An introduction to quantum information and quantum circuits

John Watrous
Institute for Quantum Computing
University of Waterloo

June 22, 2011

1 Introduction

This is an introductory paper on quantum information and quantum circuits. It is intended primarily for theoretical computer scientists (and especially complexity theorists) who know little or nothing about quantum computing and would like to have a better grasp of the basic aspects of the subject. A familiarity with basic computational complexity, including topics such as Boolean circuits, Turing machines and randomized computation, is assumed.

1.1 Motivation

This paper was (at least partly) inspired by a recent posting on Dick Lipton's blog *Gödel's Lost Letter and P = NP*, titled "Is Factoring Really in BQP? Really?" In this posting, as well as in its comments and in the follow-up posting titled "Factoring is in BQP," Lipton expressed the desire to see a careful proof that the integer factoring problem, appropriately phrased as a decision problem, is contained in the complexity class BQP (short for *bounded-error quantum polynomial time*). To be clear, Lipton was not doubting or challenging the claim that integer factoring is in BQP—he just wanted to see a detailed and carefully written proof.

As the reader likely knows, quantum computers (as modeled by quantum circuits or quantum Turing machines) can factor integers in polynomial time. This fact, first proved in 1994 by Peter Shor [Sho94, Sho97], is largely responsible for the broad interest in quantum computing that exists today. Shor's quantum algorithm for factoring integers has been studied extensively, and there is no shortage of careful and detailed presentations of it and its analysis. (Examples include the presentations in several well-known books [AB09, NC00, KSV02, KLM07], and Shor's paper [Sho97] itself contains a very clear, precise and readable presentation of the result.) There is a difference, however, between a quantum algorithm that outputs a factorization of an integer with a non-negligible probability of success (which is what a typical analysis of Shor's algorithm demonstrates) and a quantum computation that possesses the properties required to conclude that a given decision problem is in BQP. This is the issue with which Lipton was interested.

As it turns out, this issue has little to do with factoring. It is a general issue that could arise for any number of computational problems and algorithms in which samples from a quantum process of some kind are compiled, leading to a single binary-valued output. Perhaps more generally than this, the issue concerns basic manipulations of quantum circuits. Non-experts sometimes struggle

with these sorts of manipulations, while expert researchers in quantum computing view them as routine.

This paper will aim to close this gap in understanding. While it is intended to be more of an introduction than a presentation of formal proofs, it will hopefully fulfill Lipton's request for the missing details between Shor's algorithm and the fact that integer factoring is in BQP.

1.2 A choice between two descriptions

Quantum computation is based on the theory of *quantum information*, so an understanding of quantum information is, naturally, required for an understanding of quantum computation. The first half of this paper is devoted to a presentation of the basic definitions of quantum information for this reason.

The model of quantum information to be presented may be new to some readers, including readers that already know something about quantum computing. It is the *general* model of quantum information based on density matrices, general measurements and general quantum operations, as opposed to the more restricted *pure state* model that is most typically presented in introductory papers on quantum computing. (The two models and their relationship will be discussed in greater detail in the section following this one.)

A brief explanation is in order for why the general model has been chosen for this paper. The main reason for preferring the general model of quantum information over the pure state model in this particular paper is that it offers a simple and intuitive picture for how quantum circuits can be manipulated and composed with classical computations. This provides an immediate answer to the question mentioned above regarding the relationship between integer factorization as a decision problem and Shor's algorithm as a sampling procedure: in essence, the relationship is exactly the same as it would be for classical algorithms.

A second reason for preferring the general model of quantum information is that it is much more interesting than the pure state model from a mathematical point of view. It is a powerful tool for reasoning about quantum computation and quantum cryptography, and it has interesting connections to matrix theory, convex analysis, and group theory, to name just a few related branches of mathematics. It is essential for anyone who would choose to participate in current research on quantum computing to understand the general model of quantum information.

The general model of quantum information was first applied to models of quantum computation by Aharonov, Kitaev, and Nisan [AKN98], who presented several arguments in its favor.

2 Quantum information

As mentioned above, we will begin with an introduction to the fundamental notions of quantum information theory. Readers interested in learning more about this theory may find that the standard text, Nielsen and Chuang [NC00], is a good starting point.

2.1 Registers

Consider a hypothetical device to be named X. We view that X is a physical device used to store information—such as a component inside a computer or a wire connecting two devices—and with this view in mind we will refer to X as a *register*. At this point we are not necessarily considering that X is classical, quantum, or something else; this determination can be momentarily delayed. Associated with the register X is, by assumption, some finite and non-empty set of *classical states*

Σ. It is sufficient that the term *classical state* be understood at an intuitive level: if a human being opens up X and looks inside, he or she will recognize a single element of Σ as its current state. An ordered collection of registers $(\mathsf{X}_1, \ldots, \mathsf{X}_n)$ may itself be viewed as a single register X, and naturally the classical state set of this compound register X is given by the Cartesian product $\Sigma_1 \times \cdots \times \Sigma_n$, where $\Sigma_1, \ldots, \Sigma_n$ denote the classical state sets of $\mathsf{X}_1, \ldots, \mathsf{X}_n$, respectively.

In quantum computation we are usually interested in registers of the form $\mathsf{X} = (\mathsf{X}_1, \ldots, \mathsf{X}_n)$ for which each X_j has the classical state set $\{0, 1\}$, giving X the classical state set $\{0, 1\}^n$. Registers whose classical state set is $\{0, 1\}$ are called *qubits* in the quantum setting.

2.2 States

In a classical (probabilistic) setting, one's knowledge of the state of a register X whose classical state set is Σ is represented by a probability distribution over the elements of Σ. Such a distribution may be described by a *probability vector* of the form $v \in \mathbb{R}^{\Sigma}$, where $v(a)$ represents the probability of X taking the classical state $a \in \Sigma$. Naturally, one requires that $v(a) \geq 0$ for each $a \in \Sigma$, and that $\sum_{a \in \Sigma} v(a) = 1$, for such a vector to correspond to a probability distribution.

In a quantum setting, one's knowledge of X is described not by a probability vector v, but rather by a *density matrix* of the form $\rho \in \mathbb{C}^{\Sigma \times \Sigma}$. A density matrix ρ is a matrix that possesses these properties:

1. The matrix ρ is *positive semidefinite*: it must hold that $\rho = \rho^*$, where ρ^* is the *adjoint* (or *conjugate transpose*) of ρ, defined as
$$\rho^*(a, b) = \overline{\rho(b, a)}$$
for each $a, b \in \Sigma$; and moreover every eigenvalue of ρ must be non-negative.
2. The trace of ρ must equal 1:
$$\mathrm{Tr}(\rho) = \sum_{a \in \Sigma} \rho(a, a) = 1.$$

The entries of a density matrix ρ are not as intuitive or easily connected to everyday experiences as the probabilities appearing as entries of probability vectors. One way to assign an intuitive meaning to the entries of a density matrix is to consider the diagonal entries and off-diagonal entries separately:

- **Diagonal entries.** The diagonal entries of a density matrix always form a probability vector, which describes the distribution of outcomes that would result if a so-called *standard-basis measurement* were to be performed on X. (There are, in fact, a continuum of inequivalent ways in which one can measure a register, with the standard-basis measurement being just one of these ways.)
- **Off-diagonal entries.** The off-diagonal entries of a density matrix are less connected with intuition than the diagonal entries, but they can have a major impact on the results of calculations. For distinct classical states $a, b \in \Sigma$ one can reasonably view the numbers $\rho(a, b)$ and $\rho(b, a)$, which are necessarily complex conjugates of one another, as representing the extent to which the state of X is "in superposition" between a and b, along with information about the "relative phase" of these states in the superposition. Alternately, these numbers describe the characteristics with which computation paths leading from a and b may "interfere" with one another in the future.

Density matrices whose off-diagonal entries are all 0 may reasonably be viewed as classical probability distributions, and classical information theory emerges from quantum information theory through a restriction of one's attention to density matrices of this sort.

The set of all density matrices of the form $\rho \in \mathbb{C}^{\Sigma \times \Sigma}$ is a convex set, and convex combinations of density matrices represent uncertainties or random selections, just as convex combinations of probability vectors do in the classical case. For instance, a register initialized to the state ρ with probability $\lambda \in [0,1]$ and σ with probability $1-\lambda$ has a state described by the density matrix $\lambda\rho + (1-\lambda)\sigma$.

The extreme points of the set of density matrices of the form $\rho \in \mathbb{C}^{\Sigma \times \Sigma}$ are the matrices that can be expressed as $\rho = uu^*$ for some choice of a unit vector $u \in \mathbb{C}^{\Sigma}$. Such states are called *pure states*, while arbitrary states are called *mixed states*. Intuitively speaking, pure states contain no randomness: in the quantum extension of Shannon theory, these are the states with zero entropy. In the *pure state* model that was mentioned in the introduction, these are the only states considered; and rather than representing a pure state uu^* as a matrix, it is represented by the vector u (which is only determined up to the multiplication by a complex number on the unit circle, sometimes called a *global phase*, which has no effect on calculations). Even when using the general model of quantum information, we sometimes identify a given pure state with a suitable choice of a unit vector for the sake of efficiency.

2.3 Independence and correlations among multiple registers

As stated before, an ordered collection of registers may itself be considered a register. In the interest of clarity and simplicity, we will focus the discussion to follow mostly on pairs of registers (X, Y), with the classical state sets of X and Y assumed to be Σ and Γ, respectively. The picture extends to more than two registers in a straightforward way.

The classical state set of the pair (X, Y) is $\Sigma \times \Gamma$, so that density matrices representing quantum states of the pair (X, Y) take the form $\rho \in \mathbb{C}^{(\Sigma\times\Gamma)\times(\Sigma\times\Gamma)}$. Matrices of this form may be viewed as block matrices:

$$\rho = \begin{pmatrix} \rho_{a_1,a_1} & \cdots & \rho_{a_1,a_n} \\ \vdots & \ddots & \vdots \\ \rho_{a_n,a_1} & \cdots & \rho_{a_n,a_n} \end{pmatrix} \tag{1}$$

where $\Sigma = \{a_1, \ldots, a_n\}$ and where each matrix $\rho_{a,b} \in \mathbb{C}^{\Gamma\times\Gamma}$ is given by

$$\rho_{a,b}(c,d) = \rho((a,c),(b,d))$$

for every choice of $a, b \in \Sigma$ and $c, d \in \Gamma$. An alternate view is that $\mathbb{C}^{(\Sigma\times\Gamma)\times(\Sigma\times\Gamma)}$ is a concrete instantiation of the tensor product space $\mathbb{C}^{\Sigma\times\Sigma} \otimes \mathbb{C}^{\Gamma\times\Gamma}$, which consists of all elementary tensor (or Kronecker) products

$$A \otimes B = \begin{pmatrix} A(a_1,a_1)B & \cdots & A(a_1,a_n)B \\ \vdots & \ddots & \vdots \\ A(a_n,a_1)B & \cdots & A(a_n,a_n)B \end{pmatrix}$$

of matrices $A \in \mathbb{C}^{\Sigma\times\Sigma}$ and $B \in \mathbb{C}^{\Gamma\times\Gamma}$, along with all complex linear combinations of such matrices.

If the registers X and Y are *independently* prepared in the states $\rho \in \mathbb{C}^{\Sigma\times\Sigma}$ and $\sigma \in \mathbb{C}^{\Gamma\times\Gamma}$, then the state of the pair (X, Y) is given by the tensor product $\rho \otimes \sigma$. This, of course, is analogous to the classical probabilistic setting, where the independent preparations of X and Y according to the

probability vectors $v \in \mathbb{R}^{\Sigma}$ and $w \in \mathbb{R}^{\Gamma}$ results in joint probability vector $v \otimes w$ (which satisfies the familiar condition for independence: $(v \otimes w)(a, b) = v(a)w(b)$ for each $a \in \Sigma$ and $b \in \Gamma$).

If the state of the pair (X, Y) cannot be expressed as $\rho \otimes \sigma$ for density matrices ρ and σ, the registers are not independent: they are *correlated*. There are two interesting classes of correlated states in quantum information theory: *separable* states and *entangled* states. A separable state is one that can be written as a convex combination of independent states; these are states whose correlations are essentially classical. States that are not separable are, by definition, called *entangled* states. An example of an entangled state of two qubit registers is the pure state given by the vector[1]

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

The density matrix representation of this state is

$$\frac{1}{2} \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}. \tag{2}$$

Entanglement can be used as a resource in various settings, such as in cryptography and communication complexity, and its characteristics have great importance in quantum information theory.

For every possible quantum state of a pair of registers (X, Y), whether correlated or not, there are uniquely determined *reduced states* of the registers X and Y individually that, in essence, describe the states of these registers in isolation, as if the other were discarded. This is analogous to the *marginal* probability distributions of X and Y in the probabilistic case. In mathematical terms, reduced states are defined by an operation known as the *partial trace*. If it is the case that the state of the pair (X, Y) is described by a density matrix $\rho \in \mathbb{C}^{(\Sigma \times \Gamma) \times (\Sigma \times \Gamma)}$, written in block form as in (1), then the reduced states of the registers X and Y are defined as

$$\mathrm{Tr}_{\mathsf{Y}}(\rho) = \begin{pmatrix} \mathrm{Tr}(\rho_{a_1,a_1}) & \cdots & \mathrm{Tr}(\rho_{a_1,a_n}) \\ \vdots & \ddots & \vdots \\ \mathrm{Tr}(\rho_{a_n,a_1}) & \cdots & \mathrm{Tr}(\rho_{a_n,a_n}) \end{pmatrix} \in \mathbb{C}^{\Sigma \times \Sigma} \qquad \text{and} \qquad \mathrm{Tr}_{\mathsf{X}}(\rho) = \sum_{a \in \Sigma} \rho_{a,a} \in \mathbb{C}^{\Gamma \times \Gamma},$$

respectively. The notation $\mathrm{Tr}_{\mathsf{Y}}(\cdot)$ signifies that Y has been discarded, or *traced-out*, and $\mathrm{Tr}_{\mathsf{X}}(\cdot)$ has a similar meaning for X. For example, the reduced state of both the register X and the register Y for the joint state (2) is

$$\begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix}.$$

In greater generality, one defines a partial trace over X_j on a matrix space of the form

$$\mathbb{C}^{(\Sigma_1 \times \cdots \times \Sigma_n) \times (\Sigma_1 \times \cdots \times \Sigma_n)}$$

as the extension by linearity of the mapping given by

$$\mathrm{Tr}_{\mathsf{X}_j}(A_1 \otimes \cdots \otimes A_n) = \mathrm{Tr}(A_j)\, A_1 \otimes \cdots \otimes A_{j-1} \otimes A_{j+1} \otimes \cdots \otimes A_n$$

for all matrices $A_1 \in \mathbb{C}^{\Sigma_1 \times \Sigma_1}, \ldots, A_n \in \mathbb{C}^{\Sigma_n \times \Sigma_n}$.

[1] The vector is expressed using the *Dirac notation*, which is typical in quantum computing and quantum information theory: the symbols $|0\rangle$ and $|1\rangle$ represent elementary unit vectors that would typically be written as e_0 and e_1 in more standard mathematical writing, and the juxtaposition of these symbols represents a tensor product. This notation is a convenient alternative to both overloading the symbol e and putting sometimes complicated expressions in subscripts.

2.4 Operations

Quantum operations (also known as *quantum channels*) describe discrete-time changes in states of collections of registers that are, in an idealized sense, to be considered physically implementable. Upper-case Greek letters, such as Φ and Λ, will denote quantum operations in this paper.

We will consider that a given quantum operation Φ takes some register X as input and produces another register Y as output. The register X is transformed into Y by the operation: the two registers never simultaneously co-exist, so it is not meaningful to consider the joint state of X and Y in this situation. (Nothing prevents us from taking Y to be equal to X, however, and in this situation we view the operation as simply acting on X rather than transforming it into another register.) As before, the classical state sets of X and Y will be taken to be arbitrary finite and non-empty sets Σ and Γ, respectively, throughout this discussion.

To describe the collection of all valid quantum operations transforming a register X into a register Y in mathematical terms, it is helpful to first consider the classical case. Every randomized process transforming X into Y is described by some linear mapping M from $\mathbb{R}^{\Sigma}$ to $\mathbb{R}^{\Gamma}$, which we identify with a matrix $M \in \mathbb{R}^{\Gamma\times\Sigma}$. If the state of the register X is described by the probability vector $v \in \mathbb{R}^{\Sigma}$ and the process M is performed, the resulting state of Y is given by $Mv \in \mathbb{R}^{\Gamma}$. The mapping M must of course map probability vectors to probability vectors if it is to be considered a valid physical process, and this condition is equivalent to the condition that M is *stochastic*: the entries of M must be non-negative real numbers, and the entries in each column must sum to 1.

In the quantum setting, operations map density matrices to density matrices, rather than mapping probability vectors to probability vectors. Thus, an operation transforming X into Y is described by a linear mapping of the form $\Phi : \mathbb{C}^{\Sigma\times\Sigma} \to \mathbb{C}^{\Gamma\times\Gamma}$. If the state of X is $\rho \in \mathbb{C}^{\Sigma\times\Sigma}$ and the operation Φ is performed, the output register Y will be left in the state $\Phi(\rho) \in \mathbb{C}^{\Gamma\times\Gamma}$. If such a linear mapping is to map density matrices to density matrices, two conditions must be satisfied:

1. The mapping Φ must be *positive*: for every positive semidefinite matrix $A \in \mathbb{C}^{\Sigma\times\Sigma}$, it must hold that $\Phi(A)$ is positive semidefinite.
2. The mapping Φ must be *trace-preserving*: $\mathrm{Tr}(\Phi(A)) = \mathrm{Tr}(A)$ for all matrices $A \in \mathbb{C}^{\Sigma\times\Sigma}$.

It turns out, however, that a complication not found in the classical setting arises in the quantum setting: while condition 1 is certainly necessary for a mapping to describe a valid operation, it is not sufficient when combined with condition 2. This issue will be addressed shortly, but it is best done after a brief discussion of independence among quantum operations.

As is the case for states, independent quantum operations are described by tensor products. For instance, if Φ_1 transforms X_1 into Y_1, Φ_2 transforms X_2 into Y_2 and the two operations are performed *independently*, then the transformation from the pair $(\mathsf{X}_1, \mathsf{X}_2)$ into the pair $(\mathsf{Y}_1, \mathsf{Y}_2)$ is given by the mapping $\Phi_1 \otimes \Phi_2$. This is the uniquely defined linear mapping of the form

$$\Phi_1 \otimes \Phi_2 : \mathbb{C}^{(\Sigma_1\times\Sigma_2)\times(\Sigma_1\times\Sigma_2)} \to \mathbb{C}^{(\Gamma_1\times\Gamma_2)\times(\Gamma_1\times\Gamma_2)}$$

that satisfies

$$(\Phi_1 \otimes \Phi_2)(A_1 \otimes A_2) = \Phi_1(A_1) \otimes \Phi_2(A_2)$$

for every choice of matrices $A_1 \in \mathbb{C}^{\Sigma_1\times\Sigma_1}$ and $A_2 \in \mathbb{C}^{\Sigma_2\times\Sigma_2}$. This picture extends to three or more operations in the most straightforward way.

Now the complication mentioned above can be explained. Not only must a valid quantum operation Φ map density matrices to density matrices, but it must do so even when it is tensored

with other quantum operations and the joint operation is applied to multiple correlated registers. This requirement is captured precisely by the following condition, which strengthens condition 1 listed above:

1′. The mapping Φ must be *completely positive*: for every finite and non-empty set Δ, and every positive semidefinite matrix $A \in \mathbb{C}^{(\Sigma\times\Delta)\times(\Sigma\times\Delta)}$, it must hold that $(\Phi \otimes \mathrm{Id})(A)$ is positive semidefinite.

In this condition, the mapping Id is the *identity operation*, defined as $\mathrm{Id}(B) = B$ for every matrix $B \in \mathbb{C}^{\Delta\times\Delta}$.

A well-known result due to Choi [Cho75] provides a simple way to check whether a given mapping Φ is completely positive. Supposing that $\Sigma = \{a_1, \ldots, a_n\}$, one considers the block matrix

$$\begin{pmatrix} \Phi(|a_1\rangle\langle a_1|) & \cdots & \Phi(|a_1\rangle\langle a_n|) \\ \vdots & \ddots & \vdots \\ \Phi(|a_n\rangle\langle a_1|) & \cdots & \Phi(|a_n\rangle\langle a_n|) \end{pmatrix} \in \mathbb{C}^{(\Sigma\times\Gamma)\times(\Sigma\times\Gamma)}, \tag{3}$$

which is called the *Choi matrix* of Φ. (In the Dirac notation, $|a\rangle\langle b|$ is the matrix whose (a, b) entry is 1 and whose other entries are 0.) It holds that Φ is completely positive if and only if its Choi matrix (3) is positive semidefinite.

Incidentally, the Choi matrix provides one way to give an explicit expression of a given quantum operation Φ: the blocks describe the action of Φ on the basis $\{|a\rangle\langle b| : a, b \in \Sigma\}$, from which its action on any other matrix can be determined by linearity.

Another characterization of completely positive mappings is that they are precisely the mappings that can be written in the form

$$\Phi(A) = \sum_{j=1}^{m} M_j A M_j^* \tag{4}$$

for some choice of matrices $M_1, \ldots, M_m \in \mathbb{C}^{\Gamma\times\Sigma}$. Mappings of the form (4) are trace-preserving if and only if $\sum_{j=1}^{m} M_j^* M_j = \mathbb{1}$ (for $\mathbb{1}$ denoting the identity matrix in $\mathbb{C}^{\Sigma\times\Sigma}$).

A matrix $U \in \mathbb{C}^{\Sigma\times\Sigma}$ is *unitary* if it holds that $U^*U = \mathbb{1}$. For such a matrix, it holds that $\Phi_U(A) = UAU^*$ is a valid quantum operation, and we say that Φ_U is a *unitary operation*. Much like the identification of a pure state uu^* with the vector u, we sometimes identify the unitary operation Φ_U with the unitary matrix U. Similar to the vector u, the matrix U is only determined up to a global phase. Unitary operations are typically the only operations (aside from standard-basis measurements) considered in the pure state model of quantum information.

An example of a non-unitary quantum operation is the so-called *phasing-damping operation* $\Lambda : \mathbb{C}^{\Sigma\times\Sigma} \to \mathbb{C}^{\Sigma\times\Sigma}$. This operation zeroes-out the off-diagonal entries of a given matrix, leaving the diagonal entries alone:

$$\Lambda(A)(a, b) = \begin{cases} A(a, b) & \text{if } a = b \\ 0 & \text{if } a \neq b \end{cases}$$

for all $a, b \in \Sigma$. One can view Λ as a noiseless *classical* channel: it transmits classical states perfectly, but it transforms arbitrary quantum states into classical states (represented by diagonal density matrices). Another example is the trace, which can be viewed as a quantum operation from a given register X to a new register Y having a single classical state. (When a register has a single classical state, one can safely view that it is not there at all. This is because the only valid quantum state of a register with one classical state is the 1×1 matrix 1, which has no effect when tensored to any another matrix—so it can be ignored.)

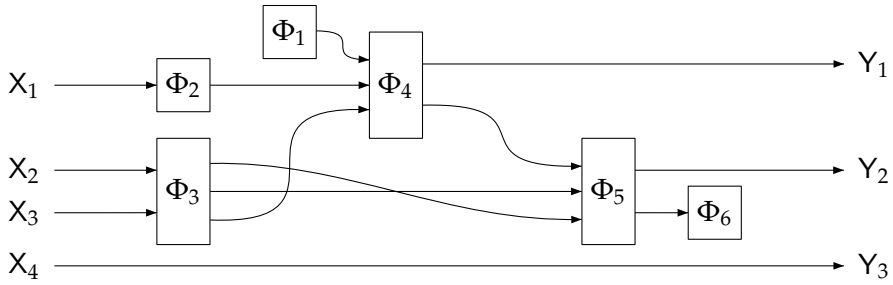

Figure 1: An example of a quantum circuit. The input qubits are labelled $X_1, \ldots, X_4$, the output qubits are labelled $Y_1, \ldots, Y_3$, and the gates are labelled by (hypothetical) quantum operations $\Phi_1, \ldots, \Phi_6$.

2.5 Measurements

The notion of a *measurement* is important in quantum information theory, but for the sake of this paper it is not necessary to consider measurements as distinct from quantum operations. There is no loss of generality in defining measurements in the following way:

1. A *standard-basis measurement* of a register X is described by the phase-damping operation Λ. When X is measured in this way, it is left in a classical state (described by a density matrix of the form $\rho = \mathrm{diag}(v)$, for $v \in \mathbb{R}^{\Sigma}$ being a probability vector) that corresponds to the outcome of the measurement.
2. A general measurement of a register X can always be described as the composition of a quantum operation Φ transforming X into Y, followed by a standard-basis measurement of Y.

It should be noted that the register that results from a measurement can still be correlated with other registers, and furthermore this output register represents the only record of the measurement outcome. If one wishes to consider a hypothetical observer that lies outside the system being modeled, it becomes necessary to condition states on measurement outcomes in a similar way to what would be done classically.

3 Quantum circuits

Having presented the basic notions of quantum information theory, we are now ready to apply them to a circuit model of computation.

A *quantum circuit* is an acyclic network of *quantum gates* connected by *wires*. The quantum gates represent quantum operations while the wires represent the qubits on which the gates act. An example of a quantum circuit having four input qubits and three output qubits is pictured in Figure 1. In general, a quantum circuit may have n input qubits and m output qubits for any choice of integers $n, m \geq 0$. Such a circuit induces some quantum operation from n qubits to m qubits, determined by composing the actions of the individual gates in the appropriate way. The *size* of a quantum circuit is the total number of gates plus the total number of wires in the circuit.

A *unitary quantum circuit* is a quantum circuit in which all of the gates correspond to unitary quantum operations. Naturally this requires that every gate, and hence the circuit itself, has an equal number of input and output qubits. It is common in the study of quantum computing

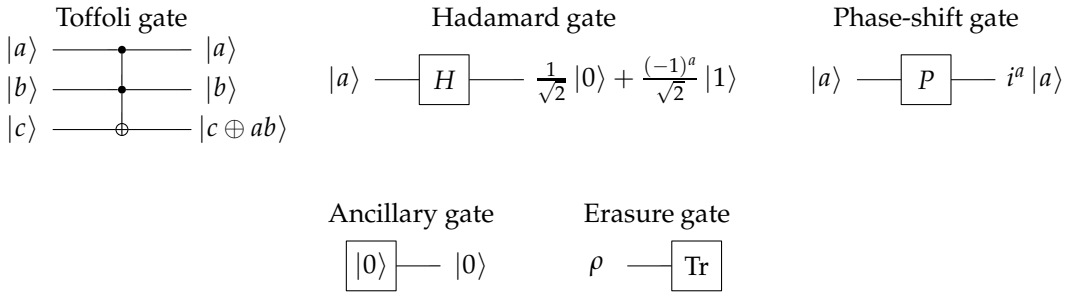

Figure 2: A universal collection of quantum gates: Toffoli, Hadamard, phase-shift, ancillary, and erasure gates.

that one works entirely with unitary quantum circuits. The unitary model and general model are closely related, as will soon be explained.

3.1 A finite universal gate set

Restrictions must be placed on the gates from which quantum circuits may be composed if the quantum circuit model is to be used for complexity theory—for without such restrictions it cannot be argued that each quantum gate corresponds to an operation with unit-cost. A typical way in which this is done is simply to fix a suitable finite set of allowable gates. For the remainder of this paper, quantum circuits will be assumed to be composed of gates from the following list (representing a standard choice for a gate set):

1. *Toffoli gates*. A Toffoli gate is a three-qubit unitary gate Φ_T identified with the unitary matrix T defined by the following action on elementary unit vectors:

$$T : |a\rangle |b\rangle |c\rangle \mapsto |a\rangle |b\rangle |c \oplus ab\rangle .$$

2. *Hadamard gates*. A Hadamard gate is a single-qubit unitary gate Φ_H identified with the unitary matrix H defined by the following action on elementary unit vectors:

$$H : |a\rangle \mapsto \frac{1}{\sqrt{2}} |0\rangle + \frac{(-1)^a}{\sqrt{2}} |1\rangle .$$

3. *Phase-shift gates*. A Phase-shift gate is a single-qubit unitary gate Φ_P identified with the unitary matrix P defined by the following action on elementary unit vectors:

$$P : |a\rangle \mapsto i^a |a\rangle .$$

4. *Ancillary gates*. Ancillary gates are non-unitary gates that take no input and produce a single qubit in the state $|0\rangle$ as output.
5. *Erasure gates*. Erasure gates are non-unitary gates that take a single qubit as input and produce no output. Their effect is represented by the partial trace on the qubit they take as input.

The symbols used to denote these gates in quantum circuit diagrams are shown in Figure 2.

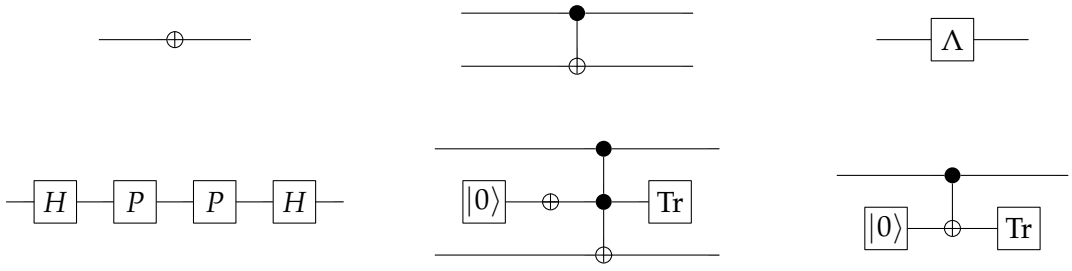

Figure 3: Three simple quantum gates: a NOT gate, a controlled-NOT gate and a phase-damping gate. For each gate, a shorthand notation is shown on the top, along with the gate's implementation using previously defined gates on the bottom.

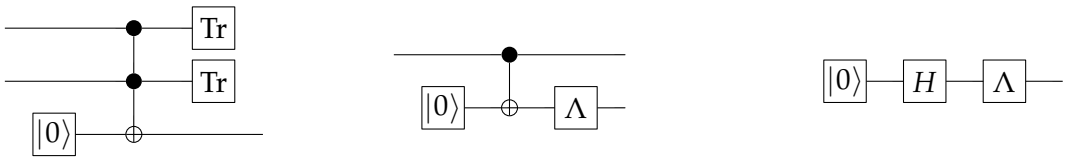

Figure 4: Three classical gates, implemented by quantum gates: an AND gate, a FANOUT or COPY gate and a random bit generator.

The above gate set is *universal* in a strong sense: every quantum operation mapping qubits to qubits can be approximated to within any desired degree of accuracy by some quantum circuit composed of gates from this set. Moreover, the size of the approximating circuit can be made to scale well with respect to the desired accuracy. (It is inevitable, however, that the size of the approximating circuit is exponential in the number of input and output qubits in the worst case [Kni95].) The following theorem expresses this fact in more precise terms.

Theorem 1 (Universality Theorem)**.** *Let Φ be an arbitrary quantum operation from n qubits to m qubits. Then for every $\varepsilon > 0$ there exists a quantum circuit Q with n input qubits and m output qubits such that Q and Φ are ε-indistinguishable. Moreover, for fixed n and m, the circuit Q may be taken to satisfy* $\text{size}(Q) = \text{poly}(\log(1/\varepsilon))$.

Readers interested in further details on the facts comprising this theorem can find them in Nielsen and Chuang [NC00] and Kitaev, Shen, and Vyalyi [KSV02]. The theorem refers to the notion of *ε-indistinguishability*, which will not be explained in this paper—it may be considered as a natural analogue of total variation distance for quantum operations rather than probability vectors, and is formalized by a norm commonly referred to as either the *completely bounded trace norm* or the *diamond norm*.

A few very simple circuit constructions are shown in Figures 3 and 4. The circuits in Figure 4 are, in particular, meant to illustrate that classical Boolean circuits making use of random bits can easily be simulated by quantum circuits.

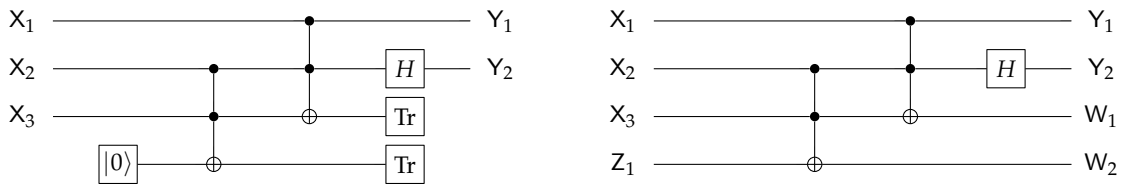

Figure 5: A general quantum circuit (left) and its unitary purification (right).

3.2 Unitary purifications of quantum circuits

The connection between general and unitary quantum circuits can be understood through the notion of a *unitary purification* of a general quantum circuit. The universal gate set described above has the effect of making the notion of a unitary purification of a general quantum circuit nearly trivial at a technical level.

Suppose that Q is a quantum circuit taking input qubits $(\mathsf{X}_1, \ldots, \mathsf{X}_n)$ and producing output qubits $(\mathsf{Y}_1, \ldots, \mathsf{Y}_m)$, and assume there are j ancillary gates and k erasure gates among the gates of Q. A new quantum circuit R may then be formed by removing the ancillary and erasure gates, and to account for the removal of these gates the circuit R takes j additional input qubits $(\mathsf{Z}_1, \ldots, \mathsf{Z}_j)$ and produces k additional output qubits $(\mathsf{W}_1, \ldots, \mathsf{W}_k)$. Figure 5 illustrates this simple process. The circuit R is said to be a *unitary purification* of Q. It holds that R is equivalent to Q, provided the qubits $(\mathsf{Z}_1, \ldots, \mathsf{Z}_j)$ are each initially set to the $|0\rangle$ state and the qubits $(\mathsf{W}_1, \ldots, \mathsf{W}_k)$ are ignored after the circuit is run. (There is no observable difference between discarding and ignoring qubits.)

Despite the simplicity of this process, it is often useful to consider the properties of unitary purifications of general quantum circuits. The fact that measurements in a quantum computation can always be simulated by unitary gates and ancillary qubits, to be either measured or ignored at the end of a computation, is sometimes called the *principle of deferred measurement*.

4 Polynomial-time quantum computations and subroutines

Having given a summary of quantum information and the quantum circuit model, we now turn to quantum complexity theory. Using the quantum circuit model, one can consider interesting quantum complexity-theoretic analogues of complexity classes such as NC, NP, AM, IP and PSPACE; but for this paper we will restrict our attention to the most basic of quantum complexity classes: BQP. The class BQP contains all decision problems solvable in *bounded-error quantum polynomial time*, and therefore represents a natural quantum analogue of BPP.

4.1 Polynomial-time uniform families of quantum circuits and BQP

We begin with a quantum circuit-based definition[2] of BQP. To define the class BQP using the quantum circuit model, a brief discussion of circuit encodings and polynomial-time uniformity is in order.

[2]Historically speaking, BQP was first defined through the use of the *quantum Turing machine* model of computation [BV93, BV97]. This model is known to be equivalent to the quantum circuit model through the work of Yao [Yao93].

Any quantum circuit formed from the universal gate set described in the previous section can be encoded as a binary string, with respect to any number of different encoding schemes. As is the case when uniform families of classical Boolean circuits are studied, many specific details of the encoding scheme are not important; and for the sake of brevity we will leave it to the reader to imagine that a sensible and efficient encoding scheme for quantum circuits has been fixed. Naturally it is assumed that a circuit's size and its encoding length are polynomially related.

Now, as any quantum circuit represents a finite computation with some fixed number of input and output qubits, quantum algorithms are modelled by *families* of quantum circuits. The most typical assumption is that a quantum circuit family that describes an algorithm contains one circuit for each possible input length. It must be possible to efficiently generate the circuits in a given family in order for that family to represent an efficient, finitely specified algorithm. The following definition formalizes this notion.

Definition 2. A collection $\{Q_n : n \in \mathbb{N}\}$ of quantum circuits is said to be *polynomial-time uniform* if there exists a polynomial-time deterministic Turing machine that, on input 1^n, outputs an encoding of Q_n (for each $n \in \mathbb{N}$).

With this definition in hand, the complexity class BQP may now be defined as the class of languages decidable with bounded-error by polynomial-time uniform families of quantum circuits in a fairly straightforward way.

Definition 3. A language $L \subseteq \{0,1\}^*$ is in BQP if there exists a polynomial-time uniform family $Q = \{Q_n : n \in \mathbb{N}\}$ of quantum circuits with the following properties.

1. Each circuit Q_n takes n qubits as input and produces one output qubit.
2. If $x \in L$, then $\Pr[Q(x) = 1] \geq 2/3$.
3. If $x \notin L$, then $\Pr[Q(x) = 0] \geq 2/3$.

(In the second and third items, for each string $x \in \{0,1\}^*$ we write $Q(x)$ to denote the random Boolean value obtained by applying $Q_{|x|}$ to the input state $|x\rangle$ and measuring the output qubit with respect to the standard-basis measurement.)

This definition may be extended from languages to promise problems [ESY84] in the most straightforward way. The function class FBQP may also be defined along similar lines, where $Q(x)$ is understood to be a random string that must agree with a given function's value $f(x)$ with probability at least $2/3$.

4.2 Circuit compositions

One of the most appealing aspects of the quantum circuit model, as it has been described in this paper, is that it allows one to compose circuits in useful ways.

As an example, consider the constant value $2/3$ appearing in the definition of BQP. As one might expect, replacing this value with any constant in the open interval $(1/2, 1)$, or even by a value of the form $1 - 2^{-\mathrm{poly}(n)}$, does not change the class defined; we still obtain the class BQP. This fact is proved in a similar way to a typical proof of the analogous fact for the class BPP. Given a circuit family Q whose probability of correctness is at least $c > 1/2$, we run the algorithm described by Q many times independently, and output the majority of the individual output values. Applying the Chernoff bound in the usual way leads to an exponentially small error probability

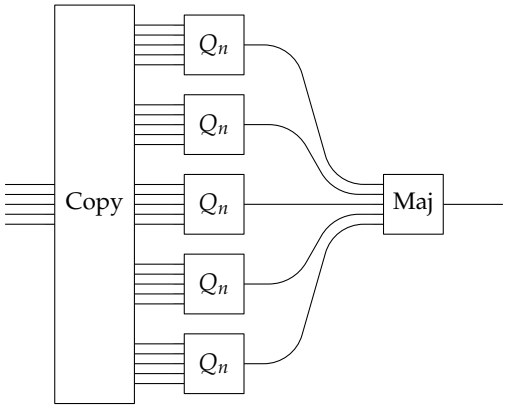

Figure 6: An illustration of error reduction for BQP. In the pictured circuit, Q_n is applied independently 5 times, but any polynomial number of repetitions could be performed instead.

in the number of independent repetitions. The aspect of this argument that is most relevant to this paper is the step in which multiple independent runs of the algorithm described by Q are performed, and the results processed to yield a final answer. This is a near-trivial step in the classical setting, but one may be concerned that this changes for quantum circuits.

Fortunately it does not: it is straightforward for quantum circuits as well. To see this, consider the circuit depicted in Figure 6. The boxed labelled Q_n represent the original circuit Q_n, while the boxes labelled "Copy" and "Maj" are quantum circuits derived from classical Boolean circuits as described in the previous section. The Copy circuit simply copies the input string x so that each circuit Q_n is provided with the right input string, and the Maj circuit computes the majority of its input bits. Uniformity of the resulting circuit family is a purely classical computational matter, and is easily argued.

A related example concerns the claim that integer factoring is BQP, which was discussed in the introduction. Let us consider the following language representation of the integer factoring problem:

$$\text{Factoring} = \{\langle N, k\rangle \,:\, N \text{ has a nontrivial factor less than } k\}\,.$$

(Here it is to be assumed that N and k are positive integers expressed in binary notation.) Typical presentations of Shor's algorithms do not explicitly construct quantum circuits that prove Factoring is in BQP. Instead, they construct circuits that output a non-trivial factor of a given input number N with some small but non-negligible probability of success.

How, then, does one reason that Factoring is in BQP? The answer is that it follows from a straightforward and essentially classical construction, as is suggested by Figure 7. The circuits labelled "Classical processing" have not been specified and are left to the reader's imagination. They can be chosen as if the circuits implementing Shor's algorithm, labelled "Shor" in the figure, are purely classical processes that take integers as input and output a non-trivial factor of that input with non-negligible probability. By making use of a polynomial number of uses of Shor's algorithm, one can obtain a complete prime factorization of N with high probability, and then compare the smallest nontrivial factor with k in the final stage of the process.

As was observed in the case of the error-reduction procedure, each of the circuits labelled "Classical processing" may be implemented by quantum gates, and a unitary purification of the entire process considered. In this way one would obtain a unitary circuit, which by definition

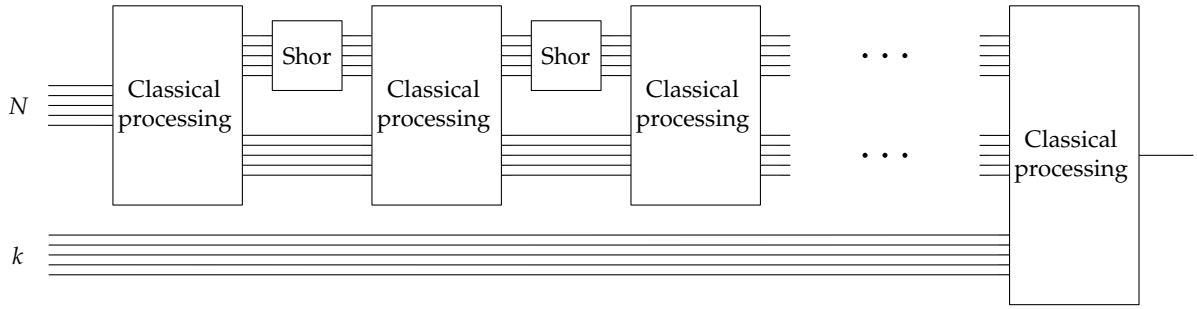

Figure 7: A quantum circuit for the Factoring decision problem, based on circuits implementing Shor's algorithm (as a sampling procedure).

contains no intermediate measurements or other non-unitary gates, that decides the Factoring language: a single measurement of the output qubit shown in the figure provides the correct answer with high probability.

4.3 Oracles and garbage-free subroutines

One final quantum circuit construction that will be discussed in this paper is the *BQP subroutine theorem* of Bennett, Bernstein, Brassard and Vazirani [BBBV97]. Technically speaking, this result has no direct connection to the factoring problem or the relationship between BQP and sampling procedures, but it is an important result that fits well into the theme of this paper.

Oracle queries are represented in the quantum circuit model by families $\{U_n : n \in \mathbb{N}\}$ of unitary quantum gates, one for each possible query length. Each gate U_n acts on $n+1$ qubits, with the effect on elementary unit vectors given by

$$U_n : |x\rangle |a\rangle \mapsto |x\rangle |a \oplus \chi_A(x)\rangle$$

for all $x \in \Sigma^n$ and $a \in \Sigma$, where χ_A is the characteristic predicate of an oracle A under consideration. When quantum computations relative to such an oracle are to be studied, quantum circuits composed of ordinary quantum gates as well as the oracle gates $\{U_n\}$ are considered, with the interpretation being that each instance of U_n in such a circuit represents one oracle query. It is critical to many results concerning quantum oracles that the above definition takes each U_n to be unitary, thereby allowing these gates to make queries "in superposition" and to take advantage of interference patterns generated by the queries.

Suppose that it is established that a particular language A is in BQP, which by definition means that there must exist an efficient quantum algorithm (represented by a family of quantum circuits) for A. It is then natural to consider the use of that algorithm as a subroutine in other quantum algorithms for more complicated problems, and ideally the algorithm for A should function as an oracle for A. A problem arises, however, when queries to an algorithm for A are made in superposition: a given BQP algorithm for A is only guaranteed to work correctly on classical inputs. It could be, for instance, that some algorithm for A begins by applying phase-damping gates to all of its input qubits, or perhaps this happens inadvertently as a result of the computation.

What the BQP subroutine theorem establishes is that it is possible, up to the introduction of an exponentially small error, to convert an arbitrary BQP algorithm to a unitary oracle. A precise statement of the theorem follows.

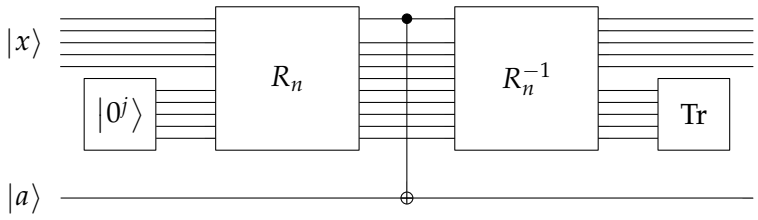

Figure 8: A quantum circuit approximating an oracle-gate implementation of a BQP computation. Here R_n is a unitary purification of a circuit having exponentially small error for some problem in BQP, while R_n^{-1} is a circuit implementing the inverse of the unitary operation performed by R_n. (Such a circuit is easily constructed, given the description of R_n.)

Theorem 4 (BQP subroutine theorem)**.** *Let A be a language in* BQP *and let $\{U_n : n \in \mathbb{N}\}$ be the collection of unitary quantum operations that act as an oracle for A as described above. For any choice of a polynomial p there exists a polynomial-time uniform family of circuits $\{Q_n : n \in \mathbb{N}\}$ with the property that Q_n and U_n are $2^{-p(n)}$-indistinguishable for each $n \in \mathbb{N}$.*

The theorem extends to promise problems, as well as functions in FBQP, in the most straightforward way.

The proof of this theorem is remarkably simple: given a BQP algorithm for A, one first uses error-reduction to obtain a circuit family $R = \{R_n : n \in \mathbb{N}\}$ having exponentially small error for A. The circuit illustrated in Figure 8 then approximates a unitary operation with the desired properties. Note that the j ancillary qubits do not really need to be traced out at the end of the computation: they may simply be ignored. Alternately, they could be used again as ancillary qubits for some other computation, for their state will be exponentially close to their original pure state $|0^j\rangle$ at the end of the computation.

5 Conclusion

This paper has discussed quantum information and quantum circuits at a very basic level that was intended to aid those with an interest, but little familiarity, with quantum computing. Many interesting topics and results in quantum complexity theory have, of course, been ignored. Readers interested in learning more about quantum complexity theory may find the survey paper [Wat09] (which admittedly has a nontrivial overlap with the current paper) to be helpful.

There are countless open questions about quantum complexity theory, quantum algorithms and quantum information, including the following ones:

1. Is the graph isomorphism problem in BQP?
2. Is BQP contained in PH? Is there an oracle relative to which BQP is not contained in PH?
3. Would there be interesting complexity-theoretic consequences (such as the collapse of PH) of the containment NP $\subseteq$ BQP?
4. Can the GCD of two integers be computed in QNC, the natural (bounded error) quantum analogue of NC?

While these are likely to be difficult problems that are not necessarily suitable for beginners, this paper will have served a useful purpose if it encourages researchers without expertise in quantum computing to consider these questions and to share their insights on them.

References

[AB09] S. Arora and B. Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009.

[AKN98] D. Aharonov, A. Kitaev, and N. Nisan. Quantum circuits with mixed states. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pages 20–30, 1998.

[BBBV97] C. Bennett, E. Bernstein, G. Brassard, and U. Vazirani. Strengths and weaknesses of quantum computing. *SIAM Journal on Computing*, 26(5):1510–1523, 1997.

[BV93] E. Bernstein and U. Vazirani. Quantum complexity theory (preliminary abstract). In *Proceedings of the 25th Annual ACM Symposium on Theory of Computing*, pages 11–20, 1993.

[BV97] E. Bernstein and U. Vazirani. Quantum complexity theory. *SIAM Journal on Computing*, 26(5):1411–1473, 1997.

[Cho75] M.-D. Choi. Completely positive linear maps on complex matrices. *Linear Algebra and Its Applications*, 10(3):285–290, 1975.

[ESY84] S. Even, A. Selman, and Y. Yacobi. The complexity of promise problems with applications to public-key cryptography. *Information and Control*, 61(2):159–173, 1984.

[KLM07] P. Kaye, R. Laflamme, and M. Mosca. *An Introduction to Quantum Computing*. Oxford University Press, 2007.

[Kni95] E. Knill. Approximation by quantum circuits. Technical Report LAUR-95-2225, Los Alamos National Laboratory, 1995. Available as arXiv.org e-Print quant-ph/9508006.

[KSV02] A. Kitaev, A. Shen, and M. Vyalyi. *Classical and Quantum Computation*, volume 47 of *Graduate Studies in Mathematics*. American Mathematical Society, 2002.

[NC00] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[Sho94] P. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings of the 35th Annual IEEE Symposium on Foundations of Computer Science*, pages 124–134, 1994.

[Sho97] P. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997.

[Wat09] J. Watrous. Quantum computational complexity. In *Encyclopedia of Complexity and System Science*. Springer, 2009.

[Yao93] A. Yao. Quantum circuit complexity. In *Proceedings of the 34th Annual IEEE Symposium on Foundations of Computer Science*, pages 352–361, 1993.