Progress in Computer Science and Applied Logic

Arthur O. Pittenger

An Introduction to Quantum Computing Algorithms

SPRINGER SCIENCE+BUSINESS
MEDIA, LLC

Progress in Computer Science and Applied Logic
Volume 19

Arthur O. Pittenger

An Introduction to Quantum Computing Algorithms

Springer Science+Business Media, LLC

Arthur O. Pittenger
Department of Mathematics and Statistics
University of Maryland, Baltimore County
Baltimore, MD 21250
U.S.A.

Library of Congress Cataloging-in-Publication Data
Pittenger, Arthur O., 1936-
An introduction to quantum computing algorithms / Arthur O. Pittenger.
p. cm. — (Progress in computer science and applied logic ; v. 19)
Includes bibliographical references and index.
ISBN 978-1-4612-7127-7 ISBN 978-1-4612-1390-1 (eBook)
DOI 10.1007/978-1-4612-1390-1
1. Quantum computers 2. Computer algorithms. I. Title. II. Series.
QA76.889.P58 1999
004.1—dc21

99-0457513
CIP

AMS Subject Classifications: 68Q10, 68Q20

Printed on acid-free paper.

Originally published by Birkhäuser Boston in 2000
Softcover reprint of the hardcover 1st edition 2000

ISBN 0-8176-4127-0 SPIN 10721226

Reformatted from author's files in LaTeX 2ε by TEXniques, Inc., Cambridge, MA.

9 8 7 6 5 4 3 2

Contents

Preface

In 1994 Peter Shor [65] published a factoring algorithm for a quantum computer that finds the prime factors of a composite integer N more efficiently than is possible with the known algorithms for a classical computer. Since the difficulty of the factoring problem is crucial for the security of a public key encryption system, interest (and funding) in quantum computing and quantum computation suddenly blossomed. Quantum computing had arrived.

The study of the role of quantum mechanics in the theory of computation seems to have begun in the early 1980s with the publications of Paul Benioff [6], [7] who considered a quantum mechanical model of computers and the computation process. A related question was discussed shortly thereafter by Richard Feynman [35] who began from a different perspective by asking what kind of computer should be used to simulate physics. His analysis led him to the belief that with a suitable class of "quantum machines" one could imitate any quantum system.

A separate line of research into the thermodynamics of computation in the 1970s and 1980s had led to the concept of "reversible programming." Motivated by mechanistic and biological models of computation, the idea was that the energy expenditures of computation could be reduced if no machine state had more than one logical precursor. (See [8] for a survey of the field at that time and associated references.) Since a quantum mechanical model of computation would involve unitary transformations, which are indeed invertible, there was the additional incentive to explore the advantages and limitations of computing devices based on quantum

mechanics. In particular it was soon noticed that the logic gates of reversible programming [74], for example, could also be modeled as specific unitary maps of collections of two-state quantum systems, as in [36] where a simple addition routine is described.

Building on these ideas David Deutsch [25] developed the thesis that quantum theory and a "universal quantum computer" constitute the correct context for an extended Church–Turing principle, that is, that "every finitely realizable physical system can be perfectly simulated by a universal model computing machine operating by finite means." He also analyzed the role of quantum parallelism, which we will see is a key element in the theory of quantum algorithms, and he commented on the role of quantum complexity theory.

In [26] Deutsch expanded on the concept of quantum computational circuits, and in a 1992 paper with Richard Jozsa [27] described an algorithm that illustrates the power of quantum computation. Other algorithms demonstrating the power of quantum parallelism and "entanglement" in the solution of theoretical computational problems soon followed, and a 1994 result of Simon [67] inspired Shor's work.

The purpose of this book is to make these quantum algorithms accessible to the mathematically literate reader who is not familiar with the subject and who may have forgotten, or has never seen, some of the diverse tools and notation of the trade. The challenge has been to provide enough background to make the context and details of the algorithms understandable, but without lapsing into encyclopedic detail.

The focus in this exposition is exclusively on the "software" side of the subject. While there is intense experimental work on the "hardware" side of quantum computation, progress has been much slower. For example, the problems involved in creating stable microphysical devices for "quantum bits" have motivated experiments that are at the boundary of currently feasible technology. Indeed, there continues to be some question about the ultimate feasibility of a quantum computer complex enough to implement Shor's algorithm for even modestly large N.

We begin in Chapter 1 by very briefly motivating the need for quantum mechanics and then developing and illustrating some of the mathematical framework of finite-dimensional quantum statics. There are no formal prerequisites, but the reader is assumed to be familiar with undergraduate linear algebra. Key definitions and theorems from linear algebra are introduced in the context of the static model and then rewritten in Dirac notation, which is used unabashedly thereafter. References are given for the reader who wishes a more complete presentation of the history of

quantum mechanics and of the physical motivation for the mathematical model.

In Chapter 2 the mathematical framework is further expanded to accommodate multi-particle models and to illustrate the motivation for some of the notation. Wiring diagrams and quantum logic gates are introduced, and some representative quantum subroutines are presented. In particular, a simple "teleportation" circuit is examined in detail.

Chapter 3 illustrates the development of quantum algorithms from the explicit to the abstract. We begin with the Deutsch–Jozsa algorithm and describe what proves to be the typical structure of most quantum algorithms. Grover's search algorithm differs somewhat from the others, and we examine it in a simple case, revealing the structure that applies in general. In a similar fashion we work through Shor's algorithm in the simple case of $N = 15$ and identify the techniques and the difficulties that occur in the more general context. A key step in Shor's factoring algorithm is the finite Fourier transform, and we illustrate its implementation as well. This leads to a discussion of the role of eigenvalues and eigenvectors, and the chapter concludes with a group theoretic model of quantum algorithms.

Since the implementation of even the simplest logic gate is experimentally challenging, it was soon recognized that error-correction techniques could play an important role in a functioning quantum computer. Seminal papers by Shor [64] and Steane [68, 69] suggested a methodology, and a theory of quantum error-correcting codes was soon in place. In Chapter 4 we trace the development of this class of algorithms, beginning with a brief discussion of the physical problem of decoherence and discussing the early models of Shor and Steane. A "truly quantum" five-"qubit" code is discussed in detail, since it provides the paradigm for the "stabilizer" codes which are described subsequently. The relation of classical codes to stabilizer codes is recorded, and we conclude with a discussion of error-correcting codes from the abstract Hilbert space perspective à la Knill and Laflamme [47].

There is much more to the subject of quantum computation and related fields than we discuss here. For example, in 1984 Charles Bennett and Gilles Brassard [10] introduced a key distribution system based on quantum mechanical principles. (For a novel extension of the ideas in [10], one which has a practical experimental realization, see Ekert [81].) Without going into detail, the goal is to enable sender and receiver (Alice and Bob) to securely construct a key sequence for encryption using the properties of photons. Thus, concurrently with quantum computers and quantum algorithms, the fields of quantum cryptography and quantum

communication were developing. This quickly led to quantum information theory and to related complexity issues.

The notation and results we present here provide a good foundation for further reading in these various subjects. A broad overview from the information-theoretic perspective is given in [70], and readers interested in learning more about these various topics will find appropriate references at the start of the bibliography section.

Acknowledgments

An Introduction to Quantum Computing Algorithms reflects my own experience in learning the mathematics and theoretical physics required for the subject. I am most grateful to many colleagues who helped in that process, especially the core members of our Quantum Computation Seminar at the Baltimore County campus of the University of Maryland (UMBC) who comprised the (usually) patient audience for the initial presentation of parts of this book. Special thanks go to Rich Davis, Keith Miller, and Mort Rubin who read all or parts of the manuscript in its various incarnations. Readers who find additional errors are encouraged to notify me at pittenge@math.umbc.edu.

I am also indebted to the many scholars who routinely made their research available on the Los Alamos web site quant-ph. The field has developed so quickly that an introduction of this nature would not have been possible otherwise.

Ann Kostant at Birkhäuser Boston was most supportive throughout the entire process, and indeed there would have been no book without her encouragement. Viet Ngo provided crucial support in preparing the various drawings, and I also received critical technical help in the preparation of the manuscript from Rouben Rostamian, Boris Alemi, and Esther M. Scheffel, colleagues at UMBC. That assistance is gratefully acknowledged.

Finally, I would like to record my appreciation to my family — Judith, Laurence, Christopher and Elise — for making the last however many years worthwhile and for their continuous support coupled, of course,

with occasional subtle reminders that my intellectual activities were not the center of the family universe.

1
Quantum Statics

To begin to describe quantum algorithms, we need an understanding of the physical phenomena that motivate quantum mechanics, a simple mathematical model of a potential quantum storage device and the notation to describe and manipulate such a model. In this chapter we present the context in which we will work, motivate the need for quantum mechanics and provide a quick introduction to the essential mathematical tools and notation to be used throughout the book. As part of the price of brevity, we will work with a static model of quantum mechanics and will defer additional mathematical definitions until they are needed. The reader who is already familiar with this material can easily skip to Chapter 2, giving only a cursory glance at Chapter 1 to identify any idiosyncratic notation.

1.1 Context

In a "classical" digital computer, information is physically stored as bits, each of which represents a zero or a one, and n-long vectors of bits are manipulated to model functional evaluation and to implement algorithms. The stability of the stored information and the reliability of these manipulations is based on classical physics and the use of error-correction techniques. In particular, it is assumed that one can examine storage devices to determine the "value" of their contents without affecting those contents.

However, if information is stored at the microphysical level where quantum mechanics prevails, the context is radically different. For example, at

the quantum level a storage bit could represent both zero and one simultaneously, and measurements and manipulations of those quantum bits turn out to be quite different processes that are modeled as matrix operations. Thus, the physical context in which information is represented becomes an important aspect of the theory.

One of the basic principles of quantum mechanics is the *correspondence principle*, whereby quantum mechanical results lead to classical results as the physical system moves from the microscopic to the macroscopic. The correspondence principle played an important role in the development of quantum mechanics, and certain classical concepts such as Hamiltonians and Poisson brackets are embedded in the conceptual framework of quantum mechanics. Thus, an introduction to even elementary quantum mechanics could well require a firm foundation in classical physics to motivate the resulting mathematical model.

The mathematical structure of quantum mechanics is quite sophisticated. Indeed the requirements and the success of the theory have stimulated the development of modern mathematics to a remarkable degree. Thus, an introduction to quantum mechanics could well presuppose a high level of mathematical sophistication and an understanding of mathematics ranging from abstract functional analysis to group representations.

In this book we take a middle road by defining a mathematical model which provides the essential ideas of quantum mechanics without requiring the full scope of mathematics and physics necessary for a complete theory. The goal is to provide the appropriate context for understanding and developing algorithms that are suitable for a prospective quantum computer. Since the model itself may appear quite arbitrary, we motivate the need for some of the axioms of quantum mechanics by giving a brief overview of two experiments. For a careful development of the conceptual meaning of quantum mechanics, the reader is referred to Peres [56]. For a comprehensive view of the subject and its development, the reader is also referred to Shankar [63] and Sakurai [60] for a good introduction to the mathematics of quantum mechanics.

1.2 Experimental motivation for quantum mechanics

Let us begin with the two-slit experiment for electrons. (For a thorough but elementary exposition of this experiment, see Volume III of Feynman's *Lectures in Physics* [35].) A source produces a mono-energetic

stream of electrons which move in the positive x-direction. An absorbing screen, perpendicular to the electron stream, has two narrow slits through which the electrons can pass on their way to counters on a second screen set beyond the first screen. If only one of the slits is open, the counters record a pattern of impacts that resembles a truncated normal distribution, i.e., the distribution one would expect from a stream of particles impinging on one slit. However, if both slits are open and are sufficiently close to one another, the distribution of impacts shows interference, that is, the same distribution of intensity that one observes from monochromatic light waves impinging on the second screen. In particular, the pattern of observed counts with two slits open is not the sum of the patterns of counts obtained by opening each slit separately.

Moreover, since the experiment can be conducted so that the impinging electrons are separated in time, the wavelike behavior of interference suggests that each electron is somehow interfering with itself, which is completely inconsistent with the particulate presumption that each electron must have gone through exactly one of the slits. And to make matters more confusing, it can be argued (cf. [35]) that no matter how cleverly one attempts to measure which slit each electron traversed, the interference pattern will be affected.

The Stern–Gerlach experiment is another of the fundamental experiments that motivate the mathematical structure of quantum mechanics. It has been found that an "elementary" particle has an intrinsic angular momentum or a "spin" which can be experimentally detected by beaming a stream of particles through an appropriate inhomogeneous magnetic field. (See, for example, [60] or [63].) Moreover, it is an experimental fact that a quantization effect is observed: instead of observing a continuum of deflections, the beam of particles is deflected in a finite number of ways. In particular, it is observed that certain particles have "spin 1/2," which means that the impinging beam of particles splits into exactly two subbeams.

More concretely, if a suitably prepared beam of particles is moving along the x-axis, and if the magnetic field is arranged to have a large inhomogeneous component in a direction perpendicular to the x-axis, say in the z-direction, then one subbeam diverges (up) in the positive z direction and the other diverges (down) in the negative z direction. Although one cannot predict whether an *individual* particle will go up or down, the two subbeams have equal intensity, and we describe the particles in the first subbeam as having z-spin up or being in a state $(z, +)$ and the particles in the second subbeam as having z-spin down or as being in a state $(z, -)$. Moreover, if a subsequent Stern–Gerlach measurement

is made on the $(z,+)$ subbeam with the same relative orientation, then only one beam emerges - the $(z,+)$ beam itself, as would be expected intuitively. Thus, prior to the second experiment the behavior of the particles in the $(z,+)$ subbeam can be predicted, they will emerge in state $(z,+)$.

The same phenomenon is observed in other directions, so that if a subsequent Stern–Gerlach experiment with the inhomogeneity in the y direction, a "y-oriented" Stern–Gerlach experiment, is performed on (say) the $(z,+)$ subbeam, two new subbeams are observed, denoted by $(y,+)$ and $(y,-)$ in the obvious notation. Again, the two new subbeams will have equal intensity although in which y-direction an individual particle goes cannot be predicted.

But now a strange thing happens. If a second z-oriented Stern–Gerlach measurement is made on the $(y,+)$ subbeam which was selected from the $(z,+)$ subbeam, both a $(z,+)$ subbeam and a $(z,-)$ subbeam are detected in equal intensity. In other words, the measurement of the y-spin orientation somehow erased the effect of the first z-spin measurement on the surviving particles, and this is not an intuitive result for particles.

As with the two-slit experiment, there is an analogy with wave phenomena. Suppose one sends a beam of light through three polarized filters: the polarization directions of the first and third filters are at right angles, and the polarization of the middle filter is at a 45° angle to each of the other two. If the middle filter is removed from the experiment, no light is observed passing through the third filter, as is to be expected. However, if the middle filter is in place, light polarized in the direction of the third filter is observed. Again, after the second filter the selective effects of the first filter on the polarization seem to have vanished. (See [35] for a nice discussion of these ideas.)

Assuming the validity of these assertions for particles and the fact that classical physics cannot account for them, several conclusions can be made. One conclusion is that, depending on the nature of the observation, particles evince wavelike behavior and, based on other experiments, that light exhibits particle-like behavior. A second conclusion is that any theory describing the behavior of matter at the microphysical level cannot ignore the effect an observation has on the physical system. Thus, for example, one cannot modify the two-slit experiment to observe the wavelike behavior of the electrons via the interference pattern, while simultaneously observing particle-like behavior by determining which slits the electrons traversed.

A third conclusion is one of indeterminism, that is, the outcome of a future measurement of a system cannot always be predicted with certainty.

Instead, at the microphysical level, one may be able to predict future behavior only with a prescribed probability. Thus, for example, in the Stern–Gerlach z-spin experiment above, one can predict with certainty that a particle in the $(z, +)$ state will go up in a subsequent z-oriented experiment but can only predict the behavior of the particle in a subsequent y-oriented experiment with a certain probability.

From the two-slit experiment, or more precisely as part of the reasoning underlying an analysis of that experiment, a fourth conclusion is that there is a limit to the accuracy of simultaneous measurements of certain physical properties such as position and momentum. This conclusion was reached by Heisenberg on physical grounds and is denoted in the quantum mechanics literature as the Heisenberg uncertainty principle.

A fifth and crucial conclusion is that one must distinguish between the evolution of a system and the measurement of a system. Sending a spin 1/2 particle through a Stern–Gerlach device is an example of the former; measuring which trajectory the particle took is an example of the latter.

These are nontrivial conclusions to draw from just two examples. Suffice it to say that there is a body of experimental evidence supporting these conclusions as documented, for example, in the cited work by Feynman [35] and in texts on the subject such as those by Bohm [14], Peres [56], Sakurai [60] and Shankar [63]. However, acceptance of these conclusions and of the subsequent mathematical models does not come easily, and the history of the debate over various aspects of the theory makes fascinating reading. (For example, see Jammer [43], Sudbery [72] and Wick [77].)

In particular, Einstein was troubled by aspects of the theory and in a famous paper in 1935 with Podolsky and Rosen [32] proposed a thought experiment which assumed an intuitively appealing principle of "local reality." The principle of local reality concerns the existence of an element of physical reality which corresponds to a measurable physical quantity, and the analysis in EPR [32] leads to the conclusion that one could assign simultaneous values to two particular quantities when quantum mechanics precludes such an assignment. We cannot go into details of the argument here, but the conclusion of EPR was that the theory of quantum mechanics was incomplete. (See [56] for a detailed discussion.)

It was shown by Bell in 1964 that the hypothesis of local reality is testable, and subsequent experiments confirmed the validity of quantum mechanics and the untenability of the local reality hypothesis. These EPR-related experiments are closely connected to the ideas of quantum computation, since they involve "entangled states" which we will see are crucial to quantum algorithms. In addition, some of the EPR experiments

have produced results that are counter-intuitive and serve as a caution against using classical reasoning with quantum algorithms. We resist the temptation to describe these results here and instead refer the reader to the literature for additional information and references. (For a discussion of the Einstein–Padolsky–Rosen paradox and the issue of "locality," see Bohm [14], Jammer [43], Peres [56] or Wick [77]. For Bell's papers on EPR and other aspects of quantum theory, see [5]. For an example of an EPR experiment producing bizarre results, see Pittman et al [57], and for a recent gedanken experiment, see Mermin [53].)

1.3 The basic model

We thus assume that a mathematical apparatus modeling quantum behavior must (1) represent physical states as mathematical entities, (2) model interference phenomena, (3) model nondeterministic predictions, (4) model the effect that a measurement has on the system being measured and (5) allow for evolution of a quantum system as distinct from its measurement. We begin by developing explicitly the standard, nonrelativistic linear model which meets these requirements. Initially we leave implicit the dynamics of the system, concentrating instead on a "static" model.

Suppose y denotes some physical object. If y is a hydrogen atom, for example, then we might be concerned with its trajectory or perhaps with its energy levels. In extreme cases we might even be concerned with the constituent quarks of the proton. The point is that when we represent y in some mathematical model, that representation may capture only some of the possible degrees of freedom of the physical system itself. Thus, the mathematical state space H is specific to the context of analysis.

Moreover, by analogy with the classical model for wave phenomena, the second constraint suggests that H should be a linear space over the complex numbers C, and that vectors correspond to physical states.

(1.1) Definition. H is a Hilbert space over the complex numbers C. The inner product is linear in the second term and complex linear in the first term so that (u, v) equals $\overline{(v, u)}$ and $(w, au + bv) = a(w, u) + b(w, v)$. If H is finite dimensional, vectors are represented as column vectors. •

(1.2) Example. As the simplest example relevant to quantum computing, let H denote a two-dimensional vector space over C with an orthogonal basis of column vectors,

$$H = span\left\{\begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}\right\},$$

where the basis vectors denote the defining states of a two-state quantum system. Since H is a linear space, it thus contains vectors that represent the quantum system as being in a linear combination of the two defining physical states. •

The third constraint requires a probabilistic component in the model, and this is an essential feature of the theory. Indeed, Peres [56] (p. 13) puts it this way: "In a strict sense, quantum theory is a set of rules allowing the computation of probabilities for the outcome of tests which follow specified preparations."

Let us use discrete probability as a motivation for the model. If an observed random phenomenon A has n real outcomes $\{a_1, \ldots, a_n\}$ which occur with respective probabilities $\{p_1, \ldots, p_n\}$, then the expected outcome (expectation) of A is given by $\sum p_k a_k$. To suggest the generalization we follow Parthasarathy [55] and calculate this expectation in another way. Let ρ denote an $n \times n$ matrix with the p_k's in order down the main diagonal, $\rho_{kk} = p_k$, and let A be an $n \times n$ matrix with the real entries a_k also in order down the main diagonal. Then it is easy to check that the expectation of the random variable coincides with the trace of the matrix product ρA.

(1.3) Definition. The *trace* $\mathrm{tr}(A)$ of a square matrix A is defined as the sum of the diagonal entries of A in a particular coordinate representation. It can then be shown that $\mathrm{tr}(A)$ is independent of the coordinate representation and that $\mathrm{tr}(AB) = \mathrm{tr}(BA)$. (See, for example, Axler [1] or any standard linear algebra textbook for proofs.) •

Thus, if U is an invertible matrix, $\mathrm{tr}(UAU^{-1}) = \mathrm{tr}(A)$. For the matrices in (1.8) and (1.9) below, $\mathrm{tr}(\rho) = 1$ and $\mathrm{tr}(A) = 0$.

To make the generalization to H, we retain the matrix representation and the key properties of ρ and A. The probability "context" or *state* of a physical system is then modeled by a positive semidefinite linear transformation ρ which has trace equal to one, imitating the fact that the total weight of a probability measure is one. As we will see below, it is possible to construct an appropriate ρ from a vector in H, and the word "state" in that context is sometimes taken to mean the vector and sometimes taken to mean the probability density ρ constructed from the vector.

The matrix A which models the experimental outcome has real eigenvalues, and its orthogonal eigenvectors span an n-dimensional space. Its generalization, an "observable" A, is taken to be a Hermitian matrix, which means that A continues to have real eigenvalues and a complete set of eigenvectors.

(1.4) Definition. A is a *Hermitian matrix* if $A = A^\dagger$, i.e., $a_{jk} = \bar{a}_{kj}$. •

Using the inner product cleverly, it is easy to confirm that A has real eigenvalues. It is standard linear algebra fare that A also has a complete set of orthonormal eigenvectors and a *spectral decomposition*

$$A = \sum_k \lambda_k P_k, \tag{1.5}$$

where the λ_k's denote the (real) eigenvalues and the P_k's denote orthogonal projections on the corresponding eigenspaces H_k. (See the continuation of Example (1.2) below (1.10).) For proofs see any standard text in linear algebra.

(1.6) Definition. An *observable* A in the (finite-dimensional) state space H is a Hermitian operator. The probability "state" or *density* of the system is a *positive semidefinite* operator ρ with trace one, and the *probability* of observing a system in a subspace H_k is defined as $\mathrm{tr}(\rho P_k)$, where P_k is the orthogonal projection on H_k. The *expectation* of the observable A relative to ρ is $\mathrm{tr}(\rho A)$. •

If A has the representation given in (1.5), it is easy to check that $\mathrm{tr}(\rho A)$ equals $\sum \lambda_k \mathrm{tr}(\rho P_k)$, so that the definitions of probability and expectation are consistent. (For example, see (1.14).) In words, a single measurement of A in state ρ produces a value λ_k with probability $\mathrm{tr}(P_k \rho)$. We call the pair (H, ρ) a *quantum probability space* (*QPS*).

(1.7) Exercise. By definition ρ is positive semidefinite if and only if for any vector u, $(u, \rho u) \geq 0$. Show that such a ρ is also Hermitian. (*Hint*: consider u's with only one or two nonzero components.) •

As an illustration continue with the context of Example (1.2). Let ρ be defined by the *outer product*

$$\rho = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \tag{1.8}$$

and let

$$A = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}. \tag{1.9}$$

Thus ρ, the state of the system, is defined by the column vector $(1,0)^t$, where $(1,0)^t$ denotes the transpose of $(1,0)$, and the expectation of A relative to ρ is 0. We will see below that this can be interpreted as a measurement of the expected y-spin of a particle in a z-spin up state.

Much of our discussion applies in a general Hilbert space, but then one has to develop the apparatus of infinite-dimensional vector spaces and general functional analysis. Since quantum computing algorithms are modeled in a finite-dimensional setting, we simplify matters by staying in the finite-dimensional context. In addition, we resist the temptation to present the consequences of our hypotheses as a generalization of a probability space. (See Parthasarathy [55].)

The physics underlying the next assumption is that after the measurement of an observable, the physical system will be in the state associated with the observed eigenvalue. For example, if a Stern–Gerlach experiment shows an electron has a particular spin orientation, then immediately after that measurement the electron will have that particular orientation with probability one. This is a fundamental assumption of quantum mechanics.

(1.10) Assumption. An observable A is "measured" by detecting one of its eigenvalues. If the physical system is not destroyed by the measurement, then it is subsequently modeled by a QPS (H, ρ'), where ρ' assigns probability one to the associated eigenspace of A. •

Again referring to the continuation of Example (1.2), A has eigenvalues 1 and -1, and the spectral theorem for Hermitian matrices guarantees A can be written in terms of its eigenvectors and eigenvalues as

$$A = 1\left(\frac{1}{\sqrt{2}}\begin{bmatrix} i \\ -1 \end{bmatrix} \cdot \frac{1}{\sqrt{2}}[-i, -1]\right) + (-1)\left(\frac{1}{\sqrt{2}}\begin{bmatrix} i \\ 1 \end{bmatrix} \cdot \frac{1}{\sqrt{2}}[-i, 1]\right)$$

$$= 1 \cdot \frac{1}{2}\begin{pmatrix} 1 & -i \\ i & 1 \end{pmatrix} + (-1) \cdot \frac{1}{2}\begin{pmatrix} 1 & i \\ -i & 1 \end{pmatrix}.$$

Suppose that a physical measurement gives $+1$ as the measurement. Then $\frac{1}{2}\begin{pmatrix} 1 & -i \\ i & 1 \end{pmatrix}$ is the probability state ρ' subsequent to that measurement, and it is easy to check that ρ' is positive semidefinite and has trace equal to one.

It is important to emphasize the sequence of events in an observation. Before the observation, the physical system is modeled by (H, ρ_0), and the probability that an observable A will have (eigen) value a is $\mathrm{tr}(\rho_0 P_a)$, where P_a is the projection on the a-eigenspace of A. Subsequent to the measurement, the physical system is modeled by (H, ρ_1), and $\mathrm{tr}(\rho_1 P_a)$ equals one. One way to describe this is to say that the system "collapses" into the measured state. An alternative description is that the probability

density subsequent to the measurement is defined by the results of the measurement.

The fifth constraint leads to an aspect of the basic model that is crucial for quantum algorithms, and that is the use of unitary mappings to model actions which modify the system without affecting the inner product of two vectors of H. It is the clever combination of unitary mappings and selected measurements that constitute the essence of quantum algorithms. To avoid too much notation at one time, however, we defer the discussion of unitary maps to the end of the chapter and instead give an example of a static model.

1.4 The basic example: spin 1/2 particles

We illustrate these definitions by modeling a spin 1/2 particle as described earlier. (Example (1.2) is actually part of this example.) Although it is unlikely a quantum computer would use spin information to represent zero and one, spin 1/2 particles present a clean and nonclassical paradigm for a two-state quantum system.

The Stern–Gerlach experiment shows the z-spin orientation of an electron is either up or down, and we take H to be C^2, a two-dimensional vector space over the complex numbers. Following the notation of Section 1.2, let $(z,+)$ denote the (vector) state of the particle when the z-spin is up and $(z,-)$ the (vector) state of the particle when the z-spin is down. The experimental evidence that the z-spin is either up or down is modeled by saying $(z,+)$ and $(z,-)$ are orthonormal, and thus the two vectors define a basis for H. The experimental measurement is modeled by defining the vectors $(z,\pm)$ as eigenvectors for a z-spin measurement which is given by S_z:

$$S_z(z,\pm) = \pm\hbar/2(z,\pm)$$

where as usual $\hbar = h/2\pi$, and h denotes Planck's constant.

Particles can be measured for spin effects in any orientation, and we can thus talk about an x-spin and a y-spin in our model. Based on the experiments described earlier, it follows that regardless of the z-spin orientation, the particle is equally likely to have its x-spin in either x-direction, and the corresponding $(x,\pm)$ states can be modeled in the same space in terms of the $(z,\pm)$ vectors. In fact, one can derive

$$\textbf{(1.11)} \quad (x,+) = ((z,+) + (z,-))/\sqrt{2} \quad (x,-) = ((z,+) - (z,-))/\sqrt{2}.$$

In a similar fashion, the y-spin states can also be represented as a linear combination of the z-spin vectors and in proper relation to the x-spin

vectors, provided one allows complex coefficients. Although a particle's intrinsic spin is not a classical concept, its mathematical representation can be derived from an analysis of the angular momentum of a particle. (See [60] or [63] for details.) Following standard conventions this gives

(1.12) $(y,+) = ((z,+) + i(z,-))/\sqrt{2} \quad (y,-) = ((z,+) - i(z,-))/\sqrt{2}.$

In a coordinate representation for which S_z is diagonal, so that $(z,+) = \begin{bmatrix}1\\0\end{bmatrix}$ and $(z,-) = \begin{bmatrix}0\\1\end{bmatrix}$, one has

(1.13) $$(x,\pm) = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\ \pm 1\end{bmatrix} \qquad (y,\pm) = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\ \pm i\end{bmatrix}.$$

(1.14) Example. Suppose we have measured the z-spin, and the particle has z-spin up. Then the model representing the state of the particle spin has $\rho_0 = \begin{pmatrix}1 & 0\\ 0 & 0\end{pmatrix}$, and we compute, for example, the probability that the y-spin is $(y,-)$ as $\mathrm{tr}(\rho_0 P_d)$, where P_d denotes the projection matrix on the one-dimensional space spanned by $(y,-)$. Since $P_d = \frac{1}{2}\begin{pmatrix}1 & i\\ -i & 1\end{pmatrix}$ in this case, we obtain $1/2$ as expected, and analogous results would be obtained for the other spin orientations. $A = 1P_u + (-1)P_d$ is the spectral representation of A given below (1.10), where P_u is the projection on $(y,+)$, and as noted in (1.6) we can confirm the consistency of the definition of the expectation:

$$\mathrm{tr}(\rho_0 A) = (1)\mathrm{tr}(\rho_0 P_u) + (-1)\mathrm{tr}(\rho_0 P_d) = 0. \qquad \bullet$$

(1.15) Example. Continuing according to our prescription, suppose that we now know the particle is in the $(y,-)$ state. Using the same basis, it follows that the appropriate quantum probability space is now (H, ρ_1) with

$$\rho_1 = \frac{1}{2}\begin{pmatrix}1 & i\\ -i & 1\end{pmatrix},$$

which is just the projection matrix P_d computed in the preceding example. If we want to measure the (new) probability of the particle having z-spin down, for example, we find T_d, the projection on $(z,-)$, and compute $\mathrm{tr}(\rho_1 T_d) = 1/2$, consistent with experiment. $\bullet$

In terms of the z-spin basis, the "z-spin operator" S_z has the matrix representation $\hbar/2\begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}$. The x-spin operator S_x maps $(x,\pm)$ to $\pm(\hbar/2)(x,\pm)$, and using the z-spin basis, it is easy to check that S_x has the representation $(\hbar/2)\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}$, and similarly S_y has the representation $(\hbar/2)\begin{pmatrix}0 & -i\\ i & 0\end{pmatrix}$. The matrices

$$\sigma_x = \begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix} \qquad \sigma_y = \begin{pmatrix}0 & -i\\ i & 0\end{pmatrix} \qquad \sigma_z = \begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}$$

are known as *Pauli spin matrices.* If we let σ_0 denote the identity matrix, it is easy to check that these four 2×2 matrices define a basis for the linear space of 2×2 matrices over the complex numbers and a basis for 2×2 Hermitian matrices over the real numbers.

(1.16) Exercise. Verify the preceding assertions and confirm that the Pauli spin matrices are Hermitian and have eigenvalues ± 1. •

The foregoing illustrates the way a QPS is defined and "evolves" as measurements are made. (Note that this "evolution" is based upon measurements and not on the dynamics of the system.) This example also illustrates the need for a tractable notation which facilitates both numerical and symbolic calculations.

1.5 Dirac notation

In developing quantum algorithms we will be working with higher dimensional spaces, and the notation used so far becomes cumbersome. In order to keep track of the physical system being modeled and to simplify the theoretical manipulations of the theory, Dirac introduced notation which has become standard in the field.

To emphasize the relationship of vectors with physical states, vectors in H are associated with physical states using the bra-ket notation. The "ket" $|\psi\rangle$ stands for a vector in H associated with the physical state ψ and would be represented by a column vector in the finite-dimensional case. The "bra" $\langle\psi|$ denotes the adjoint of $|\psi\rangle$, and in the finite-dimensional case would be represented as a row vector whose entries are complex conjugates of the entries of $|\psi\rangle$: $\langle\psi| = |\psi\rangle^\dagger$. Thus we can say that the ket $|z,+\rangle$ denotes the particle with state $(z,+)$ and the ket $|z,-\rangle$ denotes the particle in state $(z,-)$. Similarly, we use the kets $|x,\pm\rangle$ and $|y,\pm\rangle$ for the corresponding x and y spin states.

The *inner product* between a bra and a ket — between a row vector and a column vector — is denoted by the bracket notation $\langle\varphi|\psi\rangle$, (now you "c" it), and corresponds to the inner product of Definition (1.1).

(1.17) Example. To illustrate the notation, let $\{|u_k\rangle, 1 \le k \le n\}$ be an orthonormal basis for H. Then it is easy to check that the *outer product* $|u_k\rangle\langle u_k|$ is the projection operator onto the vector space spanned by $|u_k\rangle$ and that the identity operator can be written $I = \sum |u_k\rangle\langle u_k|$. Given those facts, one easily obtains the representation of a ket relative to that basis:

$$|\psi\rangle = \sum_k (|u_k\rangle\langle u_k|)|\psi\rangle = \sum_k |u_k\rangle\langle u_k|\psi\rangle.$$

(We used the outer product below (1.10) to derive the spectral representation of A.) •

(1.18) Example. Suppose a spin 1/2 system is known to have z-spin up, so that the associated ρ_0 is $|z,+\rangle\langle z,+|$. If $|z,+\rangle$ and $|z,-\rangle$ are chosen as basis elements, then the coordinate representation of ρ_0 is the matrix $\rho_0 = \begin{bmatrix}1\\0\end{bmatrix} \cdot [1,0] = \begin{pmatrix}1 & 0\\ 0 & 0\end{pmatrix}$ as illustrated in (1.8). Analogously, if the system is known to be in state $|y,-\rangle$, then the corresponding probability state is $\rho_d = |y,-\rangle\langle y,-|$, which can be represented in the z-spin basis by the matrix $\left(\frac{1}{\sqrt{2}}\begin{bmatrix}1\\-i\end{bmatrix} \cdot [1,i]\frac{1}{\sqrt{2}}\right) = \frac{1}{2}\begin{pmatrix}1 & i\\ -i & 1\end{pmatrix}$, confirming (or deriving) the expression used in Example (1.15). •

(1.19) Example. Using the z-spin vectors to define the coordinate system in a spin 1/2 system, consider the basis defined by the y-spin vectors. Then one can represent the identity matrix I as $\frac{1}{2}\begin{pmatrix}1 & -i\\ i & 1\end{pmatrix} + \frac{1}{2}\begin{pmatrix}1 & i\\ -i & 1\end{pmatrix}$, where the first term is the projection on $|y,+\rangle$ and the second term is the projection on $|y,-\rangle$. If we take $|\psi\rangle$ to be $|z,+\rangle$, then it is easy to check that

$$|z,+\rangle = |y,+\rangle \cdot \langle y,+|z,+\rangle + |y,-\rangle \cdot \langle y,-|z,+\rangle = \frac{1}{\sqrt{2}}(|y,+\rangle + |y,-\rangle) \quad \bullet$$

If we know that a system is in a particular state $|\psi\rangle$, then the density ρ must reflect that fact: a measurement of that aspect of the system should show it to be in $|\psi\rangle$ with probability one. To achieve that feature, a density ρ is routinely associated with a physical state ψ using the associated bra and ket via the outer product, so that $\rho = \frac{1}{\langle\psi|\psi\rangle}|\psi\rangle\langle\psi|$, and

thus ρ has trace one. It is for this reason that physicists associate densities ρ with kets with norm one. By the same token, if c is a complex number with norm one, both $|\psi\rangle$ and $c|\psi\rangle$ define the same ρ. (As far as expressing the state of a system via ρ, some authors identify the physical state ψ with the entire "ray" $c|\psi\rangle$, where $c \neq 0$.) We record these observations in Lemma 1.21 below.

Recall that an observable is associated with a Hermitian operator A. Then since A equals $A^\dagger$, $(A|\psi\rangle)^\dagger = \langle\psi|A^\dagger = \langle\psi|A$, and $\langle\varphi|A|\psi\rangle = \overline{\langle\psi|A|\varphi\rangle}$. Cognoscenti of Dirac notation interpret this last equation as follows. Suppose an orthonormal basis for H contains the orthonormal states $|\psi\rangle$ and $|\varphi\rangle$. Then the operator A is represented by a matrix whose $|\varphi\rangle, |\psi\rangle$'th entry is the complex conjugate of its $|\psi\rangle, |\varphi\rangle$'th entry. We also have the representation of a Hermitian operator in Dirac notation, i.e., the translation of the spectral representation of (1.5) above:

$$A = \sum \lambda_k |\psi_k\rangle\langle\psi_k|, \tag{1.20}$$

where the λ_k's are the (real) eigenvalues of the orthonormal eigenvectors $|\psi_k\rangle$ of A.

(1.21) Lemma. Let A be a Hermitian matrix, and let $|\psi\rangle$ be a ket. Let $\langle\psi|A|\psi\rangle$ denote the inner product of $\langle\psi|$ and $A|\psi\rangle$. Then $\rho = |\psi\rangle\langle\psi|$ is positive semidefinite and has trace one if and only if $|\psi\rangle$ has norm one. In addition, $\mathrm{tr}(\rho A) = \langle\psi|A|\psi\rangle$. Thus, if the system is in state $|\psi\rangle$, the *expectation* of an observable A is $\langle\psi|A|\psi\rangle$, and the *probability* of measuring the system in a subspace defined by a projection matrix P_0 is $\langle\psi|P_0|\psi\rangle$.

Proof: If $|u\rangle$ is any vector in H, $\langle u|(|\psi\rangle\langle\psi|)|u\rangle = |\langle\psi|u\rangle|^2$, so that ρ is positive semidefinite. Since the trace is independent of the choice of basis, choose $\{u_k, 1 \leq k \leq n\}$ to be an orthonormal basis for H with $u_1 = |\psi\rangle$. Taking $u = u_1$ in the foregoing equation, it is easy to see from the definition of the trace that ρ has trace one if and only if the ket $|\psi\rangle$ has norm one in H. For the last assertion, use the properties of the trace and the fact that ρ is a projection:

$$\begin{aligned} tr(\rho A) &= tr(|\psi\rangle\langle\psi|A) = tr((|\psi\rangle\langle\psi|)A(|\psi\rangle\langle\psi|)) \\ &= \langle\psi|A|\psi\rangle tr(|\psi\rangle\langle\psi|) = \langle\psi|A|\psi\rangle. \end{aligned}$$ •

The state ρ of (1.21) is called a *pure* state, since it arises from one particular state $|\psi\rangle$ of the system. A density ρ is a *mixed* state if it is defined as a convex combination of pure states as in the following example.

(1.22) Example. Let $\{p_j,\ 1 \leq j \leq n\}$ be a set of nonnegative real numbers whose sum is one. Let $\{|\varphi_j\rangle, 1 \leq j \leq n\}$ denote an orthonormal set of kets. Then

$$\rho = \sum p_j |\varphi_j\rangle\langle\varphi_j|$$

is positive semidefinite with trace one, and for Hermitian A,

$$\mathrm{tr}(\rho A) = \sum p_j \langle\varphi_j|A|\varphi_j\rangle. \quad \bullet$$

(1.23) Example. Define the rotation $R(t) = \begin{pmatrix} \cos(t) & -\sin(t) \\ \sin(t) & \cos(t) \end{pmatrix}$ relative to the z-spin basis of a spin 1/2 space. We know that $|y,+\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ i \end{bmatrix}$ and $|y,-\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -i \end{bmatrix}$ are orthonormal basis vectors. Then $R(t)|y,+\rangle$ and $R(t)|y,-\rangle$ are also orthonormal vectors. More generally, we have in Dirac notation:

$$\langle u|v\rangle = \langle u|R(t)^\dagger R(t)|v\rangle$$

for any two vectors in the space, which is another way of saying that $R(t)^\dagger = R(t)^{-1}$, the defining property of a *unitary* matrix. $\bullet$

1.6 Unitary transformations

Example (1.23) illustrates the fifth aspect of the basic (static) model. Suppose we have a given finite-dimensional space H over the complex numbers and we map H into itself with a unitary transformation U, so that $U^{-1} = U^\dagger$. Then the image space is just H itself, since dimensionality and orthogonality are preserved by a unitary transformation. However, if a vector u represents a specific physical state of the quantum system, then Uu may be a linear combination of vectors representing different physical states of the system. Alternatively, we can consider U as a change of basis mapping, so that the description of a system is changed from one perspective to a different perspective. For example, instead of using the z-spin orientation as the computational basis, we could change to the x-spin basis instead.

(1.24) Exercise. Prove that U is a unitary matrix on $H = C^2$ if and only if $U = e^{i\phi} \begin{pmatrix} \alpha & \beta \\ -\overline{\beta} & \overline{\alpha} \end{pmatrix}$, where $|\alpha|^2 + |\beta|^2 = 1$. $\bullet$

(1.25) Exercise. Suppose $\{u_1, \ldots, u_k\}$ is a set of orthonormal vectors in a complex linear space H. If U is an $n \times n$ unitary matrix, confirm that $\{Uu_1, \ldots, Uu_k\}$ is also a set of orthonormal vectors and that U takes an orthonormal basis into an orthonormal basis. •

For purposes of quantum algorithms, unitary transformations model the manipulation of information or data "without looking," while projections model the measurement of the values of the information or the data — what we see when we look at the system. These two aspects of a quantum system thus codify the distinction between manipulating a system containing information and observing a system to measure the information it contains.

(1.26) Example. Suppose we have a QPS in the (normalized) state $|\psi\rangle$ so that the density operator is $\rho = |\psi\rangle\langle\psi|$. Let U be a unitary transformation and let $|\varphi\rangle = U|\psi\rangle$. Then the new probability operator is $\rho_1 = U|\psi\rangle\langle\psi|U^\dagger$. If A is a Hermitian operator, then

$$tr(\rho_1 A) = \langle\varphi|A|\varphi\rangle = \langle\psi|U^\dagger AU|\psi\rangle = tr(\rho A_1),$$

where $A_1 = U^\dagger AU$. •

Thus, we could think of the state having changed and measure the observable A using the new state ρ_1, as on the left of the equation above, or we could think of the state having stayed the same and use that to measure the new observable A_1, as on the right of the equation above. Either way, we get the same result.

We continue to defer a discussion of Schrödinger's equation, but it is appropriate to note here that a solution of that equation involves a group of unitary matrices $\{U(t)\}$ related to the time-evolving state of the quantum system by $|\psi(t)\rangle = U(t)|\psi(0)\rangle$. Using these time-dependent operators in the preceding example, we could imagine the state evolving in time as $\rho(t) = U(t)\rho U^\dagger(t)$, which is the *Schrödinger* picture, or the observable evolving in time as $A(t) = U^\dagger(t)AU(t)$, which is the *Heisenberg* picture.

Now by using matrices to model measurements and transformations, we have introduced the feature of *noncommutativity* and that plays an important role in the theory.

(1.27) Definition. The operator $[A, B] \equiv AB - BA$ is the *commutator* of two operators and is a measure of their noncommutativity. •

Recall the Pauli spin matrices

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \qquad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix},$$

which were introduced above (1.16). It is easy to check that these matrices are both Hermitian and unitary and that $\sigma_x\sigma_y = -\sigma_y\sigma_x = i\sigma_z$, so that different spin matrices do not commute. In general,

$$[\sigma_r, \sigma_s] = \sigma_r\sigma_s - \sigma_s\sigma_r = 2i\epsilon_{rst}\sigma_t, \tag{1.28}$$

where ϵ_{rst} equals 1 if rst is a positive (i.e., even) permutation of xyz, equals -1 if rst is a negative (i.e., odd) permutation of xyz and equals 0 otherwise.

(1.29) Exercise. If $\vec{n}\cdot\vec{\sigma} \equiv n_x\sigma_x + n_y\sigma_y + n_z\sigma_z$, show that

$$(\vec{m}\cdot\vec{\sigma})(\vec{n}\cdot\vec{\sigma}) = \vec{m}\cdot\vec{n}\sigma_0 + i(\vec{m}\times\vec{n})\cdot\vec{\sigma}$$

where σ_0 is the identity matrix and $\vec{m}\times\vec{n}$ is the usual cross product of vectors. •

(1.30) Exercise. Show that any A in $SU(2)$, the set of 2×2 unitary matrices with determinant 1, can be written as

$$a_0\sigma_0 + \vec{a}\cdot i\vec{\sigma},$$

where the a_k are real and $a_0^2 + a_x^2 + a_y^2 + a_z^2 = 1$. •

(1.31) Exercise. Suppose A and B are Hermitian and $|\psi\rangle$ is a given state. Show that the commutator $[A, B]$ is anti-Hermitian, $[A, B]^\dagger = -[A, B]$, and conclude that $\langle\psi|[A, B]|\psi\rangle$ is a purely imaginary number. Let $\{A, B\} \equiv AB + BA$ denote the *anticommutator* and confirm that $\langle\psi|\{A, B\}|\psi\rangle$ is a real number. •

We have enough machinery in place to confirm that the required uncertainty relations follow as a consequence of the definitions. (See, for example, [60, 63]). Specifically, suppose a system is in state $|\psi\rangle$ and that A and B denote two Hermitian matrices. If $\Delta A \equiv A - \langle\psi|A|\psi\rangle I$ and $\Delta B \equiv B - \langle\psi|B|\psi\rangle I$, where I is the identity operator, then it is easy to verify that $\langle\psi|\Delta A|\psi\rangle = 0$ and similarly for ΔB. Thus,

$$\langle\psi|(\Delta A)^2|\psi\rangle\langle\psi|(\Delta B)^2|\psi\rangle \geq |\langle\psi|(\Delta A\Delta B)|\psi\rangle|^2$$

follows from the Cauchy–Schwarz inequality, and from elementary algebra we have

$$2\langle\psi|(\Delta A\Delta B)|\psi\rangle = \langle\psi|[A, B]|\psi\rangle + \langle\psi|\{\Delta A, \Delta B\}|\psi\rangle.$$

Using (1.31) we obtain

$$\langle\psi|(\Delta A)^2|\psi\rangle\langle\psi|(\Delta B)^2|\psi\rangle \geq \frac{1}{4}(|\langle\psi|[A, B]|\psi\rangle|^2 + |\langle\psi|\{\Delta A, \Delta B\}|\psi\rangle|^2)$$

and thus a generalization of the *Heisenberg uncertainty principle*:

$$\langle\psi|(\Delta A)^2|\psi\rangle\langle\psi|(\Delta B)^2|\psi\rangle \geq \frac{1}{4}(|\langle\psi|[A,B]|\psi\rangle|^2). \tag{1.32}$$

We have thus shown that the mathematical model developed so far includes the Heisenberg uncertainty principle as a consequence. The meaning of the inequality is that if two operators do not commute, then there is an intrinsic randomness in the outcomes of experimental measurements, a randomness which is independent of the accuracy of the measuring apparatus.

2
Basics of Quantum Computation

In this chapter we build on the basic model developed in Chapter 1 by extending the notation to handle quantum systems with multiple qubits. With that terminology in place, we can illustrate the ideas which are basic to quantum algorithms and can confirm theoretically that the unitary transformations we need can be implemented as a sequence of operations involving only one or two qubits. Quantum algorithms can be constructed using a small number of quantum gates, and we discuss those gates next. We then use quantum gates to construct an addition subroutine and complete the chapter with a teleportation subroutine, which is the first illustration of the potential importance of entanglement for communication.

2.1 Qubits and tensor products

A key component of a putative quantum computer is a quantum system with two states which can represent 0 and 1. We have used the spin state of an electron as a paradigm, and we let the z-spin up state $|z,+\rangle$ denote 0 and the z-spin down state $|z,-\rangle$ denote 1. In the basic model these states are represented by orthonormal vectors $|0\rangle$ and $|1\rangle$, respectively, and the vectors are referred to as the *computational basis*. Following Schumacher [61], we will call the mathematical model of such a physical system a *qubit*.

As of this writing, it is not clear what quantum system could be used physically to realize a practical qubit, and there is active research on the

subject. One possibility is the use of trapped ions: a finite number of similarly charged particles captured in a specially designed field which allows limited motion only along a line. Two selected energy states of an ion represent 0 and 1, and it is physically possible to operate on such an ion to achieve superposition. While it is also physically possible to trap several ions at the same time, it appears to be quite difficult to manipulate them in such a way as to implement the rudimentary logic gates we will discuss below. (See [21].) A detailed discussion of trapped ions and other possible quantum systems defining qubits would take us too far afield at this point, and we'll continue to concentrate on the theoretical context, keeping in mind that the choice of the physical system may significantly affect the implementation of algorithms. (For more on possible implementations see, for example, [37], [51] and [73].)

In order to implement any useful quantum algorithm we need to deal with many qubits in one QPS, and the appropriate model is a *tensor product* of qubits. Specifically, if we have n qubits, each with a given computational basis in a two-dimensional QPS H_i, then the tensor product is a 2^n-dimensional space with a computational basis consisting of 2^n vectors

$$|i_1 i_2 \dots i_n\rangle = |i_1\rangle \otimes |i_2\rangle \otimes \cdots \otimes |i_n\rangle,$$

where the indices i_k take on the values 0 and 1. That notation indicates that there are two ways of viewing vectors defined as tensor products. One way is as vector number $i_1 \dots i_n$ and the other way is in terms of the tensor product notation. The first representation emphasizes the fact that the n-bit index can stand for the binary representation of an integer k, $0 \leq k \leq 2^n - 1$, while the second expression emphasizes the representation as qubits.

(2.1) Example. Let $n = 2$ so that $H = H_1 \otimes H_2$ is a four-dimensional space with a basis $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$. If the encoding of information is in terms of energy levels of trapped ions, for example, then the first (high-order) bit denotes the energy level of the first ion, and the second (low-order) bit denotes the energy level of the second ion. If the encoding is in terms of z-spin, then the first (high-order) bit denotes the z-spin of the first particle, and the second (low-order) bit denotes the z-spin of the second particle. •

Once we have n qubits and the tensor product space representing them, we will want to apply unitary and Hermitian mappings. The most basic mapping is one which affects only one qubit, say the first qubit, and leaves the other qubits unchanged. If A_1 is the mapping applicable to

H_1, then the mapping $A_1 \otimes I_2 \otimes \cdots \otimes I_n$ denotes the extension to the tensor product space. In general,

$$(A_1 \otimes \cdots \otimes A_n)(|i_1\rangle \otimes \cdots \otimes |i_n\rangle) = A_1|i_1\rangle \otimes \cdots \otimes A_n|i_n\rangle$$

denotes the action of a tensor product of linear maps on a basis vector of the tensor product space. The extension of the operator to arbitrary vectors in H is defined by linearity in the obvious way.

(2.2) Example. Let H denote the space of Example (2.1) and suppose we want to operate on H by $A \otimes B$. In terms of the matrix representations in the computational basis, let

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \qquad B = \begin{pmatrix} r & s \\ t & u \end{pmatrix}$$

so that $A \otimes B$ is a 4×4 matrix which operates on $|\psi\rangle = \sum\sum a_{jk}|j\rangle|k\rangle$ in the coordinate representation by

$$A \otimes B|\psi\rangle = \begin{pmatrix} a\begin{pmatrix} r & s \\ t & u \end{pmatrix} & b\begin{pmatrix} r & s \\ t & u \end{pmatrix} \\ c\begin{pmatrix} r & s \\ t & u \end{pmatrix} & d\begin{pmatrix} r & s \\ t & u \end{pmatrix} \end{pmatrix} \begin{pmatrix} a_{00} \\ a_{01} \\ a_{10} \\ a_{11} \end{pmatrix},$$

where we have written the matrix in terms of 2×2 blocks. Note that

$$\mathrm{tr}(A \otimes B) = (ar + au + dr + du) = \mathrm{tr}(A)\mathrm{tr}(B). \qquad \bullet$$

We should make a cautionary comment here about the definition of the matrix representation of the tensor product. In order for the left bit to denote the high order bit we need to define the matrix representation as above. However, other definitions may reverse that ordering and the matrix representation of the tensor product would differ from the matrix above by a permutation of the rows and columns.

(2.3) Example. Suppose the operators A_i are each unitary (Hermitian) in H_i. Then the tensor product A is unitary (Hermitian) on the tensor product space H. We leave it to the reader to check the details, using the fact that

$$(A_1 \otimes \cdots \otimes A_n)(B_1 \otimes \cdots \otimes B_n) \equiv (A_1B_1 \otimes \cdots \otimes A_nB_n). \qquad \bullet$$

(2.4) Example. If we wish to perform a measurement on H, then as usual we do so with respect to a positive semidefinite matrix ρ with trace

one. As we have seen above, such a matrix could be defined using a state $|\psi\rangle$ in H and the outer product: $\rho = |\psi\rangle\langle\psi|$. Suppose, for example, that we chose $|\psi\rangle = |\psi_1\rangle \otimes \cdots \otimes |\psi_n\rangle$, where each of the components of the tensor product is a normalized state in the corresponding component space. Let A denote a tensor product of Hermitian matrices A_k. Then

$$\begin{aligned}\langle\psi|A|\psi\rangle &= \mathrm{tr}(|\psi\rangle\langle\psi|A) = \mathrm{tr}(|\psi_1\rangle\langle\psi_1|A_1 \otimes \cdots \otimes |\psi_n\rangle\langle\psi_n|A_n) \\ &= \prod_k \mathrm{tr}(|\psi_k\rangle\langle\psi_k|A_k) = \prod_k \langle\psi_k|A_k|\psi_k\rangle\end{aligned}$$

as we would expect in this simple case. •

2.2 The basic strategy of quantum algorithms

We can now summarize the fundamental differences between a classical computer and a quantum computer. In a classical computer, n bits can represent one integer k, where $0 \leq k \leq 2^n - 1$. However, because of superposition, it is theoretically possible for n qubits to represent simultaneously all 2^n integers in that range.

In a classical computer we can examine the n bits and read off the integer value without changing the values of the bits. However, if we attempt to read off the contents of n qubits which are in superposition, we are performing a measurement, and the theory tells us the outcome is not predictable. Moreover, once we read the contents of the qubits we lose the superposition.

This distinction between the presumed simultaneous "existence" of superposed states (an *ontological* issue) and the actual measurement of exactly one of those states (an *epistemological* issue) is at the heart of philosophical interpretations of quantum mechanics and also at the heart of quantum computing. We act as if all 2^n numbers are stored in the qubits, even though we can only know what we actually measure. We will never be able to see (i.e., measure) more than one n-bit integer at a time.

If qubit k is in the state $|\psi_k\rangle = \alpha_{k_0}|0\rangle + \alpha_{k_1}|1\rangle$ with $|\alpha_{k_0}|^2 + |\alpha_{k_1}|^2 = 1$, then relative to the tensor product state, the probability of observing the number represented by the basis vector $|i_1 i_2 \ldots i_n\rangle$ is $\prod_k \langle\psi_k|P_{i_k}^{(k)}|\psi_k\rangle = \prod_k |\alpha_{k_{i_k}}|^2$, where $P_{i_k}^{(k)}$ denotes the projection of the k'th qubit onto $|i_k\rangle$, and that will equal 2^{-n} if each component computational basis vector is equally likely. Thus, while n qubits might enable us to simultaneously represent all 2^n numbers, the probability of each one being measured is correspondingly small.

The strategy then is to take advantage of superposition, which enables us to calculate the value of a function at all 2^n integers simultaneously, while avoiding premature measurements which destroy the superposition and which don't necessarily guarantee our reading out a useful result with any significant probability.

How do we enhance the probability of getting something useful? The answer is that we can still apply unitary transformations to the qubits and try to change the initial state vector to a state vector which will significantly increase the probability of measuring something useful. In a sense, unitary transformations enable us to manipulate all of the superposed states simultaneously "without looking" until we finally arrive at a state for which a measurement is appropriate.

(2.5) Example. Suppose we have two qubits $|\psi_1\rangle$ and $|\psi_2\rangle$, each of which is in the superposed state $(|0\rangle + |1\rangle)/\sqrt{2}$ in the coordinate basis of its respective space H_i. Assume the state of the tensor product space is $|\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle$ and that we can operate on $|\psi\rangle$ by a unitary matrix U as follows:

$$\begin{aligned} |\varphi\rangle &= U|\psi\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & -1 & -1 \\ 0 & 0 & 1 & -1 \end{pmatrix} \begin{pmatrix} 1/2 \\ 1/2 \\ 1/2 \\ 1/2 \end{pmatrix} \\ &= \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ 1 \\ -1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle). \end{aligned}$$

If we now compute the probability of being in state $|10\rangle$, we find that instead of $1/4$, the probability of being in $|10\rangle$ before applying U,

$$\mathrm{tr}(|\varphi\rangle\langle\varphi|P_{10}) = \langle\varphi|P_{10}|\varphi\rangle = 1/2. \qquad \bullet$$

In Example (2.5) we used Dirac notation, tensor notation and standard matrix notation where the rows and columns were labeled in lexicographical order: 00, 01, 10, 11. The matrix U is unitary but is not a tensor product of operators on the subspaces, and the resulting state is not a tensor product of states. Such a mixture is referred to as an *entangled* state and is a crucial ingredient in our algorithms. Entangled states are central to EPR experiments and have the peculiar property that although the particles defining the two qubits may be widely separated in space and can be separately measured, they constitute one quantum state and do not possess "separate" or "local" properties (a two-particle

system rather than a system of two particles). This is the point that gave Einstein pause, and which we mentioned in Chapter 1.

If we wish to entangle two qubits, we will need to implement unitary transformations which involve those qubits and which are not tensor products of operators on the separate subspaces. In fact, quantum algorithms require that we entangle a large number of qubits, and from the brief discussion of trapped ions the reader can surmise that it would simplify the engineering if such a mapping could be implemented by a sequence of unitary operations involving no more than two qubits at a time. This is indeed mathematically possible, and we give the proof as presented in [33].

(2.6) Proposition (Deutsch 1985). Let U be a unitary matrix on a d-dimensional space. Then the action of U can be written as a product of $2d^2 - d$ unitary matrices, each of which acts within a two-dimensional subspace defined by a fixed set of d basis vectors. Thus any unitary transformation on n qubits can be realized by a sequence of unitary operations, each of which affects no more than two qubits.

Proof: Let $u_1, \ldots, u_d$ denote d orthonormal eigenvectors of U with eigenvalues $\lambda_1, \ldots, \lambda_n$. If u_1 equals $(x_1, \ldots, x_d)$ in a given basis, then using the obvious notation, the block diagonal matrix

$$A_{12} = \left(\frac{1}{\sqrt{|x_1|^2 + |x_2|^2}} \begin{pmatrix} \bar{x}_1 & \bar{x}_2 \\ -x_2 & x_1 \end{pmatrix}, \; I_{3,\ldots,n} \right)$$

effectively operates as a 2×2 matrix on the first two coordinates and maps u_1 to $(t_2, 0, x_3, \ldots, x_d)$, where $t_2 = \sqrt{|x_1|^2 + |x_2|^2}$. A total of $d - 1$ such maps A_{1k} produces $(1, 0, \ldots, 0)$. Multiplying $(1, 0, \ldots, 0)$ by the eigenvalue $\lambda_1 = e^{i\theta_1}$, a one-dimensional operation, and applying the inverses of the A_{1k}'s in reverse order has the effect of multiplying the first eigenvector u_1 by the first eigenvalue λ_1 in $2d - 1$ one- and two-dimensional unitary operations. By the orthogonality of the eigenvectors, the other eigenvectors are mapped to and then from a vector with a zero in the first component and are thus unaffected by the procedure. Repeating that process for each of the d eigenvectors, it follows that U can be modeled as a product of $d(2d - 1)$ one- and two-dimensional unitary matrices, completing the proof. •

There are two caveats to this analysis. We wish to construct algorithms that are polynomial in n, the number of qubits, but the dimension d of the underlying Hilbert space is 2^n. Proposition (2.6) guarantees the representation of a unitary map as a product of $poly(2^n)$ one- and two-dimensional maps, which is not good enough. What we need to show is

that a unitary map in a quantum algorithm is *feasible*; that is, it can be realized by a sequence of $poly(n)$ one- and two-qubit unitary mappings.

The second caveat is that most likely we cannot assume that every one- or two-qubit unitary map is physically implementable. Instead, we would have to restrict ourselves to maps that can be well approximated by sequences of maps from a (presumably) finite number of physically implementable one- and two-qubit operations. This problem was also studied by Deutsch (cf. [33]), who showed that one could define a finite collection of eight transformations that generate under composition a group dense in the set of two-dimensional unitary transformations. (Ekert and Jozsa [33] remark that the work by Bernstein and Vazirani [13] implies that a matrix implementing a rotation by a fixed irrational angle and the map $\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$ suffice to approximate an arbitrary 2×2 unitary mapping.) We revisit this issue below.

2.3 Quantum gates

A *quantum gate* is the analogue of a logic gate in a classical computer, and quantum gates are the basic units of quantum algorithms. The difference between the classical and quantum context is that a quantum gate has to be implemented unitarily and, in particular, must be a reversible operation. For example, the classical OR statement on two bits is not invertible since the four possible inputs $\{00, 01, 10, 11\}$ map onto two possible outputs $\{0, 1\}$. A quantum analogue would require four possible outputs. (See XOR below.)

The simplest examples of quantum gates are operations on one qubit. For example, in Dirac notation the NOT operation is $X|k\rangle = |k \oplus 1\rangle$, where the addition is mod(2), and a phase operation on one qubit is $A|k\rangle = e^{(-1)^k i\varphi}|k\rangle$. In matrix notation

$$X = \sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad A = \begin{pmatrix} e^{i\varphi} & 0 \\ 0 & e^{-i\varphi} \end{pmatrix}.$$

Once again we emphasize that we are not addressing the issue of physical feasibility but instead are assuming that the indicated operations could be implemented, provided they meet the theoretical requirements of unitarity.

(2.7) Example. Suppose a qubit is initially in one of the computational basis states, say, $|0\rangle$ in Dirac notation or $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ in the computational

basis notation. If

$$R = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix},$$

a unitary matrix, then the new state $|\psi\rangle = R|0\rangle$ is a superposition of the computational basis states, and each basis state now has probability $1/2$ of being observed:

$$p_i = \langle\psi|P_i|\psi\rangle = 1/2,$$

where P_i denotes the projection on $|i\rangle$. In particular,

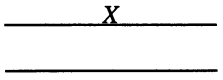
$$|0\rangle \to (|0\rangle + |1\rangle)/\sqrt{2}.$$

The map R is the *Hadamard mapping* on the space spanned by $\{|0\rangle, |1\rangle\}$, and we shall be using tensor products of R below. •

On a notational point, we have used R in lieu of the more customary H for *Hadamard*, since H also denotes a Hilbert space as well as the Hamiltonian operators in Schrödinger's equation.

(2.8) Exercise. Write R in terms of the spin matrices as $\vec{n}\cdot\vec{\sigma}$ and verify the relations $R\sigma_x R = \sigma_z$, $R\sigma_z R = \sigma_x$ and $R\sigma_y R = -\sigma_y$. •

Logic gates involving more than one qubit also have matrix representations in the computational basis, but since the dimensions grow exponentially with the number of qubits, an alternative "wiring" notation has evolved, motivated by an early paper of Feynman [36]. If one of two qubits is affected by the NOT operation, for example, we write



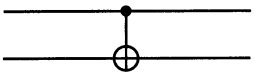
to indicate the operation on the first qubit. The intended flow of the logic is from left to right, and the transition is from $|j\rangle|k\rangle$ to $|j \oplus 1\rangle|k\rangle$.

(2.9) Example. The *controlled not* or *exclusive or* operation XOR applies a NOT gate to a target qubit only if the control qubit is in state 1, and the transition in Dirac notation is

$$|j\rangle|k\rangle \to |j\rangle|j \oplus k\rangle,$$

where the notation as usual denotes a mod(2) sum. Note that the two qubits are entangled after the XOR operation, and the state of the system

cannot be represented as a simple tensor product of the states of the individual qubits. The wiring diagram is



•

(2.10) Example. Using the computational basis with the topmost qubit representing the high-order bit, the *NOT* operation X has the matrix representation $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \otimes I$, which easily generalizes if there are more unaffected qubits. The matrix U for *XOR* is

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix},$$

and it is easy to confirm that these are unitary operations. •

(2.11) Exercise. Show that the matrix U in (2.10) is similar to

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}.$$

(*Hint*: use a change of basis matrix of the form $I \otimes R$, where R is defined in (2.7).) What is the representation of U after the basis change $R \otimes I$? Show that the basis change $R \otimes R$ reverses the roles of the first and second qubits in the *XOR* operation:

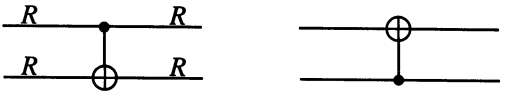
$$(R \otimes R)U(R \otimes R) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}.$$ •

It's useful to illustrate the last assertion in Exercise (2.11) using Dirac notation. Let U denote the linear operator that implements *XOR*: $U(|j\rangle|k\rangle) = |j\rangle|j \oplus k\rangle$, where the states are in the computational basis. Recall that $|0_x\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$ denotes the x-spin up state and $|1_x\rangle = (|0\rangle - |1\rangle)/\sqrt{2}$ denotes the x-spin down state. The result of (2.11)

is that if we look at U in the $\{|0_x\rangle, |1_x\rangle\}$ basis, we find that the second qubit controls the first qubit. For example:

$$\begin{aligned} U(|0_x\rangle|1_x\rangle) &= \frac{1}{2}U(|00\rangle + |10\rangle - |01\rangle - |11\rangle) \\ &= \frac{1}{2}(|00\rangle + |11\rangle - |01\rangle - |10\rangle) = |1_x\rangle|1_x\rangle. \end{aligned}$$

Two equivalent wiring diagrams are

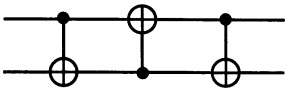


For the consequences of this observation, see the end of Chapter 4.

(2.12) Example. Combining three *XOR* gates, we can interchange the states of two qubits "without looking." Basis states change by $|j\rangle|k\rangle \rightarrow |k\rangle|j\rangle$ and thus

$$(\alpha|0\rangle + \beta|1\rangle)(\gamma|0\rangle + \delta|1\rangle) \rightarrow (\gamma|0\rangle + \delta|1\rangle)(\alpha|0\rangle + \beta|1\rangle).$$

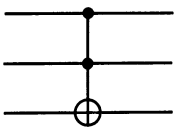
The following wiring diagram gives the schematic for the operation:



•

What's the utility of Example (2.12)? It's reasonable to suppose in a physical implementation of a quantum computer that it's easier to operate on two adjacent qubits than it is to operate on separated qubits. The circuit above shows that if we wanted to operate on separated qubits a and b, we could use operations on adjacent qubits to shift the state of qubit b to qubit c, a qubit adjacent to qubit a, perform the desired operation on qubits a and c, and then shift the resulting state in qubit c back to qubit b. Of course, the relative advantage of all of that shifting would depend on the physical implementation of the quantum computer.

The *XOR* gate can also be extended to the case when there are n bits controlling the reversal of the last qubit. Of particular use is the "2*XOR*" gate, which in Dirac notation maps $|i, j, k\rangle$ to $|i, j, k \oplus ij\rangle$; that is, the third bit is reversed only if both of the first two qubits are in state 1. The matrix representation is obvious, and the wiring diagram is



This gate is called the *Toffoli* gate ([74] or [3]) and it is a key component in quantum "circuits." We will have occasion later on to use an n-fold conditional command, that is, an operation is performed on qubit $(n+1)$ conditional on the contents of qubits 1 through n.

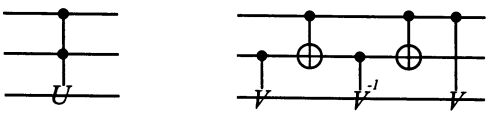
The original motivation for looking at such operations was to define logic gates for "reversible programming," and it was noted that the Toffoli gate suffices for the implementation of classical logic gates. Since the *not* and *and* gates are sufficient, it is enough to demonstrate that they can be implemented by the Toffoli gate.

(2.13) Exercise. Show that the Toffoli gate models the *not* gate by mapping $(1, 1, a)$ to $(1, 1, \neg a)$ and the *and* gate by

$$(a, b, 0) \rightarrow (a, b, a \wedge b). \qquad \bullet$$

If it proves to be easy to work with three qubits, the Toffoli gate could be implemented as above and could be considered a basic operation. However, if unitary operations on only one or two qubits are possible, then the Toffoli gate would have to be realized using more fundamental operations. In fact, using an earlier result of Deutsch, DiVincenzo [28] showed that the *XOR* gate and unitary operations on one qubit suffice for quantum programming. Such matters are discussed in detail in [3], and we illustrate some of the results of that paper in the following lemmas.

(2.14) Lemma. Let U denote a 2×2 unitary matrix and let *2XU* denote the analogue of *2XOR*, i.e., U is applied to the third qubit only if the first two qubits are 1. Suppose V is unitary and $V^2 = U$. Then *2XU* can be realized by a circuit in which at most two qubits at a time are affected. Specifically, the following two wiring diagrams are equivalent:



Proof: Neither of the first two qubits is different at the end of the routine, and if the first and second qubits are in state 1, then V^2 is applied to the third qubit. If either of the first two qubits is in state 0, then either the identity I, $V^{-1}V$ or VV^{-1} is applied to the third bit, completing the proof. $\bullet$

(2.15) Example. The Toffoli gate can be implemented physically by two-qubit operations, provided the controlled *XOR* and V can be imple-

mented, where V is defined as

$$V = \frac{e^{\frac{i\pi}{4}}}{\sqrt{2}} \begin{pmatrix} 1 & -i \\ -i & 1 \end{pmatrix}. \quad \bullet$$

Even if operations on two qubits are feasible, they probably will not be implementable in practice in the generality of (2.14). However, at the cost of additional one-qubit operations, it is possible to reduce our requirements for two-qubit operations to only the *XOR* operation.

(2.16) Proposition. Let W be a 2×2 unitary matrix with determinant 1; that is, W is in *SU(2)*. Then a controlled W operation can be implemented by the following circuit, where A, B and C are also in *SU(2)*.

Proof: Using the results of (1.24), we see that any 2×2 unitary matrix W can be written as a product of matrices in the form:

$$\begin{aligned} W &= e^{i\phi} \begin{pmatrix} e^{-i(\gamma+\delta)/2}\cos(\theta/2) & e^{i(-\gamma+\delta)/2}\sin(\theta/2) \\ -e^{i(\gamma-\delta)/2}\sin(\theta/2) & e^{i(\gamma+\delta)/2}\cos(\theta/2) \end{pmatrix} \\ &= e^{i\phi} \begin{pmatrix} e^{-i\gamma/2} & 0 \\ 0 & e^{i\gamma/2} \end{pmatrix} \begin{pmatrix} \cos(\theta/2) & \sin(\theta/2) \\ -\sin(\theta/2) & \cos(\theta/2) \end{pmatrix} \begin{pmatrix} e^{-i\delta/2} & 0 \\ 0 & e^{i\delta/2} \end{pmatrix}. \end{aligned}$$

If W is in *SU(2)*, the phase factor $e^{i\phi}$ has to be ± 1 and can be absorbed into the middle matrix.

Let us denote the first and last matrices in the last line as $R_z(\gamma)$ and $R_z(\delta)$, which we will see below can be interpreted as rotations about the z-axis in a three-dimensional space by angles γ and δ, respectively, while the middle matrix $R_y(\theta)$ can be interpreted as a rotation about the y-axis by an angle θ. (We expand on this terminology below.) If we set

$$C = R_z(\gamma)R_y(\theta/2) \quad B = R_y(-\theta/2)R_z(-(\gamma+\delta)/2) \quad A = R_z((\delta-\gamma)/2),$$

then $CBA = I$. Since

$$R_y(\theta/2)\sigma_x R_y(-\theta/2)\sigma_x = R_y(\theta)$$

and

$$\sigma_x R_z(-(\gamma+\delta)/2)\sigma_x = R_z((\gamma+\delta)/2),$$

we have

$$C\sigma_x B\sigma_x A = R_z(\gamma)R_y(\theta)R_z((\gamma+\delta)/2)R_z((\delta-\gamma)/2) = W.$$

It is then immediate that the two wiring diagrams are equivalent: if the first qubit is in state $|0\rangle$, CBA is applied to the second qubit, while if the first qubit is in state $|1\rangle$, $C\sigma_x B\sigma_x A$ is applied. •

To handle all 2×2 unitary matrices, it remains to show that a conditional multiplication by the phase factor $e^{i\phi}$ can be implemented without a conditional operation.

(2.17) Lemma. Let $S = \begin{pmatrix} e^{i\phi} & 0 \\ 0 & e^{i\phi} \end{pmatrix}$ and $E = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{pmatrix}$. Then the following circuits are equivalent and consequently any 2×2 unitary matrix can be implemented with no more than four one-qubit operations and two *XORs*.

Proof: Simply confirm that the same 4×4 unitary matrix

$$U = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & e^{i\phi} & 0 \\ 0 & 0 & 0 & e^{i\phi} \end{pmatrix}$$

is represented by both wiring diagrams. •

There is a well-developed theory justifying the interpretation of matrices in *SU(2)* as rotations in a three-dimensional space, and we summarize just enough of that theory here to justify the terminology. (See, for example, the physics texts cited earlier or Sternberg [71].)

The physical phenomenon of *spin* arises in a relativistic context, and it is appropriate to use real vectors $x = (x_0, x_1, x_2, x_3)$ with *Lorentz metric*

$$\|x\|^2 \equiv x_0^2 - x_1^2 - x_2^2 - x_3^2$$

to denote points in space-time. The relationship with the spin matrices is obtained by defining a bijective linear mapping from this real space of four-vectors to the linear space of 2×2 Hermitian matrices over the reals by

$$\tau(x) = \begin{pmatrix} x_0 + x_3 & x_1 - ix_2 \\ x_1 + ix_2 & x_0 - x_3 \end{pmatrix}$$

so that $\tau(x) = x_0\sigma_0 + x_1\sigma_x + x_2\sigma_y + x_3\sigma_z$. (Recall that in (1.16) we noted that $\{\sigma_0, \sigma_x, \sigma_y, \sigma_z\}$ is a basis for the linear space over the real numbers of Hermitian matrices.) Moreover, $\det(\tau(x)) = \|x\|^2$.

Using this identification of vectors with matrices, for any 2×2 unitary matrix U we can define a linear mapping $T(U)$ on x by

$$T(U)x = \tau^{-1}(U\tau(x)U^\dagger),$$

a valid definition since $U\tau(x)U^\dagger$ is Hermitian. Since U is unitary, $\|x\|^2$ equals $\|T(U)x\|^2$, and from the definition of $\tau(x)$ it can be shown that the x_0 component of x is unchanged by $T(U)$. If we simply set $x_0 = 0$, we can consider $T(U)$ to be a mapping on R^3. Building on these ideas, it can be shown that T maps *SU(2)* in a two-to-one fashion onto *SO(3)*, the group of orthogonal matrices on R^3 with determinant equal to one.

As an example, suppose $U = \begin{pmatrix} e^{-i\gamma/2} & 0 \\ 0 & e^{i\gamma/2} \end{pmatrix}$. It is easy to confirm that

$$\tau(T(U)x) = \begin{pmatrix} x_0 + x_3 & e^{-i\gamma}(x_1 - ix_2) \\ e^{i\gamma}(x_1 + ix_2) & x_0 - x_3 \end{pmatrix}$$

and thus the x_3 coordinates are unchanged. The interpretation is that $T(U)$ rotates the x_1, x_2 plane through an angle γ about the x_3 axis, and that explains the notation $R_z(\gamma)$ in Proposition (2.16).

(2.18) Exercise. Let $B = \begin{pmatrix} \cos(\theta/2) & \sin(\theta/2) \\ -\sin(\theta/2) & \cos(\theta/2) \end{pmatrix}$. If e_2 denotes the four-vector $(0, 0, 1, 0)$, show that $T(B)e_2 = e_2$ and that in general $T(B)$ represents a rotation of the x_3, x_1 plane through angle θ about the $y = x_2$ axis, explaining the notation of $R_y(\theta)$ in Proposition (2.16). •

If we could implement $R_y(\theta_0)$ for $\theta_0/2\pi$ irrational, then any $R_y(\theta)$ could be approximated arbitrarily well by sufficiently many iterates of $R_y(\theta_0)$. If we could also implement $R_z(\delta_0)$ for $\delta_0/2\pi$ irrational, then it follows from (2.16) that any U in *SU(2)* could be approximated arbitrarily closely by iterates of just two implementable operations, and therefore by (2.15) and (2.16) two one-qubit operations and the two-qubit *XOR* are minimally sufficient to realize the logic gates needed for quantum computation.

Finally, we note that the R_y and R_z mappings are conjugate to one another so that if a basis change operation is available, then only one rotation is needed. To see this, define the *SU(2)* matrix $U = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & i \\ i & 1 \end{pmatrix}$ and verify that $U\sigma_x U^\dagger = \sigma_x$ and $U\sigma_y U^\dagger = -\sigma_z$. Hence,

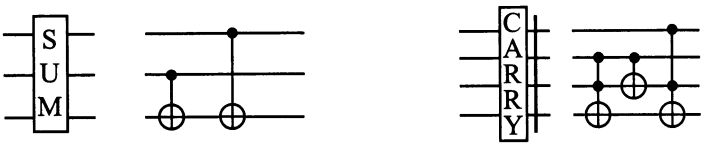
$$U\begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix} U^\dagger = U(\cos(\theta)\sigma_0 + i\sin(\theta)\sigma_y)U^\dagger = \begin{pmatrix} e^{-i\theta} & 0 \\ 0 & e^{i\theta} \end{pmatrix},$$

confirming the relationship.

2.4 Quantum subroutines: addition on a quantum computer

By patching quantum logic gates together, we can construct basic, unitary computational units. To give the reader an idea of the methodology, we present in detail the wiring diagrams required for addition mod(N).

To begin, consider the following two wiring diagrams:



The *SUM* subroutine adds the bits from the top two qubits to the third qubit,

$$|i, j, k\rangle \rightarrow |i, j, i \oplus j \oplus k\rangle,$$

and is unitary since it's the product of two unitary maps. *SUM* is also its own inverse.

CARRY is used in an addition subroutine to be described below and as usual reads from left to right. If we label the four qubits from top to bottom as c, a, b and d, respectively, then c will be used to denote a lower-order carry, and a and b will denote the bits to be added. If d is zero at first, d at the output will be the next carry bit. Under *CARRY*,

$$|c, a, b, d\rangle \rightarrow |c, a, a \oplus b, d \oplus ab \oplus ac \oplus bc\rangle.$$

(2.19) RCARRY. As part of the addition subroutine it will be necessary to run *CARRY* from right to left, and in that case the position of the vertical bar indicates the reversal. That is, the three operations defining *CARRY* are carried out in reverse order. This "reversed carry" effects

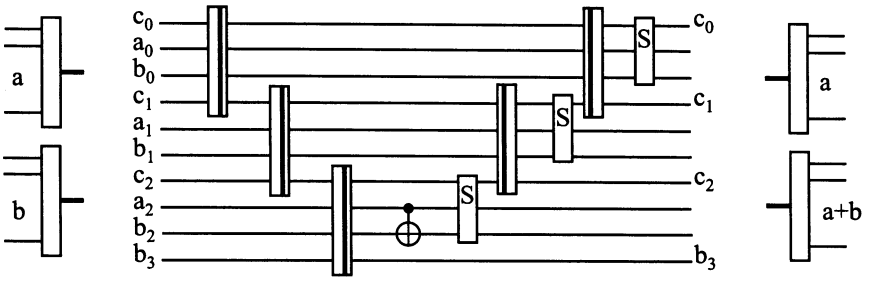
$$|u, x, y, v\rangle \rightarrow |u, x, x \oplus y, v \oplus uy \oplus xy \oplus x\rangle,$$

and we leave it to the reader to check that this is in fact the inverse of the *CARRY* operation. Again, both operations are unitary since they're the product of unitary maps. •

The routines for addition, addition mod(N), controlled multiplication mod(N) and exponentiation mod(N) are defined in complete detail in [76]. There is no claim that the proposed routines are the simplest possible, but they do give some understanding of the complexities of programming with unitary matrices. To keep the presentation manageable,

we shall study in detail only the first two routines. The multiplication and exponentiation mod (N) routines are central to Shor's algorithm, and we refer the reader to [76] for the wiring diagrams. (See also [4].)

We give the explicit wiring diagram for the addition of two three-bit numbers, and since the generalization to two n-bit numbers is obvious, we confine the analysis to the example. The flow is from left to right, as usual, and since each of the component operations is unitary, so is the entire addition routine. Note that the *CARRY* subroutine is run forward three times and in reverse order twice. The *SUM* subroutine is used three times and *XOR* once.



The input is $a_2a_1a_0$ and $0b_2b_1b_0$, while the output is $a_2a_1a_0$ and $a+b$, where the b_3 line denotes the possible high-order carry qubit. The three carry qubits, denoted by c_i, are initially zero and are also zero at the end of the routine, as we shall see. The b_3 qubit is considered to be an input qubit and is also equal to zero initially. The virtue of setting the carry qubits back to zero is that those qubits can serve as carry qubits for a subsequent addition and are disentangled from the a and b qubits.

At the end of the third *CARRY* subroutine, c_1, c_2 and b_3 contain the appropriate carry information, the a_i qubits are unchanged and the other b_i qubits are in state $(a_i \oplus b_i)$. Applying *XOR* next affects only the b_2 qubit, restoring it to its input state, so that the subsequent *SUM* operation puts the mod(2) sum of c_2, a_2 and b_2 into the b_2 qubit, as required for addition. Having used the carry information contained in c_2, the first reversed *CARRY* returns c_2 and b_1 back to their input states without affecting the other qubits, so that the next significant bit of the sum can be put into the b_1 qubit. The process is repeated once more, completing the calculation of $a+b$ and restoring the first three carry qubits to their input state of zero. Note that b_3, the high-order carry qubit, is not restored to its input state.

Since the *ADDER* routine has to be invertible, we need to examine the mapping when the qubit designated for the high-order carry isn't initially

in state zero, i.e., when b_3 isn't initially zero. If we denote the final value of the high-order carry bit as r and let (b) and $(a+b)$ denote the terms excluding the high-order carry bit, then the four possibilities are

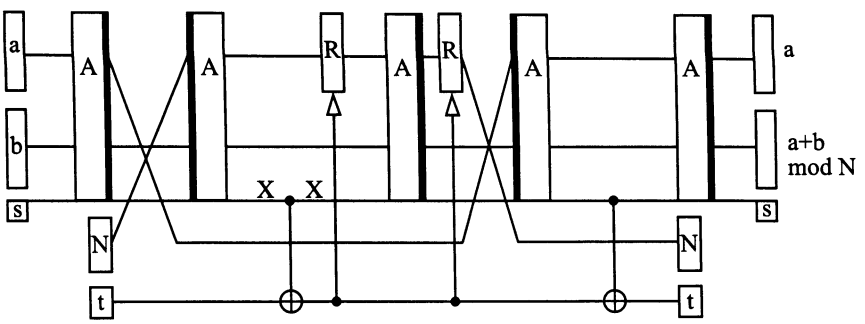
$$ADDER: \begin{bmatrix} |a,(b),0\rangle \\ |a,(b),1\rangle \\ |a,(b),0\rangle \\ |a,(b),1\rangle \end{bmatrix} \rightarrow \begin{bmatrix} |a,(a+b),0\rangle,(r=0) \\ |a,(a+b),1\rangle,(r=0) \\ |a,(a+b),1\rangle,(r=1) \\ |a,(a+b),0\rangle,(r=1) \end{bmatrix}.$$

(2.20) R-ADDER. As with the *CARRY* operator, *ADDER* can be run backwards, defining (almost) a *subtracter*. Suppose u and v are the inputs, with u and v *strictly less* than 2^n. That is, we separate out the high-order qubit. Then since ADDER is invertible, we have

$$R-ADDER: \begin{bmatrix} |u,v,0\rangle,(u \leq v) \\ |u,v,1\rangle,(u \leq v) \\ |u,v,0\rangle,(u > v) \\ |u,v,1\rangle,(u > v) \end{bmatrix} \rightarrow \begin{bmatrix} |u,v-u,0\rangle \\ |u,v-u,1\rangle \\ |u,2^n+v-u,1\rangle \\ |u,2^n+v-u,0\rangle \end{bmatrix} \bullet$$

In the next example, a and b are less than N, and we want to compute $(a+b)$ mod (N) on a quantum computer. We follow the approach given in [76], which requires five applications of the *ADDER* subroutine, two of which are run in reverse. Schematically, the routine is represented in the diagram below. (As before, we use the vertical line to indicate the direction of the *ADDER*.) We assume that a and b are n-bit numbers less than N, that N is less than 2^{n+1} and that s denotes the state of a qubit associated with an $(n+2)nd$ carry bit in the *ADDER* subroutine. We allow $n+1$ qubits for each of a, b and N, and s is the separated high-order carry bit. As with *ADDER*, we index beginning with 0.

ADDITION MOD N



The input to the routine in order is a, b, the high-order "s-qubit" which is initially in state 0, the number N stored in another bank of $n+1$ qubits and the "t-qubit," also preset to 0. The action of the first adder is $|a,b,0,N,0\rangle \rightarrow |a,a+b,0,N,0\rangle$. In the operation of the first reversed adder, N plays the role of u, and using the results of the example in (2.20), we have

$$\begin{bmatrix} |a,a+b,0,N,0\rangle, N \leq a+b \\ |a,a+b,0,N,0\rangle, N > a+b \end{bmatrix} \rightarrow \begin{bmatrix} |a,a+b-N,0,N,0\rangle \\ |a,2^{n+1}+a+b-N,1,N,0\rangle \end{bmatrix}.$$

The state of the s-qubit carries the information about the relative size of $a+b$ and N, and that information is next copied into the unused t-qubit, whose state is the fifth entry in the state vector, so that $t=1$ if and only if $N \leq a+b$. The vertical arrow denotes a conditional operation requiring the utility qubits in R which are initially set equal to zero. If the t-qubit is in state 0, there is no interchange of the N- and R-qubits, and *ADDER* reverses the prior subtraction of N. If the t-qubit is in state 1, the N and R registers are interchanged before *ADDER*, so that 0 is the "a" input, and *ADDER* has no effect. In either case, subsequent to that addition, the conditional interchange is applied once more, returning the qubits initially holding N to their original state and guaranteeing that each of the R-qubits is in state 0. The result is the following mapping:

$$\begin{bmatrix} |a,a+b-N,0,N,1\rangle \\ |a,2^{n+1}+a+b-N,1,N,0\rangle \end{bmatrix} \rightarrow \begin{bmatrix} |a,a+b \bmod (N),0,N,1\rangle \\ |a,a+b \bmod (N),0,N,0\rangle \end{bmatrix}.$$

Thus, we have computed $(a+b) \bmod (N)$ but have entangled the t-qubit in the process and need to undo that if *ADDITION MOD N* is to be used as a subroutine in other calculations. The remainder of the computation achieves that purpose. We first run *ADDER* in reverse once again, this time with a and $(a+b) \bmod (N)$ as the inputs, and obtain

$$\begin{bmatrix} |a,a+b \bmod (N),0,N,1\rangle \\ |a,a+b \bmod (N),0,N,0\rangle \end{bmatrix} \rightarrow \begin{bmatrix} |a,2^{n+1}+b-N,1,N,1\rangle \\ |a,b,0,N,0\rangle \end{bmatrix},$$

where the form of the right-hand side is forced by virtue of the prior calculations. The intervening *XOR* command converts the t-qubit to its input state of zero, and a final application of *ADDER* reverses the prior subtraction, leaving the system in $|a,a+b \bmod (N),0,N,0\rangle$ and the utility qubits in their initial states.

(2.21) Exercise. The last two applications of *ADDER* — forward and in reverse — were necessary to disentangle the t-qubit from the output.

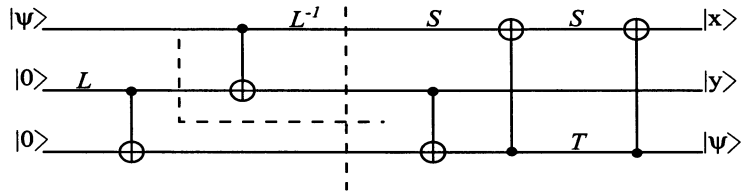
Is there a less expensive way to accomplish this? (There is no school solution.) •

In [76] the subroutines described above are used to construct a controlled multiplication mod(N) routine and subsequently an exponentiation mod(N) routine. The interested reader can inspect the flow charts presented in that paper for the details.

2.5 Quantum subroutines: a teleportation circuit

In 1984 Bennett and Brassard [10] suggested the use of entangled states as a means of public key distribution, and in [11] the use of entangled states as a means of quantum "teleportation" was proposed. At the simplest level, the idea of teleportation is that if a sender and a receiver share an entangled state of the form $(|00\rangle+|11\rangle)/\sqrt{2}$, then it is theoretically possible for the sender to transmit an arbitrary one-qubit superposition to the receiver using only classical means of communication. This idea quickly leads into questions of quantum communication theory and quantum information theory, and we limit the discussion to one simple example. The official rationale for including it here is that this example could be relevant to a quantum computer as an internal circuit. The real reason is that it is a very clever idea which illustrates the bizarre consequences of quantum theory.

We follow the presentation in [17], beginning with an explicit wiring diagram



where

$$L = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix} \quad S = \begin{pmatrix} i & 0 \\ 0 & 1 \end{pmatrix} \quad T = \begin{pmatrix} -1 & 0 \\ 0 & -i \end{pmatrix}.$$

The vertical dashed line in the center of the diagram is intended to demarcate the area of influence of the sender (left) and the receiver (right), or *Alice* and *Bob* in the standard parlance. Sender and receiver could denote different parts of a quantum computer or, more dramatically, Alice and Bob could be separated by a continent. The right-angled dashed

line denotes a potential delay in communication, which will be discussed below.

Let us trace the effect of the circuit. We begin with an input to the sender of $(\alpha|0\rangle + \beta|1\rangle)|0\rangle|0\rangle$ which is $(\alpha|0\rangle + \beta|1\rangle)(|00\rangle + |11\rangle)$ after the first two operations, ignoring the normalizing factor, and completing the first stage. At this point, the sender might transmit the third qubit to the receiver for storage, retaining the second qubit in storage for later use. Alternatively, the processing might continue as sketched in the wiring diagram. In either case, a word of caution is in order. Because the second and third qubits are entangled, they operate as one physical state and must be treated that way even though they may be separated in space. While this seems counterintuitive, it has been verified by experiment, and it is this entanglement that is the basis for the teleportation.

The second stage consists of the next two operations in which the sender completes the processing of the input, acting on the first qubit, which has not been needed until now. Continuing to ignore the normalizing factor, we first have

$$(\alpha|0\rangle + \beta|1\rangle)(|00\rangle + |11\rangle) \rightarrow \alpha(|000\rangle + |011\rangle) + \beta(|110\rangle + |101\rangle).$$

Then an application of L^{-1} gives

$$\alpha(|000\rangle - |100\rangle + |011\rangle - |111\rangle) + \beta(|010\rangle + |110\rangle + |001\rangle + |101\rangle)$$

completing the second stage of the circuit.

Suppose the first two qubits are transmitted as qubits to the receiver, who then applies S to the first qubit and XOR to the second and third qubits, obtaining

$$\alpha(i|000\rangle - |100\rangle + i|010\rangle - |110\rangle) + \beta(i|011\rangle + |111\rangle + i|001\rangle + |101\rangle).$$

After the next XOR, the S and T mappings and the final XOR the system is in the state

$$\alpha(|000\rangle + |100\rangle + |010\rangle + |110\rangle) + \beta(|011\rangle + |111\rangle + |001\rangle + |101\rangle),$$

which can be written as

$$(|0\rangle + |1\rangle) \otimes (|0\rangle + |1\rangle) \otimes (\alpha|0\rangle + \beta|1\rangle).$$

In other words, the superposition entering at the first qubit now appears in the third qubit, and the first two qubits contain the same uniform superposition of the basis states. Thus, the circuit effects the mapping $|\psi\rangle|0\rangle|0\rangle \rightarrow |\varphi\rangle|\varphi\rangle|\psi\rangle$, where $|\varphi\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$.

In Example (2.12) we showed how to interchange the contents of two qubits, retaining superposition, so the result above may not appear too surprising. However, suppose that at the end of the second stage, just before the vertical dashed line, the sender measured the first two qubits and sent via classical communication the results of those measurements. If the receiver loads into quantum states the transmitted classical information, then just past the dashed line, the system is in one of the four states

$$\begin{array}{ll} |0\rangle|0\rangle(\alpha|0\rangle + \beta|1\rangle) & |0\rangle|1\rangle(\alpha|1\rangle + \beta|0\rangle) \\ |1\rangle|0\rangle(-\alpha|0\rangle + \beta|1\rangle) & |1\rangle|1\rangle(-\alpha|1\rangle + \beta|0\rangle). \end{array}$$

And here's the remarkable feature of the wiring. If the receiver now completes the circuit, the original input again appears in the third qubit, and moreover the transmitted classical states appear in the same positions at the end of the computation!

(2.22) Exercise. Verify that, after the second procedure, the input does in fact appear in the third qubit at the end of the computation. •

If the technology permitted it, here's how one could construe the result as teleportation. The sender creates an initial entangled state and transmits one of the entangled states to the receiver, both of them storing their share of the entanglement for future use. Sometime later the sender receives a superposed state. Operating as indicated, the sender completes the second stage locally and measures the results, transmitting over classical channels the results of the measurements. The receiver inputs that classical information into the latter part of the circuit and reproduces the initial quantum superposition without receiving any additional quantum mechanical information. That is, using the resources of a prior entanglement and the communication of two classical bits, the sender can transmit an arbitrary qubit.

3
Quantum Algorithms

We are now ready to assemble quantum subroutines into quantum algorithms. A reasonable first question is: Are there problems which can be solved efficiently on a quantum computer? In his analysis of a universal quantum computer [25], Deutsch illustrated quantum parallelism by constructing a procedure to compute the parity $f(0) \oplus f(1)$ of a function on one bit. In a generalization of this context, Deutsch and Jozsa [27] defined a quantum algorithm that could solve a problem more efficiently than is possible on a classical computer. This problem helped motivate the search for algorithms that could solve "real" problems, and Simon [67] provided an example that inspired Shor's subsequent work on the factoring problem and on the discrete logarithm problem.

We begin this chapter by describing the Deutsch–Jozsa algorithm and then describe Simon's algorithm, emphasizing the structure that serves as a motivation for other quantum algorithms such as Shor's algorithms and part of Grover's search algorithm. An important component of Shor's algorithms is the finite Fourier transform, and we also examine that algorithm in detail. Having worked through these significant algorithms, we discuss their common features, completing the chapter with a general theory which we apply to the discrete log problem.

3.1 Deutsch–Jozsa algorithm

Suppose we are challenged to distinguish between two different classes of functions which map V^n to $\{0, 1\}$, where V^n denotes the linear space of n-

long binary vectors. Functions in one class are constant while functions in the second class are balanced: exactly 2^{n-1} vectors map to 0 and exactly 2^{n-1} vectors map to 1. Given a particular function f, we would have to evaluate f on at least two vectors and on at most $2^{n-1}+1$ vectors in order to determine with certainty the class to which f belonged. If n is large, certainty may not be possible with classical computers, even though the probability of making an error decreases exponentially with the number of evaluations of the function. However, on a quantum computer, only one call of the function is required.

Here's the context. We assume n qubits initially in state $|0\rangle$ and another qubit which is also in state $|0\rangle$. We assume there is an algorithm which actually evaluates $f(x)$ via a quantum algorithm for a given input vector x and stores the result in one additional qubit. The trick is to use the *Hadamard transform* $R^{(n)}$ before and after an evaluation of the function f, and we first examine that operation in detail.

Specifically, suppose we have a state $|k\rangle = |k_{n-1}\rangle \otimes \cdots \otimes |k_0\rangle$, where the k_i denote the usual values in the binary expansion of k and $0 \le k \le 2^n - 1$. Then the Hadamard transform $R^{(n)}$ of $|k\rangle$ is defined as

$$|k\rangle \to \frac{1}{2^{n/2}} \sum_{u=0}^{2^n-1} (-1)^{k\cdot u} |u\rangle, \tag{3.1}$$

where $k \cdot u = \sum k_i u_i \bmod (2)$ is the mod(2) dot product of the n-long binary vectors defined by the binary representations of k and u.

(3.2) Lemma. The Hadamard transform $R^{(n)}$ is a feasible operation, implemented by n applications of $R = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ first defined in Example (2.7).

Proof: If we apply R to each of the n qubits, we map $|k\rangle$ to

$$R|k_{n-1}\rangle \otimes \cdots \otimes R|k_0\rangle = \frac{1}{2^{n/2}} (|0\rangle + (-1)^{k_{n-1}}|1\rangle) \otimes \cdots \otimes (|0\rangle + (-1)^{k_0}|1\rangle).$$

When we multiply out the right-hand side, we get 2^n terms which can be indexed by the choice of $|0\rangle$ or $|1\rangle$ from each of the n factors. Using the same ordering as in the definition of k, let u be the resulting integer with binary terms u_i. Then, as advertised, $|k\rangle$ maps to

$$\frac{1}{2^{n/2}} \sum_{u=0}^{2^n-1} (-1)^{k_{n-1}u_{n-1}+\cdots+k_0u_0} |u\rangle = \frac{1}{2^{n/2}} \sum_{u=0}^{2^n-1} (-1)^{k\cdot u} |u\rangle = R^{(n)}|k\rangle. \quad \bullet$$

The special case when $k = 0$ is worth recording separately, and we note that the effect is a "randomization" or the construction of a uniform superposition over all of the integers $0 \leq k \leq 2^n - 1$:

$$|0\rangle = |0\rangle \otimes \cdots \otimes |0\rangle \to \frac{1}{2^{n/2}} \sum_{j=0}^{2^n-1} |j\rangle. \tag{3.3}$$

Here's how Deutsch and Jozsa [27] determine the character of f in two calls of the function. (We show later in the section how only one call of the function suffices.)

(3.4) Deutsch–Jozsa Algorithm.

Step 1: Randomize the initial setting by applying R to each of the first n qubits:

$$|0, \ldots, 0\rangle|0\rangle \to \frac{1}{2^{n/2}} \sum_{j=0}^{2^n-1} |j\rangle|0\rangle.$$

Step 2: Evaluate the function and store the result:

$$\frac{1}{2^{n/2}} \sum_{j=0}^{2^n-1} |j\rangle|0\rangle \to \frac{1}{2^{n/2}} \sum_{j=0}^{2^n-1} |j\rangle|f(j)\rangle.$$

Since the 2^n states are in superposition, we have in some sense computed f simultaneously on all of the states with one call of the f-subroutine.

Step 3: Apply the unitary operator $U = \sigma_z$ to the last qubit, obtaining

$$\frac{1}{2^{n/2}} \sum_{j=0}^{2^n-1} |j\rangle|f(j)\rangle \to \frac{1}{2^{n/2}} \sum_{j=0}^{2^n-1} |j\rangle(-1)^{f(j)}|f(j)\rangle.$$

Step 4: Again evaluate the function and add the result into the last qubit:

$$\frac{1}{2^{n/2}} \sum_{j=0}^{2^n-1} |j\rangle(-1)^{f(j)}|f(j)\rangle \to \frac{1}{2^{n/2}} \sum_{j=0}^{2^n-1} |j\rangle(-1)^{f(j)}|0\rangle.$$

This step has the effect of disentangling the last qubit from the first n qubits, a requisite part of the algorithm.

Step 5: Apply $R^{(n)}$ a second time to the first n qubits:

$$\frac{1}{2^{n/2}} \sum_{j=0}^{2^n-1} |j\rangle(-1)^{f(j)}|0\rangle \to \frac{1}{2^n} \sum_{u=0}^{2^n-1} |u\rangle|0\rangle \sum_{j=0}^{2^n-1} (-1)^{u \cdot j}(-1)^{f(j)}.$$

Now suppose that f is a constant function. Then the summation over j produces zero for $u \neq 0$ and 2^n for $u = 0$. Hence, a measurement of the first n qubits gives $u = 0$ with probability one. If f is a balanced function and $u = 0$, then the summation over j is zero and a measurement over the first n qubits gives some $u \neq 0$ with probability one. It follows that the measurement at *Step 5* distinguishes between the classes with certainty, completing the algorithm. •

As we see next, a clever modification of *Step 2* makes it unnecessary to call f a second time, so that the class to which f belongs can be determined with only one call of the function. This trick appears in [22] and improves on the original solution. However, as the reader will recall from the examples in Chapter 2, the evaluation of a function on a quantum computer also involves the restoration of the ancillary states to their original form, and qualitatively that can make the algorithm twice as long as its classical counterpart. As another illustration of this technique, see Example (3.7) below.

(3.5) Lemma. Only one call of f is required in the Deutsch–Jozsa algorithm.

Proof: The point of *Step 3* above is to obtain the factor $(-1)^{f(j)}$ and the point of *Step 4* is to disentangle the last qubit from the first n qubits. Both steps are unnecessary if the $(n+1)$st qubit is initially in a superposition of the basis states and the value of $f(j)$ is stored differently at *Step 2*:

$$\begin{aligned} (|0\rangle - |1\rangle)/\sqrt{2} &\rightarrow (|0 \oplus f(j)\rangle - |1 \oplus f(j)\rangle)/\sqrt{2} \\ &= (-1)^{f(j)} \left((|0\rangle - |1\rangle)/\sqrt{2} \right). \end{aligned}$$

The multiplicative phase factor can then be associated with the first n qubits, which in turn have no entanglement with the state of the $(n+1)$st qubit. •

3.2 Simon's algorithm

We have now shown that there is a problem that can be solved more efficiently on a quantum computer than on a classical computer. The key ingredients were the two applications of the Hadamard transform separated by an entanglement created by the evaluation of a function. The effect was to change the probability amplitudes of the states so that a subsequent measurement gave definitive information.

Suppose we relax the requirement that one measurement tells all, and allow several runs of the quantum algorithm with the cumulative information acquired sufficing to solve a problem. As an example, suppose there is an unknown n-bit vector ξ which we want to find. Suppose also that there is a function or an *oracle* f on all n-bit binary vectors x which has the property that $f(x) = f(y)$ if and only if $x = y \oplus \xi$, where $\oplus$ denotes as usual coordinate-wise mod(2) addition. In group-theoretic terms, $K = \{0, \xi\}$ is a subgroup of F_2^n, the group of binary n-vectors, and f is constant on cosets of K and takes different values on different cosets. *Simon's algorithm* uses the properties of such a function to enhance the probability of measuring something useful in determining ξ.

To do this we need three registers of qubits: an n-qubit register for the domain of f, a j-qubit register for the value of f with $j \geq n-1$ and an ancillary set of qubits which are left implicit. Here is the algorithm.

(3.6) Simon's Algorithm.

Step 1: Initialize each of the $n+j$ qubits to the computational basis state $|0\rangle$.

Step 2: Randomize the n-input qubits using $R^{(n)}$ as in (3.3), so that with $N = 2^n$,

$$|0, \ldots, 0\rangle|0\rangle \to \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} |k\rangle|0\rangle = |\psi\rangle|0\rangle.$$

Step 3: Evaluate the function f:

$$|\psi\rangle|0\rangle \to \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} |k\rangle|f(k)\rangle = |\psi, f\rangle.$$

Step 4: Apply the Hadamard transform $R^{(n)}$ to the first n qubits a second time:

$$\begin{aligned} |\psi, f\rangle &\to \frac{1}{\sqrt{N}} \sum_{u=0}^{N-1} |u\rangle \frac{1}{\sqrt{N}} \sum_{k=0}^{2^n-1} (-1)^{k \cdot u} |f(k)\rangle \\ &= \frac{1}{N} \sum_{u=0}^{N-1} |u\rangle \sum_{m} |f(m)\rangle ((-1)^{m \cdot u}) \left(1 + (-1)^{\xi \cdot u}\right) = |\varphi\rangle, \end{aligned}$$

where m denotes a representative of one of the 2^{n-1} cosets, and $f(m)$ is the common value of f on that coset. Note that if u is not orthogonal to ξ, that is, if $u \cdot \xi$ is not equal to zero, then the coefficient of $|u\rangle$ is zero.

Step 5: Measure the first n qubits, obtaining a particular state v orthogonal to ξ with probability

$$\langle\varphi|P_v|\varphi\rangle = \frac{1}{N^2}\sum_m \langle f(m)|f(m)\rangle \left|\left(1+(-1)^{\xi\cdot v}\right)\right|^2 = \frac{1}{2^{n-1}}.$$

That is, each of the 2^{n-1} binary vectors orthogonal to the unknown ξ is measured with equal probability.

Step 6: Repeat the process until r linearly independent vectors orthogonal to ξ have been obtained.

Step 7: Classically solve for vectors orthogonal to the space spanned by $v_1, \ldots, v_r$. If that orthogonal subspace has sufficiently small dimension, ξ can be determined by appropriate secondary testing. (And if $r = n-1$, no secondary testing is necessary!) •

The underlying idea of this algorithm is that periodic properties of the function f can be realized in the phase factor of a state and then transformed into enhanced probabilities of states which can subsequently provide information about the unknown vector ξ. The solution of the actual problem may require secondary calculations and the basic quantum computation may have to be repeated several times. This approach can be abstracted in terms of an underlying Abelian group structure and the characters of that group, but at this point we confine ourselves to specific examples.

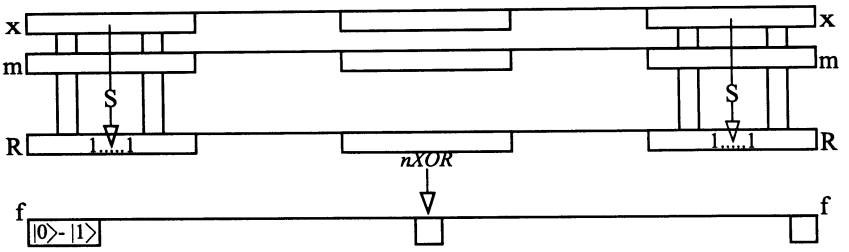
3.3 Grover's algorithm

Before describing Shor's factoring algorithm, which also translates periodicity into increased probabilities, we describe a slightly different use of the Hadamard transform which Grover [41] developed for an exhaustive search algorithm. Suppose we have an oracle which is a "needle-in-the-haystack" function f defined on integers k, $0 \le k \le 2^n - 1$, so that $f(k) = 0$ for all k except for $k = k_0$, and $f(k_0) = 1$. If we search through the domain of f, classically evaluating $f(k)$ until we find k_0, we can expect to make about $N/2$ evaluations of f, where $N = 2^n$. Grover's algorithm is a quantum algorithm which estimates that $O\left(\sqrt{N}\right)$ evaluations will be required to find k_0. An analysis of Grover's algorithm by [16] identified an underlying geometry in the algorithm and generalized the approach to the case when f takes the value one t times, that is, when there are t needles in the haystack.

As we saw in the preceding section, simple subroutines can be rather involved, and assuming that f can be evaluated is assuming a large amount

of quantum programming. Nonetheless, we shall make that assumption and work through the details of Grover's algorithm to illustrate in detail the implementation of the basic strategy of a potentially useful quantum algorithm: compute in superposition and maximize the probability of a useful outcome.

(3.7) Example. Suppose the problem consisted of comparing an n-bit number x with a given n-bit number m and letting $f(x)$ equal one if $x = m$ and equal zero otherwise. The following wiring diagram illustrates the kind of computation assumed as a subroutine in Grover's algorithm as well as the use of the state $(|0\rangle - |1\rangle)$ in the utility qubit. (We suppress the normalizing factor of $1/\sqrt{2}$.)



The qubits containing x are in the top n qubits, and those containing the given m are in the second array of n qubits. R denotes n qubits initially in state 1. The first operation consists of n distinct *SUMs* which have the effect of keeping each qubit in R equal to 1 if and only if the corresponding x and m qubits are equal. The *nXOR* operation denotes an n-fold *XOR* operation on the f qubit. Thus, the f-qubit contains a phase factor of (-1) if and only if x equals m. The rest of the operations map the R qubits back to their initial fills. Note that *nXOR* can be realized using unitary maps which affect only a few qubits at a time, so that it is a feasible operation. See [3] for specific details. •

As with Simon's algorithm, we present Grover's algorithm as a sequence of steps represented in both Dirac notation and the N-dimensional computational basis notation, where $N = 2^n$ as before. The basic principles are easily illustrated in a two-qubit context, which we analyze in detail; the generalization to n qubits is immediate. Finally, we discuss the feasibility of the algorithm; that is, we show that the unitary operations of the algorithm can be implemented by sequences of unitary maps involving only a few qubits at a time.

(3.8) Grover's Algorithm.

Step 1: Initialize n qubits to 0 and the last qubit to $|\chi\rangle \equiv (|0\rangle - |1\rangle)/\sqrt{2}$.

Step 2: Randomize the n input "domain" qubits as above, so that

$$|0,\ldots,0\rangle|\chi\rangle \rightarrow \sum_{k=0}^{N-1} a_k|k\rangle|\chi\rangle = |\psi\rangle|\chi\rangle,$$

where $a_k = 1/\sqrt{N}$.

Step 3: Perform steps a and b m times, where m will be defined in (3.15):

a. Compute the value of f via a unitary map U_f and add it into the last qubit to obtain the phase factor $(-1)^{f(k)}$:

$$|\psi\rangle|\chi\rangle \rightarrow |\psi\rangle U_f|\chi\rangle = \sum_{k=0}^{N-1} a_k|k\rangle(-1)^{f(k)}|\chi\rangle.$$

We denote this transformation as T.

b. Apply the "diffusion" operator $D = -I + 2J/N$ to the n domain qubits, where J is the all-ones matrix. It is easy to check that D is orthogonal and hence unitary. We show below that D equals $R^{(n)}SR^{(n)}$, where $R^{(n)}$ is the Hadamard transform and $S = -I$ except for $S(0,0) = 1$:

$$\sum_{k=0}^{N-1} a_k(-1)^{f(k)}|k\rangle|\chi\rangle \rightarrow \sum_{k=0}^{N-1} a_k^{(1)}|k\rangle|\chi\rangle.$$

Step 4: Measure the domain qubits and determine a state k. This is not a unitary operation, and superposition is lost after the measurement.

Step 5: Evaluate $f(k)$. If $f(k) = 1$, quit. Otherwise go to *Step 1*. •

Note that the use of $|\chi\rangle$ as the state of the utility qubit enables us to use the trick described in Lemma (3.5) and to complete *Step 3* with m calls of the function.

Since the initialization and randomization steps can be accomplished by operating on the qubits one at a time, they are feasible steps. Thus, it suffices to examine the iterative step to understand the algorithm and to establish its feasibility. We begin with the two-qubit case, so that $N = 4$, and for specificity we also assume that $k_0 = 2$. Then the initialization and randomization give

$$|0,0\rangle|\chi\rangle \rightarrow \sum_{k=0}^{3} \frac{1}{2}|k\rangle|\chi\rangle.$$

The operations in *Steps 3a–b* can be modeled by a matrix multiplication. T is a diagonal matrix with diagonal entries $(-1)^{f(k)} = (1 - 2f(k))$, and

both T and D are 4×4 matrices. Then it is easy to compute the entries of the mapping DT:

$$M \equiv DT = \begin{pmatrix} -0.5 & 0.5 & -0.5 & 0.5 \\ 0.5 & -0.5 & -0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & -0.5 & -0.5 \end{pmatrix}.$$

If we were to measure the system before applying M, we would find each of the four states with equal probability. However, after applying M we have

$$|\varphi\rangle = M|\psi\rangle = \begin{pmatrix} -0.5 & 0.5 & -0.5 & 0.5 \\ 0.5 & -0.5 & -0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & -0.5 & -0.5 \end{pmatrix} \begin{pmatrix} 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}.$$

Now if we measure the system, we find state k_0 with probability 1.

If we were to run the iterative step twice, we would find

$$M^2|\psi\rangle = 0.5|2\rangle - \sum_{k \neq 2} 0.5|k\rangle,$$

which means that doing twice the work actually produces a lower probability of finding the correct state. Thus, we have to specify in advance the value of m, the number of times the iterative step is run, in order to maximize the probability of finding the correct state.

To get a better idea of the effect of the iterative step, let's continue with the example when $N = 4$. We begin the iterative step with all of the numerical coefficients equal to 0.5. However, abstract a bit by letting r_0 denote the initial common value of the coefficients of the incorrect states and s_0 the coefficient of the correct state. Then an application of M gives

$$\begin{pmatrix} -0.5 & 0.5 & -0.5 & 0.5 \\ 0.5 & -0.5 & -0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & -0.5 & -0.5 \end{pmatrix} \begin{pmatrix} r_0 \\ r_0 \\ s_0 \\ r_0 \end{pmatrix} = \begin{pmatrix} 0.5r_0 - 0.5s_0 \\ 0.5r_0 - 0.5s_0 \\ 1.5r_0 + 0.5s_0 \\ 0.5r_0 - 0.5s_0 \end{pmatrix} = \begin{pmatrix} r_1 \\ r_1 \\ s_1 \\ r_1 \end{pmatrix}$$

so that the incorrect states also have a common coefficient after the mapping, and the evolution of the *coefficients* can be summarized in a 2×2 matrix:

$$\begin{pmatrix} r_1 \\ s_1 \end{pmatrix} = \begin{pmatrix} 0.5 & -0.5 \\ 1.5 & 0.5 \end{pmatrix} \begin{pmatrix} r_0 \\ s_0 \end{pmatrix}.$$

It is also easy to check that $3r_1^2+s_1^2 = 3r_0^2+s_0^2 = 1$, which is the necessary normalization.

We can rewrite the last equation using that normalization. If we define $a_i = \sqrt{3}r_i$ and $b_i = s_i$, the r-s equation above is equivalent to

$$\begin{pmatrix} a_1 \\ b_1 \end{pmatrix} = \begin{pmatrix} \sqrt{3} & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0.5 & -0.5 \\ 1.5 & 0.5 \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{3}} & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} a_0 \\ b_0 \end{pmatrix} = R\left(\frac{\pi}{3}\right) \begin{pmatrix} a_0 \\ b_0 \end{pmatrix},$$

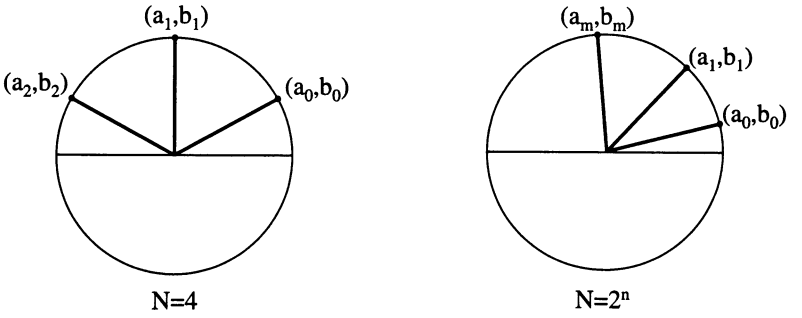
where

$$R\left(\frac{\pi}{3}\right) = \begin{pmatrix} 0.5 & -\sqrt{3}/2 \\ \sqrt{3}/2 & 0.5 \end{pmatrix},$$

denotes a counterclockwise rotation through the indicated angle. The interpretation is that we are looking at the evolution of the probability amplitudes of the *incorrect* and the *correct subspaces*. Since the *initial subspace probability amplitude* is

$$\begin{pmatrix} a_0 \\ b_0 \end{pmatrix} = \begin{pmatrix} \cos\left(\frac{\pi}{6}\right) \\ \sin\left(\frac{\pi}{6}\right) \end{pmatrix},$$

we have an explanation of the phenomena noticed above. One application of M, as encoded in $R(\frac{\pi}{3})$, rotates the subspace amplitude vector by 60°, guaranteeing that a measurement will detect the correct solution, since the initial amplitude vector is at an angle of 30°. However, a second application of M rotates the amplitude another 60° so that the probability of measuring a correct solution is random. Pictorially, the situation looks like this:



Thus, we need to specify in advance the value of m, the number of times the iterative step is taken. That value turns out to be dependent on N and also on t, the number of correct solutions. As in the two-qubit case, an incorrect choice of m in the general case could lead to doing

too much work *and* the possible reduction of the probability of finding a solution.

The general case of n qubits is immediate. As before, we compute the function f, obtaining the factor $(-1)^{f(k)}$ and defining the mapping T. The overall effect of *Step 3* is $M = DT$:

$$\begin{matrix} M(k_0,k_0) = 1 - 2/N \\ M(k_0,k) = 2/N,\ k \neq k_0 \end{matrix} \qquad k \neq k_0 \Rightarrow \begin{cases} M(k,k) = -1 + 2/N \\ M(k,k_0) = -2/N \\ M(k,j) = 2/N,\ j \neq k, k_0 \end{cases}$$

Again, it is easy to see that if the incorrect states have equal coefficients before applying M, they will have equal coefficients after applying M and that the values are related by

$$\begin{pmatrix} r_{i+1} \\ s_{i+1} \end{pmatrix} = \begin{pmatrix} 1 - 2/N & -2/N \\ 2 - 2/N & 1 - 2/N \end{pmatrix} \begin{pmatrix} r_i \\ s_i \end{pmatrix}. \tag{3.9}$$

This time the normalization is $(N-1)r_i^2 + s_i^2 = 1$, and the subspace amplitudes are $a_i = \sqrt{N-1}r_i$ and $b_i = s_i$. The r-s matrix again translates into an amplitude rotation with

$$R(\theta) = \begin{pmatrix} 1 - 2/N & -2\sqrt{N-1}/N \\ 2\sqrt{N-1}/N & 1 - 2/N \end{pmatrix} \tag{3.10}$$

and the initial probability amplitudes are

$$\begin{pmatrix} a_0 \\ b_0 \end{pmatrix} = \begin{pmatrix} \sqrt{1 - 1/N} \\ \sqrt{1/N} \end{pmatrix} = \begin{pmatrix} \cos(\theta/2) \\ \sin(\theta/2) \end{pmatrix}. \tag{3.11}$$

Since the number of iterations m should satisfy

$$\theta(m + 1/2) \approx \pi/2,$$

it follows that if N is large, so that θ is approximately $2/\sqrt{N}$, then

$$m \approx \sqrt{N}\pi/4 - 1/2. \tag{3.12}$$

(3.13) Exercise. Confirm the assertions above and also confirm that the angle in the rotation matrix is twice the angle of the initial probability amplitude. (*Hint*: use a trigonometric identity.) •

The preceding analysis carries over to the case when there are exactly t distinct values of k for which $f(k) = 1$. The algorithm has been presented

to accommodate more than one correct solution, and the only difficulty is finding the entries of M and confirming that the "equal coefficient property" holds in this case. Conceptually there is nothing new, and we leave the calculation of the initial probability amplitude and the subspace rotation matrix to the reader.

(3.14) Exercise. Show that if there are t solutions, then

$$\frac{\theta}{2} = \arcsin\left(\sqrt{t/N}\right)$$

is the angle of the initial amplitude vector. Again each application of M rotates the subspace amplitude vector through twice the initial angle:

$$\begin{pmatrix} a_m \\ b_m \end{pmatrix} = R^{(m)}(\theta)\begin{pmatrix} a_0 \\ b_0 \end{pmatrix}$$

and

$$\begin{pmatrix} a_0 \\ b_0 \end{pmatrix} = \begin{pmatrix} \cos(\theta/2) \\ \sin(\theta/2) \end{pmatrix}. \quad \bullet$$

(3.15) Calculation of m. The representation of M in terms of a rotation enables us to specify the value of m, the number of iterations in Grover's algorithm, for known t. Assuming that t is much less than N, we iterate until the correct subspace amplitude vector is at an angle of approximately $\pi/2$, and it is easy to check that $m \approx \pi\sqrt{N/16t}$. For a discussion of how to specify m when the number of correct solutions is not known, see [16]. $\bullet$

Grover's algorithm illustrates the fact that a calculation may not succeed and may have to be repeated. Thus, in assessing how many times the function is evaluated, we have to take into account the number of times the entire algorithm has to be repeated.

(3.16) Work Factor. Suppose there are t correct solutions, and m is computed as in (3.15). Then the number of times f is evaluated is m times the expected number of repetitions of Grover's algorithm. The probability that a correct solution is found after applying Grover's algorithm once is $u = \sin^2(\alpha)$, where $\alpha = (2m+1)\theta$, so the expected number of repetitions R is

$$E(R) = \sum_{n=1}^{\infty} nu(1-u)^{n-1} = 1/u.$$

Since u can be bounded well away from 0, it follows that the number of calls of f is $O(m)$ and thus $O\left(\sqrt{N/t}\right)$. (See [9] and [16] for comments that Grover's algorithm is within a constant factor of the theoretical optimal solution.) •

It remains to demonstrate the feasibility of Grover's algorithm. Specifically, we have to show that each of the unitary matrices defined in the iterative step can be implemented in a (small) finite number of unitary operations involving a small number of qubits. We have already assumed that f can be so calculated via U_f, and we have seen how T can be implemented as part of the call of the function f.

D is defined as $-I + 2J/N$, and the claim is that $D = R^{(n)}SR^{(n)}$, where $R^{(n)}$ is the Hadamard transform, the n-fold tensor product of R, and S is $-I$ except that $S(0,0) = 1$. Since $R^{(n)}$ is its own inverse, the claim is equivalent to

$$SR^{(n)} = -R^{(n)} + \frac{2}{N}R^{(n)}J.$$

Since J is the all-ones matrix, the entries of $R^{(n)}J$ are zero everywhere except for the first row, and it is then obvious that the matrix on the right equals $-R^{(n)}$ except in the first row where the entries equal those of $R^{(n)}$. It is trivial to check that $SR^{(n)}$ achieves the same result.

As we saw in (3.2), $R^{(n)}$ is a tensor product of unitary matrices and can be implemented by operating on each of the n "domain" qubits in turn. S is also feasible, although the analysis is more involved. One way to realize S is to apply σ_x to each of the first n qubits and then use an $nXOR$ gate to change a utility qubit from $|0\rangle$ to $|1\rangle$ if and only if the system was originally in $|0,\ldots,0\rangle$. Using the utility qubit to conditionally multiply one of the domain qubits by -1 distinguishes the original all-zero state. A second application of $nXOR$ resets the utility qubit to its initial state, and another application of σ_x to each of the first n qubits restores them to their original state. We have achieved $-S$ with these steps so that an unconditional operation of $\begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}$ on one of the qubits will complete the proof that S is feasible.

(3.17) Exercise. Confirm that the description above does in fact implement S. If possible, find a more efficient implementation, perhaps by using the technique of Lemma (3.5). •

There is another way of viewing the effect of the iteration step in Grover's algorithm which sometimes goes under the rubric of "amplitude amplification." Instead of lumping the states together into the "correct"

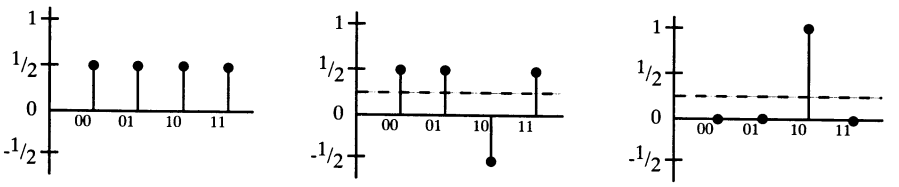
and "incorrect" subspaces, let $a_k(n)$ denote the (real) amplitude of state $|k\rangle$ after n iterations. For example, in the two-qubit case with $k_0 = 2$,

$$\begin{aligned}(k \neq 2) &\Rightarrow a_k(0) = 0.5 \xrightarrow{T} a'_k(0) = 0.5 \xrightarrow{D} 0.0 = a_k(1)\\ (k = 2) &\Rightarrow a_2(0) = 0.5 \xrightarrow{T} a'_2(0) = -0.5 \xrightarrow{D} 1.0 = a_2(1)\end{aligned}$$

and

$$\begin{aligned}(k \neq 2) &\Rightarrow a_k(1) = 0.0 \xrightarrow{T} a'_k(1) = 0.0 \xrightarrow{D} -0.5 = a_k(2)\\ (k = 2) &\Rightarrow a_2(1) = 1.0 \xrightarrow{T} a'_2(1) = -1.0 \xrightarrow{D} 0.5 = a_2(2).\end{aligned}$$

This can be viewed as a "reflection about the mean." After the T-mapping, compute the average of the amplitudes. Then the D-mapping reflects about that average as illustrated schematically below for the mapping $a_k(0) \rightarrow a_k(1)$.



(3.18) Amplitude Amplification. Confirm the generality of the preceding discussion for any n by showing that if $a_k(n) \xrightarrow{T} b_k(n)$, the D-mapping effects

$$b_k(n) \xrightarrow{D} \bar{b}(n) - \big(b_k(n) - \bar{b}(n)\big) = a_k(n+1)\,,$$

where $\bar{b}(n) \equiv (\sum_k b_k(n))/N$. •

3.4 Shor's algorithm: factoring $N = 15$

The potentially practical importance of quantum computers was first demonstrated by Shor (see [65]), who developed an algorithm for factoring a number N with a work factor which is polynomial in $\log(N)$ in contrast to the work factor of $\exp(c(\log(N))^{1/3}(\log(\log(N)))^{2/3})$ of the number field sieve approach [50]. Since the difficulty of factoring numbers N which are the products of two large primes is crucial to the security of some public key encryption schemes, there was enhanced interest in

the feasibility of quantum computers. In this section we present Shor's algorithm in a simple case that illustrates the underlying concepts. In the next section, we present and analyze the general algorithm. The discussion is based on the presentations in [65] and in Ekert and Jozsa [33], which is also a good introduction to the subject of quantum computation.

Before describing the algorithm, we review some elementary number theory. Let N denote the large, positive integer we wish to factor. (For the illustration, 15 will be defined as large.) Suppose we choose another positive integer y which is less than N and which, presumably, has $(y, N) = 1$, where (a, b) denotes the *greatest common divisor* of the integers a and b. (If $(y, N) > 1$, Euclid's algorithm can be used efficiently to find a factor of N, namely (y, N), and we don't need to use a quantum computer!) It is known that if $(y, N) = 1$, then there is a smallest positive integer $r \leq N - 1$ such that $y^r = 1 \bmod (N)$, and we define r as the *period* of y. For example, if $y = 13$, then modulo 15 the powers of y equal

$$13, 4, 7, 1, 13, 4, 7, 1, \ldots$$

and 13 has period $r = 4$ with respect to 15.

Now suppose that the period is even: $r = 2s$, so that $y^{2s} - 1 = 0 \bmod (N)$. Suppose further that $y^s \neq -1 \bmod (N)$. (We know that y^s can't equal $+1 \bmod (N)$ since s is half the period.) Then for some nonzero integer k,

$$(y^s + 1)(y^s - 1) = kN,$$

and we can use the Euclidean algorithm to find $(N, y^s \pm 1)$, obtaining nontrivial divisors of N. In our example, $13^2 - 1 = 168$, $13^2 + 1 = 170$, $(168, 15) = 3$, and $(170, 15) = 5$, so that we can find the factors of $N = 15$ using the period of 13. Thus, the problem of factoring N reduces to finding even periods $r = 2s$ for which the term $y^s + 1$ is not degenerate modulo N.

If N is very large, it is impractical to compute r classically, since that might require evaluating $O(N)$ powers of y. On a quantum computer, however, one could evaluate all of the powers of y simultaneously, and then the challenge would be one of adjusting the probability amplitudes to obtain the value of r with a reasonably high probability. Shor shows how to do this using two facts. First, the powers of $y \bmod (N)$ define a periodic function. Second, as in Simon's algorithm, the finite Fourier transform can transform the cyclic behavior of a periodic function into enhanced probability amplitudes of certain states.

We illustrate these basic ideas in the special case $N = 15$, introducing some of the general notation in the process. We keep track of two different

registers — one to represent the values of the exponents of y and the other to represent the values of $f(k) = y^k \bmod (N)$. A utility register of qubits is presumed, but not specified.

(3.19) Shor's Algorithm for $N = 15$

Step 1: Choose $n = 4$ so that $15 \leq S \equiv 2^4 = 16$. Choose y so that $(y, 15) = 1$. For example, if $y = 13$, $(13, 15) = 1$.

Step 2: Initialize two four-qubit registers to state 0: $|\psi_0\rangle = |0\rangle|0\rangle$.

Step 3: Randomize the first register:

$$|\psi_0\rangle \rightarrow |\psi_1\rangle = \sum_{k=0}^{15} a_k |k\rangle|0\rangle,$$

where $a_k = 1/\sqrt{16}$.

Step 4: Unitarily compute the function $f(k) = 13^k \bmod (15)$:

$$|\psi_1\rangle \rightarrow |\psi_2\rangle = \sum_{k=0}^{15} a_k |k\rangle|f(k)\rangle.$$

Step 5: Operate on the first four qubits by the finite Fourier transform $F = F_{16}$. (See the discussion below.) That operation gives

$$|k\rangle \rightarrow \frac{1}{\sqrt{16}} \sum_{u=0}^{15} \exp(2\pi i u k/16)|u\rangle$$

and is implementable unitarily. Thus,

$$|\psi_2\rangle \rightarrow |\psi_3\rangle = \frac{1}{16} \sum_{u=0}^{15} |u\rangle \sum_{k=0}^{15} \exp(2\pi i u k/16)|f(k)\rangle.$$

Step 6: Since 13 has (unknown) period r, the function f is periodic. In this case the period happens to divide 16, so that we can write $k = m + jr$, where $0 \leq m < r$ and $0 \leq j < 16/r$, and $f(k)$ can be written as $f(m)$. Then the new state can be expressed as

$$|\psi_3\rangle = \frac{1}{16} \sum_{u=0}^{15} |u\rangle \sum_{m=0}^{r-1} |f(m)\rangle \exp(2\pi i u m/16) \sum_{j=0}^{(16/r)-1} \exp(2\pi i u r j/16).$$

Calculating the summation over j and including the lead factor of $1/16$, we obtain finally

$$|\psi_3\rangle = \sum_{u=0}^{15} |u\rangle b_u \sum_{m=0}^{r-1} |f(m)\rangle \exp(2\pi i u m/16),$$

where $b_u = 1/r$ if 16 divides ur, or $16|ur$, and is zero otherwise.

Step 7: Measure the state of the first register. If P_u denotes the projection mapping onto the state denoting the value u, then, as defined in (1.21), the probability of observing u is

$$\langle\psi_3|P_u|\psi_3\rangle = r|b_u|^2 = \begin{cases} \frac{1}{r} & \text{if } 16|ur \\ 0 & \text{otherwise.} \end{cases}$$

Step 8: Use the known value of u and the known value of 16 to deduce a putative value of r:

a. If no inference can be made or if r is odd, return to *Step 2* and repeat.

b. If $r = 2s$ is even, and $y^s = -1 \bmod (15)$, return to *Step 2* and repeat.

c. If $(15, y^s \pm 1) > 1$, quit; otherwise, return to *Step 2* and repeat. •

Let's see what occurs in our special case. In this case $r = 4$, r divides 16 and the only u's with a positive probability of being observed are multiples of 4: $u = 0$, 4, 8, and 12, each one occurring at *Step 7* with probability 0.25. Since $ur = 16k$, for some integer k, we can infer nothing about r if $u = 0$. The remaining cases give equations for r of the form $r = 4k$, $r = 2k$ and $3r = 4k$. In two of those cases the presumption that r and k are relatively prime is correct, leading to the putative value of $r = 4$ for the period. We know that value to be correct and that 15 has to divide the product of $13^2 - 1$ and $13^2 + 1$, enabling us to find both of its factors.

In the remaining case the presumption would lead to $r = 2$, giving values of 12 and 14 for $y^s \pm 1$. Because the numbers are so small, we happen to discover a factor of 15 in this case, since one of three consecutive numbers is divisible by 3. However, $r = 2$ isn't the period, and in general we're not guaranteed to obtain a factor of N if we get the wrong value of r. Hence, we can say that we know we succeed in factoring $N = 15$ after only one run through the procedure with a probability of 0.5. Following the same reasoning as in (3.16), the work factor is thus proportional to 2, the expected number of calls of the procedure.

3.5 Shor's algorithm: factoring $N = pq$

Shor's algorithm works for odd N that are not powers of primes. However, to simplify the presentation a bit and since the theory extends in a straightforward manner to the general case, we will assume $N = pq$, the product of two odd primes, and the letters p and q will denote those primes henceforth.

There are three difficulties in Shor's algorithm. First of all, the y that we choose may have an odd period or else the period r may equal $2s$ but $y^s = -1 \bmod (N)$. For example, if we had chosen $y = 14$ above, we would have found $r = 2$ and $y^1 = -1 \bmod (15)$. (As we will see below, for $N = 15$, the period in that case always divides 8, and so is odd only for $y = 1$.) Second, it is most likely that r does not divide 2^n, where n denotes the register length, and we must adjust the algorithm accordingly. Third, it may happen that y has a suitable period r but we cannot deduce the true value of r from the measurement.

We begin the discussion of the first difficulty in the case $N = pq$ by reviewing the relevant number theory. The well-informed and/or impatient reader can skip to the statement of Lemma (3.25) for the key result. For omitted details, see, for example, Hardy and Wright [42].

(3.20) Exercise. Let (p, q) denote the greatest common divisor of the integers p and q. Then show that if $(p, q) = 1$, there exist integers r and s such that $1 = rp + sq$. (*Hint*: show that the minimum element in $\{m : 0 < m = ip + jq\}$ is precisely (p, q).) •

(3.21) Exercise. (*Chinese remainder theorem*) Suppose $(p, q) = 1$ and let $0 \leq a < p$ and $0 \leq b < q$. Show that there is a unique x, $0 \leq x < pq$, such that $x = a \bmod (p)$ and $x = b \bmod (q)$. (*Hint*: choose n so that $0 \leq x = brp + asq + npq < pq$.) •

The effect of (3.21) is that we can uniquely specify an $x \bmod (pq)$ by specifying its residues $\bmod(p)$ and $\bmod(q)$. Put another way, if $\{x : 0 \leq x < pq\}$ is given a uniform distribution, then its $\bmod(p)$ and $\bmod(q)$ residues can be treated as independent random variables.

(3.22) Definition and Properties of ϕ. The *Euler ϕ function*, or the *totient function*, is defined as follows: $\phi(n)$ equals the number of y, $0 < y < n$, such that $(y, n) = 1$. Thus, $\phi(p) = p - 1$ for p prime, and it is easy to check that $\phi(p^k) = p^{k-1}(p - 1)$ and $\phi(pq) = (p - 1)(q - 1)$ if p and q are different primes. Using those results, it can be shown that ϕ is multiplicative: if $(m, n) = 1$, $\phi(mn) = \phi(m)\phi(n)$ for positive integers m and n. The *Euler–Fermat theorem* states that if $(y, n) = 1$, then $y^{\phi(n)} = 1 \bmod(n)$. In addition, if p is prime, then $\{z^k \bmod (p), 1 \leq k < p\}$ equals $\{a : 1 \leq a < p\}$ for some z. (See [42], Theorem 111.) •

(3.23) Definition. Suppose $(y, n) = 1$. Then the *period* of y with respect to n is the smallest positive integer r such that $y^r = 1 \bmod (n)$. It follows that if $f(a)$ is defined as $y^a \bmod (n)$, then f is a periodic function with period r. •

(3.24) Exercise. (a) If $1 \leq y < n$, then show that r, the period of y with respect to n, divides $\phi(n)$.

(b) Suppose $n = pq$, p and q prime, and s and t denote the periods of y with respect to p and to q, respectively. Then $r = \text{lcm}(s,t)$, the *least common multiple* of s and t. •

For example, if $(y, 15) = 1$, the period of y with respect to 15 has to be a power of 2, since $\phi(15) = 8$. We saw above that $f(a) = 13^a \bmod (15)$ has period 4, and it is easy to check that 13 has period 2 with respect to 3 and period 4 with respect to 5.

We are now ready to compute the proportion of y's whose periods are suitable for a successful completion of Shor's algorithm.

(3.25) Lemma. Suppose $N = pq$ and $S = \{y : 1 \leq y < N, (y, N) = 1\}$. Then at least $1/2$ of the integers y in S have even period $2k$ and satisfy $y^k \neq -1 \bmod (N)$.

Proof: Suppose y is in S and has period $r = \text{lcm}(s,t)$, where s and t are the periods of y with respect to p and to q, respectively. If we write $s = 2^i u$ and $t = 2^j v$, where u and v are odd integers, then $r = 2^{\max(i,j)}\text{lcm}(u,v)$, and r will be odd if and only if both i and j are zero.

Suppose $r = 2k$ is even. From (3.21), $y^k = -1 \bmod (N)$ if and only if $y^k = -1 \bmod (p)$ and $y^k = -1 \bmod (q)$. But if (say) $i < j$, then k is a *multiple* of s, and $y^k = 1 \bmod (p)$, a contradiction. A similar result holds if $j < i$, and thus it is easy to show that $y^k = -1 \bmod (N)$ if and only if the periods s and t have the same number of factors of 2.

Let $p - 1 = 2^m x$ with x odd. From the last sentence in (3.22) we know that the nonzero integers mod(p) are generated by the $p - 1$ powers of some z with $(z, p) = 1$. It follows that an integer b will have an odd period with respect to p if and only if b equals z to a power $2^m w$, $1 \leq w \leq x$, and the proportion of such integers is therefore 2^{-m}. Furthermore, precisely those b's equal to z to a power $2^{m-k}w$, w odd, will have a period with exactly k powers of 2. It follows that w can take odd values from 1 through $2^k x - 1$, and there are precisely $2^{k-1}x$ such integers. That is, the proportion of integers in S whose period has exactly k powers of 2 is 2^{k-1-m}. For example, if $p = 7$, then $m = 1$ and half of the integers have period 2 or 6 and half have period 1 or 3.

Suppose then that $p-1 = 2^m x$ and $q-1 = 2^n w$, where $1 \leq m \leq n$ and x and w are odd. Using the probability interpretation of Exercise (3.21) we see that the proportion of integers in S which have an odd period or

which have even period $2u$ with $y^u = -1 \bmod (N)$ is

$$\left(\frac{1}{2}\right)^{m+n} + \sum_{k=1}^{m} \left(\frac{1}{2}\right)^{m+n-2k+2} = \left(\frac{1}{2}\right)^{m+n} \left(1 + \sum_{j=0}^{m-1} 4^j\right) \leq \frac{1}{2},$$

and the assertion follows from this inequality. •

(3.26) Exercise. Show that the bound $1/2$ is exact if $N = 77$. If $N = 119$, show that only $1, 18, 33, 86, 101$, and 118 fail to have even period with $y^{r/2} \neq -1 \bmod N$. •

We thus see that at least half of the y's that might be chosen will have a suitable period, and we turn to the second difficulty - that the period of y does not divide 2^n, where n is the number of qubits in the first register. To handle this case Shor requires that n be chosen so that 2^n is significantly larger than N, thus requiring more "domain" qubits than might be strictly necessary in special cases. Here's the general algorithm, which is a modest modification of the algorithm for $N = 15$.

(3.27) Shor's Algorithm for Odd $N = pq$.

Step 1: Choose n so that $N^2 \leq S = 2^n < 2N^2$. Choose y so that $(y, N) = 1$.

Step 2: Initialize two n-qubit registers to state 0: $|\psi_0\rangle = |0\rangle|0\rangle$.

Step 3: Randomize the first n "domain" qubits:

$$|\psi_0\rangle \to |\psi_1\rangle = \sum_{k=0}^{S-1} \frac{1}{\sqrt{S}} |k\rangle|0\rangle.$$

Step 4: Evaluate the function $f(k) \equiv y^k \bmod (N)$:

$$|\psi_1\rangle \to |\psi_2\rangle = \sum_{k=0}^{S-1} \frac{1}{\sqrt{S}} |k\rangle|f(k)\rangle.$$

Step 5: Transform the first n qubits using the finite Fourier transform $F = F_S$. That operation maps

$$|k\rangle \to \frac{1}{\sqrt{S}} \sum_{u=0}^{S-1} \exp(2\pi i u k/S)|u\rangle$$

and is implementable unitarily and feasibly, as we will see later:

$$|\psi_2\rangle \to |\psi_3\rangle = \frac{1}{S} \sum_{u=0}^{S-1} |u\rangle \sum_{k=0}^{S-1} |f(k)\rangle \exp(2\pi i u k/S).$$

Step 6: Since y has (unknown) period r, the function f is periodic with period r. Because the period need not divide S, we write $k = m + jr$, where $0 \leq m < r$ and $0 \leq j < A$, with A equaling $\lceil \frac{S}{r} \rceil$, the smallest integer bigger than or equal to S/r. Again, $f(k)$ equals $f(m)$. Then the new state can be expressed as

$$|\psi_3\rangle = \frac{1}{S} \sum_{u=0}^{S-1} |u\rangle \sum_{m=0}^{r-1} |f(m)\rangle \exp(2\pi i u m/S) \sum_{j=0}^{A-1} \exp(2\pi i u r j/S) I_{(m+rj<S)},$$

where the last term indicates that the summation doesn't exceed $S-1$. Including the lead factor in the summation over j, we obtain

$$|\psi_3\rangle = \sum_{u=0}^{S-1} |u\rangle \sum_{m=0}^{r-1} |f(m)\rangle \exp(2\pi i u m/S) b_{u,m},$$

where

$$b_{u,m} = \frac{1}{S} \sum_{j=0}^{A-1} \exp(2\pi i u r j/S) I_{(m+rj<S)}.$$

If $A = S/r$, then $I_{(m+rj<S)} = 1$ for all m and j in the summations. It is easy to confirm that if $S = (A-1)r + k$ with $0 < k \leq r-1$, then $I_{(m+rj<S)} = 1$ unless $j = A-1$ and $k \leq m \leq r-1$. Ignoring the small error when the j-summation excludes an $A-1$ term, we take as the final state:

$$|\psi_3\rangle = \sum_{u=0}^{S-1} \sum_{m=0}^{r-1} b_u \exp(2\pi i u m/S) |u\rangle |f(m)\rangle,$$

where

$$b_u = \frac{1}{S} \frac{1 - \exp(2\pi i u r A/S)}{1 - \exp(2\pi i u r/S)} \approx b_{u,m},$$

using the limiting value if S divides ur.

Step 7: Measure the first register, obtaining as before a value u with probability

$$\langle \psi_3 | P_u | \psi_3 \rangle = r b_u^2.$$

Step 8: Interpret the known value of u vis-a-vis the known value of S to deduce a putative value of r.

a. If no inference can be made or if r is odd, return to *Step 2* and repeat.

b. If $r = 2s$ is even, and $y^s = -1 \bmod (N)$, return to *Step 2* and repeat.

c. Use the Euclidean algorithm to compute $(N, y^s \pm 1)$. If the result is bigger than 1, quit. Otherwise, return to *Step 2* and repeat. •

The work factor in computing the function $f(k) \equiv y^k \bmod (N)$ and in taking the finite Fourier transform can be shown to be polynomial in $n = \log(N)$, the number of bits of N. (See [76] for the former and the comments after Proposition (3.31) for the latter.) The estimated number of repetitions will be shown to be polynomial in $(\log(N))$, confirming the advantage of this approach to the number field sieve algorithm.

We begin the analysis of the output of the algorithm by examining values of u for which there is an integer k such that

$$-\frac{r}{2} \leq ur - kS \leq \frac{r}{2}. \tag{3.28}$$

Since S/r is approximately A and u is bounded above by S, there are approximately r such u's corresponding more or less to multiples of A. (If r divides S, these are exact values.) For such a u, we calculate

$$rb_u^2 = \frac{r}{S^2}\left(\frac{1-\cos(2\pi urA/S)}{1-\cos(2\pi ur/S)}\right) = \frac{r}{S^2}\left(\frac{\sin(\pi urA/S)}{\sin(\pi ur/S)}\right)^2$$

again using limiting values if necessary. In Exercise (3.30) we approximate the right-hand side and find a lower bound essentially of the form

$$rb_u^2 \geq \frac{1}{r}\frac{4}{\pi^2}\left(1-\frac{2}{N}\right) \geq \frac{0.4}{r} \tag{3.29}$$

for large N. Thus, since there are approximately r such u's satisfying (3.28), with a probability greater than 0.4 the measurement will result in one of those u's.

In the case when r divides S exactly, as we saw for $N = 15$, we have $ur = kS$ for such a u, so that u/S in lowest terms equals k/r. In the general case the inequality (3.28) can be written as

$$\left|\frac{u}{S} - \frac{k}{r}\right| \leq \frac{1}{2S}$$

and interpreted as the *known* fraction u/S being approximated to within $1/(2S)$ by the *unknown* fraction k/r with denominator less than N. Since

$1/(2S) < 1/(2N^2)$, there can be at most one such approximation and that can be found in polynomial time using a continued fraction expansion of the known fraction u/S. (See, for example, Hardy and Wright, [42], Chapter X and Knuth [48].) We have thus dealt with the second difficulty, that is, when the period doesn't divide S.

(3.30) Exercise. (*Verification of (3.29)*) The restrictions of (3.28) are sufficient for the lower bound in (3.29). By assumption, we can write $ur/S = k + t$, where $|t| \leq 1/2$. Defining $g(y) = \sin(y)/y$, we have

$$\sin^2(\pi ur/S) = \sin^2(\pi t) = (\pi t)^2 g^2(\pi t) \leq (\pi t)^2,$$

since $|g|$ is bounded by 1. Next observe that

$$-\frac{Ar}{2S} \leq Aur/S - Ak = At \leq \frac{Ar}{2S},$$

and since $A - 1 < S/r \leq A$ implies $1 \leq Ar/S < A/(A-1)$, it follows that

$$-\frac{\pi}{2}\left(1 + \frac{1}{A-1}\right) < \pi At < \frac{\pi}{2}\left(1 + \frac{1}{A-1}\right).$$

Since $r < N$ and $N^2 \leq S$, $N < S/r \leq A$, and it follows that

$$-\frac{\pi}{2}\left(1 + \frac{1}{N-1}\right) < \pi At < \frac{\pi}{2}\left(1 + \frac{1}{N-1}\right).$$

It follows that $|\pi At|$ is in $[0, \pi]$, where the function $g(y) = \sin(y)/y$ is decreasing. Hence,

$$\sin^2(\pi urA/S) = (\pi At)^2 g^2(\pi At) \geq (\pi At)^2 g^2\left(\frac{\pi}{2}\left(1 + \frac{1}{N-1}\right)\right)$$

and

$$rb_u^2 \geq \frac{rA^2}{S^2}\frac{4}{\pi^2}\left(\frac{N-1}{N}\right)^2 \sin^2\left(\frac{\pi}{2}\left(1 + \frac{1}{N-1}\right)\right).$$

We leave the final steps as an exercise: show that

$$rb_u^2 \geq \frac{1}{r}\frac{4}{\pi^2}\left(1 - \frac{2}{N} - \frac{1}{N^2}\left(\frac{\pi^2}{8} - 1\right)\right). \quad \bullet$$

Now for the third difficulty: computing r from this information. If k is relatively prime with respect to r, then the denominator computed from

u/S by continued fractions equals r; otherwise, that denominator is a factor of r. Hence, assuming a uniform distribution on the value of k, the probability that we correctly read off r from u/S is the probability that k is relatively prime with respect to r, and that is $\phi(r)/r$. Again quoting Hardy and Wright [42], (Theorem 328), we can obtain $\phi(r)/r \geq \delta/\log(\log(r))$ for some positive δ, so that the probability of measuring a "usable" u and correctly calculating r from that measurement is at least $0.4\delta/\log(\log(r)) \geq 0.4\delta/\log(\log(N))$. Thus, repeating the procedure $O(\log(\log(N)))$ times will give a high probability of success, *provided* we have chosen a y whose period satisfies the factoring requirements. Since that occurs at least half the time, we can repeat the foregoing with different y's, if necessary, maintaining the polynomial time result.

In [65], the author discusses variations on the theme that he and others have suggested since the algorithm was first published. For example, if the denominator computed from u/S is not r, it is a factor of r, and one could use multiples of that number for the period. Or, if two putative values of r have been computed, the least common multiple could be tried. In addition, it has been noticed that the physical implementation might be simplified by making partial measurements during the finite Fourier transform, and we discuss that briefly in the next section. Finally, a different perspective of the algorithm gives an alternative motivation for the actual steps, and that is discussed in Sections 3.7 and 3.8 below.

3.6 The finite Fourier transform

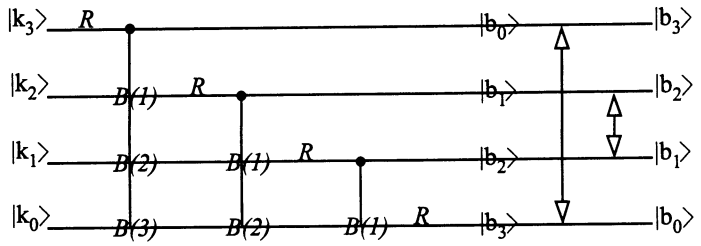
The practicality of Shor's factoring algorithm depends on the feasibility of implementing the finite Fourier transform, and we concentrate on that issue next. The quantum implementation is motivated by the classical fast Fourier transform, and we elaborate on the presentation in [33], illustrating the concept by giving an explicit wiring diagram for four qubits. Ekert and Jozsa [33] attribute this particular algorithm to Coppersmith [24] and Deutsch. (See [33], p. 741 for additional references and comments and [22] for an independent analysis of the finite Fourier transform similar to the one presented here.)

Let R denote the now-familiar rotation of (2.7) which acts on one qubit. Let $B(k)$, $0 \leq k < n$, denote an operator acting on two qubits with the following matrix representation in the computational basis:

$$B(k) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \beta_k \end{pmatrix},$$

where $\beta_k = \exp(i\pi 2^{-k})$. Thus $B(k)$ represents the controlled phase operation $|a\rangle|b\rangle \rightarrow \exp(i\pi ab2^{-k})|a\rangle|b\rangle$.

Now consider the following wiring diagram for four qubits:



where the implementation of the $B(k)$'s is controlled by the indicated qubit. As in the case of the Hadamard transform, the initial state is $|k\rangle = |k_3\rangle|k_2\rangle|k_1\rangle|k_0\rangle$, where $k = 2^3k_3 + 2^2k_2 + 2k_1 + k_0$, so that the high-order bit in the wiring diagram is on the top, and the tensor product notation is implicit.

The first application of R and the first set of controlled phase changes yield

$$\begin{aligned} |k\rangle &\rightarrow \frac{1}{\sqrt{2}}(|0\rangle + \alpha(0,k)|1\rangle)|k_2\rangle|k_1\rangle|k_0\rangle \\ &= \frac{1}{\sqrt{2}}\sum_{b_0=0}^{1}\exp\left(\frac{2\pi i k b_0}{2^4}\right)|b_0\rangle|k_2\rangle|k_1\rangle|k_0\rangle, \end{aligned}$$

where

$$\alpha(0,k) = \exp\left(i\pi\left(k_3 + \frac{k_2}{2} + \frac{k_1}{2^2} + \frac{k_0}{2^3}\right)\right) = \exp(2\pi i k/2^4).$$

Notice that after these two mappings, the basis state $|k\rangle$, which is a tensor product, has been transformed into another tensor product, a fact that is clarified by the notation on the left and obscured by the notation on the right. From the recursive structure of the wiring diagram up to the switching operations, we can conclude that the final state will also be a tensor product.

At the second stage, we rotate the second qubit and apply the conditional phase shifts, obtaining the superposition $\frac{1}{\sqrt{2}}(|0\rangle + \alpha(1,k)|1\rangle)$ in the second qubit, where

$$\alpha(1,k) = \exp\left(i\pi\left(k_2 + \frac{k_1}{2} + \frac{k_0}{2^2}\right)\right) = \exp\left(2\pi i \cdot 2k/2^4\right).$$

Notice that the high-order bit k_3 enters with a factor of $2\pi i$ and thus doesn't affect the value of the constant. Letting b_1 denote the new index for the states of the rotated second qubit, we have, after the second set of operations,

$$\frac{1}{\sqrt{2^2}} \sum_{b_0=0}^{1} \sum_{b_1=0}^{1} \exp\left(\frac{2\pi i k}{2^4}(b_0 + 2b_1)\right) |b_0\rangle|b_1\rangle|k_1\rangle|k_0\rangle,$$

and the pattern is clear. At the end of the four rotations, the original state has been transformed into the state

$$\frac{1}{\sqrt{2^4}} \sum_{b_0=0}^{1} \sum_{b_1=0}^{1} \sum_{b_2=0}^{1} \sum_{b_3=0}^{1} \exp\left(\frac{2\pi i k}{2^4}(b_0 + 2b_1 + 4b_2 + 8b_3)\right) |c\rangle\,,$$

where $|c\rangle = |b_0\rangle|b_1\rangle|b_2\rangle|b_3\rangle$.

This is almost the finite Fourier transform F_{16} of $|k\rangle$; the only problem is that the high-order bit is in the bottom position instead of the top position. To correct that, we perform two swap operations, denoted by the arrows and using the logic of Example (2.12), and obtain the final form of the transform:

$$|k\rangle \to \frac{1}{\sqrt{2^4}} \sum_{b=0}^{2^4-1} \exp(2\pi i k b/2^4)|b\rangle\,.$$

The generalization of the algorithm and the wiring diagram to arbitrary n is obvious, and we record the final result without a formal proof.

(3.31) Proposition. Given n qubits in the state $|k\rangle = |k_{n-1}\rangle \ldots |k_0\rangle$, we define the finite Fourier transform F_{2^n} on basis states by

$$F_{2^n}|k\rangle \to \frac{1}{\sqrt{2^n}} \sum_{b=0}^{2^n-1} \exp(2\pi i k b/2^n)|b\rangle. \qquad \bullet$$

It is easy to see that F_{2^n} can be effected by n applications of R to single qubits, $n(n-1)/2$ controlled phase operations involving two qubits each and $[n/2]$ two-qubit swaps each requiring three *XORs*. Thus, the work factor of implementing the finite Fourier transform is polynomial in n. Moreover, F_{2^n} maps tensor product basis states to tensor products.

A close look at the factoring algorithm shows that there was no measurement of the qubits containing the value of the exponential function

whose role was, in effect, to transform the probability amplitudes. Furthermore, the only actual measurement of the first register that was made occurred just after the implementation of the finite Fourier transform. Now we have observed that once the rotation and the first set of conditional phase changes involving the top qubit are made, that qubit is not involved in subsequent operations until the swap operation. If we delete the swap from the algorithm, then the next time the top qubit is affected occurs when the measurement is made.

Griffiths and Niu [40] observed that if one measured the domain qubits sequentially during the finite Fourier transform, rather than waiting until the transform was completed to measure them at the same time, one could implement the phase mappings as one-qubit operations whose implementation is based on a measurement, rather than as two-qubit unitary operations. The practicality of this approach remains to be seen, but it does offer the potential of shifting some of the technical problems to the classical domain.

We illustrate the idea using four domain qubits as in Section 3.4 and explicitly expressing the dependence of the function on the binary expansion of the integers k. Suppose we have computed f and are beginning the finite Fourier transform. We first apply the rotation R to the top qubit and, ignoring normalizing factors, obtain the superposition of

$$|0\rangle \left(\sum |k_2\rangle |k_1\rangle |k_0\rangle |f(0, k_2, k_1, k_0)\rangle + \sum |k_2\rangle\, |k_1\rangle |k_0\rangle |f(1, k_2, k_1, k_0)\rangle \right)$$

and

$$|1\rangle \left(\sum |k_2\rangle |k_1\rangle |k_0\rangle |f(0, k_2, k_1, k_0)\rangle - \sum |k_2\rangle |k_1\rangle\, |k_0\rangle\, |f(1, k_2, k_1, k_0)\rangle \right).$$

Next measure the high-order qubit and, according to the wiring diagram, effect the phase changes as one-qubit operations on the three low-order qubits, conditional on the results of the measurement. We thus obtain for the state of the system either the parenthetical factor in the first expression unchanged by any phase factors or else the corresponding factor in the second expression, where each term is modified by the phase change $\exp\left(i\pi\left(\frac{k_2}{2} + \frac{k_1}{2^2} + \frac{k_0}{2^3}\right)\right)$. In either case the state of the system can be written as the normalization of

$$\sum_k |k_2\rangle |k_1\rangle |k_0\rangle \sum_{j=0}^{1} \exp\left(\frac{2\pi i k b_0}{2^4}\right) f(j, k_2, k_1, k_0)\,,$$

where b_0 is the known state of the high-order qubit. (Compare this with the expression for $|\psi_3\rangle$ in *Step 5* of Shor's algorithm for $N = 15$.) Continuing this same approach on the next three qubits, we will ultimately

find a four-tuple $b_0b_1b_2b_3$ with the same probability as if we had made all four measurements at the end of the transform.

There is another consideration to be noted, and that is the practicality of actually performing controlled small-phase operations. This particular point is addressed by Coppersmith [24] in his definition of an approximate finite Fourier transform, and he shows that for the purpose of the factoring algorithm, one can omit implementing *B(k)* for sufficiently large k. Again, the wiring diagram suggests a way of calculating the effect of ignoring small phase changes, and we use the four-qubit example to illustrate the methodology.

Let F denote the complete transform and suppose we didn't implement the high-order phase change $\exp(\pi i/8)$. Let G denote the finite Fourier transform without that factor. Then $F|k\rangle$ is the tensor product of terms of the form $\frac{1}{\sqrt{2}}(|0\rangle + \alpha(j,k)|1\rangle)$, where

$$\alpha(j,k) = \exp(2\pi i k 2^{j-4}),$$

and $G|k\rangle$ is a tensor product of similar terms $\frac{1}{\sqrt{2}}(|0\rangle + \beta(j,k)|1\rangle)$, where

$$\beta(0,k) = \alpha(0,k)\exp\left(-\pi i k_0/2^3\right)$$

and $\beta = \alpha$ otherwise. Then $(F-G)|k\rangle$ can be written as

$$\frac{1}{\sqrt{2}}\left(1-\exp\left(-\pi i k_0/2^3\right)\right)|1\rangle \otimes_{j=1}^{3}\left(\frac{1}{\sqrt{2}}(|0\rangle + \alpha(j,k)|1\rangle)\right),$$

and it follows that

$$\langle k|(F-G)^2|k\rangle = \frac{\left|1-\exp\left(-\pi i k_0/2^3\right)\right|^2}{2} = 2\left(\sin^2\left(\pi k_0/2^4\right)\right) \leq \pi^2 2^{-7}.$$

We can abstract this approach to apply to the case of n qubits and the omission of the r smallest phase changes, obtaining as an upper bound on the L^2 error a sum of terms similar to the square root of the expression above. Since no new concepts are involved, we forego the exercise.

3.7 Eigenvalues in quantum algorithms

There is a common structure in at least part of each of the algorithms presented so far: apply a finite Fourier transform to a "domain" set of n qubits, compute a given function and thus entangle the domain states, apply a second finite Fourier transform to the domain qubits and then

make a measurement. In [46] Kitaev gave an approach to a general class of problems, which includes the factoring problem, using the estimation of certain eigenvalues as a central theme. As nicely presented by Cleve et al. [22], the use of eigenvalues of unitary operators depending on given functions gives a revealing perspective of the common structure of quantum algorithms and also relates it to interferometry experiments, so that quantum algorithms could be considered multiple-particle interferometry. In this section we present some of the results in [22] and recommend the entire paper to the reader.

We have already noted this perspective in Lemma (3.5). Recall that we could replace several steps in the Deutsch-Jozsa algorithm with

$$\begin{aligned}(|0\rangle - |1\rangle)/\sqrt{2} &\rightarrow \frac{1}{\sqrt{2}}\left(|0 \oplus f(j)\rangle - |1 \oplus f(j)\rangle\right) \\ &= (-1)^{f(j)}(|0\rangle - |1\rangle)/\sqrt{2}.\end{aligned}$$

Now $f(j)$ has to be computed by a unitary transformation $U_{f(j)}$, so that we could represent the shortcut by

$$U_{f(j)}\,(|0\rangle - |1\rangle) = (-1)^{f(j)}(|0\rangle - |1\rangle)\,. \tag{3.32}$$

In other words, $|0\rangle - |1\rangle$ is an eigenvector for the unitary mapping $U_{f(j)}$ and the factor $(-1)^{f(j)}$ used in the calculation is the corresponding eigenvalue. We used the same trick in the iterative step of Grover's algorithm.

The Deutsch–Jozsa algorithm showed the existence of a better-than-classical algorithm, but seemed a little contrived since the outcome only identified the class to which the unknown function belonged. However, one can actually identify the function if additional structure is assumed, and (3.32) is again an integral part of the algorithm.

(3.33) Example. (Bernstein and Vazirani [13]) Let F_2^n denote the vector space of n-long binary vectors over $Z_2 = \{0, 1\}$ and suppose $f : F_2^n \rightarrow Z_2$ is defined by $f(x) = (a \cdot x) \oplus b$, where the unknown a is in F_2^n, the unknown b is in Z_2 and the notation denotes a mod(2) dot product of the vectors and mod(2) scalar addition as usual. Then the vector a can be computed in one pass of the Deutsch–Jozsa algorithm. (We ignore normalizing factors throughout.)

$$\begin{aligned}|0\ldots0\rangle(|0\rangle - |1\rangle) &\rightarrow \sum_k |k\rangle(|0\rangle - |1\rangle) \rightarrow \sum_k |k\rangle(-1)^{(a\cdot k)\oplus b}(|0\rangle - |1\rangle) \\ &\rightarrow (-1)^b \sum_u |u\rangle \sum_k (-1)^{(a\oplus u)\cdot k}(|0\rangle - |1\rangle) \\ &= (-1)^b 2^n |a\rangle(|0\rangle - |1\rangle),\end{aligned}$$

where the Hadamard transform $R^{(n)}$ is applied at the first and third mappings and the unitary mapping implementing f is applied in between. A measurement of the first n qubits then returns a with probability 1. •

There is an immediate generalization to higher dimensions which uses the same eigenvector trick in a somewhat more involved fashion.

(3.34) Example. Let A be an unknown $m \times n$ binary matrix and b a binary m-vector. Let f be the affine map from F_2^n to F_2^m defined by $f(x) = Ax \oplus b$, where this time the addition is mod(2) addition of binary m-vectors and the matrix multiplication is also mod(2). Assuming f is computable, the problem is to determine A using fewer calls of f than is possible classically. This is an extension of (3.33), and m calls of f are necessary. The approach is to compute m different n-vectors of the form $c^{(s)}A$, for a suitable set of m-long row vectors $c^{(s)}$. This can be done by initializing m utility qubits with the fill $|c_1 c_2 \dots c_m\rangle$, where we drop the superscript, and by employing an additional Hadamard transform at the start of the algorithm. Note that we use the fact that the Hadamard transform of $|c_1 c_2 \dots c_m\rangle$ is also a tensor product:

$$|0 \dots 0\rangle|c_1 c_2 \dots c_m\rangle \to \sum_k |k\rangle((|0\rangle + (-1)^{c_1}|1\rangle) \otimes \cdots \otimes (|0\rangle + (-1)^{c_m}|1\rangle)).$$

For a given k, $f(k) = (Ak) \oplus b = (f_1, \dots, f_m)$ is an m-vector, and we can store each component from the action of $U_{f(k)}$ using the corresponding eigenvector:

$$|0 \oplus f_r\rangle + (-1)^{c_r}|1 \oplus f_r\rangle = (-1)^{f_r c_r}(|0\rangle + (-1)^{c_r}|1\rangle).$$

Combining all of the phase factors gives the final factor of $(-1)^{c\cdot(Ak\oplus b)}$.

Applying a second Hadamard transform to the first n qubits gives each state $|u\rangle$ a numerical coefficient of $(-1)^{c\cdot b}\sum_k(-1)^{(cA\oplus u)\cdot k}$, which is zero unless $u = cA$. Thus, a subsequent measurement of the first n states gives cA with probability 1. •

Example (3.34) could have been solved using specific vectors c, but the virtue of the generality is that we have illustrated the construction of an *eigenstate* $|\psi_c\rangle$ with *eigenvalue* $(-1)^{c\cdot(Ak\oplus b)}$ relative to the unitary operator $U_{f(k)}$ which implements the mapping f from F_2^n to F_2^m. Moreover, the methodology applies more generally.

(3.35) Exercise. Assume f maps F_2^n to F_2^m. Let U_f denote a unitary map which computes and stores each bit of the value of $f(k)$ as in (3.34). Then show that for each m-vector c there is a ket $|\psi_c\rangle$ such that $U_{f(k)}|\psi_c\rangle = (-1)^{c\cdot f(k)}|\psi_c\rangle$. •

(3.36) Example. (*Simon's problem*) Suppose f maps F_2^n to F_2^m in such a way that $f(a) = f(a+k)$ for every k in a subgroup K of order 2^{n-m} in F_2^n; that is, f is constant on cosets of K. Find K with a minimal number of evaluations of f. (This is a generalization of Simon's problem analyzed earlier.) Taking the context and notation of (3.34) and (3.35), we begin by following the same approach:

$$|0\ldots0\rangle|c_1c_2\ldots c_m\rangle \to \sum_a |a\rangle|\psi_c\rangle \to \sum_a |a\rangle(-1)^{c\cdot f(a)}|\psi_c\rangle.$$

Taking the Hadamard transform as usual and rearranging terms gives

$$\sum_u |u\rangle|\psi_c\rangle \sum_a (-1)^{u\cdot a}(-1)^{c\cdot f(a)} = \sum_u |u\rangle|\psi_c\rangle \sum_y (-1)^{u\cdot y \oplus c\cdot f(y)} \sum_k (-1)^{k\cdot u}$$

with the k-summation over K and the y-summation over the cosets of K. It follows that a measurement on the domain qubits gives only those u that *annihilate* K: for every k in K, $k \cdot u = 0 \bmod (2)$. Then, as in (3.6), K can be calculated from its annihilator. Note that we have not used all of the information available, since the y-summation must also be nonzero, and that could be useful in special cases. •

The utility of this general approach depends in part on the ease with which the associated eigenvector can be prepared. From our earlier work we have seen that $|0\rangle - |1\rangle$ is a feasible initial state as is the tensor product $|c_1c_2\ldots c_m\rangle$ and its Hadamard transform. However, for some problems a known eigenstate can't be prepared easily, and that may require approximations that lead to obtaining the desired information with probability less than one. In addition, the unitary map U may not be known explicitly, although the action of U may be computable on a quantum computer, and it also may be necessary to know the value of the phase ϕ of the eigenvalue $e^{2\pi i\phi}$ of U, where ϕ need not be a binary rational number and hence can only be approximated by a finite number of bits.

As a specific example, let us revisit the factoring problem analyzed earlier. Recall that the goal is to find the period r of an integer y with respect to a relatively prime integer N, that is, $y^r = 1 \bmod (N)$, where $0 < y < N$. Suppose we could construct an initial state

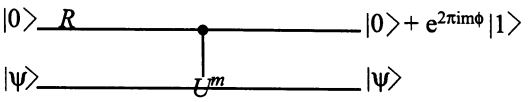
$$|\psi\rangle = \sum_{j=0}^{r-1} \exp(-2\pi ij/r)|y^j \bmod (N)\rangle.$$

Let $f(k) = ky \bmod (N)$. As we noted in Chapter 2, there is a rather complicated routine developed in [76] which effects f, and this is an example of

a unitary map U which can be implemented without really being known explicitly as a matrix. Then $|\psi\rangle$ is an eigenstate of U, since it is easy to confirm that

$$U|\psi\rangle = \sum_{j=0}^{r-1} \exp(-2\pi ij/r)|y^{j+1} \bmod (N)\rangle = \exp(2\pi i/r)|\psi\rangle.$$

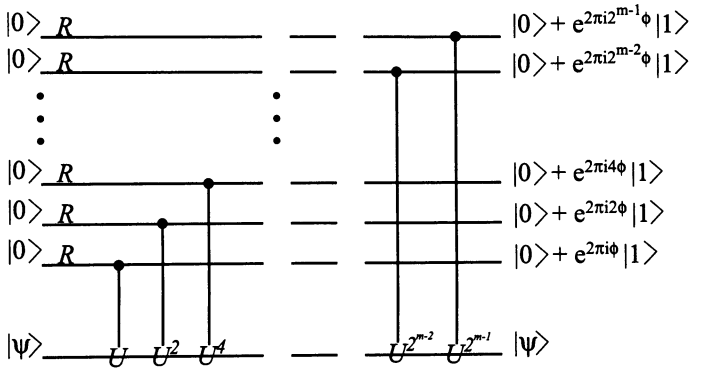
Now suppose that we could feasibly iterate an arbitrary U with eigenvector $|\psi\rangle$ and eigenvalue $e^{2\pi i\phi}$ and could apply it conditionally as illustrated in the following circuit:



with R denoting the usual rotation. Then as indicated in the diagram above, up to a scaling factor the effect of the circuit is to alter the input state by

$$|0\rangle|\psi\rangle \rightarrow |0\rangle|\psi\rangle + e^{2\pi im\phi}|1\rangle|\psi\rangle = (|0\rangle + e^{2\pi im\phi}|1\rangle)|\psi\rangle.$$

Assuming then that such a U could be implemented, we generalize the foregoing and apply selected powers of U conditionally to an initial preparation $|0\ldots0\rangle|\psi\rangle$ with m input qubits, as represented by the next circuit.



(3.37) Exercise. Show that up to the scaling factor the effect of the preceding circuit is the mapping

$$|0\ldots0\rangle|\psi\rangle \rightarrow \sum_{b=0}^{2^m-1} \exp(2\pi ib\phi)|b\rangle|\psi\rangle. \qquad \bullet$$

Now return to the factoring problem and apply the preceding scenario in that case. If r were such that

$$\phi = 1/r = 2^{-m} \sum_{j=0}^{m-1} a_j 2^j,$$

then the output of the circuit could also be written as

$$\sum_{b=0}^{2^m-1} \exp(2\pi i a b / 2^m) |b\rangle |\psi\rangle.$$

Comparing this with the results in Proposition (3.31), we see that applying the inverse Fourier transform to the first m qubits produces $|a\rangle|\psi\rangle$, and we can read off the value of $1/r$ in this case. If $1/r$ is not of that form, we can still read off the best binary m-tuple approximation with probability approximately $4/\pi^2$, as verified in [22] with the same kind of analysis as used in (3.30).

It remains to consider the construction of the eigenstate $|\psi\rangle$, and as noted in [22] there seems to be no simple way to do this. However, if we consider the family of eigenstates whose members are defined as

$$|\psi_k\rangle = \sum_{j=0}^{r-1} \exp(-2\pi i j k / r) |y^j \bmod (N)\rangle,$$

the same analysis shows that $|\psi_k\rangle$ is an eigenvector of U with eigenvalue $\exp(2\pi i k/r)$, and we would get an estimate of k/r instead of $1/r$. Again, this is one of the issues addressed in the discussion of Shor's algorithm.

Although the states $|\psi_k\rangle$ may be difficult to prepare, the state $|1\rangle$ is easy to prepare. But then it is straightforward to check that, up to a normalizing factor,

$$|1\rangle = \sum_{k=1}^{r} |\psi_k\rangle$$

and that $\{|\psi_k\rangle, 1 \le k \le r\}$ is an orthogonal set of eigenvectors with the same weight. If we were to apply the conditional powers of U as above but with $|\psi\rangle$ replaced by $|1\rangle$, after the application of the inverse Fourier transform the system would be in the general form

$$\sum_{k=1}^{r} (a_k |[k 2^m / r]\rangle + b_k |\varphi_k\rangle) |\psi_k\rangle,$$

where $|a_k|^2 > 0.4/r$. Each value $[k2^m/r]$ is equally likely to result from a measurement of the first m qubits, and the inference of r from $[k2^m/r]$ and the estimate of $|a_k|^2$ are also features of Shor's algorithm. Thus, the entire factoring algorithm can be derived from the perspective of estimating the phases of the eigenvalues of a particular unitary transformation.

(3.38) Exercise. Show that $\langle\psi_k|\psi_k\rangle$ is the same for each k and confirm the orthogonality of $\{|\psi_k\rangle, 1 \le k \le r\}$. Verify that, up to normalization,

$$|1\rangle = \sum_{k=1}^{r} |\psi_k\rangle. \qquad \bullet$$

3.8 Group theory and quantum algorithms

In Section 3.7 we viewed some of the quantum algorithms from the perspective of eigenvectors of unitary operators and found that phases of the eigenvectors contained the desired information. In this section we emphasize the group-theoretic foundations of the eigenvector approach and develop a general theory for which many of the known quantum algorithms are special cases. There are a variety of references for this analysis including [22], Jozsa [44], [45] and a recent paper by Mosca and Ekert [54]. See also [65] as well as the web sites of John Preskill at the California Institute of Technology and Umesh Vazirani at the University of California at Berkeley.

To motivate the notation, take a slight modification of Simon's problem, as discussed in (3.36). Let G denote the Abelian group F_2^n and let K be a subgroup of order 2^{n-m} on which a given function f is to be constant. This time, however, we assume the range of f is modeled by the states of a quantum register $|f(a)\rangle$ for each a in G. We continue to assume that f is constant and distinct on different cosets of K, so that f induces a one-to-one map between G/K, the quotient group of K, and the states of the register.

A second abstraction is the assumption that for each x in G, the controlled unitary map $U_{f(x)}$ can be implemented on each state $|f(a)\rangle$:

$$U_{f(x)}|f(a)\rangle = |f(x+a)\rangle. \tag{3.39}$$

For example, if $f(a) = Aa$, then $U_{f(x)}|Aa\rangle = |A(x+a)\rangle$, as in Simon's algorithm. In Shor's algorithm $f(a) = |y^a \bmod (N)\rangle$ and

$$U_{f(x)}|y^a \bmod (N)\rangle = |y^x y^a \bmod (N)\rangle = |y^{(x+a)} \bmod (N)\rangle.$$

Now revisit Simon's algorithm with this notation. We start with two registers in the initial state $|0\rangle|f(0)\rangle$, where 0 denotes the identity element in $G = F_2^n$, and follow the usual ritual: successively apply $R^{(n)}$ to the first register, U_f to the entire system and $R^{(n)}$ a second time to the first register:

$$\begin{aligned} |0\rangle|f(0)\rangle &\xrightarrow{R^{(n)}} \sum_{a\in G} |a\rangle|f(0)\rangle \\ &\xrightarrow{U_f} \sum_{a\in G} |a\rangle|f(a)\rangle \\ &\xrightarrow{R^{(n)}} \sum_{b\in G} |b\rangle \sum_{a\in G} |f(a)\rangle(-1)^{b\cdot a}. \end{aligned} \tag{3.40}$$

If we let H' denote a set of representatives of the cosets $[h]$ in G/K, we can decompose the summation over a into summations over H' and K. Since f is constant on cosets of K, we can misuse f by writing either $f(h)$ or $f([h])$ as required. Thus,

$$|0\rangle|f(0)\rangle \to \sum_{b\in G} |b\rangle \sum_{h\in H'} |f(h)\rangle(-1)^{b\cdot h} \sum_{k\in K} (-1)^{b\cdot k}.$$

As was the case before, the last factor is zero unless b is in $K^\perp$, the annihilator of K, and thus up to constant multiples

$$|0\rangle|f(0)\rangle \to \sum_{b\in K^\perp} |b\rangle|\psi_b\rangle,$$

where we define

$$|\psi_b\rangle = \sum_{h\in H'} |f(h)\rangle(-1)^{b\cdot h}. \tag{3.41}$$

Thus, a measurement of the first register gives the representation of a b in $K^\perp$.

An additional result of this format is that we have constructed $|\psi_b\rangle$ in the second register, and $|\psi_b\rangle$ is an eigenvector for each $U_{f(x)}$:

$$U_{f(x)}|\psi_b\rangle = \sum_h (-1)^{b\cdot h}|f(x+h)\rangle = (-1)^{b\cdot x} \sum_h (-1)^{b\cdot h}|f(h)\rangle.$$

Since b is in $K^\perp$, $(-1)^{b\cdot x}$ is also constant on cosets of K, and the measurement of the first register equivalently gives the factor b in the phase

of the eigenvalue. Note that we could also write the eigenvector in (3.41) in terms of the quotient group G/K.

To abstract this methodology to arbitrary finite Abelian groups G, we need the *dual group* G^* of G.

(3.42) Definition. G^* denotes the set of homomorphisms of G to the multiplicative group of complex numbers with modulus 1. •

The elements of G^* are called *characters*, and if we define an operation $\varphi_1 \cdot \varphi_2$ by

$$(\varphi_1 \cdot \varphi_2)(g) = \varphi_1(g) \cdot \varphi_2(g),$$

then G^* is a group in its own right with identity $\varphi_0(g) = 1$ for every g in G and $\varphi^{-1}(g) = \varphi(-g)$. Since G is finite, each element g has finite period, and it follows that $\varphi(g)$ is a root of unity for each g.

For example, if $G = Z_p$, the integers modulo p, then

$$\varphi_k(g) = \exp(2\pi i g k/p)$$

defines a character for each k, $0 \leq k < p$, and in fact

$$Z_p^* = \{\varphi_k, 0 \leq k < p\}.$$

If $G = F_2^n$, then for each φ in G^* there is an n-vector $b = (b_1, \ldots, b_n)$ such that

$$\varphi(x) = \prod_{k=1}^{n} (-1)^{b_k x_k} = (-1)^{b \cdot x}$$

for each $x = (x_1, \ldots, x_n)$ in F_2^n. Thus, $G^* = \{\varphi_b, b \in F_2^n\}$, with the obvious notation. It can be shown in general that there is a group isomorphism between G and G^*, as illustrated by

$$(\varphi_a \cdot \varphi_b)(g) = \varphi_{a+b}(g),$$

and in particular the cardinality of the two groups is the same: $|G| = |G^*|$.

The set of characters defines an orthogonal basis for an inner product space of complex-valued functions on G, and in particular

$$\textbf{(3.43)} \qquad \sum_{g \in G} \varphi(g) = \begin{cases} |G| & \text{if } \varphi = \varphi_0 \\ 0 & \text{otherwise} \end{cases}$$

and

$$\textbf{(3.44)} \qquad \sum_{\varphi \in G^*} \varphi(g) = \begin{cases} |G^*| & \text{if } g = 0 \\ 0 & \text{otherwise,} \end{cases}$$

where 0 denotes the identity of G and φ_0 the identity of G^*. The related Fourier transform F_G is defined using the normalizations of the characters:

$$\sqrt{|G|}F_G f(b) = \sum_{g \in G} \varphi_b(g) f(g), \tag{3.45}$$

and maps functions on G to functions on G^*. (For precise definitions, details and additional results see, for example, Dym and McKean [31].)

In Simon's problem we needed to write $\varphi_b([x])$ when φ_b is identically equal to one on the subgroup K; that is, we wanted a character on $H = G/K$, the quotient group of K. Indeed, each character in H^* can be identified with a representative of a coset of characters in G^*, each of which equals one on elements of K.

We now have the tools to generalize the steps in Simon's algorithm. First, note that whatever the difference in the representation of an a in G and a character φ_a in G^*, the identity elements 0 and φ_0 will have the same representation $(0, \ldots, 0)$ in the first register. We can now follow the steps of (3.40). Thus, applying the inverse Fourier transform as the first step, we have

$$|\varphi_0\rangle|f(0)\rangle \xrightarrow{F_G^{-1}} \sum_{a \in G} |a\rangle|f(0)\rangle. \tag{3.46}$$

Next we apply the controlled unitary map to the entire system:

$$\sum_{a \in G} |a\rangle|f(0)\rangle \xrightarrow{U_f} \sum_{a \in G} |a\rangle|f(a)\rangle, \tag{3.47}$$

and then the Fourier transform to the first register:

$$\sum_{a \in G} |a\rangle|f(a)\rangle \xrightarrow{F_G} \sum_{\varphi \in G^*} |\varphi\rangle \sum_{a \in G} |f(a)\rangle\varphi(a). \tag{3.48}$$

Since f is constant on cosets of K, we can again decompose the summation over G into sums over representations h of cosets $[h]$ of G/K and over K itself:

$$\sum_{a \in G} |f(a)\rangle\varphi(a) = \sum_{h} |f(h)\rangle\varphi(h) \sum_{k \in K} \varphi(k).$$

The last factor is $|K|$ or zero, as φ is in the annihilator $K^\perp$ or not, where

$$K^\perp = \{\varphi \in G^* : \varphi(k) = 1, k \in K\}. \tag{3.49}$$

(As noted above, $K^\perp$ is isomorphic to $(G/K)^*$.)

Thus, up to a multiplicative constant, the three transformations of (3.46), (3.47) and (3.48) combine to give

$$|\varphi_0\rangle|f(0)\rangle \to \sum_{\varphi\in K^\perp} |\varphi\rangle|\psi_\varphi\rangle, \tag{3.50}$$

where

$$|\psi_\varphi\rangle = \sum_{h\in H} \varphi(h)|f(h)\rangle \tag{3.51}$$

is an eigenvector of $U_{f(x)}$ with eigenvalue $\varphi(-[x])$.

There is an equivalent derivation of (3.50) which emphasizes the role of the eigenvectors and which was put forward by Kitaev [46]. We leave the details to the reader in Exercise (3.53).

Take, for example, Shor's algorithm when the period r of a generator y in Z_N divides the order q of the group $G = Z_q$, as in Section 3.4. Then the subgroup K is

$$K = \{g : y^g = 1 \bmod (N)\} = \{g : r \text{ divides } g\},$$

so that φ_b is in $K^\perp$ if for all j

$$1 = \varphi_b(rj) = \exp(2\pi i b r j/q).$$

In particular, $br/q = k$, an integer. Hence, the measured state $|b\rangle = |\varphi_b\rangle$ gives the now-familiar result

$$\frac{b}{q} = \frac{k}{r}.$$

Thus, the output of Shor's algorithm when r divides q is a special case of a general theory. As discussed in Section 3.5, additional analysis is necessary if r doesn't divide q.

As another application of the general theory, let us describe Shor's algorithm for the discrete log problem, again in the special case when the order of K divides the order of G. For the additional analysis necessary when that condition isn't satisfied, see [65].

Given a prime p, let b be a generator of the multiplicative group $\tilde{Z}_p$ consisting of the nonzero elements of Z_p. Then for every a in $\tilde{Z}_p$ there is an m in $\{0, 1, \ldots, p-2\}$ such that $a = b^m$. It is easy to compute a given p and m, but hard to compute m given a and p, and that's the discrete

log problem. For example, if $p = 17$, then $3^4 = 13 \bmod (17)$ is easy and finding m for $a = 7$ is harder.

To use the general theory, let $G = Z_q \times Z_q$ and for the sake of simplicity assume that $p = q + 1$, where q^2 is the order of G. The function f on G is defined by

$$f(x, y) = a^x b^y \bmod (p),$$

where b is the assumed generator of $\tilde{Z}_p$, and we want to find the exponent m associated with a given a. In this case,

$$\begin{aligned} K &= \{(x, y) : a^x b^y = 1 \bmod (p)\} \\ &= \{(x, y) : y = -mx \bmod (p-1)\}, \end{aligned}$$

and $|K| = p - 1$. We assume that f is unitarily implementable as usual:

$$U_{f_{(c,d)}} |f(x, y)\rangle = |a^c b^d a^x b^y \bmod (p)\rangle = |f(x + c, y + d)\rangle.$$

If the characters of G are represented by

$$\varphi_{(c,d)}(x, y) = \exp\left(\frac{2\pi i}{q}(xc + yd)\right),$$

then the general theory gives

$$|\varphi_{(0,0)}\rangle |f(0, 0)\rangle \rightarrow \sum_{\varphi \in K^\perp} |\varphi\rangle |\psi_\varphi\rangle,$$

where $\varphi_{(c,d)} \in K^\perp$ if and only if

$$\exp(\frac{2\pi i}{q}(xc + yd)) = 1$$

for all (x, y) in K. Equivalently,

$$cx - mxd + j(x)(p - 1) = 0 \bmod (q),$$

where $j(x)$ depends on x and m. In the simplified case when $p = q + 1$, choosing $x = 1$ gives

$$\textbf{(3.52)} \qquad c = md \bmod (p - 1).$$

Thus, a measurement of the first register gives a pair (c, d) satisfying (3.52), and if d and $p - 1$ are relatively prime, we can solve for m:

$$m = cd^{-1} \bmod (p - 1).$$

That condition is met with positive probability, so that repeated runs of the algorithm will produce a solution m with an extra multiplicative work factor, as in (3.16).

(3.53) Exercise. Using the notation of the general theory and (3.51), prove that, up to a multiplicative constant,

$$|f(0)\rangle = \sum_{\varphi \in K^\perp} |\psi_\varphi\rangle,$$

so that

$$|\varphi_0\rangle|f(0)\rangle = \sum_{\varphi \in K^\perp} |\varphi_0\rangle|\psi_\varphi\rangle. \tag{3.54}$$

Apply the steps of (3.46), (3.47) and (3.48) to the right-hand side of (3.54) and use the fact that the $|\psi_\varphi\rangle$ are eigenvectors of U_f to derive (3.50). •

4
Quantum Error-Correcting Codes

4.1 Quantum dynamics and decoherence

So far we have been dealing with a highly idealized situation in which physical states have been assumed to be stable and independent of time and in which unitary transformations on these states can be effected in a reliable fashion. The real situation is considerably more complicated and requires a model for the dynamics of a quantum system.

The main addition to our idealized static model is *Schrödinger's equation*, which describes the time evolution of the physical system subject to the various forces involved. Assume that the underlying Hilbert space H is now an appropriate, infinite-dimensional function space and that we have a physical system ψ with a ket $|\psi\rangle$ which is a function of both time and position in H. The evolution of the state is assumed to be given by

$$i\hbar\frac{\partial}{\partial t}|\psi\rangle = H_0|\psi\rangle, \tag{4.1}$$

where H_0 denotes the *Hamiltonian*, a Hermitian operator appropriate to the physical situation, and as usual $\hbar = h/2\pi$.

(4.2) Example. In the special case where $|\psi\rangle$ is a finite-dimensional, time-dependent vector and H_0 is a time-independent Hermitian matrix, it is easy to check that $|\psi\rangle(t) = \exp(-itH_0/\hbar)|\psi\rangle(0)$ satisfies Schrödinger's equation, where exp( ) denotes the usual infinite series expansion. •

It goes well beyond the scope of this book to develop a mathematical framework suitable for a sensible discussion of Schrödinger's equation and of the way the Hamiltonian is defined. (The reader is referred to [60] and [63] for the basic development of the theory.) However, we can indicate one of the consequences of Schrödinger's equation. Suppose the operator H_0 does not depend explicitly on t and has a countable number of eigenvectors $|\psi_n\rangle$ with real eigenvalues λ_n. Then using a separation of variables approach we would find

$$|\psi\rangle(t) = \sum_n a_n \exp(-it\lambda_n/\hbar)|\psi_n\rangle$$

with the proviso that $\sum_n |a_n|^2 = 1$. The interpretation is that the system is in a linear combination of eigenstates whose probability amplitudes have a time-dependent phase. Since some of the quantum gates are sensitive to perturbations of the phase factor of the coefficients of the computational basis states, control and correction of the phase factor becomes an issue for both hardware and software.

(4.3) Example. Let $|\psi\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$ be the spin state of an electron with respect to the canonical (z-spin) computational basis. Suppose there is a phase perturbation so that the new state is

$$|\phi\rangle = (e^{i\alpha}|0\rangle + e^{-i\alpha}|1\rangle)/\sqrt{2}.$$

Then the probability distribution of the system's z-spin remains the same, but the probability distribution of the x-spin and y-spin depend on α. For example, using the results of (1.15) we can compute the probability of the y-minus spin as

$$\mathrm{tr}(|\phi\rangle\langle\phi|P_d) = \frac{1}{4}\mathrm{tr}\left(\begin{bmatrix} 1 & e^{2i\alpha} \\ e^{-2i\alpha} & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & i \\ -i & 1 \end{bmatrix}\right) = \frac{1}{2}(1 + \sin(2\alpha)),$$

which is certainly dependent on the phase change. •

A consequence of considering the environment of a quantum system or its additional degrees of freedom is that the physical state representing one of the qubits can decay over time, so that encoded information becomes incorrect or even so that the physical system is in a third state. For example, if two energy levels of a particle were used as the computational basis states, the particle could make a transition to a third energy level. Given the difficulty of working at the microphysical level, it seems likely that such problems will be inherent in any quantum computer.

One result of a quantum system interacting with its environment is the loss of superposition, and the effect is referred to as the *decoherence* of the system. Decoherence is of great concern for the development of quantum computers because its effect is to undo the superpositions and entanglements of states on which quantum algorithms rely, replacing them instead with mixtures of states. Its explanation has proved to be a source of some controversy in the theoretical physics community, and the reader is referred to Zurek [79] for one presentation and subsequent rebuttals. (See also [75].)

To appreciate the effect of decoherence, we discuss a simple example, following the approach in [79]. (Since there are scholarly tomes dedicated to this topic, the informed reader can consider the attempt to describe this phenomenon in so few words as an example of quantum chutzpah.) As illustrated in Example (4.3), the measurement of the state of a quantum system is modeled by using the trace operation, projection matrices and the density matrix of the system. What is added here is the introduction of states representing the measuring device and the environment to better describe the actual process of measurement.

For example, suppose the state of a two-state quantum system is the superposition $\alpha|0\rangle + \beta|1\rangle$, where the coefficients satisfy the usual normalization $|\alpha|^2 + |\beta|^2 = 1$. Suppose this state becomes entangled with a measuring device which is to be subsequently read:

$$(\alpha|0\rangle + \beta|1\rangle)|D\rangle \rightarrow |\psi_e\rangle = \alpha|0\rangle|D_0\rangle + \beta|1\rangle|D_1\rangle.$$

Then the resulting density matrix of the system *before* the measurement is

$$\begin{aligned}\rho_e &= |\psi_e\rangle\langle\psi_e| = \alpha\bar{\alpha}|0\rangle|D_0\rangle\langle 0|\langle D_0| + \alpha\bar{\beta}|0\rangle|D_0\rangle\langle 1|\langle D_1| \\ &\quad +\beta\bar{\alpha}|1\rangle|D_1\rangle\langle 0|\langle D_0| + \beta\bar{\beta}|1\rangle|D_1\rangle\langle 1|\langle D_1|.\end{aligned}$$

Since the measuring device is "classical" it will either be in state D_0 or state D_1. In the first case, the measurement produces

$$\langle D_0|\rho_e|D_0\rangle = |\alpha|^2|0\rangle\langle 0|,$$

which we interpret as saying that the probability of finding the quantum system in state $|0\rangle$ is $|\alpha|^2$. In the second case we obtain

$$\langle D_1|\rho_e|D_1\rangle = |\beta|^2|1\rangle\langle 1|,$$

with an analogous probabilistic interpretation.

This approach distinguishing the role of classical systems was postulated by von Neumann. (See [72, p. 51] however.) The tacit assumption that $\langle D_1|D_0\rangle = 0$ can be reformulated as saying that after entanglement with a classical measuring device, the density is not given by ρ_e but rather by the *mixed density*

$$\rho_r = |\alpha|^2|0\rangle|D_0\rangle\langle 0|\langle D_0| + |\beta|^2|1\rangle|D_1\rangle\langle 1|\langle D_1|,$$

which leads to the same probabilistic interpretation given above. Thus ρ_r differs from ρ_e by the elimination of the off-diagonal terms and can be considered to be the density of the system before the measuring device is read. The advantage is that the coefficients $|\alpha|^2$ and $|\beta|^2$ can be interpreted as classical probabilities.

Note that the density of the system after the measurement but before the results of the measurement are known could be modeled by

$$\rho_m = \langle D_0|\rho_e|D_0\rangle + \langle D_1|\rho_e|D_1\rangle = |\alpha|^2|0\rangle\langle 0| + |\beta|^2|1\rangle\langle 1|.$$

This operation is called a "tracing-out" of a density over one of the variables, in this case the measuring apparatus. More generally, a pure density of the form

$$\rho = \left(\sum_{k=1}^{r} \alpha_k|\phi_k\rangle|\chi_k\rangle|\psi_k\rangle\right)\left(\sum_{k=1}^{r} \bar{\alpha}_k\langle\phi_k|\langle\chi_k|\langle\psi_k|\right)$$

traces out over the third component to give a mixed density:

$$\begin{aligned} \textbf{(4.4)} \qquad \sum_{k=1}^{r}\langle\psi_k|\rho|\psi_k\rangle &= \sum_{k=1}^{r}(\alpha_k|\phi_k\rangle|\chi_k\rangle)(\bar{\alpha}_k\langle\phi_k|\langle\chi_k|)\langle\psi_k|\psi_k\rangle \\ &= \sum_{k=1}^{r}|\alpha_k|^2|\phi_k\rangle|\chi_k\rangle\langle\phi_k|\langle\chi_k|. \end{aligned}$$

Now suppose that in addition to the two-state quantum system and the two-state measuring device, we also take into account the rest of the "environment" with which the system and the detector can interact. Then the environment becomes entangled as well, and the overall density is

$$\rho_c = (\alpha|0\rangle|D_0\rangle|E_0\rangle + \beta|1\rangle|D_1\rangle|E_1\rangle)(\bar{\alpha}\langle 0|\langle D_0|\langle E_0| + \bar{\beta}\langle 1|\langle D_1|\langle E_1|).$$

If the environment is considered to be an apparatus over which we have no control, then following von Neumann's approach, we must trace over the

environmental states as in (4.4) to model the environmental "measurement." Assuming orthogonality and equal weight of the environmental states,

$$\langle E_0|\rho_c|E_0\rangle + \langle E_1|\rho_c|E_1\rangle = \rho_r.$$

That is, even if the measuring device is not classical, the uncontrolled "measurement" by the environment leads to the reduced density matrix ρ_r postulated by von Neumann and thus the loss of some of the entanglement on which our quantum algorithms rely.

More sophisticated models show the loss of the off-diagonal terms as functions of time, which is intuitively consistent with the idea that the more isolated the quantum system is from its environment, the longer entanglement should persist. We should also note that the foregoing model is basis dependent. Decoherence can take different forms in different bases, as illustrated in (4.3).

In any case, the conclusion is that errors will inevitably occur, and the reliability of a quantum computer will depend on detecting and correcting those errors. Remarkably enough, there are already quantum error-correction techniques available, and we discuss some of those algorithms next.

4.2 Error correction

To begin the discussion of quantum error correction, let us recall the use of various error-correcting codes to address the problem of controlling errors when transmitting information classically. For example, if before transmission the bit 1 is encoded as (111) and 0 as (000), then after the transmission, the value of the original bit could be assumed to be the majority value of the three received bits. If at most one error occurred during the transmission, this procedure produces the correct initial value. A well-developed theory of error-correcting codes deals with such problems and is presented in texts such as [52].

Originally, it was thought that analogous error-correction techniques would not apply to the transmission of qubits. For one thing, a measurement forces the collapse of the qubit into a specific state, and superposition is lost. Thus, sending three copies of a qubit and taking a majority vote at the receiving end would lose the original superposition which was to be preserved. Equally important was the observation that there is a problem with copying or "cloning" an arbitrary quantum state in the first place. Since the proof is easy, we present a version of the "no-cloning" theorem of Wootters and Zurek [78].

(4.5) Proposition. There does not exist a perfect, linear amplifying device which clones an arbitrary state of a quantum system.

Proof: Suppose there were such a device so that

$$|A_b\rangle|s\rangle|0\rangle \rightarrow |A_s\rangle|s\rangle|s\rangle,$$

where the subscripts denote the state of the apparatus before and after the operation. Then from

$$|A_b\rangle|0\rangle|0\rangle \rightarrow |A_0\rangle|0\rangle|0\rangle \qquad |A_b\rangle|1\rangle|0\rangle \rightarrow |A_1\rangle|1\rangle|1\rangle,$$

we would have from the assumed linearity

$$\begin{aligned} |A_b\rangle(\alpha|0\rangle + \beta|1\rangle)|0\rangle &\rightarrow |A_m\rangle(\alpha|0\rangle + \beta|1\rangle)(\alpha|0\rangle + \beta|1\rangle) \\ &= \alpha|A_0\rangle|0\rangle|0\rangle + \beta|A_1\rangle|1\rangle|1\rangle. \end{aligned}$$

Measuring the amplifying device in the first expression on the right gives mixed terms such as $|0\rangle|1\rangle\langle 0|\langle 1|$ in the resulting density, but the same measurement of the density of the second expression produces no terms of that form. Hence, for general α and β cloning is not possible. •

However, as we saw in the discussion of the *XOR* gate, it *is* possible to clone a basis state, giving

$$(\alpha|0\rangle + \beta|1\rangle)|0\rangle \rightarrow \alpha|0\rangle|0\rangle + \beta|1\rangle|1\rangle.$$

In other words, while we can't clone arbitrary superpositions, we can create entangled states which contain duplicates of basis states, and it might be possible to use such entanglements to create an error-correcting mechanism.

The idea of using entangled states as the mechanism for error correction was put forward independently by Shor [64] and by Steane [68] and is the basis for a burgeoning industry in quantum error-correcting codes. Because of the rapid development of the field, it is impossible to provide the chronology and all of the results to date in a limited space. Inevitably, that also means that not every contribution to the field of quantum error correction will be mentioned and not every contributor will be cited. What we will do instead is examine specific examples of quantum error-correcting codes and illustrate some of the emerging theory using these examples, giving references that contain a more complete account of the current state of the art. As particularly apt examples of such references, we should mention Bennett et al. [12], Calderbank et al. [18], Knill and Laflamme [47], Steane [69] and the thesis of Gottesman [39],

which contains an extensive bibliography as well as a theoretical context for a large number of error-correcting codes.

The kinds of errors that can occur will certainly depend on the physical implementation of qubits, so at least initially one has to idealize the errors to be addressed. A basic simplifying assumption is that errors on qubits are defined independently of one another. Then, at the very least, one error which should be included is a switch in the computational basis, that is, $|k\rangle \to |k \oplus 1\rangle$ or

$$\alpha|0\rangle + \beta|1\rangle \to \alpha|1\rangle + \beta|0\rangle.$$

A second type of error which should be included is the phase error:

$$\alpha|0\rangle + \beta|1\rangle \to \alpha|0\rangle - \beta|1\rangle.$$

And if we're going to be mathematical about this, we should also include the possibility that both errors occur:

$$\alpha|0\rangle + \beta|1\rangle \to \alpha|1\rangle - \beta|0\rangle.$$

These errors are closely related to the Pauli spin matrices which were introduced above (1.16) and used in Section 2.3. If we write the state $\alpha|0\rangle + \beta|1\rangle$ in the computational basis, then the errors can be represented as the results of matrix multiplication:

$$\sigma_x \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \beta \\ \alpha \end{pmatrix} \quad \sigma_z \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \alpha \\ -\beta \end{pmatrix} \quad \sigma_y \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = i \begin{pmatrix} -\beta \\ \alpha \end{pmatrix}.$$

Thus, a switch of the basis state is modeled by multiplication by σ_x, a phase error of π is modeled by multiplication by σ_z and the combination of the two is modeled by multiplication by σ_y, modulo the factor of i.

Moreover, we saw in (1.30) that any matrix in *SU(2)* can be expressed as a linear combination of the Pauli matrices and the 2×2 identity matrix, and that fact will enable us to correct any one-qubit error represented as multiplication by a matrix in *SU(2)*. (See Theorem (4.14).) Consequently, when modeling errors to be corrected, we have an incentive to think of the errors in terms of the Pauli spin matrices, and that is a key assumption for a theory of quantum error-correcting codes.

Detection of these errors requires measurements, and measurements require determining eigenvalues of Hermitian operators. For example, suppose a σ_x error affects the first qubit in an entangled state:

$$|\psi\rangle = (|00\rangle + |11\rangle)/\sqrt{2} \to (|10\rangle + |01\rangle)/\sqrt{2} = |\psi'\rangle.$$

If we had measured the eigenvalue of the operator $\sigma_z \otimes \sigma_z$ when the system was in the state $|\psi\rangle$, we would have found $\lambda = 1$; while if we performed the same measurement after the error, we would have found $\lambda = -1$. In other words, in the computational basis a σ_x error affects the *parity* of the contents of qubits and can be detected by making measurements of σ_z operators.

Suppose, however, that the original system was changed by a σ_z error on the first qubit, so that the new state is

$$|\psi''\rangle = (|00\rangle - |11\rangle)/\sqrt{2}.$$

This time a post-error measurement of the eigenvalue of $\sigma_z \otimes \sigma_z$ gives $+1$, and there is no change of eigenvalue; i.e., a measurement in the computational basis doesn't reveal the σ_z error. (This same result was also noted in Example (4.3)). However, suppose we were to make the *x-direction* the computational basis direction. Physically, this amounts to making measurements along the x-spin axis or else rotating the physical system appropriately so that a z-spin measurement of the rotated system is equivalent to an x-spin measurement of the original system. Mathematically, it means making a (unitary) change of basis mapping of both qubits using the R of (2.7) and based on (2.8):

$$|k\rangle \rightarrow |0\rangle + (-1)^k|1\rangle$$

for $k = 0, 1$ so that the state $|\psi''\rangle$ is mapped as follows:

$$|\psi''\rangle = (|00\rangle - |11\rangle)/\sqrt{2} \rightarrow (|01\rangle + |10\rangle)/\sqrt{2} = |\tilde{\psi}\rangle.$$

But now a measurement of $\sigma_z \otimes \sigma_z$ in the context of the rotated system gives an eigenvalue of (-1), and the phase error can be detected! Moreover, a σ_y error would be detected in either basis.

(4.6) Example. We can also compute the eigenvalues using the outer product notation, so that

$$|\psi\rangle\langle\psi| = \begin{pmatrix} 0.5 & 0 & 0 & 0.5 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0.5 & 0 & 0 & 0.5 \end{pmatrix}$$

and

$$\sigma_z \otimes \sigma_z = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

in the original computational basis. Then $\mathrm{tr}(|\psi\rangle\langle\psi|\sigma_z \otimes \sigma_z) = 1$. As an exercise, represent $|\psi'\rangle\langle\psi'|$ in terms of the computational basis and show that $\mathrm{tr}(|\psi'\rangle\langle\psi'|\sigma_z \otimes \sigma_z) = -1$. •

Of course, the difficulty with this approach is that the measurements described above would destroy any superposition of even and odd parity states that existed before the measurement. Thus, we have to see if there is a way to make parity measurements which can detect Pauli-type errors while preserving superpositions. In fact, the idea is very simple. Suppose we were to add a third qubit in a known state and perform two *XOR* operations on the two-qubit states $|\psi\rangle$ and $|\psi'\rangle$. Then, for example, $|i\rangle|j\rangle|0\rangle \rightarrow |i\rangle|j\rangle|i \oplus j\rangle$, so that we have three entangled states. We can then take the spin measurement of the *last* qubit and find the parity in the first two states *without knowing* their actual values.

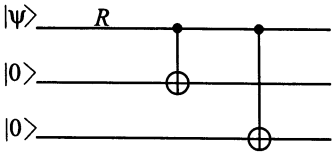
4.3 Shor's nine-qubit error-correcting code

We begin with the example of Shor [64] in which the quantum analogue of the classical three-bit repetition scheme is developed. One part of the structure of the code is designed to detect bit changes while the other part is designed to detect phase changes. Specifically, Shor presumes that one qubit is encoded into nine qubits via the following scheme:

$$\begin{aligned}\alpha|0\rangle + \beta|1\rangle \quad \rightarrow \quad & \alpha(|000\rangle + |111\rangle) \otimes (|000\rangle + |111\rangle) \otimes (|000\rangle + |111\rangle) \\ & +\beta(|000\rangle - |111\rangle) \otimes (|000\rangle - |111\rangle) \otimes (|000\rangle - |111\rangle),\end{aligned}$$

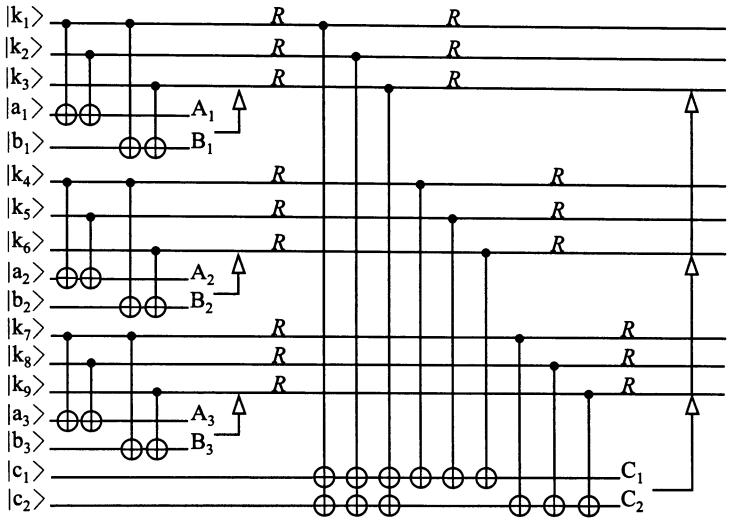
where we continue the practice of ignoring constant factors which can be recovered from the normalization. The nine qubits are transmitted to a receiver, and the goal is to recover the original superposition, assuming that no more than one "Pauli-type" error has occurred during transmission. The trick is to use specified entanglements with additional or *ancillary* qubits and measures of those ancillary qubits to correct the error. Moreover, that has to be done so that the original superposition is retained in some way. As with the classical majority vote algorithm, the recovery process does not necessarily work if more than one error occurs.

The first problem is the encoding. Suppose we are given a qubit in the superposition. We are also given eight qubits initially in the zero state. Then if we apply the following schema to the first three qubits, we obtain some of the desired entanglement. (We use the usual *XOR* operation and *R* mappings.)



(4.7) Exercise. Verify that the wiring diagram above leads to part of Shor's entanglement. Then construct the complete encoding of the three groups of three qubits. •

We begin by describing the following wiring diagram:



Suppose the encoded nine qubits denoted by $|k_i\rangle$ are transmitted and that at most one of the Pauli-type errors occurred. The trick is to measure the computational basis parity of two pairs of each of the three sets of three qubits, projecting the entanglement of each triple from an eight-dimensional subspace to a two-dimensional subspace in the process. The results of the parity measurements will show which qubit, if any, in each group of three was affected by a basis change, i.e., a σ_x error. After those six measurements, two additional measurements are possible, and they can be used to determine if a σ_z (phase) error occurred. The coding is stable under other error changes, but first let's describe the circuitry.

The eight ancillary qubits initially in the zero state are grouped in four pairs of two qubits. Each of the first three pairs of ancillary qubits is used to measure the basis parity of pairs of the related group of three

transmitted qubits. The last pair of ancillary qubits is used to measure the phase parity of the three triples. The additional piece of notation is the writing of two distinct *XOR*s involving each of the first three qubits (the long vertical lines).

As we noted above, a measurement of $|a_1\rangle$ in the computational basis is a measure of the eigenvalues of $\sigma_{z1} \cdot \sigma_{z2}$, where the second subscript denotes the relevant qubit, and the eigenvalues are $(-1)^{k_1+k_2}$. We let "0" denote even parity and "1" denote odd parity. Suppose, for example, that the result of the first pair of measurements (A_1, B_1) is $(0, 1)$. This means that in the computational basis the first two qubits have the same parity and the first and third bits have opposite parity. Under the assumption that at most one error has occurred, we can conclude that the third qubit is in the wrong computational basis state and thus a σ_{x3} map will undo the damage. Note that we do not know the actual state of the third qubit, only that an error has occurred. The initial superposition is preserved in a two-dimensional subspace involving the first three qubits.

As with classical codes the dibit pattern of the measurements (A, B) is called the *error syndrome* of the measurement and determines the corrective mapping, which is denoted schematically by the arrow leading back up to the related three qubits:

$$\begin{array}{cc|cc} (A,B) & U & (A,B) & U \\ (0,0) & I & (1,0) & \sigma_{x2} \\ (0,1) & \sigma_{x3} & (1,1) & \sigma_{x1} \end{array}$$

In fact, this procedure will correct the state error caused by one σ_x or σ_y error in each triple.

Having corrected a possible state error, the remainder of the circuit corrects a phase error caused by a σ_z or σ_y error. Again, we want to change bases, and the mapping R takes the state $|k\rangle$ to $|0\rangle + (-1)^k|1\rangle$, where we continue to ignore the normalizing factor in favor of emphasizing the 0-1 patterns of the states.

(4.8) Exercise. Show that up to normalizing factors the mapping $R \otimes R \otimes R$ gives

$$|000\rangle + |111\rangle \rightarrow |000\rangle + |011\rangle + |101\rangle + |110\rangle$$

and

$$|000\rangle - |111\rangle \rightarrow |111\rangle + |100\rangle + |010\rangle + |001\rangle. \quad \bullet$$

We next add the first six qubits into the seventh ancillary qubit and the first three and last three qubits into the last ancillary qubit, so that a subsequent measurement in the computational basis gives the parity of the two sums. But, as we see from (4.8), each of the words in the rotated qubit triples representing $|0\rangle$ should have even parity and each of the words in the rotated qubit triples representing $|1\rangle$ should have odd parity. If one of the minus signs of one of the triples is in error, that will show up as a computational basis parity error in the (C_1, C_2) measurement. The error can be corrected by applying an appropriate σ_z mapping to one of the qubits in the triple with the incorrect parity after the R mappings have returned the nine qubits to their prior state. Note that we cannot distinguish which qubit in a triple suffered the phase error but we can still correct the effect.

Thus, assuming that these parity operations haven't introduced additional errors into the nine transmitted qubits, we have been able to correct a possible σ_z error in each of the groups of three qubits and one phase error in the nine qubits. This scheme also corrects one σ_y error in the process, without our actually knowing the original state.

(4.9) Exercise. Recall from (2.8) that $\sigma_x = R\sigma_z R$ and that $-\sigma_y = R\sigma_y R$. Find analogous unitary mappings S and T that transform σ_y to σ_x and to σ_z, respectively, leaving the third Pauli matrix invariant, possibly up to a factor. •

There is a practical aspect to this exercise. We have written Shor's nine-qubit error correction as if it were feasible to measure the state of a qubit in only one way. But if it were physically possible to measure a qubit with respect to different bases, then one could reduce the number of operations in an error-correction algorithm. For example, instead of performing eighteen R operations in the nine-qubit algorithm, we would be able to assume that the requisite measurements of σ_z could be made directly. In particular, Steane [69] emphasized this perspective in his work on quantum error-correcting codes.

4.4 A seven-qubit quantum error-correcting code

We have seen how Shor used the structure of the three-bit parity code, the simplest classical code, to construct an encoding that would correct single-qubit errors in the encoding qubits. Steane [68], [69] and Calderbank and Shor [20] also used classical codes to construct quantum error-correcting codes. We illustrate the ideas using their example, which en-

codes one qubit in seven qubits and which can detect and correct a single error affecting any of the seven qubits.

The idea is to use classical coding theory to prescribe suitable measurements in both the z-spin basis and the x-spin basis. For example, suppose one encoded $|k\rangle$ in the (computational) z-spin basis as the following equally-weighted linear combinations:

$$\begin{aligned} |0\rangle \;\rightarrow\; & |0000000\rangle + |0001111\rangle + |0110011\rangle + |0111100\rangle \\ & + |1101001\rangle + |1010101\rangle + |1100110\rangle + |1011010\rangle \end{aligned}$$

and

$$\begin{aligned} |1\rangle \;\rightarrow\; & |1111111\rangle + |1110000\rangle + |1001100\rangle + |1000011\rangle \\ & + |0010110\rangle + |0101010\rangle + |0011001\rangle + |0100101\rangle. \end{aligned}$$

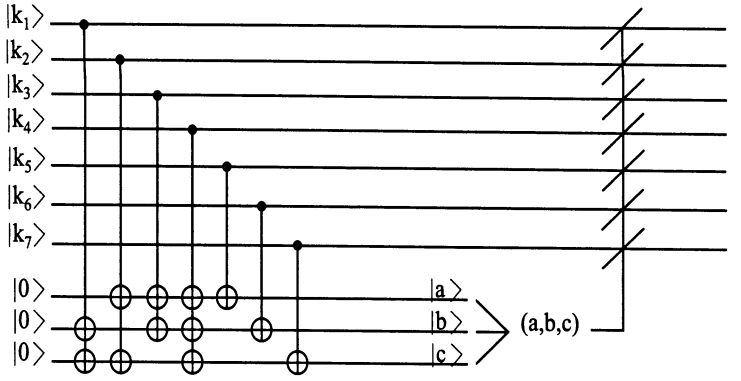
The reader familiar with classical error-correcting codes will recognize that the encoding is based on the classical [7,4,3] Hamming code, where the notation means that the code words are 7-long vectors defining a 4-dimensional linear subspace over *GF(2)*, and that the minimal number of differences between any two code words is 3. $|0\rangle$ is encoded as the equally weighted sum of the eight seven-bit words with even parity, and $|1\rangle$ is encoded as the equally weighted sum of the complements of those eight seven-bit words. Alternatively, $|0\rangle$ is encoded as a sum over a subspace of the Hamming code, and $|1\rangle$ is encoded as a sum over a translation of that subspace.

(4.10) Exercise. Verify that the set {(0111100),(1011010),(1101001)} defines a basis for the space of eight words representing $|0\rangle$. •

Let C denote the subspace defined by the [7,4,3] Hamming code. If a σ_x error has occurred in one of the qubits, then the parity of *each* word in C is changed, and that can be detected using the classical parity matrix, which is defined by basis vectors for $C^{\perp}$, the linear space defined by the dual code of C. ($C^{\perp}$ denotes the subspace of 7-long vectors whose mod(2) dot product with every vector in C is zero.) In this case, the dual code is the [7,3,4] simplex code which is precisely the set of even weight vectors of the Hamming code. It follows from (4.10) that we can use {(0111100),(1011010),(1101001)} as the basis code words for $C^{\perp}$.

The end result of those observations is that *every* word in the encoding of $|0\rangle$ and of $|1\rangle$ must have even parity in the three sets of positions {2,3,4,5}, {1,3,4,6} and {1,2,4,7}. But that means we can check for a one-qubit σ_x error by performing three measurements in the computational basis and use the resulting three-bit error syndrome to correct the error.

Moreover, the choice of basis vectors of $C^{\perp}$ defines the following wiring diagram for three measurements on the seven qubits. (The last vertical line with seven slashes denotes the correction mappings based on the error syndrome.)



(4.11) Exercise. Verify that $C^{\perp}$ consists of the eight even-weight vectors in the Hamming code. Using the indicated measurements, find the error syndrome for a σ_x error in qubit k, $1 \leq k \leq 7$. Confirm that distinct one-qubit errors have distinct syndromes. •

If we were to rotate each qubit by applying the operator R, we would obtain a representation (in the original z-spin basis) of the encoding of $|0\rangle$ and $|1\rangle$ in the x-spin basis. In this particular case, we obtain sums of words with exactly the same 0-1 patterns as the 0-1 patterns of the original words. The compulsive reader can confirm directly that using $|k\rangle \rightarrow |0\rangle + (-1)^k|1\rangle$ we obtain:

$$\begin{aligned} |0\rangle \rightarrow\ & |0000000\rangle + |0001111\rangle + |0110011\rangle + |0111100\rangle \\ & + |1101001\rangle + |1010101\rangle + |1100110\rangle + |1011010\rangle \\ & + |1111111\rangle + |1110000\rangle + |1001100\rangle + |1000011\rangle \\ & + |0010110\rangle + |0101010\rangle + |0011001\rangle + |0100101\rangle \end{aligned}$$

and

$$\begin{aligned} |1\rangle \rightarrow\ & |0000000\rangle + |0001111\rangle + |0110011\rangle + |0111100\rangle \\ & + |1101001\rangle + |1010101\rangle + |1100110\rangle + |1011010\rangle \\ & - (|1111111\rangle + |1110000\rangle + |1001100\rangle + |1000011\rangle) \\ & - (|0010110\rangle + |0101010\rangle + |0011001\rangle + |0100101\rangle). \end{aligned}$$

But then the parity patterns of the new words are the same as those in the original representation and that means we can detect and correct a one-qubit σ_z error by making the same three measurements after the rotations. Since a σ_y error will be detected as an error in both bases, this encoding will detect the type and location of any single-qubit error.

Abstracting this example gets a bit tricky, since it requires two distinct but closely related classical codes, and we provide the details in Section 4.8. (See especially Example (4.32).) The key to the whole approach is that the Hadamard transform took a z-spin representation to an x-spin representation, and we confirm the generality of that fact.

(4.12) Proposition [68], [20]. Suppose C denotes the subspace, including the all-zeros word, of n-bit words defined by a linear code over *GF(2)*. Suppose $|\psi\rangle$ denotes an equally weighted superposition of the n-qubit words whose bit patterns in the z-spin basis correspond to those of the words in C. Then in the x-spin basis $|\psi\rangle$ is an equally weighted superposition of n-qubit words whose bit patterns correspond to $C^{\perp}$, the dual code of C defined as the set of n-bit words orthogonal to C, using the mod(2) inner product on binary vectors.

Proof: By assumption, and ignoring the normalizing constant,

$$|\psi\rangle = \sum_{w \in C} |w_1 \dots w_n\rangle = \sum_{w \in C} |w_1\rangle \otimes \cdots \otimes |w_n\rangle.$$

The representation in the x-spin basis can be modeled by writing each qubit in the x-spin basis or by applying R to each qubit. (This is just the Hadamard transform $R^{(n)}$ as used in Grover's algorithm and elsewhere.) Again, using $|k\rangle \rightarrow |0\rangle + (-1)^k|1\rangle$, we can collect terms in the calculation as follows:

$$\sum_{w \in C} R|w_1\rangle \otimes \cdots \otimes R|w_n\rangle = \sum_{i=0}^{2^n-1} \sum_{w \in C} (-1)^{i_1 w_1 + \cdots i_n w_n} |i_1\rangle \dots |i_n\rangle,$$

where the i_k's denote the bits in the binary expansion of i, with i_n now the low-order bit. Since the linear code is generated by k basis elements $v_1, \dots, v_k$, we can also express this as a product:

$$\sum_{w \in C} R|w_1\rangle \otimes \cdots \otimes R|w_n\rangle = \sum_{i=0}^{2^n-1} |i\rangle \prod_{r=1}^{k} \left(1 + (-1)^{i \cdot v_r}\right)$$

using the usual dot product over *GF(2)*. But then it is immediate that a state has the same nonzero coefficient if and only if i is in $C^{\perp}$, completing the proof. •

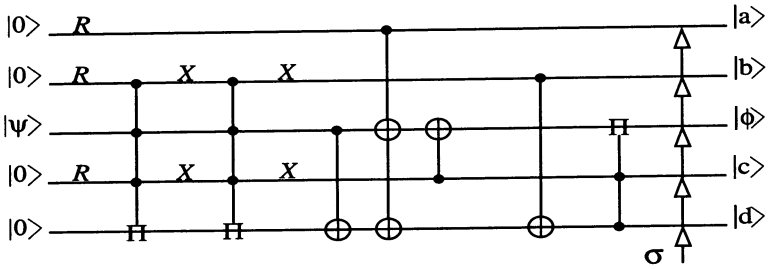
This approach using classical codes not only leads to the definition of the wiring diagram for the measurements, but it also enables one to define the initial encoding. Since both of those procedures can be determined as special cases of the stabilizer approach to be discussed below, we omit the details and defer the generalization of this approach to Section 4.8.

4.5 A five-qubit error-correcting code

Shor's encoding scheme requires nine qubits for each encoded piece of information whereas the Calderbank-Shor-Steane (CSS) code uses seven, and it's logical to ask if one could do as well with fewer qubits. For example, suppose one used a total of n qubits to encode one qubit's worth of information and wanted to be able to detect one error to the system which could be any of the three Pauli-type errors to each of the n qubits as well as the condition that no errors occurred. Having n qubits means that $n-1$ measurements are possible, giving 2^{n-1} error patterns, so a *perfect* code would have the minimal number of qubits: $3n+1 = 2^{n-1}$ or $n = 5$.

Motivated by this calculation and by the seven-qubit code given in [68], Laflamme et al. [49] found a five-qubit quantum error-correcting code that can detect the 16 possible one-qubit error occurrences and is not based on a classical error-correcting code. The strategy is similar to that used above: information from one qubit is encoded in words based on five qubits and a mechanism for detecting a subsequent error via four measurements is provided. In fact, the detection algorithm in this case is precisely the inverse of the encoding algorithm. That is, an input state is mapped into a linear combination of five-qubit words via an explicit wiring diagram. If a single Pauli error then occurs, running the circuit backwards and taking measurements on the four supplementing qubits defines a four-bit error syndrome that indicates which error occurred and the state of the middle qubit which is still in superposition. A subsequent unitary mapping transforms the middle qubit to its original form.

A five-qubit coding scheme was also presented independently in [12] and the relationship to the code in [49] is discussed by DiVincenzo and Shor [30]. We first begin with the algorithm of [49] and present the encoding scheme in the first diagram below. The actual code words differ slightly from the published version by a reversal of the signs of each of the first two words in the encoding of $|0\rangle$ and $|1\rangle$.



The notation follows past usage with the addition of "Π" denoting multiplication by (-1). The vertical arrows denote the possibility of a σ-error occurring *after* the encoding.

The initial encoding of $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ represented by the wiring diagram defines the mappings of the data qubits into a superposition of five-qubit words:

$$\begin{aligned} |0\rangle \;\rightarrow\; & |00000\rangle - |01111\rangle - |10011\rangle + |11100\rangle \\ & + |00110\rangle + |01001\rangle + |10101\rangle + |11010\rangle \end{aligned}$$

and

$$\begin{aligned} |1\rangle \;\rightarrow\; & |11111\rangle - |10000\rangle + |01100\rangle - |00011\rangle \\ & + |11001\rangle + |10110\rangle - |01010\rangle - |00101\rangle. \end{aligned}$$

assuming the encoding scheme above works *without* errors. Thus, $|\psi\rangle$ maps into a linear combination of the code words above.

(4.13) Exercise. Confirm that the asserted encoding of $|0\rangle$ and $|1\rangle$ is correct (optional). Note that there is a common even parity among a fixed pattern of four of the five qubits. •

Now assume that we wish to correct for a possible Pauli-type error which occurred after the encoding, as denoted by the arrows. In this algorithm we *first* run the (assumed error-free) network backwards to obtain $|a'\rangle|b'\rangle|\phi'\rangle|c'\rangle|d'\rangle$ and *then* measure the contents of the four appended qubits, obtaining the error syndrome $(a'b'c'd')$. That four-tuple indicates which single error occurred and also indicates the corresponding form of the middle qubit.

For example, suppose the syndrome is (0110). That signifies that a σ_x error affected the last qubit with the result that $|\phi'\rangle = -\alpha|1\rangle - \beta|0\rangle$, so that an appropriate unitary map can restore the original superposition. The calculations to determine the significance of all 16 syndromes and the related status of $|\phi'\rangle$ are not particularly enlightening and are omitted.

(Table 1 in [49] relates the syndromes to the possible Pauli errors and gives examples of the form of $|\phi'\rangle$.)

In (4.13) we asserted that a particular four-bit pattern has even parity in each of the five-qubit words resulting from the encoding. If a σ_x or σ_y error had occurred in one of those four qubits after the encoding which altered that parity, then that change would be detectable by a measurement of the eigenvalue of the product of the four related σ_z operators. In their study of the five-qubit code of [12], DiVincenzo and Shor [30] related the two encodings and showed that a four-bit error syndrome could be defined by four such parity checks, one of which was based on the parity in the [49] representation mentioned in (4.13). The encoding algorithm differs from the measurement algorithm and turns out to be preferable from the standpoint of minimizing the propagation of errors during the measurement process, a topic we touch on briefly below.

So far we have concentrated on correcting one-qubit errors which are modeled by an application of a Pauli spin matrix. In fact, we also have the following result, which shows in particular that any unitary one-qubit error can be corrected by the algorithm above.

(4.14) Theorem. Suppose we have an encoding, an entangling with ancillary qubits, a measurement scheme of the ancillary qubits and a corrective unitary mapping algorithm that distinguishes and corrects each of the four possible errors to one particular qubit caused by one of the Pauli spin operators or the identity operator. Then that schema will also correct an error modeled by an application of any unitary matrix to that qubit.

Proof. If A is in *SU(2)*, we have from Exercise (1.30)

$$A = a_0\sigma_0 + a_1 i\sigma_x + a_2 i\sigma_y + a_3 i\sigma_z,$$

where the coefficients are real and $\sum |a_k|^2 = 1$. Suppose we have an encoding $|\varphi\rangle$ and ancillary qubits initially in the zero state, so that the system is in state $|\varphi\rangle|0\rangle$. Assume the error A occurs so that

$$|\varphi\rangle|0\rangle \to a_0\sigma_0|\varphi\rangle|0\rangle + a_1 i\sigma_x|\varphi\rangle|0\rangle + a_2\sigma_y i|\varphi\rangle|0\rangle + a_3 i\sigma_z|\varphi\rangle|0\rangle.$$

By assumption, there is an entanglement such that a subsequent measurement of the ancillary qubits will distinguish among the σ's. Symbolically, this means that after the entanglement the system is in the state

$$a_0\sigma_0|\varphi\rangle|w_0\rangle + a_1 i\sigma_x|\varphi\rangle|w_1\rangle + a_2 i\sigma_y|\varphi\rangle|w_2\rangle + a_3 i\sigma_z|\varphi\rangle|w_3\rangle,$$

where the $|w_k\rangle$'s denote four orthogonal states. Once a measurement of the ancillary bits is made, w_k is known and the remaining part of the

system is in a state caused by the Pauli "error" associated with w_k. Because such an error is unitarily correctable, it follows that the system can be restored to its original state, and since any 2×2 unitary matrix can be written as a phase factor times a matrix in *SU(2)*, the proof is complete. •

4.6 Stabilizers and the five-qubit code

The five-qubit encoding representation of [12] was found by a computer search, but subsequently a general theory was developed which explains both the encoding and the error syndrome. The theory goes under the rubric of *stabilizer codes* and is presented from slightly different perspectives in papers such as [18], [19], [38] and [39]. We begin with a development of the five-qubit code from the stabilizer point of view, amplifying somewhat some of the related results as presented in [39] and providing a paradigm for general stabilizer codes. One possible novelty is the discussion between (4.20) and (4.21), although the conclusion itself is standard fare.

Each qubit defines a two-dimensional, "initial coordinate" subspace of a 2^5-dimensional complex Hilbert space H_5 defined by the tensor product. Encoding one "data" qubit is the same as mapping that two-dimensional, initial coordinate subspace to another two-dimensional subspace of H_5 with the proviso that $|0\rangle$ and $|1\rangle$ are encoded as orthonormal basis vectors of the two-dimensional subspace H_0: $H_0 = span(|0_e\rangle, |1_e\rangle)$, using the obvious notation. Assuming the encoding mapping is linear, the information in a superposition $\alpha|0\rangle + \beta|1\rangle$ is thus spread out across five qubits.

Let G_5 denote the set of matrices defined by five-fold tensor products of Pauli matrices:

$$\{\alpha\sigma_{k_1} \otimes \sigma_{k_2} \otimes_{k_3} \otimes\sigma_{k_4} \otimes \sigma_{k_5},\ k_i \in \{0, x, y, z\},\ \alpha \in \{\pm 1, \pm i\}\},$$

where σ_0 denotes the identity. Thus, together with scalar multiples of 1, -1, i and $-i$, G_5 represents the set of possible qubit errors with which we will be concerned, and the context is that the operators of G_5 act on the vectors of H_5.

(4.15) Exercise. With the operation between two elements of G_5 defined as the product of tensor products, prove that G_5 is a group of $4 \cdot 4^5$ Hermitian matrices. Show that if M and N are elements of G_5, then $MM = \pm I$ and $MN = \pm NM$. Find L, M and N such that $LMN = iI$,

thus showing the necessity of including the constant factors α in the definition of G_5. •

Following the precedent of making measurements of operators which are the tensor products of Pauli matrices, we select the following four elements of G_5:

$$\begin{aligned} M_1 &= \sigma_x \otimes \sigma_0 \otimes \sigma_x \otimes \sigma_z \otimes \sigma_z \\ M_2 &= \sigma_x \otimes \sigma_z \otimes \sigma_z \otimes \sigma_x \otimes \sigma_0 \\ M_3 &= \sigma_z \otimes \sigma_x \otimes \sigma_0 \otimes \sigma_x \otimes \sigma_z \\ M_4 &= \sigma_0 \otimes \sigma_x \otimes \sigma_z \otimes \sigma_z \otimes \sigma_x. \end{aligned} \tag{4.16}$$

(The reasoning behind the numbering of the subscripts will be explained below.) Each operator M_i has the property that $M_i^2 = I$, and using the fact that different Pauli matrices anticommute, it is easy to check that these four operators commute with one another. For example,

$$\begin{aligned} M_1 M_2 &= \sigma_0 \otimes \sigma_z \otimes (-i\sigma_y) \otimes i\sigma_y \otimes \sigma_z \\ &= \sigma_0 \otimes \sigma_z \otimes i\sigma_y \otimes (-i\sigma_y) \otimes \sigma_z = M_2 M_1. \end{aligned}$$

Since none of the four operators is the product of the others, it is also easy to check that they generate an Abelian subgroup S of order 16. The operators in S will be shown below to define an encoding such that each element in S leaves vectors in H_0 invariant. In fact, S contains all such mappings and is called the *stabilizer* of H_0.

(4.17) Exercise. Verify that the operator $\bar{X}$ also commutes with every M in S, where

$$\bar{X} = \sigma_x \otimes \sigma_x \otimes \sigma_x \otimes \sigma_x \otimes \sigma_x.$$

Optional: Find explicit representations of the members of the subgroup S generated by M_1, M_2, M_3 and M_4. Note that factors of σ_y always occur in pairs, so that factors of i occur in even powers. •

Now, suppose that we define the encoding of $|0\rangle$ by

$$|0_e\rangle = \sum_{M \in S} M|00000\rangle$$

and that of $|1\rangle$ by

$$|1_e\rangle = \bar{X} \sum_{M \in S} M|00000\rangle.$$

Using the explicit representation of the 16 elements in S, we are then able to derive the encoding of [12]:

$$\begin{aligned}|0_e\rangle &= |00000\rangle + |10100\rangle + |01010\rangle + |00101\rangle + |10010\rangle + |01001\rangle \\ &\quad - (|11000\rangle + |01100\rangle + |00110\rangle + |00011\rangle + |10001\rangle) \\ &\quad - (|01111\rangle + |10111\rangle + |11011\rangle + |11101\rangle + |11110\rangle)\end{aligned}$$

and

$$\begin{aligned}|1_e\rangle &= |11111\rangle + |01011\rangle + |10101\rangle + |11010\rangle + |01101\rangle + |10110\rangle \\ &\quad - (|00111\rangle + |10011\rangle + |11001\rangle + |11100\rangle + |01110\rangle) \\ &\quad - (|10000\rangle + |01000\rangle + |00100\rangle + |00010\rangle + |00001\rangle).\end{aligned}$$

Note that each word in $|0_e\rangle$ has even parity and each word in $|1_e\rangle$ has odd parity.

(4.18) Exercise. Use the definition of $|k_e\rangle$ and the fact that S is an Abelian group to confirm that for each M in S, $M|k_e\rangle = |k_e\rangle$, where k is 0 or 1. Thus, H_0 is in the $\lambda = 1$ eigenspace of each M in S. In addition, verify that $\langle 0_e|1_e\rangle = 0$. •

There is method in the foregoing madness. Suppose we let E denote one of the 15 "one-qubit-error" elements of G_5; that is, E is a tensor product of four identities and one Pauli matrix. Then it is easy to check that there is at least one generator M_k such that E and M_k anticommute: $EM_k = -M_kE$. For example, if

$$E = \sigma_x \otimes \sigma_0 \otimes \sigma_0 \otimes \sigma_0 \otimes \sigma_0,$$

then $EM_3 = -M_3E$. Since

$$\langle j_e|E|k_e\rangle = \langle j_e|EM_3|k_e\rangle = -\langle j_e|M_3E|k_e\rangle = -\langle j_e|E|k_e\rangle$$

for an arbitrary choice of j and k, E maps H_0 into a two-dimensional space H_a orthogonal to H_0. In fact, if E and F denote operators for different one-qubit errors, then there is an M_k in S such that EF also anticommutes with M_k, and the analogous calculation gives $\langle j_e|EF|k_e\rangle = 0$. Thus, each of the 15 one-qubit error operators maps H_0 into one of 15 mutually orthogonal two-dimensional subspaces of H_5.

(4.19) Example. The table below gives the set of one-qubit mappings of G_5 which anticommute with the corresponding generator. (We indicate the type of Pauli error together with the index of the qubit on which it

occurs.) Given two different one-qubit error mappings E and F, verify that there is a generator of S which anticommutes with the product EF.

Generator	$S_k = \{E : EM_k = -M_kE\}$
$M_1 = \sigma_x \otimes \sigma_0 \otimes \sigma_x \otimes \sigma_z \otimes \sigma_z$	$\{\sigma_{y1}, \sigma_{z1}, \sigma_{y3}, \sigma_{z3}, \sigma_{x4}, \sigma_{y4}, \sigma_{x5}, \sigma_{y5}\}$
$M_2 = \sigma_x \otimes \sigma_z \otimes \sigma_z \otimes \sigma_x \otimes \sigma_0$	$\{\sigma_{y1}, \sigma_{z1}, \sigma_{x2}, \sigma_{y2}, \sigma_{x3}, \sigma_{y3}, \sigma_{y4}, \sigma_{z4}\}$
$M_3 = \sigma_z \otimes \sigma_x \otimes \sigma_0 \otimes \sigma_x \otimes \sigma_z$	$\{\sigma_{x1}, \sigma_{y1}, \sigma_{y2}, \sigma_{z2}, \sigma_{y4}, \sigma_{z4}, \sigma_{x5}, \sigma_{y5}\}$
$M_4 = \sigma_0 \otimes \sigma_x \otimes \sigma_z \otimes \sigma_z \otimes \sigma_x$	$\{\sigma_{y2}, \sigma_{z2}, \sigma_{x3}, \sigma_{y3}, \sigma_{x4}, \sigma_{y4}, \sigma_{y5}, \sigma_{z5}\}$

Suppose $|\psi\rangle = \alpha|0_e\rangle + \beta|1_e\rangle$ is the encoded superposition, so that $|\psi\rangle$ is in H_0 and $M_k|\psi\rangle = |\psi\rangle$. If no error or an error not in S_1 has occurred, we would continue to measure an eigenvalue of $+1$, but if one of the errors in S_1 has occurred, the system is in the state $E|\psi\rangle$ so that $M_1E|\psi\rangle = -E|\psi\rangle$, and we would measure an eigenvalue of -1. Using "0" as code for eigenvalue $+1$ and "1" for eigenvalue -1, the results of measuring each of the eigenvalues of the four generators produces a code that uniquely specifies which one-qubit error occurred. For example, 1001 corresponds to $S_1 \cap S_2^c \cap S_3^c \cap S_4 = \{\sigma_{x4}\}$. Thus, assuming we can do the four required measurements and still preserve a superposition, we have related a unique four-bit error syndrome with a one-qubit error and the qubit on which it occurred. Note that if we allowed a larger class of errors, each four-bit error syndrome would correspond to a set of errors. •

(4.20) Exercise. Verify that the error syndromes 1100 and 1101 correspond to $\{\sigma_{z1}\}$ and $\{\sigma_{y3}\}$, respectively. *Optional*: Construct the table of error syndromes and the related Pauli errors. •

In order to compute the error syndrome, we need to make four measurements, one for each of the generators, and we also need to demonstrate an appropriate wiring diagram. Again, the structure of the generators is crucial. For example, if $|\psi\rangle$ is an eigenvector for M_2, then

$$M_2|\psi\rangle = (\sigma_x \otimes \sigma_z \otimes \sigma_z \otimes \sigma_x \otimes \sigma_0)|\psi\rangle = \pm|\psi\rangle.$$

Now define $\tilde{M}_2 = \sigma_z \otimes \sigma_z \otimes \sigma_z \otimes \sigma_z \otimes \sigma_0$ and use Exercise (2.8) to obtain $R\sigma_x = \sigma_z R$ and thus

$$(R \otimes \sigma_0 \otimes \sigma_0 \otimes R \otimes \sigma_0)M_2 = \tilde{M}_2(R \otimes \sigma_0 \otimes \sigma_0 \otimes R \otimes \sigma_0).$$

This gives the key result

$$\tilde{M}_2|\phi\rangle = \pm|\phi\rangle \Leftrightarrow M_2|\psi\rangle = \pm|\psi\rangle,$$

where $|\phi\rangle = (R \otimes \sigma_0 \otimes \sigma_0 \otimes R \otimes \sigma_0)|\psi\rangle$. In other words, to determine the eigenvalue of $|\psi\rangle$ with respect to M_2, it is equivalent to rotate the

first and fourth qubits, obtaining $|\phi\rangle$, and then determine the eigenvalue of $|\phi\rangle$ with respect to the operator $\tilde{M}_2$. But $\tilde{M}_2$ operates on a five-qubit word by multiplying it by $(-1)^k$, where k is the number of 1's in the first four positions of the word. It follows that each word in the presentation of an encoding state *after the two rotations* must have the same parity in the first four positions. In particular, if the eigenvalue is $+1$, each word must have even parity.

(4.21) Example. Let $|\phi_k\rangle = (R \otimes \sigma_0 \otimes \sigma_0 \otimes R \otimes \sigma_0)|k_e\rangle$, referring to the encoding above (4.18). Then it can be painfully shown that

$$\begin{aligned} |\phi_0\rangle \;=\; & |00000\rangle + |10011\rangle + |00110\rangle + |00111\rangle + |01011\rangle + |10010\rangle \\ & - (|01010\rangle + |01100\rangle + |01101\rangle + |00001\rangle + |10100\rangle + |11110\rangle) \\ & + |10101\rangle + |11000\rangle + |11001\rangle + |11111\rangle \end{aligned}$$

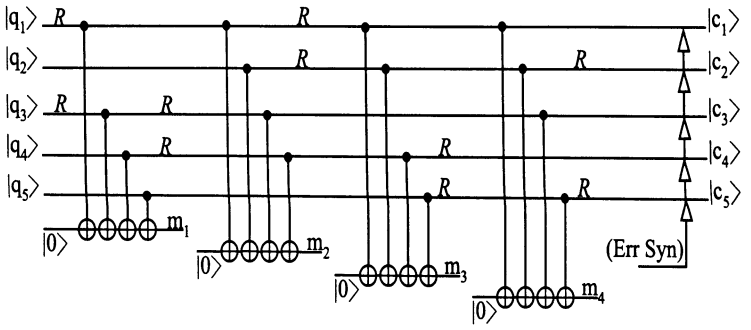
and

$$\begin{aligned} |\phi_1\rangle \;=\; & |11111\rangle + |10010\rangle + |00111\rangle + |11001\rangle + |11110\rangle + |01101\rangle \\ & - (|00001\rangle + |00110\rangle + |11000\rangle + |00000\rangle + |10011\rangle) \\ & - (|10100\rangle + |10101\rangle + |01010\rangle + |01011\rangle + |01100\rangle) \end{aligned}$$

confirming what we believed in the first place: each word has even parity in the first four positions. (This approach differs slightly from that in [30], where an initial rotation on the encoded vectors led to eight words in the presentation.) •

This analysis also suggests the way to define a wiring diagram for the four measurements. Given the encoding defined above, for each generator we rotate appropriate pairs of qubits, apply four *XOR* gates to an ancillary qubit and measure the result. The resulting error syndrome defines which Pauli matrix to apply to which qubit to correct a one-qubit error. *And* the reason for the subscripts on the generators is that we wished to minimize the number of times that R is required. We rotate qubits 3 and 1 for the first measurement, 3 and 4 for the second, 1 and 2 for the third and 4 and 5 for the fourth. Performing the measurements in that order means we only need to rotate and "unrotate" each qubit once.

The result is the diagram below, which is essentially the same as that in [30].



The five-qubit array is augmented by four ancillary qubits initially set to 0. The first two rotations are applied to qubits 1 and 3, and the first measurement is made. Rotations are applied to qubits 3 and 4, and the second measurement made and so forth. After the final measurement, the last two rotations are applied and a one-qubit error-correcting mapping is made based on the four-bit error syndrome. The result is the correction of any one-qubit error.

It remains to construct a wiring diagram implementing the encoding, and that can also be deduced from the generators. First of all, note that the choice of generators is not unique, since we could use appropriate multiples of the original generators to define a new set of generators. For example, we will see later the motivation for choosing

$$\begin{aligned} N_1 &= M_2M_3M_4 = \sigma_y \otimes \sigma_z \otimes \sigma_0 \otimes \sigma_z \otimes \sigma_y \\ N_2 &= M_4 = \sigma_0 \otimes \sigma_x \otimes \sigma_z \otimes \sigma_z \otimes \sigma_x \\ N_3 &= M_1M_2M_3M_4 = \sigma_z \otimes \sigma_z \otimes \sigma_x \otimes \sigma_0 \otimes \sigma_x \\ N_4 &= M_3M_4 = \sigma_z \otimes \sigma_0 \otimes \sigma_z \otimes \sigma_y \otimes \sigma_y. \end{aligned}$$

The encoding scheme works equally well with the N_k's as generators, and we observe that

$$|0_e\rangle = \sum_{N \in S} N|00000\rangle = \prod_{k=1}^{4}(I + N_k)|00000\rangle.$$

(We used essentially the same trick in the proof of Proposition (4.12).)

Recall that $|1_e\rangle$ is obtained from $|0_e\rangle$ by applying $\bar{X} = \sigma_x \otimes \sigma_x \otimes \sigma_x \otimes \sigma_x \otimes \sigma_x$. Equivalently,

$$|1_e\rangle = \sum_{N \in S} N\bar{X}|00000\rangle = \sum_{N \in S} NM\bar{X}|00000\rangle,$$

where M is any mapping in S, and we will see the motivation for choosing M to be $N_1N_2N_3N_4$. Defining $\tilde{X} = M\bar{X}$, we have

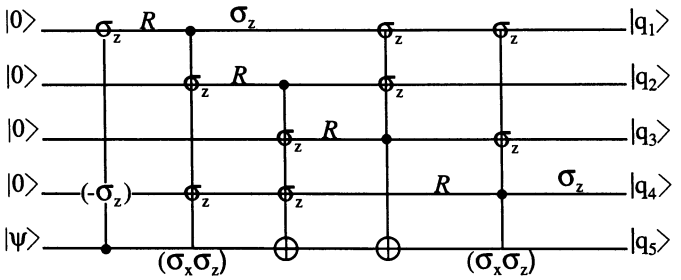
$$\tilde{X} = (-1)\sigma_z \otimes \sigma_0 \otimes \sigma_0 \otimes \sigma_z \otimes \sigma_x,$$

and the encoding can be expressed as

$$\alpha|0\rangle + \beta|1\rangle \to \prod_k (I + N_k)\left(\alpha|00000\rangle + \beta\tilde{X}|00000\rangle\right),$$

which will help define an explicit wiring diagram.

Each of the N_k's contains a bit flip operation only on the k'th qubit and on the last qubit, and $\tilde{X}$ contains a bit flip on the last qubit and nowhere else. Thus, if we begin with the initial state $\alpha|00000\rangle + \beta\tilde{X}|00000\rangle$ and sequentially apply in order $I + N_k$, we will know that the k'th qubit is in state $|0\rangle$ prior to the k'th mapping. But then it is easy to confirm that the action of $I + N_k$ can be implemented by applying the rotation R to the k'th qubit and then the conditional operations defined by N_k. This leads to the following wiring diagram for the initial encoding of $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$. (Note the analogous structure of the finite Fourier transform in Section 3.6.)



The diagram is written so that several conditional operations are collected on one vertical line, and the preceding explanation rationalizes each operation except for the application of the two unconditional σ_z operations on qubits 1 and 4 and the use of $\sigma_x\sigma_z$ in place of σ_y. Both N_1 and N_4 flip bits using a σ_y mapping rather than a σ_x mapping. Since there are two σ_y mappings in each of N_1 and N_4, there are two factors of i unaccounted for, and the unconditional application of σ_z together with the conditional use of $\sigma_x\sigma_z$ makes the necessary correction to the phase.

(4.22) Exercise. Verify that the use of the rotation R, of σ_z, and the subsequent conditional command actually does implement the operator $I + N_1$. •

(4.23) Exercise. Confirm that $\bar{Z} = \sigma_z \otimes \sigma_z \otimes \sigma_z \otimes \sigma_z \otimes \sigma_z$ also commutes with each of the generators. Show that $\bar{X}$ and $\bar{Z}$ act like σ_x and σ_z on the encoded states $|0_e\rangle$ and $|1_e\rangle$ and in particular that they map the span of the encoded vectors to itself. •

4.7 Theoretical aspects of stabilizer codes

In working through the five-qubit code from the perspective of its stabilizer, we have seen how the generators defined the encoding scheme and its wiring diagram, the error measurement wiring diagram and the error syndrome. None of these capabilities is peculiar to that particular example, and in this section we isolate the relevant theoretical properties that permit the generalization of the approach. Although some of the theoretical results we derive are standard fare, we provide details for completeness and note that the paper of [18] provides the linkage with a different perspective. In addition [19] gives a thorough analysis of quantum error-correcting codes using *GF(4)* as the base field.

Let H_n denote a 2^n-dimensional Hilbert space over the complex numbers, and let G_n denote the group of matrices defined as n-fold tensor products of the Pauli spin matrices, including the 2×2 identity matrix σ_0, together with the scalar factors ± 1 and $\pm i$. As before, G_n models the errors on individual qubits and acts on the qubit space H_n.

(4.24) Lemma. Suppose S is a subgroup of G_n which fixes each vector in a subspace H_0 of H_n. Then S is an Abelian group of order 2^d for some integer d and is generated by a set of d matrices M_1, M_2,..., M_d. Each M in S is self-invertible, and if M is in S, then aM is not in S unless $a = 1$. In particular, $-I$ is not in S.

Proof: Since G_n has order $4 \cdot 4^n$, it follows that the order of S is also a power of 2. As in Exercise (4.15), $MM = \pm I$ and $MN = \pm NM$, and since mappings in S fix the vectors in H_0, it follows that MM must be $+I$ for M in S. Similarly, MN must equal NM, and if both M and aM are in S, then $aMM = aI$ is also in S. But aI doesn't fix the vectors in H_0 unless a equals 1. For the last assertion, note if M_1 is in S, then $\{M_1\}$ generates a subgroup of S of order 2. If M_2 is in $S - \{I, M_1\}$, then it is easy to check that $\{M_1, M_2\}$ generates a larger subgroup of order 4. Continuing in this fashion, we ultimately find d matrices which generate all of S, completing the proof. •

The subspace H_0 will be the space of k encoded qubits and, as in Section 4.6, will actually be defined by a subgroup S with the properties of (4.24). Henceforth we refer to a group with those properties as a *stabilizer*

group. The relevant fact is that H_0 will be 2^k dimensional, where $d+k = n$, thus relating d, the number of generators, and n, the number of qubits available, to k, the number of qubits that can be encoded.

(4.25) Lemma. Suppose S is a stabilizer subgroup of G_n with d generators. Then there is a 2^{n-d}-dimensional space H_0 whose vectors are 1-eigenvectors for each M in S. Moreover, M is in S if and only if every vector in H_0 is a 1-eigenvector for M, and every vector that is a 1-eigenvector of every M in S is in H_0.

Proof: Each of the d generators M_k of S is a tensor product of Pauli matrices, and it is easy to check that M_k has eigenvalues ± 1 and that each eigenspace is 2^{n-1} dimensional, proving the first assertion for $d = 1$. Since the matrices in S commute, they can be simultaneously diagonalized. Thus, H_n can be decomposed into four subspaces on which the eigenvalue pattern of M_1 and M_2 is $(1,1)$, $(1,-1)$, $(-1,1)$ and $(-1,-1)$, respectively. Since $M_1 M_2$ also has eigenvalues ± 1, it is easy to check that each of those subspaces has dimension 2^{n-2}. Continuing in this way, we find that the d generators define 2^d eigenspaces of dimension 2^{n-d}, each of which is defined by the eigenvalue pattern of the generators. In particular, H_0, the common 1-eigenspace of all d generators, must have dimension 2^{n-d}. Since every matrix in S is a multiple of the generators, it follows that H_0 is also a 1 -eigenspace for every M in S.

Finally, suppose every vector in H_0 is a 1-eigenvector for some M and M is not in S. Then M would define a $(d+1)$st generator, and the preceding argument shows that M would have to have (-1)-eigenvectors in H_0, completing the proof. •

We saw in Exercise (4.23) for the five-qubit example that $\bar{X}$ and $\bar{Z}$ commuted with each M in S but did not fix the vectors in H_0. However, both $\bar{X}$ and $\bar{Z}$ do map H_0 to itself. A relevant group-theoretic definition and result are contained in the next lemma.

(4.26) Lemma. Let S be a stabilizer group with d generators. Let *C(S)* denote $\{N\colon MN = NM$ for all M in $S\}$, the *centralizer* of S, and let *N(S)* denote $\{N\colon NS = SN\}$, the *normalizer* of S. Then *N(S)* = *C(S)*, and operators in *N(S)* map H_0 to itself.

Proof: Since $MN = \pm NM$ for every M and N, if N is in *N(S)* - *C(S)*, there must be an M in S for which $MN = -NM$. Since N is in the normalizer, that means that $-NM$ must be in S, contradicting the assumption that S is a stabilizer group. If N is in *C(S)*, it is easy to see that for every v in H_0, Nv is a 1-eigenvector for every M in S; hence Nv must be in H_0, completing the proof. •

Up to the factor of i, σ_y can be represented as $\sigma_x\sigma_z$. Thus, we can think of σ_z as $(0,1)$, σ_x as $(1,0)$ and σ_y as $(1,1)$. Each generator M_k can then be represented as two binary n-vectors, one of which indexes the occurrences of σ_x while the other indexes the occurrences of σ_z. If both vectors have a 1 in the same position, we can interpret that as an occurrence of σ_y. Thus, for example, the two vectors associated with the mapping $M_1 = \sigma_x \otimes \sigma_0 \otimes \sigma_x \otimes \sigma_z \otimes \sigma_z$ of (4.19) in the five-qubit code are $X_{M_1} = (1,0,1,0,0)$ and $Z_{M_1} = (0,0,0,1,1)$. In that same section $(0,0,0,1,1)$ and $(1,0,1,1,1)$ are the respective X- and Z-vectors of

$$N_4 = M_3M_4 = \sigma_z \otimes \sigma_0 \otimes \sigma_z \otimes \sigma_y \otimes \sigma_y.$$

Now N_4 was derived by multiplying two of the original generators together, and its X- and Z-vectors could have been derived by performing *GF(2)* arithmetic coordinate by coordinate on the X- and Z-vectors of the generators. Conversely, combining the X- and Z-vectors of the generators defines N_4 up to factors of i and (-1). Since we require S to be Abelian, we don't have to worry about factors of (-1) arising from issues of commutativity. Since the σ_y factors appear an even number of times in each element of S, the number of σ_y factors determines the sign of the product arising from factors of i, and that can also be inferred from the X- and Z-vectors. Note also that each element in S can be represented as a tensor product of real matrices with appropriate factors of (-1). Those remarks constitute an outline of the proof of the next result.

(4.27) Lemma. Suppose S is a real stabilizer group, i.e., factors of σ_y appear an even number of times in each operator in S. If M and N are in S and $R = MN$, then $(X_R, Z_R) = ((X_M \oplus X_N), (Z_M \oplus Z_N))$. •

This observation means that we can represent operators in S as binary vectors and group operations in S as linear operations on the representing binary vectors. In particular, if we represent the d generators of S in a $d \times 2n$ matrix, we are allowed to perform row operations on the matrix to obtain a standard representation of the generators. (As noted earlier, these mathematical insights are articulated in [18] and in detail in [23] and [39].) In the particular example of the five-qubit code, we can represent the X- and Z-vectors of the four M and N generators as

M	X	Z
1	10100	00011
2	10010	01100
3	01010	10001
4	01001	00110

$\rightarrow$

N	X	Z
1	10001	11011
2	01001	00110
3	00101	11000
4	00011	10111

where the right-hand side is obtained using elementary row operations and mod(2) arithmetic. Note that we may also permute columns, if necessary, since that is only a relabeling of the qubits.

We now treat the general case, following the approach in [39]. Suppose we have d generators of a real stabilizer group S. Then we can use row operations, and column permutations if necessary, to obtain the following $d \times 2n$ array:

	X-Block			*Z-Block*		
	r	$d-r-s$	$n-d+s$	r	$d-r-s$	$n-d+s$
r	I	A_2	A_3	B_1	0	B_3
$d-r-s$	0	0	0	C_1	I	C_3
s	0	0	0	D	0	0

The $(d-r-s) \times (d-r-s)$ identity in the *Z-block* was defined using row operations on generators whose σ_x factors had been eliminated in the derivation of the $r \times r$ identity in the *X-block*. If s is positive, then there is a mapping in S whose only nonidentity factors are σ_z operators in the first r positions. But such mappings cannot commute with the reduced operators of the first r rows; hence, D is the zero matrix. Since we assume the generators are independent, we can't have a row of zeros and that means $s = 0$.

As an example, the reduced form for the five-qubit code is in this standard form in the "N-table" above, where $r = d = 4$ and the last bit of the five-bit X- and Z-vectors is in the "$n-d$" column. The 4×4 array defined by the first four bits of the X-vectors of each of the four N-operators corresponds to the $r \times r$ identity in the *X-block* of the standard form, and that was the rationale for defining the N-operators in the first place.

It is easy to confirm that the commutativity of two operators can be expressed in terms of their X-Z representation as

$$\sum_k ((X_M(k) \cdot Z_N(k)) \oplus (Z_M(k) \cdot X_N(k))) = 0,$$

using the mod(2) inner product and mod(2) addition. In terms of the submatrices above, one can show that this becomes

$$(I, A_2, A_3, B_1, 0, B_3) \cdot (C_1, I, C_3, 0, 0, 0)^{\mathrm{t}} = 0,$$

where the superscript denotes the transpose, which also applies to the submatrices, i.e., a column of transposed matrices. Equivalently,

$$C_1 = (C_3 \cdot A_3^{\mathrm{t}}) \oplus A_2^{\mathrm{t}}.$$

It's obvious that reduced generators with only σ_z components commute with themselves, and the commutativity of the reduced generators in the first r rows is summarized by

$$(I, A_2, A_3, B_1, 0, B_3) \cdot (B_1, 0, B_3, I, A_2, A_3)^t(i, j) = 0$$

for all $i \neq j$. The remaining condition is that σ_y factors appear an even number of times, and that translates to the mod(2) equation

$$B_1(i,i) = \sum_k A_3(i,k)B_3(i,k)$$

for each of the first r rows.

We can milk some additional information from this approach and characterize the normalizer *N(S)*. The resulting representation will then enable us to encode k qubits. Rather than go through the derivation, we instead summarize a variation of the results of the analysis in [39] and confirm the existence of an additional $2k$ operators which, together with S, generate *N(S)*.

(4.28) Proposition. Let S be a real stabilizer group defined by d generators in G_n. Then *N(S)* has $n + k$ generators, where $k = n - d$, and defines an encoding scheme mapping k qubits into n qubits. S and *N(S)* can be represented in the following standard reduced form of the *X-Z* representation, where the submatrices satisfy the indicated constraints:

	X-Block			*Z-Block*		
	r	$d-r$	k	r	$d-r$	k
r	I	A_2	A_3	B_1	0	B_3
$d-r$	0	0	0	C_1	I	C_3
k	0	C_3^t	I	B_3^t	0	0
k	0	0	0	A_3^t	0	I

and using mod(2) arithmetic $C_1 = (C_3 \cdot A_3^t) \oplus A_2^t$, $B_1 \oplus (B_3 \cdot A_3^t)$ is symmetric, and finally

$$B_1(i,i) = \sum_k A_3(i,k)B_3(i,k) \bmod (2)$$

for $1 \leq i \leq r$.

Proof: We have already confirmed the validity of the first two row blocks. The condition on C_1 is equivalent to commutativity between the generators in those blocks and the second condition is equivalent to commutativity within the first row block. The third condition states that an even number of σ_y factors appears in each reduced generator.

Think of the third row block as denoting k generators for $\bar{X}$ -type operators in *N(S)* and the fourth row block as denoting k generators for $\bar{Z}$-type operators. Then it is easy to check that those generators commute with every reduced generator of S and furthermore do not contain any σ_y terms. By construction these generators are independent and satisfy the commutativity condition except for the i'th pair in each block, since $\bar{X}_i$ and $\bar{Z}_i$ anticommute: $\bar{X}_i \cdot \bar{Z}_i = -\bar{Z}_i \cdot \bar{X}_i \bmod (2)$.

Finally, suppose N is a mapping in *N(S)*, and append its X-Z representation as a new row. If N is in S, we know that row can be reduced to a row of zeros. If N is not in S, we can still apply row operations using S generators to transform its first r X-entries and its middle $d-r$ Z entries to zero. Using the k-type generators we can map the last k X-entries to 0 and then use the k-type generators to map the last k Z-entries to 0. Thus, the reduced vector looks like $(0u0v00)$, where u and v denote $d-r$- and r-long vectors, respectively. We leave it to the reader to confirm that the requirement of commutativity with the generators of S forces those vectors to equal zero as well. Hence, every N in *N(S)* can be represented as the product of $n-k$ S generators and $2k$ *N(S)* - S generators, completing the proof. •

The foregoing analysis for the five-qubit case gave $r = d = 4$, so that there are no reduced Z generators. We denote the corresponding dimension as zero and the entries as null entries. The last two rows denote the reduced form of $\bar{X}$ and $\bar{Z}$, respectively, and the representation of *N(S)* for the five-qubit code has the following canonical form:

	X-Block			*Z-Block*		
	r	$d-r$	k	r	$d-r$	k
N_1	1000	ϕ	1	1101	ϕ	1
N_2	0100	ϕ	1	0011	ϕ	0
N_3	0010	ϕ	1	1100	ϕ	0
N_4	0001	ϕ	1	1011	ϕ	1
$d-r$	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ
$\bar{X}$	0000	ϕ	1	1001	ϕ	0
$\bar{Z}$	0000	ϕ	0	1111	ϕ	1

where we use ϕ to denote "no entry."

(4.29) Example. (*The CSS seven-qubit code*) Refer back to the wiring diagram of Section 4.4 and rearrange the ordering of the qubits from (1234567) to (5671243). In the new ordering we can read off the X-vectors of the first three generators from the three measurements as (1000111),

(0101011), and (0011110) and the Z-vectors of the last three generators have the same pattern. In this case d equals 6, r equals 3 and k equals 1. We leave it to the reader to confirm that the canonical (reduced) representation of the generators of $N(S)$ in this case is given by the 8×14 array below. The fact that the code is based on separate X and Z measurements remains clear, but the definition of the canonical representation obscures the fact that they have the same structure.

	X-Block			*Z-Block*		
	r	$d-r$	k	r	$d-r$	k
N_1	100	011	1	000	000	0
N_2	010	101	1	000	000	0
N_3	001	111	0	000	000	0
N_4	000	000	0	101	100	1
N_5	000	000	0	011	010	1
N_6	000	000	0	111	001	0
$\bar{X}$	000	110	1	000	000	0
$\bar{Z}$	000	000	0	110	000	1

We have now reached the point where we can see how the five-qubit case served as a paradigm for defining the encoding wiring diagram. Denote the k $\bar{X}$-type generators as $\bar{X}_i$, where the subscript corresponds to the i'th coordinate in a k-long qubit to be encoded. Then following the same sequential strategy of encoding, prepare the states to be encoded by the map

$$\sum_{j=0}^{2^k-1} \alpha_j |j\rangle \to \sum_{j=0}^{2^k-1} \alpha_j \bar{X}_1^{j_{k-1}} \cdot \ldots \cdot \bar{X}_1^{j_0} |0 \ldots 0\rangle,$$

where the superscripts correspond to the binary expansion of the state index. Since each $\bar{X}$-type operator makes a base change in only one of the last k qubits, we can use the same techniques used in the five-qubit case to define a wiring diagram. Note that these operators may also introduce base changes in the middle $d-r$ qubits but not in the first r qubits. Since the operators also commute with the generators of S, it follows as before that the overall encoding is linear and that the same strategy of applying the rotation R and using conditional operations on the first r qubits will lead directly to a feasible wiring diagram encoding k qubits in n qubits. (See [23] and [39] for more details.)

In a similar fashion, the techniques used in defining the measurements to detect errors in the five-qubit case also apply in general. We determine

sets of error mappings that anticommute with each of the generators, and make d measurements by applying a suitable tensor product of rotation operators to bring the system to the appropriate state for a parity check in the computational basis. If the resulting error syndrome uniquely defines one of the allowed potential errors, then a corrective operation can be made.

4.8 CSS codes

In the preceding section we showed the centrality of the generators of the stabilizer group to the analysis and implementation of a quantum error-correcting code. In this section we follow the ideas in Calderbank and Shor [20] and in Steane [68], [69] and generalize the example of Section 4.4 to show how to use classical codes to define the generators of a quantum error-correcting code. The basic idea is to use two classical codes, one to correct σ_x errors and one to correct σ_z errors, in such a way that the corrections of one type of error do not affect the corrections of the other type of error.

Here is the context. Assume that C_1 and C_2 are two classical linear error-correcting codes in F_2^n, the linear space of n-long vectors over *GF(2)*, and that $C_2 \subset C_1$, or equivalently $C_1^\perp \subset C_2^\perp$. (Recall that if C is a set of vectors, $C^\perp$ is the linear space of orthogonal vectors relative to the usual mod(2) dot product. The notation $[n, k, d]$ denotes a code whose words are n-bits long in a k-dimensional subspace, where the number of differences between any two code words is at least d.) In the example of Section 4.4, C_1 is the [7,4,3] Hamming code and C_2 is the [7,3,4] simplex code. Assume that $C_2^\perp$ is an $[n, n-r, e_2]$ code and that C_1 is an $[n, n-s, e_1]$ code, so that necessarily $s + r \leq n$. If t_i is defined as $\lfloor \frac{e_i - 1}{2} \rfloor$, the largest integer less than or equal to $(\frac{e_i - 1}{2})$, then $C_2^\perp$ is a t_2 error-correcting code and C_1 is a t_1 error-correcting code.

(4.30) Theorem. Let C_1 and C_2 be as above and suppose $r + s < n$. Then there is a quantum, $\min(t_1,t_2)$ error-correcting stabilizer code C which has $r+s$ generators and which encodes $n-r-s$ qubits. In addition, C corrects up through t_1 σ_z errors and t_2 σ_x errors.

Proof: Let $\{u_1, \ldots, u_r\}$ be a basis of n-vectors for C_2, and let $\{v_1, \ldots, v_s\}$ be a basis for $C_1^\perp$. For each vector w in F_2^n define

$$M_w(x) = \sigma_x^{w_1} \otimes \cdots \otimes \sigma_x^{w_n},$$

where $\sigma_x^0 = \sigma_0$, and define $M_w(z)$ analogously using σ_z. If $|b\rangle$ is a qubit with an n-long pattern of 0's and 1's corresponding to b, then it is easy

to check that

$$M_w(x)|b\rangle = |b \oplus w\rangle \quad \text{and} \quad M_w(z)|b\rangle = (-1)^{b\cdot w}|b\rangle. \tag{4.31}$$

Define S as the subgroup generated by

$$\{M_{u_i}(z), M_{v_j}(x), 1 \le i \le r, 1 \le j \le s\}.$$

Since each u_i is in C_2 and each v_j is in $C_1^\perp \subset C_2^\perp$, using mod(2) arithmetic it follows that $u_i \cdot v_j = 0$. That means that the number of positions in which $M_{u_i}(z)$ and $M_{v_j}(x)$ both have nonidentity factors is even, and hence the operators commute. Since each of the generators is its own inverse, we can conclude that S is an Abelian group, and that each M in S can be written as the product $M_u(z)M_v(x)$ for some n-long vectors u in C_2 and v in $C_1^\perp$. It is then easy to see that if both M and aM are in S, $a = 1$ and we can conclude that S is a stabilizer group.

$C_1^\perp$ is a normal subgroup of $C_2^\perp$ and if w is in $C_2^\perp$, then $w \oplus C_1^\perp$ is a coset of $C_1^\perp$ in $C_2^\perp$. For each such w define

$$|s_w\rangle = \sum_{v \in C_1^\perp} |v \oplus w\rangle = \sum_{v \in C_1^\perp} M_v(x)|w\rangle,$$

ignoring the normalizing constant. If $w_1 \oplus w_2$ is not in $C_1^\perp$, it is easy to check that

$$\langle s_{w_1}|s_{w_2}\rangle = \sum_{v_1 \in C_1^\perp} \sum_{v_2 \in C_1^\perp} \langle v_1 \oplus w_1|v_2 \oplus w_2\rangle = 0,$$

while if $w_1 \oplus w_2$ is in $C_1^\perp$, it is also easy to check that $|s_{w_1}\rangle = |s_{w_2}\rangle$. Hence, there are as many orthogonal states $|s_w\rangle$ as there are distinct cosets of $C_1^\perp$ in $C_2^\perp$, and that is precisely $2^{\dim(C_1)-\dim(C_2)} = 2^{n-r-s}$. It follows that using a suitable identification of coset representatives with orthogonal computational states of $n-r-s$ qubits, we can consider the span of the $|s_w\rangle$'s to be the encoding space H_0. Since S has $r+s$ generators, if we can show that each M in S leaves each state in H_0 invariant, then by the general results of the preceding section, we know that S is the stabilizer group for H_0.

To do that, assume v_0 is in $C_1^\perp$. Then for every w in $C_2^\perp$,

$$M_{v_0}(x)|s_w\rangle = M_{v_0}(x) \sum_{v \in C_1^\perp} |v \oplus w\rangle = \sum_{v \in C_1^\perp} |v_0 \oplus v \oplus w\rangle = |s_w\rangle.$$

Suppose that u is in C_2 and again that w is in $C_2^\perp$. Then

$$M_u(z)|s_w\rangle = M_u(z) \sum_{v \in C_1^\perp} |v \oplus w\rangle = \sum_{v \in C_1^\perp} (-1)^{u\cdot(v\oplus w)}|v \oplus w\rangle = |s_w\rangle.$$

Hence the generators of S leave each state in H_0 invariant, and S is the stabilizer group of H_0.

Recall that we measure σ_x errors by performing σ_z measurements in the computational basis. Since $C_2^\perp$ is a classical t_2 error-correcting code, the r measurements using the $M_{u_i}(z)$ operators will detect t_2 σ_x errors. Analogously, we detect σ_z phase errors by using σ_x measurements and since C_1 is a classical t_1 error-correcting code, measurements involving the $M_{v_j}(x)$ operators will detect t_1 σ_z errors. Since a quantum error-correcting code must also detect σ_y errors, the resulting code is a $\min(t_1,t_2)$ error-correcting code, and the proof is complete. •

(4.32) Example. In the special case discussed in Section 4.4, C_1 is the $[7,4,3]$ Hamming code, C_2 is the $[7,3,4]$ simplex code and the roles of the codes and their "perps" is a bit tricky. For example, the encoding of $|0\rangle$ involves a sum over $C_1^\perp$ in this section, but in Section 4.4 it appears as if the sum were over C_2. Since $C_1^\perp$ equals C_2 in this case, the encodings coincide. •

(4.33) Exercise. We have assumed that the $|s_w\rangle$ are defined in the computational basis. To correct σ_z errors, we transform to the x-spin basis using the Hadamard transform $R^{(n)}$. Show that $R^{(n)}$ takes $|s_w\rangle$ to

$$|c_w\rangle = \sum_{u \in C_1} (-1^{u \cdot w})|w\rangle.$$

(See Proposition (4.12).) •

(4.34) Exercise. In the notation of this section, the encoding of $|0\rangle$ is

$$|s_0\rangle = \sum_{v \in C_1^\perp} |v\rangle.$$

Show that this representation is equivalent to the stabilizer encoding

$$|0_e\rangle = \sum_{N \in S} N|0 \ldots 0\rangle.$$ •

4.9 Abstract quantum error correction

In the preceding sections we have concentrated on specific encoding algorithms and the related wiring diagrams. In this section we take a different perspective and consider an abstract theory of quantum error-correcting

codes. This perspective is intimately related to quantum information theory and to quantum communication, but it is possible to develop a general theory without first defining and discussing salient topics such as entropy and channel capacity. Our goal is to give necessary and sufficient conditions for a given encoding to correct errors modeled by a given set of operators. We follow the development given in Knill and Laflamme [47], noting that related analyses appear in Ekert and Macchiavello [34] and in [12].

Suppose that we are given as a coding space a Hilbert space H which contains an encoding space, a sub-Hilbert space H_0 which contains encoded information and which we wish to preserve intact. We will be working with a class A of error-causing operators, each of which maps H_0 into H. In order to correct the errors caused by operators in A, we may need an ancillary space $H(anc)$ and operations which unitarily map $H \otimes H(anc)$ to itself. Subsequently, measurements can be made on the ancillary space, and based on those measurements corrective unitary mappings of H made. The assumption we make is that from the perspective of H, the measurements and corrective actions can be modeled by *recovery mappings*, which are projections followed by unitary maps.

For example, consider the five-qubit, one error-correcting *QECC* discussed in Section 4.6. In this example the coding space H is the 32-dimensional space defined by the five qubits, and the encoding space H_0 is the 2-dimensional subspace defined by the encoding of $|0\rangle$ and $|1\rangle$. The set A contains 16 operators: the 15, one-qubit error operators A_a and the identity operator A_0. We have seen that A_a maps H_0 to H_a and that those 16 2-dimensional spaces are orthogonal. Subsequent to a putative error mapping, unitary mappings involving auxiliary qubits are applied, and four measurements are made which have the effect of identifying a and projecting the system onto one of the H_a. The corrective unitary mapping M_a in this particular case coincides with $A_a^{-1} = A_a$, since each of the error mappings is defined on all of H and is its own inverse.

Let R denote the set of recovery mappings, which in the five-qubit example consists of 16 recovery mappings

$$R_r = M_r(|0_r\rangle\langle 0_r| + |1_r\rangle\langle 1_r|)$$

and the index set for the recovery mappings happens to coincide with that of the error mappings. Then the error-followed-by-recovery process in the five-qubit case can be modeled as

$$|k_0\rangle = \sum_r R_r A_a |k_0\rangle = \sum_r M_r(|0_r\rangle\langle 0_r| + |1_r\rangle\langle 1_r|)A_a|k_0\rangle$$

for each a and for each of the two values of k_0. Since we can always normalize a state, we could generalize somewhat by allowing the error-followed-by-recovery process to produce a multiple of the original state. Equivalently, if ρ_i is any density matrix defined by states in H_0, such as $\sum_{k_0} p_{k_0}|k_0\rangle\langle k_0|$, we could require for each error mapping A_a

$$\alpha(a,\rho_i)\rho_i = \mathbb{S}(A_a\rho_i A_a^\dagger) \equiv \sum_r R_r A_a \rho_i A_a^\dagger R_r^\dagger, \tag{4.35}$$

where $\mathbb{S}$ is the "superoperator" defined on densities ρ as $\sum_r R_r\rho R_r^\dagger$, $\alpha(a,\rho_i)$ is a positive constant and ρ_i is any H_0 density. Note that in the example the recovery mappings also have the additional property that

$$\sum_r R_r^\dagger R_r = \sum_r P_r M_r^\dagger M_r P_r = I, \tag{4.36}$$

where the P_r are the appropriate orthogonal projections and I is the identity operator on H.

We have motivated the definition of $\mathbb{S}$ using a linear mapping of positive semidefinite matrices to positive semidefinite matrices. With a slightly strengthened definition of the positivity-preserving property, such operators are called *superoperators*, and it can be shown, as for example in [62], that one can always find mappings R_r so that $\mathbb{S}(\rho)$ has an "operator-sum representation" as above. That being the case, we take the general operator-sum representation with condition (4.36) as part of the definition.

(4.37) Definition. A linear operator $\mathbb{S}$ mapping positive semidefinite matrices of H to positive semidefinite matrices of H is called a *superoperator* on H if there is a set of linear mappings $R = \{R_r\colon r \text{ in } I(\mathbb{S})\}$, where $I(\mathbb{S})$ is an index set, such that $\mathbb{S}(\rho) = \sum_r R_r\rho R_r^\dagger$ and $\sum_r R_r^\dagger R_r = I$. •

We now have the requisite level of abstraction. The encoding space H_0 sits inside the coding space H. There is a family A of *error-causing operators* and a set R of *recovery mappings* which defines a superoperator $\mathbb{S}$ on H satisfying (4.35) for every H_0-density matrix. We wish to determine the relationship between A and R with H_0 being fixed, and we will find conditions on the error-causing operators which are both necessary and sufficient for the existence of a recovery superoperator.

Let us begin with the assumption that a recovery set $R = \{R_r\}$ exists and let $A(H_0,R)$ denote the set of error operators corrected by R. If A_a is in $A(H_0,R)$, then for $\rho_i = |\varphi_0\rangle\langle\varphi_0|$ with $|\varphi_0\rangle$ in H_0,

$$\alpha(A_a,\varphi_0)|\varphi_0\rangle\langle\varphi_0| = \sum_r R_r A_a|\varphi_0\rangle\langle\varphi_0|A_a^\dagger R_r^\dagger,$$

where $\alpha = \alpha(A_a, \varphi_0)$ is a positive constant depending on A_a and $|\varphi_0\rangle$. If $|\psi\rangle$ is any other state in H, then it is easy to see that

$$\alpha|\langle\psi|\varphi_0\rangle|^2 = \sum_r |\langle\psi|R_r A_a|\varphi_0\rangle|^2.$$

In particular, for every state $|\psi\rangle$ orthogonal to $|\varphi_0\rangle$, we conclude that $|\psi\rangle$ is also orthogonal to $R_r A_a|\varphi_0\rangle$ for every r. But that means that $R_r A_a|\varphi_0\rangle$ is in the span of $|\varphi_0\rangle$ and hence for some $\lambda(r, A_a, \varphi_0)$,

$$R_r A_a|\varphi_0\rangle = \lambda(r, A_a, \varphi_0)|\varphi_0\rangle. \tag{4.38}$$

Using the linearity of the mappings $R_r A_a$, it is easy to confirm that the eigenvalue $\lambda(r, A_a, \varphi_0)$ is the same for every state in H_0 and we denote the common value by $\lambda(r, a)$.

(4.39) Exercise. Verify that $\lambda(r, A_a, \varphi_0)$ doesn't depend on $|\varphi_0\rangle$. By choosing $|\psi\rangle = |\varphi_0\rangle$ show that $\sum_r |\lambda(r, a)|^2 = \alpha(A_a, \varphi_0)$ and conclude that α is also independent of the state $|\varphi_0\rangle$. (The proof is similar in spirit to that of the no-cloning theorem, Proposition (4.5).) •

Next assume that A_a and A_b are two error-causing operators corrected by R. Then for any two states in H_0

$$\langle\psi|\, A_a^\dagger A_b\, |\varphi\rangle = \sum_r \langle\psi|\, A_a^\dagger R_r^\dagger R_r A_b\, |\varphi\rangle = \langle\psi\,|\varphi\rangle \sum_r \bar{\lambda}\,(r, a)\,\lambda\,(r, b)\,,$$

leading to the following lemma.

(4.40) Lemma. Let $|\varphi\rangle$ and $|\psi\rangle$ be normalized states in H_0 and let A_a and A_b be error-causing operators corrected by R. Then

$$\langle\psi|\varphi\rangle = 0 \Rightarrow \langle\psi|A_a^\dagger A_b|\varphi\rangle = 0,$$

$c(a, b) \equiv \langle\varphi|A_a^\dagger A_b|\varphi\rangle$ is independent of the (normalized) state $|\varphi\rangle$, and $c(a, a)$ is strictly positive for each operator A_a corrected by R. •

(4.41) Exercise. In the five-qubit example let C denote the 16×16 matrix with entries $c(a, b)$ defined as in (4.40). Verify that C is the identity matrix. •

We can now state the main result of this section.

(4.42) Theorem. Let H_0 denote an encoding sub-Hilbert space of H. Then there exists a recovery superoperator $\mathbb{S}$ for a finite set of error-causing mappings A if and only if the mappings in A satisfy the conditions described in (4.40).

Proof: We have already confirmed the necessity of (4.40), even without the assumption of finiteness, so let us assume a finite set A satisfying (4.40) and let C denote the matrix with entries $c(a,b)$. It is easy to check from the definition that C is Hermitian and moreover that C is positive semidefinite:

$$w^\dagger C w = \sum_{a,b} \bar{w}(a) c(a,b) w(b) \geq 0$$

for any vector w. It then follows from the standard linear algebra results that C has real, nonnegative eigenvalues, and is unitarily diagonalizable,

$$D = U^\dagger C U,$$

in such a way that the resulting diagonal matrix D has strictly positive entries in the first r_0 positions on the main diagonal and zero entries elsewhere.

Let $\{|k\rangle, 1 \leq k \leq d\}$ be an arbitrary but fixed orthonormal basis for H_0 and let V^k denote the span of $\{A_a|k\rangle, 1 \leq a \leq a_0\}$, where we let a_0 denote the number of error mappings in A. For each r, $1 \leq r \leq r_0$, define the mapping Q_r on H_0 by

$$Q_r|k\rangle = \sum_a A_a|k\rangle u(a,r),$$

where $u(a,r)$ is the (a,r)'th entry of the diagonalizing matrix U . It is obvious that for each k, $Q_r|k\rangle$ is in V^k, and, since $U^{-1} = U^\dagger$,

$$A_b|k\rangle = \sum_r Q_r|k\rangle \bar{u}(b,r)$$

so that we also have $V^k = span(Q_r|k\rangle, 1 \leq r \leq r_0)$. Using the assumed properties of the error mappings,

$$\begin{aligned} \langle k|Q_r^\dagger Q_s|k\rangle &= \sum_{a,b} \bar{u}(a,r)\langle k|A_a^\dagger A_b|k\rangle u(b,s) \\ &= \sum_{a,b} \bar{u}(a,r) c(a,b) u(b,s) = D(r,s), \end{aligned}$$

and it follows that $\{Q_r|k\rangle, 1 \leq r \leq r_0\}$ is an orthogonal basis for V^k.

The effect of the preceding calculations is that we have defined linear combinations of the "physical" error mappings in A which act like the error mappings of the five-qubit example, the images of the encoding

space are mapped onto orthogonal subspaces of H. The verification of that fact uses the other property of A described in (4.40), and we leave the details as an exercise within the proof.

(4.43) Exercise. Confirm that $\langle j|Q_r^\dagger Q_s|k\rangle = \delta(j,k)D(r,s)$. •

The significance of (4.43) is that we can now easily construct the recovery superoperator $\mathbb{S}$. To define the set R, let W_r denote the range of Q_r and note from (4.43) that Q_r maps the given orthonormal basis of H_0 to the orthogonal basis $\{Q_r|k\rangle, 1 \leq k \leq d\}$. Hence we can define a unitary map M_r from W_r to H_0 which maps the *normalization* of the Q_r image of each $|k\rangle$ back to $|k\rangle$ and which can be extended to be a unitary map on all of the coding space H. The r'th recovery mapping is then defined as $R_r = M_r P_r$, where P_r is the projection onto W_r. The recovery set R is the set of those r_0 mappings together with the additional mapping R_c defined as the projection on the orthogonal complement of $W^1 \oplus \cdots \oplus W^{r_0}$, i.e., on the orthogonal complement of the range of the error mappings.

It is easy to verify that $I = R_c^\dagger R_c + \sum_r R_r^\dagger R_r$ and that R defines a superoperator $\mathbb{S}$ mapping densities to densities. To see that $\mathbb{S}$ corrects errors in A, we choose a basis state of H_0, use the fact that $A_b|k\rangle = \sum_r Q_r|k\rangle\bar{u}(b,r)$ for every b , and simply trace out the definitions, noting that R_c plays no role in the calculation:

$$\begin{aligned}
\mathbb{S}(A_a|k\rangle\langle k|A_a^\dagger) &= \sum_r R_r A_a|k\rangle\langle k|A_a^\dagger R_r^\dagger = \sum_r M_r P_r A_a|k\rangle\langle k|A_a^\dagger P_r M_r^\dagger \\
&= \sum_r M_r \bar{u}(a,r) Q_r|k\rangle\langle k|Q_r^\dagger u(a,r) M_r^\dagger \\
&= |k\rangle\langle k| \sum_r \bar{u}(a,r)D(r,r)u(a,r) = |k\rangle\langle k|c(a,a).
\end{aligned}$$

Since this holds for an arbitrary basis state of H_0, it holds for any density with support in H_0, completing the proof. •

The five-qubit encoding algorithm illustrates the case when the error-causing mappings are in one-to-one correspondence with the recovery mappings. Knill and Laflamme give an example when there are more error-causing mappings than recovery mappings, and we reproduce a slight modification of that example to complete this section.

(4.44) Example. Let H denote the Hilbert space modeling two qubits and let H_0 denote the space spanned by $\{|00\rangle, |11\rangle\}$. A consists of three error-mappings parametrized by q, where $0 < q < 0.5$. Using the usual

basis for H, the matrix representations for the A_a are

$$A_1 = \begin{pmatrix} \sqrt{1-2q} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \sqrt{1-2q} \end{pmatrix}$$

and

$$A_2 = \begin{pmatrix} \sqrt{q/2} & 0 & 0 & 0 \\ 0 & 1 & 0 & \sqrt{q/2} \\ \sqrt{q/2} & 0 & 1 & 0 \\ 0 & 0 & 0 & \sqrt{q/2} \end{pmatrix} \qquad A_3 = \begin{pmatrix} \sqrt{q/2} & 0 & 0 & 0 \\ 0 & 1 & 0 & -\sqrt{q/2} \\ -\sqrt{q/2} & 0 & 1 & 0 \\ 0 & 0 & 0 & \sqrt{q/2} \end{pmatrix}.$$

The images of $|00\rangle$ are

$$\left\{\sqrt{1-2q}|00\rangle, \sqrt{q/2}(|00\rangle + |10\rangle), \sqrt{q/2}(|00\rangle - |10\rangle)\right\}$$

and those of $|11\rangle$ are

$$\left\{\sqrt{1-2q}|11\rangle, \sqrt{q/2}(|01\rangle + |11\rangle), \sqrt{q/2}(-|10\rangle + |11\rangle)\right\}$$

under the mappings of A, and these sets are obviously linearly dependent. It is easy to check that A satisfies the conditions of (4.40), and letting t denote $\sqrt{q(1-2q)/2}$, we find that

$$C = \begin{pmatrix} 1-2q & t & t \\ t & q & 0 \\ t & 0 & q \end{pmatrix}$$

and

$$U = \begin{pmatrix} 0 & \sqrt{(1-2q)/(1-q)} & -\sqrt{q/(1-q)} \\ 1/\sqrt{2} & \sqrt{q/2(1-q)} & \sqrt{(1-2q)/2(1-q)} \\ -1/\sqrt{2} & \sqrt{q/2(1-q)} & \sqrt{(1-2q)/2(1-q)} \end{pmatrix},$$

where the columns of U correspond, respectively, to the eigenvalues q, $1-q$ and 0 of C.

Following the recipe in (4.42) we obtain the mappings Q_1 and Q_2 on H_0:

$$\begin{aligned} Q_1|00\rangle &= \sqrt{q}|10\rangle \qquad Q_2|00\rangle = \sqrt{1-q}|00\rangle \\ Q_1|11\rangle &= \sqrt{q}|01\rangle \qquad Q_2|11\rangle = \sqrt{1-q}|11\rangle \end{aligned}$$

and thus $R_1 = |00\rangle\langle 10| + |11\rangle\langle 01|$ and $R_2 = |00\rangle\langle 00| + |11\rangle\langle 11|$. •

(4.45) Exercise. Confirm the statements in (4.44) and verify that $\{R_1, R_2\}$ corrects the error mappings of A. •

4.10 Further aspects of quantum error-correcting codes

The study of quantum error-correcting codes has developed rapidly, and this introduction is not intended as a comprehensive presentation of the field. In particular, there are a number of significant topics that have not yet been discussed. We mention some of them below, indicating the ideas and citing the literature for further study.

Generation of Quantum Error-Correcting Codes. We have described in detail how an appropriate set of generators of a code defines the wiring diagrams for encoding and error detection. The trick then is to select the set of generators for an appropriate encoding. In Section 4.8 we showed how CSS codes can be defined using classical coding theory. Another approach involves modifying existing codes, and Gottesman's thesis [39] contains a wealth of detail on such techniques as well as analyses of other aspects of quantum error-correcting codes. For an example of a code that is not a stabilizer code, see [59].

(4.46) Exercise. Define a *perfect*, n-qubit, one-error-correcting code as one in which d qubits are encoded in n qubits, and $n-d$ measurements distinguish exactly among the $3n+1$ possible single errors. The five-qubit code of Section 4.6 is an example with $d=1$. Could there be an example with $d=2$? How about $d=15$? (See [39].) •

Computations on Encoded Qubits. The operators in *N(S)-S* map the encoded subspace to itself and could be used to perform logical operations without decoding. For example, in the five-qubit case, $\bar{X}$ interchanges $|0_e\rangle$ and $|1_e\rangle$, while $\bar{Z}$ performs the analogue of σ_z on the encoded qubits. Zurek and Laflamme [80] discuss such "qubyte" operations in the context of Steane's encodings; [39] contains other references and also includes a related discussion in the context of fault-tolerant computation.

Fault-Tolerant Computations. The five-qubit code of [49] has the virtue of combining the encoding and the error detection in one wiring but the liability of facilitating the propagation of errors. Errors can propagate both "forwards" and "backwards," and the overuse of the same qubit in computations should be avoided. Thus, for example, Steane's use of ancillary qubits for measurements inhibits some but not all of that error propagation.

(4.47) Example. Suppose an *XOR* operation is to be applied to $|i\rangle|k\rangle$ with the first qubit acting as the control qubit. If an error has occurred in the first qubit, so that the actual state is $|i\oplus 1\rangle|k\rangle$, then after the

XOR operation, the second qubit will be in the state $|i \oplus k \oplus 1\rangle$ and the error has propagated "forwards." If the system were in the state $(\alpha|0\rangle + \beta|1\rangle)(|0\rangle \pm |1\rangle)$, where the plus/minus sign indicates a potential error σ_z, then after the *XOR* operation the system can be written as $(\alpha|0\rangle \pm \beta|1\rangle)(|0\rangle \pm |1\rangle)$, which can be interpreted as saying the second qubit is unchanged and the control qubit now has a σ_z error. (See the discussion after Exercise (2.11).) •

The use of entangled ancillary qubits can further reduce error propagation, as noted in [66], and aspects of fault-tolerant computation are described, for example, in [20] and [39]. The construction of fault-tolerant algorithms appears to be crucial for the development of feasible algorithms for quantum computation, and a nice discussion of the subject is given in [58].

Abstract Theory of Quantum Error-Correcting Codes. One can abstract the context and machinery of quantum error-correcting codes, examining the problem from a more general perspective, and we mention just a few of the relevant papers. An early paper by Schumacher [61] took an information-theoretic approach and introduced the idea of fidelity in the quantum domain. (Schumacher also introduced the term "qubit" to the literature.) Knill and Laflamme [47] emphasize the Hilbert space context and the role of "superoperators," and in the preceding section we gave their necessary and sufficient conditions for a code to correct a given set of errors. (See also [34].) The relationships among quantum error-correcting codes, quantum cryptography and EPR entangled states are emphasized in [12], and a group-theoretic framework is used in [18] and [19]. For additional references, see [39] and the ever-changing LANL (Los Alamos) web site quant-ph.

Afterword

Since the penultimate draft of this manuscript was prepared, additional references have become available which may be of interest to the reader interested in algorithms or other topics not covered in this book. For example, John Preskill has prepared notes and problems for a course at the California Institute of Technology and has made them available at his web site: preskill@theory.caltech.edu. Similarly, notes from a course taught by Umesh Vazirani are also available at vazirani@cs.berkeley.edu.

The proceedings from a 1996 conference at Santa Barbara contain some of the references cited here, including [22], as well as many other relevant articles: *Proc. Roy. Soc. London A*, vol. 454 (Jan. 1998). There have been some special issues of journals dedicated to aspects of quantum computing, such as *SIAM J. Comput.*, vol. 26, no. 5 (Oct. 1997) which contains [9].

Other recent publications include a book on the subject (Colin Williams and Scott Clearwater, *Explorations in Quantum Computing*, Springer-Telos (1997)) and an edited volume to appear: *Introduction to Quantum Computation and Information*, edited by H.-K. Lo, S. Popescu and T. P. Spiller, World Scientific, Singapore (1998), which contains [58].

Many of the references cited in the text appeared first on the "quant-ph" site of the Los Alamos National Laboratory and subsequently in proceedings or journals. If possible, both references are given, and papers with only a quant-ph reference may well have been published elsewhere since this bibliography was prepared.

References

[1] Sheldon Axler, *Linear Algebra Done Right*, 2nd ed., Springer, New York (1997).

[2] Adriano Barenco, "Quantum physics and computation," Clarendon Laboratory, Oxford, UK, preprint (1995–96).

[3] Adriano Barenco, Charles Bennett, Richard Cleve, David DiVincenzo, Norman Margolus, Peter Shor, Tycho Sleator, John Smolin, Harold Weinfurter, "Elementary gates for quantum computation," Phys. Rev. A, vol. 52, no. 5, 3457–3467 (Nov. 1995).

[4] David Beckman, Amalavoyal N. Chari, Srikrishna Devabhaktuni, John Preskill, "Efficient networks for quantum factoring", Phys. Rev. A, vol. 54, no. 2, 1034–1063 (Aug. 1996).

[5] John Bell, *Speakable and Unspeakable in Quantum Mechanics*, Cambridge Univ. Press, London and New York (1993).

[6] Paul Benioff, "The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines," J. Stat. Physist, vol. 22, no. 5, 563–590 (1980).

[7] Paul Benioff, "Quantum-mechanical Hamiltonian models of Turing machines that dissipate no energy," Phys. Rev. Lett., vol. 48, 1581–1585 (1982).

[8] Charles H. Bennett, "The thermodynamics of computation — a review," Int. J. of Theo. Phys., vol. 21, no. 12, 905–940 (1982).

[9] Charles H. Bennett, Ethan Bernstein, Gilles Brassard, Umesh Vazirani, "Strengths and weaknesses of quantum computing," SIAM J. Comput., vol. 26, no. 5, 1510–1523 (Oct. 1997).

[10] Charles H. Bennett, Gilles Brassard, "Quantum cryptography: Public key distribution and coin tossing," in Proc. IEEE International Conference on Computers, Systems, and Signal Processing, Bangalore, India, 175–179 (1984).

[11] Charles H. Bennett, Gilles Brassard, Claude Crepeau, Richard Jozsa, Asher Peres, William Wootters, "Teleporting an unknown quantum state via dual classical and Einstein–Podolsky–Rosen channels," Phys. Rev. Lett., vol. 70, 1895–1899 (1993).

[12] Charles H. Bennett, David P. DiVincenzo, John A. Smolin, William K. Wootters, "Mixed state entanglement and quantum error correction," Phys. Rev. A, vol. 54, 3824–3851 (1996); quant-ph/9604024 v2 (Aug. 1996).

[13] E. Bernstein, U. Vazirani, "Quantum complexity theory," in Proc. 25th ACM Symposium on Theory of Computing, ACM, New York (1993).

[14] David Bohm, *Quantum Theory*, Prentice-Hall, Englewood Cliffs, NJ (1951).

[15] D. Boneh, R. J. Lipton, "Quantum cryptanalysis of hidden linear functions," extended abstract, Lecture Notes in Computer Science, 963, 424–437, Springer-Verlag (1995).

[16] Michel Boyer, Gilles Brassard, Peter Hoyer, Alain Tapp, "Tight bounds on quantum searching," Forstschritte Der Physik, Special issue on quantum computing and quantum cryptography, vol. 4, 820–831 (1998); quant-ph/9605034 (May 1996).

[17] Gilles Brassard, "Teleportation as a quantum computation", Physica D120, 43–47 (1998); quant-ph/9605035 (May 1996).

[18] A. R. Calderbank, E. M. Rains, P. W. Shor, N. J. Sloane, "Quantum error correction and orthogonal geometry," Phys. Rev. Lett., vol. 78, 405–408 (1997); quant/ph/96/05/005 (May 1996).

[19] A. R. Calderbank, E. M. Rains, P. W. Shor, N. J. Sloane, "Quantum error correction via codes over GF(4)," IEEE Trans. Inform. Theory, to appear; quant/ph/96/08/006 v5 (10 Sept. 1997).

[20] A. R. Calderbank, P. W. Shor, "Good quantum error-correcting codes exist," Phys. Rev. A., vol. 54, 1098–1105 (1996).

[21] J. I. Cirac, P. Zoller, "Quantum computation with cold trapped ions," Phys. Rev. Lett., vol. 74, no. 20, 4091–4094 (May 1995).

[22] R. Cleve, A. Ekert, C. Macchiavello, M. Mosca, "Quantum algorithms revisited," Proc. Roy. Soc. London A, vol. 454, 339–354 (1998); quant-ph/9708016 (Aug. 1997).

[23] Richard Cleve, Daniel Gottesman, "Efficient computations of encodings for error correction," quant-ph/9607030 (July 1996).

[24] D. Coppersmith, "An approximate Fourier transform useful in quantum factoring," IBM Research Report no. RC 19642 (1994).

[25] David Deutsch, "Quantum theory, the Church–Turing principle and the universal quantum computer," Proc. Roy. Soc. London A, vol. 400, 97–117 (1985).

[26] David Deutsch, "Quantum computational networks," Proc. Roy. Soc. London A, vol. 425, 73–90 (1989).

[27] David Deutsch, Richard Jozsa, "Rapid solution of problems by quantum computation," Proc. Roy. Soc. London A, vol. 439, 553–558 (1992).

[28] David DiVincenzo, "Two-bit gates are universal for quantum computation," Phys. Rev. A, vol. 51, 1015–1022 (1995).

[29] David DiVincenzo, "Quantum gates and circuits", in Proc. ITP Conference on Quantum Coherence and Decoherence, December 1996, Proc. Roy. Soc. London A, (Jan. 1998); quant-ph9705009 (May 1997).

[30] David DiVincenzo, Peter Shor, "Fault-tolerant error correction with efficient quantum codes," Phys. Rev. Lett., vol. 77, no. 15, 3260–3263 (Oct. 1996).

[31] H. Dym, H. P. McKean, *Fourier Series and Integrals*, Academic Press, New York (1972).

[32] A. Einstein, B. Podolsky, N. Rosen, "Can quantum-mechanical description of physical reality be considered complete?" Phys. Rev., vol. 47, 777 (1935).

[33] Artur Ekert, Richard Jozsa, "Quantum computation and Shor's algorithm," Rev. Modern Phys., vol. 68, no. 3, 733–753 (July 1996).

[34] Artur Ekert, Chiara Macchiavello, "Quantum error correction for communication," Phys. Rev. Lett., vol. 77 (12), 2585–2588 (Sept. 1996).

[35] Richard Feynman, *Lectures in Physics, Vol III*, Addison-Wesley, Reading, MA (1965).

[36] Richard Feynman, "Quantum mechanical computers," Found. Phys., vol. 16, no. 6, 507–531 (1986).

[37] Neil Gershenfeld, Isaac Chuang, "Bulk spin-resonance quantum computation," Science, vol. 275 (17 Jan. 1997).

[38] Daniel Gottesman, "A class of quantum error-correcting codes saturating the quantum Hamming bound," Phys. Rev. A, vol. 54, 1862 (1996); quant-ph/96/04/038 (Apr. 1996).

[39] Daniel Gottesman, "Stabilizer codes and quantum error correction," thesis, Calif. Inst. Tech., Pasadena, CA, quant-ph/97/05/052 (1997).

[40] R. B. Griffiths and C. S. Niu, "Semiclassical Fourier transform for quantum computation," Phys. Rev. Lett., vol.76, 3228–3230 (1996).

[41] Lov Grover, "A fast quantum-mechanical algorithm for database search," in Proc. 28th Annual ACM Symposium on Theory of Computing, ACM, New York (1996).

[42] G. H. Hardy, E. M. Wright, *An Introduction to the Theory of Numbers*, 5th ed., Oxford Univ. Press, New York (1979).

[43] Max Jammer, *The Philosophy of Quantum Mechanics*, Wiley, New York (1974).

[44] Richard Jozsa, "How can we find new quantum algorithms?" Institute for Theoretical Physics, Univ. Calif. Santa Barbara, Santa Barbara CA, preprint (Aug. 1996).

[45] Richard Jozsa, "Quantum algorithms and the Fourier transform," Proc. Roy. Soc. London A, vol. 454, 323–337 (1998).

[46] A. Yu. Kitaev, "Quantum measurements and the Abelian stabilizer problem," Landau Institute for Theoretical Physics, Moscow, Russia, preprint (June 1996); quant-ph/9511026 (1995).

[47] Emanuel Knill, Raymond Laflamme, "Theory of quantum error-correcting codes," Phys. Rev. A, vol. 55 (2), 900–911 (Feb. 1997).

[48] D. E. Knuth, *The Art of Computer Programming, Vol. 2: Seminumerical Algorithms*, 2nd ed., Addison-Wesley, Reading, MA (1980).

[49] R. Laflamme, C. Miquel, J-P. Paz, W. H. Zurek, "A perfect quantum error-correcting code," Phys. Rev. Lett., vol. 7, 198–201 (July 1996).

[50] A. K. Lenstra and H. W. Lenstra Jr., eds., *The Development of the Number Field Sieve*, Lecture Notes in Math., vol. 1554, Springer, Berlin and New York (1993).

[51] Seth Lloyd, "Quantum-mechanical computers," Scientific American, 44–49 (Oct. 1995).

[52] F. J. MacWilliams, N. J. A. Sloane, *The Theory of Error-Correcting Codes*, North-Holland, Amsterdam (1977).

[53] David Mermin, "What's wrong with these elements of reality?" Physics Today, 9–11 (June 1990).

[54] Michele Mosca, Artur Ekert, "The hidden subgroup problem and eigenvalue estimation on a quantum computer," quant-ph/9903071 (Mar. 1999).

[55] K. R. Parthasarathy, *An Introduction to Quantum Stochastic Calculus*, Monographs Math., vol. 85, Birkhäuser, Basel (1992).

[56] Asher Peres, *Quantum Theory: Concepts and Methods*, Kluwer, Dordrecht (1995).

[57] T. B. Pittman, Y. H. Shih, D. V. Strekalov, A. V. Sergienko, "Optical images by means of 2-photon quantum entanglement," Phys. Rev. A, vol. 52, R3429–32 (1995).

[58] John Preskill, "Fault-tolerant error computation," in *Introduction to Quantum Computation and Information*, edited by H.-K. Lo, S. Popescu, and T. P. Spiller, World Scientific, Singapore (1998); quant-ph/9712048 (Dec. 1997).

[59] Eric M. Rains, R. H. Hardin, Peter W. Shor, N. J. A. Sloane, "A non-additive quantum code," Phys. Rev. Lett., vol. 79, 953–954 (1997); quant-ph/9703002 V2 (May 1997).

[60] J. J. Sakurai, *Modern Quantum Mechanics*, rev. ed., Addison-Wesley, New York (1994).

[61] Benjamin Schumacher, "Quantum Coding," Phys. Rev. A, vol. 51, 2738–2747 (Apr. 1995).

[62] Benjamin Schumacher, "Sending entanglement through noisy channels," quant-ph/9604023 (Apr. 1996).

[63] Ravi Shankar, *Principles of Quantum Mechanics*, 2nd ed., Plenum Press, New York (1994).

[64] Peter Shor, "Scheme for reducing decoherence in quantum computer memory," Phys. Rev. A, vol. 52, no. 4, R2493–R2496 (Oct. 1995).

[65] Peter Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," SIAM J. Comput., vol. 26, no. 5, 1484–1509 (Oct. 1997); quant-ph950802/V2 (Jan. 1996).

[66] Peter Shor, "Fault-tolerant quantum computation," in Proc. 37th Symposium on Foundation of Computing, IEEE Computer Society Press, 56–65 (1996).

[67] Daniel Simon, "On the power of quantum computation," in Proc. 35th Annual Symposium on Foundations of Computer Science, IEEE Computer Society Press, Los Alamitos, CA, 116–123 (1994).

[68] A. M. Steane, "Error correcting codes in quantum theory," Phys. Rev. Lett., vol. 77, no. 5, 793–797 (July 1996).

[69] A. M. Steane, "Multiple-particle interference and quantum error correction," Proc. Roy. Soc. London A, vol. 452, 2551–2577 (1996).

[70] A. M. Steane, "Quantum computing," qaunt-ph/9708022 (Aug. 1997).

[71] S. Sternberg, *Group Theory and Physics*, Cambridge Univ. Press, London and New York (1994).

[72] Anthony Sudbery, *Quantum Mechanics and the Particles of Nature: An Outline for Mathematicians*, Cambridge Univ. Press, London and New York (1986).

[73] Gary Taubes, "Putting a quantum computer to work in a cup of coffee," Science, vol. 275, 307–309 (17 Jan. 1997).

[74] T. Toffoli, "Reversible computing," in Automata, Languages, and Programming, Seventh Colloquium, Lecture Notes in Computer Science vol. 84, eds. J. W. de Bakker and J. van Leeuwen, Springer, New York, 632–644 (1980).

[75] W. G. Unruh, "Maintaining coherence in quantum computers," Phys. Rev. A, vol. 51, no. 2, 992–997 (Feb. 1995).

[76] Vlatko Vedral, Adriano Barenco, Artur Ekert, "Quantum networks for elementary arithmetic operations," Phys. Rev. A, vol. 54, 147 (1996).

[77] David Wick, *The Infamous Boundary — Seven Decades of Controversy in Quantum Physics*, Birkhäuser, Boston, MA (1995).

[78] W. K. Wootters, W. H. Zurek, "A single quantum cannot be cloned," Nature, vol. 299, 802 (1982).

[79] Wojciech Zurek, "Decoherence and the transition from quantum to classical," Physics Today, 36–45 (Oct. 1991); replies in Physics Today (Apr. 1993).

[80] Wojciech Zurek, Raymond Laflamme, "Quantum logical operations on encoded qubits," quant-ph/9605013 (May 1996).

[81] Artur Ekert, "Quantum cryptography based on Bell's theorem," Phys. Rev. Lett., vol. 67, no. 6, 661–663 (Aug. 1991).

Index

Progress in Computer Science and Applied Logic

PCS 1 Mathematics for the Analysis of Algorithms, 3rd Edition
Daniel H. Greene & Donald E. Knuth

PCS 2 Applied Probability–Computer Science: The Interface, Volume I
Edited by Ralph L. Disney & Teunis J. Ott

PCS 3 Applied Probability–Computer Science: The Interface, Volume II
Edited by Ralph L. Disney & Teunis J. Ott

PCS 4 Notes on Introductory Combinatorics
George Pólya, Robert E. Tarjan, & Donald R. Woods

PCS 5 The Evolution of Programs
Nachum Dershowitz

PCS 6 Lecture Notes on Bucket Algorithms
Luc Devroye

PCS 7 Real-Time Control of Walking
Marc D. Donner

PCS 8 Logic for Computer Scientists
Uwe Schöning

PCS 9 Feasible Mathematics
Edited by Samuel R. Buss & Philip J. Scott

PCS 10 Graph-Based Proof Procedures for Horn Clauses
Stan Raatz

PCS 11 A Proof Theory of General Unification
Wayne Snyder

PCS 12 Logical Methods: In Honor of Anil Nerode's Sixtieth Birthday
Edited by John E. Crossley, Jeffrey B. Remmel, Richard A. Shore, & Moss E. Sweedler

PCS 13 Feasible Mathematics II
Edited by P. Clote & J.B. Remmel

PCS 14 Learning and Geometry: Computational Approaches
Edited by David W. Kueker & Carl H. Smith

PCS 15 Symbolic Rewriting Techniques
Manuel Bronstein, Johannes Grabmeier, & Volker Weispfenning

PCS 16 Bounded Queries in Recursion Theory
William I. Gasarch & Georgia A. Martin

PCS 17 Number Theoretic Methods in Cryptography: Complexity lower bounds
I. Shparlinski

PCS 18 Interpolating Cubic Splines
Gary D. Knott

PCS 19 An Introduction to Quantum Computing Algorithms
Arthur O. Pittenger