caught dead doing anything relevant. This was certainly true in Turing's case – indeed, he spent the years after his PhD, from 1939 to 1943, studying certain abstruse symmetry transformations on a 26-letter alphabet.

Anyway, we say that problem A is *Turing reducible* to problem B, if A is solvable by a Turing machine given an oracle for B. In other words, "A is no harder than B": if we had a hypothetical device to solve B, then we could also solve A. Two problems are *Turing equivalent* if each is Turing reducible to the other. So, for example, the problem of whether a statement can be proved from the axioms of set theory is Turing equivalent to the halting problem: if you can solve one, you can solve the other.

could do the first step of a computation in one second, the next step in a half second, the next step in a quarter second, the next step in an eighth second, and so on. Then in two seconds you'll have done an infinite amount of computation! Well, as stated it sounds a bit silly, so maybe sex it up by throwing in a black hole or something. How could the hidebound Turing reactionaries possibly object? (It reminds me of the joke about the supercomputer that was so fast, it could do an infinite loop in 2.5 seconds.)

We should immediately be skeptical that, if Nature was going to give us these vast computational powers, she would do so in a way that's so mundane, so uninteresting. Without making us sweat or anything. But admittedly, to *really* see why the hypercomputing proposals fail, you need the entropy bounds of Bekenstein, Bousso, and others – which are among the few things the physicists think they know about quantum gravity, and which we'll say something about later in the book. So the Church–Turing Thesis – even its original, nonextended version – really is connected to some of the deepest questions in physics. But in my opinion, neither quantum computing, nor analog computing, nor anything else, has mounted a serious challenge to that thesis in the 75 years since it was formulated.

A closely-related objection to this computation by geometric series is that we do sort of understand why this model isn't physical: we believe that the very notion of time starts breaking down when you get down to around 10^{-43} seconds (the Planck scale). We don't know exactly what happens there. Nevertheless, the situation seems not the slightest bit analogous to quantum computing (for example). In quantum computing, as we'll see, no one has any quantitative idea of where the theory could break down and the computer could stop working – which leads to the conjecture that maybe it *won't* stop working.

Once you get to the Planck scale, you might say we're getting into a really sophisticated argument. Why not just say you're always limited in practice by noise and imperfection?

The question is why are you limited? *Why* can't you store a real number in a register? I think that if you try to make the argument precise, ultimately, you're going to be talking about the Planck scale.