Quantum Computational Complexity

John Watrous

Institute for Quantum Computing and School of Computer Science
University of Waterloo, Waterloo, Ontario, Canada.

Article outline

Glossary

Quantum circuit.

A quantum circuit is an acyclic network of quantum gates connected by wires: the gates represent quantum operations and the wires represent the qubits on which these operations are performed. The quantum circuit model is the most commonly studied model of quantum computation.

Quantum complexity class.

A quantum complexity class is a collection of computational problems that are solvable by a chosen quantum computational model that obeys certain resource constraints. For example, BQP is the quantum complexity class of all decision problems that can be solved in polynomial time by a quantum computer.

Quantum proof.

A quantum proof is a quantum state that plays the role of a witness or certificate to a quantum computer that runs a verification procedure. The quantum complexity class QMA is defined by this notion: it includes all decision problems whose yes-instances are efficiently verifiable by means of quantum proofs.

Quantum interactive proof system.

A quantum interactive proof system is an interaction between a verifier and one or more provers, involving the processing and exchange of quantum information, whereby the provers attempt to convince the verifier of the answer to some computational problem.

I Definition of the subject and its importance

The inherent difficulty, or *hardness*, of computational problems is a fundamental concept in computational complexity theory. Hardness is typically formalized in terms of the resources required by different models of computation to solve a given problem, such as the number of steps of a deterministic Turing machine. A variety of models and resources are often considered, including deterministic, nondeterministic and probabilistic models; time and space constraints; and interactions among models of differing abilities. Many interesting relationships among these different models and resource constraints are known.

One common feature of the most typically studied computational models and resource constraint is that they are *physically motivated*. This is quite natural, given that computers are physical devices, and to a significant extent it is their study that motivates and directs research on computational complexity. The predominant example is the class of polynomial-time computable functions, which ultimately derives its relevance from physical considerations; for it is a mathematical abstraction of the class of functions that can be efficiently computed without error by physical computing devices.

In light of its close connection to the physical world, it seems only natural that modern physical theories should be considered in the context of computational complexity. In particular, *quantum mechanics* is a clear candidate for a physical theory to have the potential for implications, if not to computational complexity then at least to computation more generally. Given the steady decrease in the size of computing components, it is inevitable that quantum mechanics will become increasingly relevant to the construction of computers—for quantum mechanics provides a remarkably accurate description of extremely small physical systems (on the scale of atoms) where classical physical theories have failed completely. Indeed, an extrapolation of Moore's Law predicts subatomic computing components within the next two decades [83, 78]; a possibility inconsistent with quantum mechanics as it is currently understood.

That quantum mechanics should have implications to computational complexity theory, however, is much less clear. It is only through the remarkable discoveries and ideas of several researchers, including Richard Feynman [50], David Deutsch [41, 42], Ethan Bernstein and Umesh Vazirani [30, 31], and Peter Shor [93, 94], that this potential has become evident. In particular, Shor's polynomial-time quantum factoring and discrete-logarithm algorithms [94] give strong support to the conjecture that quantum and classical computers yield differing notions of computational hardness. Other quantum complexity-theoretic concepts, such as the efficient verification of quantum proofs, suggest a wider extent to which quantum mechanics influences computational complexity.

It may be said that the principal aim of quantum computational complexity theory is to understand the implications of quantum physics to computational complexity theory. To this end, it considers the hardness of computational problems with respect to models of quantum computation, classifications of problems based on these models, and their relationships to classical models and complexity classes.

II Introduction

This article surveys quantum computational complexity, with a focus on three fundamental notions: polynomial-time quantum computations, the efficient verification of quantum proofs, and quantum interactive proof systems. Based on these notions one defines quantum complexity classes, such as BQP, QMA, and QIP, that contain computational problems of varying hardness.

Properties of these complexity classes, and the relationships among these classes and classical complexity classes, are presented. As these notions and complexity classes are typically defined within the quantum circuit model, this article includes a section that focuses on basic properties of quantum circuits that are important in the setting of quantum complexity. A selection of other topics in quantum complexity, including quantum advice, space-bounded quantum computation, and bounded-depth quantum circuits, is also presented.

Two different but closely related areas of study are not discussed in this article: *quantum query complexity* and *quantum communication complexity*. Readers interested in learning more about these interesting and active areas of research may find the surveys of Brassard [35], Cleve [36], and de Wolf [107] to be helpful starting points.

It is appropriate that brief discussions of computational complexity theory and quantum information precede the main technical portion of the article. These discussions are intended only to highlight the aspects of these topics that are non-standard, require clarification, or are of particular importance in quantum computational complexity. In the subsequent sections of this article, the reader is assumed to have basic familiarity with both topics, which are covered in depth by several text books [14, 44, 61, 68, 84, 87].

II.1 Computational complexity

Throughout this article the binary alphabet $\{0, 1\}$ is denoted Σ, and all computational problems are assumed to be encoded over this alphabet. As usual, a function $f : \Sigma^* \to \Sigma^*$ is said to be *polynomial-time computable* if there exists a polynomial-time deterministic Turing machine that outputs $f(x)$ for every input $x \in \Sigma^*$. Two related points on the terminology used throughout this article are as follows.

1. A function of the form $p : \mathbb{N} \to \mathbb{N}$ (where $\mathbb{N} = \{0, 1, 2, \ldots\}$) is said to be a *polynomial-bounded function* if and only if there exists a polynomial-time deterministic Turing machine that outputs $1^{p(n)}$ on input 1^n for every $n \in \mathbb{N}$. Such functions are upper-bounded by some polynomial, and are efficiently computable.

2. A function of the particular form $a : \mathbb{N} \to [0, 1]$ is said to be *polynomial-time computable* if and only if there exists a polynomial-time deterministic Turing machine that outputs a binary representation of $a(n)$ on input 1^n for each $n \in \mathbb{N}$. References to functions of this form in this article typically concern bounds on probabilities that are functions of the length of an input string to some problem.

The notion of *promise problems* [45, 53] is central to quantum computational complexity. These are decision problems for which the input is assumed to be drawn from some subset of all possible input strings. More formally, a promise problem is a pair $A = (A_{\text{yes}}, A_{\text{no}})$, where $A_{\text{yes}}, A_{\text{no}} \subseteq \Sigma^*$ are sets of strings satisfying $A_{\text{yes}} \cap A_{\text{no}} = \varnothing$. The strings contained in the sets A_{yes} and A_{no} are called the *yes-instances* and *no-instances* of the problem, and have answers *yes* and *no*, respectively. *Languages* may be viewed as promise problems that obey the additional constraint $A_{\text{yes}} \cup A_{\text{no}} = \Sigma^*$. Although complexity theory has traditionally focused on languages rather than promise problems, little is lost and much is gained in shifting one's focus to promise problems. Karp reductions (also called polynomial-time many-to-one reductions) and the notion of completeness are defined for promise problems in the same way as for languages.

Several classical complexity classes are referred to in this article, and compared with quantum complexity classes when relations are known. The following classical complexity classes, which should hereafter be understood to be classes of promise problems and not just languages, are among those discussed.

P	A promise problem $A = (A_{\text{yes}}, A_{\text{no}})$ is in P if and only if there exists a polynomial-time deterministic Turing machine M that accepts every string $x \in A_{\text{yes}}$ and rejects every string $x \in A_{\text{no}}$.
NP	A promise problem $A = (A_{\text{yes}}, A_{\text{no}})$ is in NP if and only if there exists a polynomial-bounded function p and a polynomial-time deterministic Turing machine M with the following properties. For every string $x \in A_{\text{yes}}$, it holds that M accepts (x, y) for some string $y \in \Sigma^{p(\lvert x\rvert)}$, and for every string $x \in A_{\text{no}}$, it holds that M rejects (x, y) for all strings $y \in \Sigma^{p(\lvert x\rvert)}$.
BPP	A promise problem $A = (A_{\text{yes}}, A_{\text{no}})$ is in BPP if and only if there exists a polynomial-time probabilistic Turing machine M that accepts every string $x \in A_{\text{yes}}$ with probability at least $2/3$, and accepts every string $x \in A_{\text{no}}$ with probability at most $1/3$.
PP	A promise problem $A = (A_{\text{yes}}, A_{\text{no}})$ is in PP if and only if there exists a polynomial-time probabilistic Turing machine M that accepts every string $x \in A_{\text{yes}}$ with probability strictly greater than $1/2$, and accepts every string $x \in A_{\text{no}}$ with probability at most $1/2$.
MA	A promise problem $A = (A_{\text{yes}}, A_{\text{no}})$ is in MA if and only if there exists a polynomial-bounded function p and a probabilistic polynomial-time Turing machine M with the following properties. For every string $x \in A_{\text{yes}}$, it holds that $\Pr[M \text{ accepts } (x, y)] \geq \frac{2}{3}$ for some string $y \in \Sigma^{p(\lvert x\rvert)}$; and for every string $x \in A_{\text{no}}$, it holds that $\Pr[M \text{ accepts } (x, y)] \leq \frac{1}{3}$ for all strings $y \in \Sigma^{p(\lvert x\rvert)}$.
AM	A promise problem $A = (A_{\text{yes}}, A_{\text{no}})$ is in AM if and only if there exist polynomial-bounded functions p and q and a polynomial-time deterministic Turing machine M with the following properties. For every string $x \in A_{\text{yes}}$, and at least $2/3$ of all strings $y \in \Sigma^{p(\lvert x\rvert)}$, there exists a string $z \in \Sigma^{q(\lvert x\rvert)}$ such that M accepts (x, y, z); and for every string $x \in A_{\text{no}}$, and at least $2/3$ of all strings $y \in \Sigma^{p(\lvert x\rvert)}$, there are no strings $z \in \Sigma^{q(\lvert x\rvert)}$ such that M accepts (x, y, z).
SZK	A promise problem $A = (A_{\text{yes}}, A_{\text{no}})$ is in SZK if and only if it has a statistical zero-knowledge interactive proof system.
PSPACE	A promise problem $A = (A_{\text{yes}}, A_{\text{no}})$ is in PSPACE if and only if there exists a deterministic Turing machine M running in polynomial space that accepts every string $x \in A_{\text{yes}}$ and rejects every string $x \in A_{\text{no}}$.
EXP	A promise problem $A = (A_{\text{yes}}, A_{\text{no}})$ is in EXP if and only if there exists a deterministic Turing machine M running in exponential time (meaning time bounded by 2^p, for some polynomial-bounded function p), that accepts every string $x \in A_{\text{yes}}$ and rejects every string $x \in A_{\text{no}}$.
NEXP	A promise problem $A = (A_{\text{yes}}, A_{\text{no}})$ is in NEXP if and only if there exists an exponential-time non-deterministic Turing machine N for A.

$$\begin{tikzcd}
& \mathrm{NEXP} \arrow[d, dash] & \\
& \mathrm{EXP} \arrow[d, dash] & \\
& \mathrm{PSPACE} \arrow[dl, dash] \arrow[dr, dash] & \\
\mathrm{AM} \arrow[d, dash] \arrow[drr, dash] & & \mathrm{PP} \arrow[d, dash] \\
\mathrm{SZK} \arrow[d, dash] & & \mathrm{MA} \arrow[dll, dash] \arrow[d, dash] \\
\mathrm{BPP} \arrow[dr, dash] & & \mathrm{NP} \arrow[dl, dash] \\
& \mathrm{P} \arrow[d, dash] & \\
& \mathrm{NC} \arrow[d, dash] & \\
& \mathrm{PL} &
\end{tikzcd}$$

Figure 1: A diagram illustrating known inclusions among most of the classical complexity classes discussed in this paper. Lines indicate containments going upward; for example, AM is contained in PSPACE.

PL A promise problem $A = (A_{\text{yes}}, A_{\text{no}})$ is in PL if and only if there exists a probabilistic Turing machine M running in polynomial time and logarithmic space that accepts every string $x \in A_{\text{yes}}$ with probability strictly greater than $1/2$ and accepts every string $x \in A_{\text{no}}$ with probability at most $1/2$.

NC A promise problem $A = (A_{\text{yes}}, A_{\text{no}})$ is in NC if and only if there exists a logarithmic-space generated family $C = \{C_n : n \in \mathbb{N}\}$ of poly-logarithmic depth Boolean circuits such that $C(x) = 1$ for all $x \in A_{\text{yes}}$ and $C(x) = 0$ for all $x \in A_{\text{no}}$.

For most of the complexity classes listed above, there is a standard way to attach an *oracle* to the machine model that defines the class, which provides a subroutine for solving instances of a chosen problem $B = (B_{\text{yes}}, B_{\text{no}})$. One then denotes the existence of such an oracle with a superscript—for example, P^B is the class of promise problems that can be solved in polynomial time by a deterministic Turing machine equipped with an oracle that solves instances of B (at unit cost). When classes of problems appear as superscripts, one takes the union, as in the following example:

$$\mathrm{P}^{\mathrm{NP}} = \bigcup_{B \in \mathrm{NP}} \mathrm{P}^B.$$

II.2 Quantum information

The standard general description of quantum information is used in this article: mixed states of systems are represented by density matrices and operations are represented by completely positive trace-preserving linear maps. The choice to use this description of quantum information is deserving of a brief discussion, for it will likely be less familiar to many non-experts than the simplified picture of quantum information where states are represented by unit vectors and operations are represented by unitary matrices. This simplified picture is indeed commonly used in the study of both quantum algorithms and quantum complexity theory; and it is often adequate. However, the general picture has major advantages: it unifies quantum information with classical probability theory, brings with it powerful mathematical tools, and allows for simple and intuitive descriptions in many situations where this is not possible with the simplified picture.

Classical simulations of quantum computations, which are discussed below in Section IV.5, may be better understood through a fairly straightforward representation of quantum operations by matrices. This representation begins with a representation of density matrices as vectors based on the function defined as $\operatorname{vec}(|x\rangle\langle y|) = |x\rangle|y\rangle$ for each choice of $n \in \mathbb{N}$ and $x, y \in \Sigma^n$, and extended by linearity to all matrices indexed by Σ^n. The effect of this mapping is to form a column vector by reading the entries of a matrix in rows from left to right, starting at the top. For example,

$$\operatorname{vec}\begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix} = \begin{pmatrix} \alpha \\ \beta \\ \gamma \\ \delta \end{pmatrix}.$$

Now, the effect of a general quantum operation Φ, represented in the typical Kraus form as

$$\Phi(\rho) = \sum_{j=1}^{k} A_j \rho A_j^*,$$

is expressed as a matrix by means of the equality

$$\operatorname{vec}(\Phi(\rho)) = \left(\sum_{j=1}^{k} A_j \otimes \overline{A_j}\right) \operatorname{vec}(\rho).$$

The matrix

$$M_\Phi = \sum_{j=1}^{k} A_j \otimes \overline{A_j}$$

is sometimes called the *natural representation* (or *linear representation*) of the operation Φ. Although this matrix could have negative or complex entries, one can reasonably view it as being analogous to a stochastic matrix that describes a probabilistic computation.

For example, the complete phase-damping channel for a single qubit can be written

$$D(\rho) = |0\rangle\langle 0|\rho|0\rangle\langle 0| + |1\rangle\langle 1|\rho|1\rangle\langle 1|.$$

The effect of this mapping is to zero-out the off-diagonal entries of a density matrix:

$$D\begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix} = \begin{pmatrix} \alpha & 0 \\ 0 & \delta \end{pmatrix}.$$

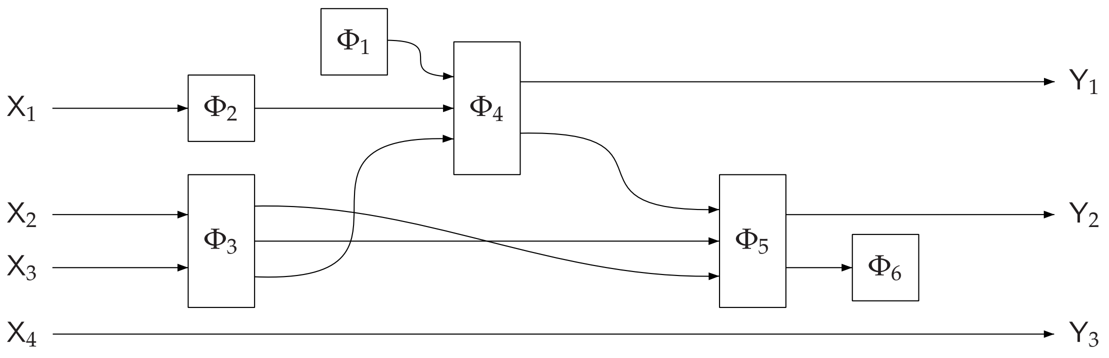


Figure 2: An example of a quantum circuit. The input qubits are labelled $\mathsf{X}_1, \ldots, \mathsf{X}_4$, the output qubits are labelled $\mathsf{Y}_1, \ldots, \mathsf{Y}_3$, and the gates are labelled by (hypothetical) quantum operations $\Phi_1, \ldots, \Phi_6$.

The natural representation of this operation is easily computed:

$$M_D = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

The natural matrix representation of quantum operations is well-suited to performing computations. For the purposes of this article, the most important observation about this representation is that a composition of operations corresponds simply to matrix multiplication.

III The quantum circuit model

III.1 General quantum circuits

The term *quantum circuit* refers to an acyclic network of *quantum gates* connected by *wires*. The quantum gates represent general quantum operations, involving some constant number of qubits, 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 2. 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 that one works entirely with unitary quantum circuits. The unitary model and general model are closely related, as will soon be explained.

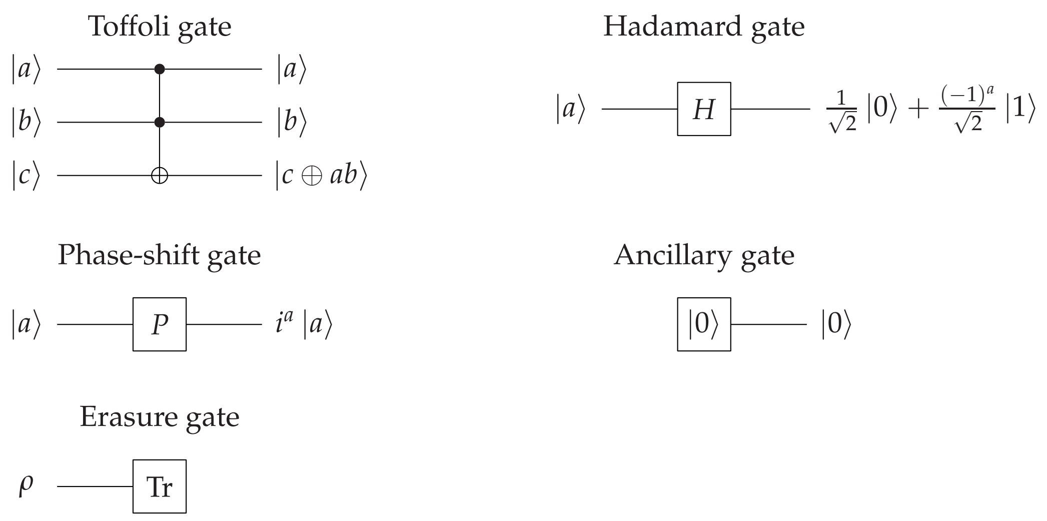


Figure 3: A universal collection of quantum gates: Toffoli, Hadamard, phase-shift, ancillary, and erasure gates.

III.2 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. The usual way in which this is done is simply to fix a suitable finite set of allowable gates. For the remainder of this article, quantum circuits will be assumed to be composed of gates from the following list:

1. *Toffoli gates*. Toffoli gates are three-qubit unitary gates defined by the following action on standard basis states:
$$T : |a\rangle\,|b\rangle\,|c\rangle \mapsto |a\rangle\,|b\rangle\,|c \oplus ab\rangle\,.$$
2. *Hadamard gates*. Hadamard gates are single-qubit unitary gates defined by the following action on standard basis states:
$$H : |a\rangle \mapsto \frac{1}{\sqrt{2}}\,|0\rangle + \frac{(-1)^a}{\sqrt{2}}\,|1\rangle\,.$$
3. *Phase-shift gates*. Phase-shift gates are single-qubit unitary gates defined by the following action on standard basis states:
$$P : |a\rangle \mapsto i^a\,|a\rangle\,.$$
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 space corresponding to the qubit they take as input.

The symbols used to denote these gates in quantum circuit diagrams are shown in Figure 3. Some additional useful gates are illustrated in Figure 4, along with their realizations as circuits with gates from the chosen basis set.

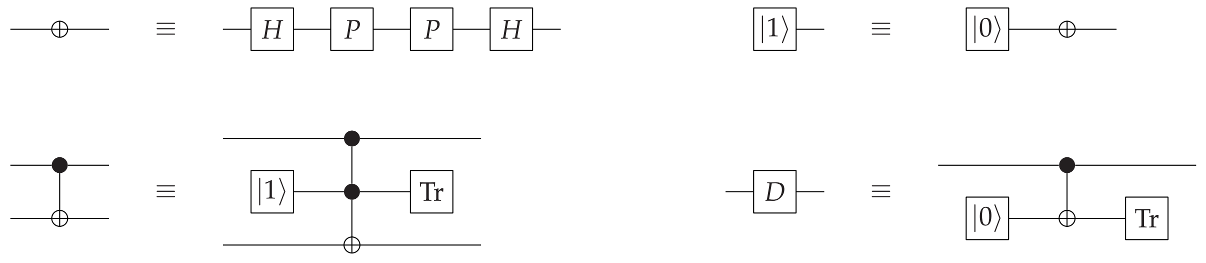


Figure 4: Four additional quantum gates, together with their implementations as quantum circuits. Top left: a NOT gate. Top right: a constant $|1\rangle$ ancillary gate. Bottom left: a controlled-NOT gate. Bottom right: a phase-damping (or decoherence) gate.

The above gate set is *universal* in a strong sense: every quantum operation can be approximated to within any desired degree of accuracy by some quantum circuit. Moreover, the size of the approximating circuit scales well with respect to the desired accuracy. Theorem 1, stated below, expresses this fact in more precise terms, but requires a brief discussion of a specific sense in which one quantum operation approximates another.

A natural and operationally meaningful way to measure the distance between two given quantum operations Φ and Ψ is given by

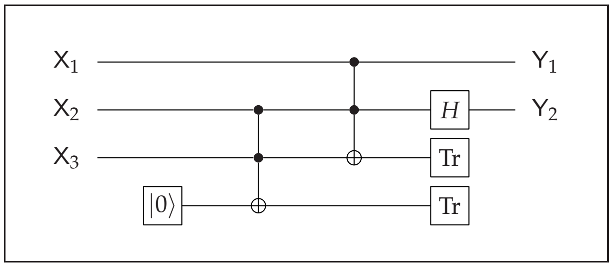
$$\delta(\Phi, \Psi) = \frac{1}{2} \left\| \Phi - \Psi \right\|_{\diamond},$$

where $\|\cdot\|_{\diamond}$ denotes a norm usually known as the *diamond norm* [66, 68]. A technical discussion of this norm is not necessary for the purposes of this article and is beyond its scope. Instead, an intuitive description of the distance measure $\delta(\Phi, \Psi)$ will suffice.

When considering the distance between quantum operations Φ and Ψ, it must naturally be assumed that these operations agree on their numbers of input qubits and output qubits; so assume that Φ and Ψ both map n qubits to m qubits for nonnegative integers n and m. Now, suppose that an arbitrary quantum state on n or more qubits is prepared, one of the two operations Φ or Ψ is applied to the first n of these qubits, and then a general measurement of all of the resulting qubits takes place (including the m output qubits and the qubits that were not among the inputs to the chosen quantum operation). Two possible probability distributions on measurement outcomes arise: one corresponding to Φ and the other corresponding to Ψ. The quantity $\delta(\Phi, \Psi)$ is simply the maximum possible total variation distance between these distributions, ranging over all possible initial states and general measurements. This is a number between 0 and 1 that intuitively represents the *observable difference* between quantum operations. In the special case that Φ and Ψ take no inputs, the quantity $\delta(\Phi, \Psi)$ is simply one-half the trace norm of the difference between the two output states; a common and useful ways to measure the distance between quantum states.

Now the Universality Theorem, which represents an amalgamation of several results that suits the needs of this article, may be stated. In particular, it incorporates the Solovay–Kitaev Theorem, which provides a bound on the size of an approximating circuit as a function of the accuracy.

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 $\delta(\Phi, Q) < \varepsilon$. Moreover, for fixed n and m, the circuit Q may be taken to satisfy* $\operatorname{size}(Q) = \operatorname{poly}(\log(1/\varepsilon))$.



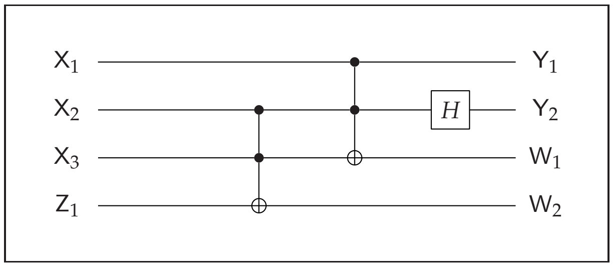


Figure 5: A general quantum circuit (left) and its unitary purification (right).

Note that it is inevitable that the size of Q is exponential in n and m in the worst case [70]. Further details on the facts comprising this theorem can be found in Nielsen and Chuang [84] and Kitaev, Shen, and Vyalyi [68].

III.3 Unitary purifications of quantum circuits

The connection between the general and unitary quantum circuits can be understood through the notion of a *unitary purification* of a general quantum circuit. This may be thought of as a very specific manifestation of the *Stinespring Dilation Theorem* [95], which implies that general quantum operations can be represented by unitary operations on larger systems. It was first applied to the quantum circuit model by Aharonov, Kitaev, and Nisan [10], who gave several arguments in favor of the general quantum circuit model over the unitary model. The term *purification* is borrowed from the notion of a purification of a mixed quantum state, as the process of unitary purification for circuits is similar in spirit. The universal gate described in the previous section 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 k ancillary gates and l erasure gates among the gates of Q to be labelled in an arbitrary order as $G_1, \ldots, G_k$ and $K_1, \ldots, K_l$, respectively. A new quantum circuit R may then be formed by removing the gates labelled $G_1, \ldots, G_k$ and $K_1, \ldots, K_l$; and to account for the removal of these gates the circuit R takes k additional input qubits $(\mathsf{Z}_1, \ldots, \mathsf{Z}_k)$ and produces l additional output qubits $(\mathsf{W}_1, \ldots, \mathsf{W}_l)$. Figure 5 illustrates this process. The circuit R is said to be a unitary purification of Q. It is obvious that R is equivalent to Q, provided the qubits $(\mathsf{Z}_1, \ldots, \mathsf{Z}_k)$ are initially set to the $|0\rangle$ state and the qubits $(\mathsf{W}_1, \ldots, \mathsf{W}_l)$ are traced-out, or simply ignored, after the circuit is run—for this is precisely the meaning of the removed gates.

Despite the simplicity of this process, it is often useful to consider the properties of unitary purifications of general quantum circuits.

III.4 Oracles in the quantum circuit model

Oracles play an important, and yet uncertain, role in computational complexity theory; and the situation is no different in the quantum setting. Several interesting oracle-related results, offering some insight into the power of quantum computation, will be discussed in this article.

Oracle queries are represented in the quantum circuit model by an infinite family

$$\{R_n \,:\, n \in \mathbb{N}\}$$

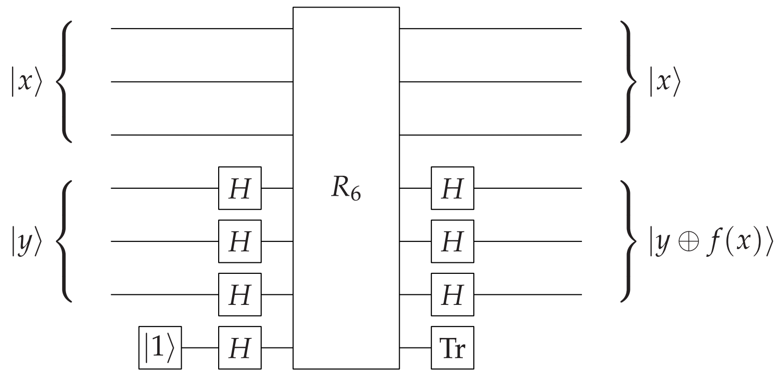


Figure 6: The Bernstein–Vazirani algorithm allows a multiple-bit query to be simulated by a single-bit query. In the example pictured, $f : \Sigma^3 \rightarrow \Sigma^3$ is a given function. To simulate a query to this function, the gate R_6 is taken to be a standard oracle gate implementing the predicate $A(x, z) = \langle f(x), z \rangle$, for $\langle \cdot, \cdot \rangle$ denoting the modulo 2 inner product.

of quantum gates, one for each possible query length. Each gate R_n is a unitary gate acting on $n + 1$ qubits, with the effect on computational basis states given by

$$R_n \, |x, a\rangle = |x, a \oplus A(x)\rangle \tag{1}$$

for all $x \in \Sigma^n$ and $a \in \Sigma$, where A is some predicate that represents the particular oracle 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 $\{R_n\}$ are considered; the interpretation being that each instance of R_n in such a circuit represents one oracle query.

It is critical to many results concerning quantum oracles, as well as most results in the area of quantum query complexity, that the above definition (1) takes each R_n to be unitary, thereby allowing these gates to make queries “in superposition”. In support of this seemingly strong definition is the fact, discussed in the next section, that any efficient algorithm (quantum or classical) can be converted to a quantum circuit that closely approximates the action of these gates.

Finally, one may consider a more general situation in which the predicate A is replaced by a function that outputs multiple bits. The definition of each gate R_n is adapted appropriately. Alternately, the Bernstein–Vazirani algorithm [31] may be used to simulate multiple-bit queries with single-bit queries at unit cost, as illustrated in Figure 6.

IV Polynomial-time quantum computations

This section focuses on *polynomial-time quantum computations*. These are the computations that are viewed, in an abstract and idealized sense, to be efficiently implementable by the means of a quantum computer. In particular, the complexity class BQP (short for *bounded-error quantum polynomial time*) is defined. This is the most fundamentally important of all quantum complexity classes, as it represents the collection of decision problems that can be efficiently solved by quantum computers.

IV.1 Polynomial-time generated circuit families and BQP

To define the class BQP using the quantum circuit model, it is necessary to briefly discuss encodings of circuits and the notion of a polynomial-time generated circuit family.

It is clear that any quantum circuit formed from the gates described in the previous section could be *encoded* as a binary string using any number of different encoding schemes. Such an encoding scheme must be chosen, but its specifics are not important so long as the following simple restrictions are satisfied:

1. The encoding is sensible: every quantum circuit is encoded by at least one binary string, and every binary string encodes at most one quantum circuit.

2. The encoding is efficient: there is a fixed polynomial-bounded function p such that every circuit of size N has an encoding with length at most $p(N)$. Specific information about the structure of a circuit must be computable in polynomial time from an encoding of the circuit.

3. The encoding disallows compression: it is not possible to work with encoding schemes that allow for extremely short (e.g., polylogarithmic-length) encodings of circuits; so for simplicity it is assumed that the length of every encoding of a quantum circuit is at least the size of the circuit.

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 typical assumption is that a quantum circuit family that describes an algorithm contains one circuit for each possible input length. Precisely the same situation arises here as in the classical setting, which is that it should 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. Let $S \subseteq \Sigma^*$ be any set of strings. Then a collection $\{Q_x : x \in S\}$ of quantum circuits is said to be *polynomial-time generated* if there exists a polynomial-time deterministic Turing machine that, on every input $x \in S$, outputs an encoding of Q_x.

This definition is slightly more general than what is needed to define BQP, but is convenient for other purposes. For instance, it allows one to easily consider the situation in which the input, or some part of the input, for some problem is hard-coded into a collection of circuits; or where a computation for some input may be divided among several circuits. In the most typical case that a polynomial-time generated family of the form $\{Q_n : n \in \mathbb{N}\}$ is referred to, it should be interpreted that this is a shorthand for $\{Q_{1^n} : n \in \mathbb{N}\}$. Notice that every polynomial-time generated family $\{Q_x : x \in S\}$ has the property that each circuit Q_x has size polynomial in $|x|$. Intuitively speaking, the number of quantum and classical computation steps required to implement such a computation is polynomial; and so operations induced by the circuits in such a family are viewed as representing *polynomial-time quantum computations*.

The complexity class BQP, which contains those promise problems abstractly viewed to be efficiently solvable using a quantum computer, may now be defined. More precisely, BQP is the class of promise problems that can be solved by polynomial-time quantum computations that may have some small probability to make an error. For decision problems, the notion of a polynomial-time quantum computation is equated with the computation of a polynomial-time generated quantum circuit family $Q = \{Q_n : n \in \mathbb{N}\}$, where each circuit Q_n takes n input qubits, and produces one output qubit. The computation on a given input string $x \in \Sigma^*$ is obtained by first applying the circuit $Q_{|x|}$ to the state $|x\rangle\langle x|$, and then measuring the output qubit with respect to the standard basis. The measurement results 0 and 1 are interpreted as *yes* and *no* (or *accept* and *reject*), respectively. The events that Q accepts x and Q rejects x are understood to have associated probabilities determined in this way.

BQP Let $A = (A_{\text{yes}}, A_{\text{no}})$ be a promise problem and let $a, b : \mathbb{N} \rightarrow [0,1]$ be functions. Then $A \in \text{BQP}(a, b)$ if and only if there exists a polynomial-time generated family of quantum circuits $Q = \{Q_n : n \in \mathbb{N}\}$, where each circuit Q_n takes n input qubits and produces one output qubit, that satisfies the following properties:

1. if $x \in A_{\text{yes}}$ then $\Pr[Q \text{ accepts } x] \geq a(|x|)$, and
2. if $x \in A_{\text{no}}$ then $\Pr[Q \text{ accepts } x] \leq b(|x|)$.

The class BQP is defined as $\text{BQP} = \text{BQP}(2/3, 1/3)$.

Similar to BPP, there is nothing special about the particular choice of error probability 1/3, other than that it is a constant strictly smaller than 1/2. This is made clear in the next section.

There are several problems known to be in BQP but not known (and generally not believed) to be in BPP. Decision-problem variants of the integer factoring and discrete logarithm problems, shown to be in BQP by Shor [94], are at present the most important and well-known examples.

IV.2 Error reduction for BQP

When one speaks of the flexibility, or *robustness*, of BQP with respect to error bounds, it is meant that the class $\text{BQP}(a, b)$ is invariant under a wide range of "reasonable" choices of the functions a and b. The following proposition states this more precisely.

Proposition 3 (Error reduction for BQP)**.** *Suppose that $a, b : \mathbb{N} \rightarrow [0,1]$ are polynomial-time computable functions and $p : \mathbb{N} \rightarrow \mathbb{N}$ is a polynomial-bounded function such that $a(n) - b(n) \geq 1/p(n)$ for all but finitely many $n \in \mathbb{N}$. Then for every choice of a polynomial-bounded function $q : \mathbb{N} \rightarrow \mathbb{N}$ satisfying $q(n) \geq 2$ for all but finitely many $n \in \mathbb{N}$, it holds that*

$$\text{BQP}(a, b) = \text{BQP} = \text{BQP}\left(1 - 2^{-q}, 2^{-q}\right).$$

The above proposition may be proved in the same standard way that similar statements are proved for classical probabilistic computations: by repeating a given computation some large (but still polynomial) number of times, overwhelming statistical evidence is obtained so as to give the correct answer with an extremely small probability of error. It is straightforward to represent this sort of repeated computation within the quantum circuit model in such a way that the requirements of the definition of BQP are satisfied.

IV.3 Simulating classical computations with quantum circuits

It should not be surprising that quantum computers can efficiently simulate classical computers—for quantum information generalizes classical information, and it would be absurd if there were a loss of computational power in moving to a more general model. This intuition may be confirmed by observing the containment $\text{BPP} \subseteq \text{BQP}$. Here an advantage of working with the general quantum circuit model arises: for if one truly believes the Universality Theorem, there is almost nothing to prove.

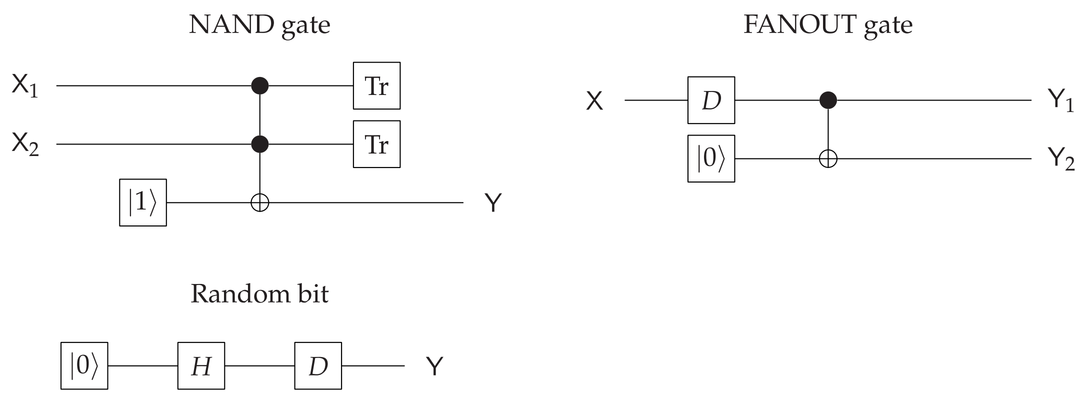


Figure 7: Quantum circuit implementations of a NAND gate, a FANOUT gate, and a random bit. The phase-damping gates, denoted by D, are only included for aesthetic reasons: they force the purely classical behavior that would be expected of classical gates, but are not required for the quantum simulation of BPP.

Observe first that the complexity class P may be defined in terms of Boolean circuit families in a similar manner to BQP. In particular, a given promise problem $A = (A_{\text{yes}}, A_{\text{no}})$ is in P if and only if there exists a polynomial-time generated family $C = \{C_n : n \in \mathbb{N}\}$ of Boolean circuits, where each circuit C_n takes n input bits and outputs 1 bit, such that

1. $C(x) = 1$ for all $x \in A_{\text{yes}}$, and
2. $C(x) = 0$ for all $x \in A_{\text{no}}$.

A Boolean circuit-based definition of BPP may be given along similar lines: a given promise problem $A = (A_{\text{yes}}, A_{\text{no}})$ is in BPP if and only if there exists a polynomial-bounded function r and a polynomial-time generated family $C = \{C_n : n \in \mathbb{N}\}$ of Boolean circuits, where each circuit C_n takes $n + r(n)$ input bits and outputs 1 bit, such that

1. $\Pr[C(x,y) = 1] \geq 2/3$ for all $x \in A_{\text{yes}}$, and
2. $\Pr[C(x,y) = 1] \leq 1/3$ for all $x \in A_{\text{no}}$,

where $y \in \Sigma^{r(|x|)}$ is chosen uniformly at random in both cases.

In both definitions, the circuit family C includes circuits composed of constant-size Boolean logic gates—which for the sake of brevity may be assumed to be composed of NAND gates and FANOUT gates. (FANOUT operations must be modelled as gates for the sake of the simulation.) For the randomized case, it may be viewed that the random bits $y \in \Sigma^{r(|x|)}$ are produced by gates that take no input and output a single uniform random bit. As NAND gates, FANOUT gates, and random bits are easily implemented with quantum gates, as illustrated in Figure 7, the circuit family C can be simulated gate-by-gate to obtain a quantum circuit family $Q = \{Q_n : n \in \mathbb{N}\}$ for A that satisfies the definition of BQP. It follows that BPP $\subseteq$ BQP.

IV.4 The BQP subroutine theorem

There is an important issue regarding the above definition of BQP, which is that it is not an inherently "clean" definition with respect to the modularization of algorithms. The *BQP subroutine theorem* of Bennett, Brassard, Bernstein and Vazirani [28] addresses this issue.

Suppose that it is established that a particular promise problem 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 one would like to be able to do this without worrying about the specifics of the original algorithm. Ideally, the algorithm for A should function as an oracle for A, as defined in Section III.4.

A problem arises, however, when queries to an algorithm for A are made in superposition. Whereas it is quite common and useful to consider quantum algorithms that query oracles 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. Perhaps it is too much to ask that the existence of a BQP algorithm for A admits a subroutine having the characteristics of an oracle for A?

The BQP subroutine theorem establishes that, up to exponentially small error, this is not too much to ask: the existence of an arbitrary BQP algorithm for A implies the existence of a "clean" subroutine for A with the characteristics of an oracle. A precise statement of the theorem follows.

Theorem 4 (BQP subroutine theorem)**.** *Suppose $A = (A_{yes}, A_{no})$ is a promise problem in* BQP. *Then for any choice of a polynomial-bounded function p there exists a polynomial-bounded function q and a polynomial-time generated family of unitary quantum circuits $\{R_n : n \in \mathbb{N}\}$ with the following properties:*

1. *Each circuit R_n implements a unitary operation U_n on $n + q(n) + 1$ qubits.*
2. *For every $x \in A_{yes}$ and $a \in \Sigma$ it holds that*
$$\langle x, 0^m, a \oplus 1 | U_n | x, 0^m, a \rangle \geq 1 - 2^{-p(n)}$$
for $n = |x|$ and $m = q(n)$.
3. *For every $x \in A_{no}$ and $a \in \Sigma$ it holds that*
$$\langle x, 0^m, a | U_n | x, 0^m, a \rangle \geq 1 - 2^{-p(n)}$$
for $n = |x|$ and $m = q(n)$.

The proof of this theorem is remarkably simple: given a BQP algorithm for A, one first uses Proposition 3 to obtain a circuit family Q having exponentially small error for A. The circuit illustrated in Figure 8 then implements a unitary operation with the desired properties. This is essentially a bounded-error quantum adaptation of a classic construction that allows arbitrary deterministic computations to be performed reversibly [27, 97].

The following corollary expresses the main implication of the BQP subroutine theorem in complexity-theoretic terms.

Corollary 5. $\text{BQP}^{\text{BQP}} = \text{BQP}$.

IV.5 Classical upper bounds on BQP

There is no known way to efficiently simulate quantum computers with classical computers—and there would be little point in seeking to build quantum computers if there were. Nevertheless,

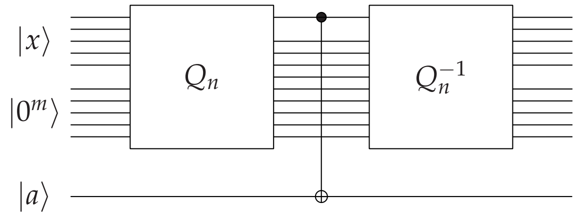


Figure 8: A unitary quantum circuit approximating an oracle-gate implementation of a BQP computation. Here Q_n is a unitary purification of a circuit having exponentially small error for some problem in BQP.

some insight into the limitations of quantum computers may be gained by establishing containments of BQP in the smallest classical complexity classes where this is possible.

The strongest containment known at this time is given by *counting complexity*. Counting complexity began with Valiant's work [99] on the complexity of computing the permanent, and was further developed and applied in many papers (including [22, 48, 96], among many others).

The basic notion of counting complexity that is relevant to this article is as follows. Given a polynomial-time nondeterministic Turing machine M and input string $x \in \Sigma^*$, one denotes by $\#M(x)$ the number of *accepting* computation paths of M on x, and by $\#\overline{M}(x)$ the number of *rejecting* computation paths of M on x. A function $f : \Sigma^* \rightarrow \mathbb{Z}$ is then said to be a *GapP function* if there exists a polynomial-time nondeterministic Turing machine M such that $f(x) = \#M(x) - \#\overline{M}(x)$ for all $x \in \Sigma^*$.

A variety of complexity classes can be specified in terms of GapP functions. For example, a promise problem $A = (A_{\text{yes}}, A_{\text{no}})$ is in PP if and only if there exists a function $f \in$ GapP such that $f(x) > 0$ for all $x \in A_{\text{yes}}$ and $f(x) \leq 0$ for all $x \in A_{\text{no}}$. The remarkable closure properties of GapP functions allows many interesting facts to be proved about such classes. Fortnow's survey [51] on counting complexity explains many of these properties and gives several applications of the theory of GapP functions. The following closure property is used below.

GapP–multiplication of matrices. Let $p, q : \mathbb{N} \rightarrow \mathbb{N}$ be polynomial-bounded functions. Suppose that for each $n \in \mathbb{N}$ a sequence of $p(n)$ complex-valued matrices is given:

$$M_{n,1},\ M_{n,2},\ \ldots,\ M_{n,p(n)},$$

each having rows and columns indexed by strings in $\Sigma^{q(n)}$. Suppose further that there exist functions $f, g \in$ GapP such that

$$f(1^n, 1^k, x, y) = \operatorname{Re}\left(M_{n,k}[x, y]\right)$$
$$g(1^n, 1^k, x, y) = \operatorname{Im}\left(M_{n,k}[x, y]\right)$$

for all $n \in \mathbb{N}$, $k \in \{1, \ldots, p(n)\}$, and $x, y \in \Sigma^{q(n)}$. Then there exist functions $F, G \in$ GapP such that

$$F(1^n, x, y) = \operatorname{Re}\Big((M_{n,p(n)} \cdots M_{n,1})[x, y]\Big)$$
$$G(1^n, x, y) = \operatorname{Im}\Big((M_{n,p(n)} \cdots M_{n,1})[x, y]\Big)$$

for all $n \in \mathbb{N}$ and $x, y \in \Sigma^{q(n)}$. In other words, if there exist two GapP functions describing the real and imaginary parts of the entries of a polynomial sequence of matrices, then the same is true of the product of these matrices.

Now, suppose that $Q = \{Q_n : n \in \mathbb{N}\}$ is a polynomial-time generated family of quantum circuits. For each $n \in \mathbb{N}$, let us assume the quantum circuit Q_n consists of gates $G_{n,1}, \ldots, G_{n,p(n)}$ for some polynomial bounded function p, labelled in an order that respects the topology of the circuit. By tensoring these gates with the identity operation on qubits they do not touch, and using the natural matrix representation of quantum operations, it is possible to obtain matrices $M_{n,1}, \ldots, M_{n,p(n)}$ with the property that

$$M_{n,p(n)} \cdots M_{n,1} \operatorname{vec}(\rho) = \operatorname{vec}(Q_n(\rho))$$

for every possible input density matrix ρ to the circuit. The probability that Q_n accepts a given input x is then

$$\langle 1,1| \left(M_{n,p(n)} \cdots M_{n,1} \right) |x, x\rangle ,$$

which is a single entry in the product of the matrices.

By padding matrices with rows and columns of zeroes, it may be assumed that each matrix $M_{n,k}$ has rows and columns indexed by strings of length $q(n)$ for some polynomial-bounded function q. The assumption that the family Q is polynomial-time generated then allows one to easily conclude that there exist GapP functions f and g so that

$$f(1^n, 1^k, x, y) = \frac{1}{2} \operatorname{Re}\left(M_{n,k}[x, y]\right)$$
$$g(1^n, 1^k, x, y) = \frac{1}{2} \operatorname{Im}\left(M_{n,k}[x, y]\right)$$

for all $n \in \mathbb{N}$, $k \in \{1, \ldots, p(n)\}$, and $x, y \in \Sigma^{q(n)}$. (Note that his fact makes use of the observation that the natural matrix representation of all of the gates listed in Section III.2 have entries whose real and imaginary parts come from the set $\{-1, -1/2, 0, 1/2, 1\}$. The numbers $\pm 1/2$ are only needed for the Hadamard gates.) By the property of GapP functions above, it follows that there exists a GapP function F such that

$$\Pr[Q \text{ accepts } x] = \frac{F(x)}{2^{p(|x|)}}.$$

The containment BQP $\subseteq$ PP follows easily. This containment was first proved by Adleman, DeMarrais, and Huang [8] using a different method, and was first argued using counting complexity along similar lines to the above proof by Fortnow and Rogers [52]. There are two known ways that this upper bound can be improved. Using the fact that BQP algorithms have bounded error, Fortnow and Rogers [52] proved that BQP is contained in a (somewhat obscure) counting complexity class called AWPP, which implies the following theorem.

Theorem 6. $\text{PP}^{\text{BQP}} = \text{PP}$.

Another improvement comes from the observation that the above proof that BQP $\subseteq$ PP makes no use of the bounded-error assumption of BQP. It follows that an unbounded error variant of BQP is equal to PP.

PQP Let $A = (A_{\text{yes}}, A_{\text{no}})$ be a promise problem. Then $A \in$ PQP if and only if there exists a polynomial-time generated family of quantum circuits $Q = \{Q_n : n \in \mathbb{N}\}$, where each circuit Q_n takes n input qubits and produces one output qubit, that satisfies the following properties. If $x \in A_{\text{yes}}$ then $\Pr[Q \text{ accepts } x] > 1/2$; and if $x \in A_{\text{no}}$ then $\Pr[Q \text{ accepts } x] \leq 1/2$.

Theorem 7. PQP = PP.

IV.6 Oracle results involving BQP

It is difficult to prove separations among quantum complexity classes for apparently the same reasons that this is so for classical complexity classes. For instance, one clearly cannot hope to prove BPP $\neq$ BQP when major collapses such as NC $=$ PP or P $=$ PSPACE are still not disproved. In some cases, however, separations among quantum complexity classes can be proved in relativized settings, meaning that the separation holds in the presence of some cleverly defined oracle. The following oracle results are among several that are known:

1. There exists an oracle A such that BQP$^A \not\subseteq$ MA^A [18, 31, 101]. Such an oracle A intuitively encodes a problem that is solvable using a quantum computer but is not even efficiently verifiable with a classical computer. As the containment BPP $\subseteq$ BQP holds relative to every oracle, this implies that BPP$^A \subsetneq$ BQP^A for this particular choice of A.

2. There is an oracle A such that NP$^A \not\subseteq$ BQP^A [28]. In less formal terms: a quantum computer cannot find a needle in an exponentially large haystack in polynomial time. This result formalizes a critically important idea, which is that a quantum computer can only solve a given search problem efficiently if it is able to exploit that problem's structure. It is easy to define an oracle search problem represented by A that has no structure whatsoever, which allows the conclusion NP$^A \not\subseteq$ BQP^A to be drawn. It is not currently known whether the NP-complete problems, in the absence of an oracle, have enough structure to be solved efficiently by quantum computers; but there is little hope and no indication whatsoever that this should be so. It is therefore a widely believed conjecture that NP $\not\subseteq$ BQP.

3. There is an oracle A such that SZK$^A \not\subseteq$ BQP^A [1, 6]. This result is similar in spirit to the previous one, but is technically more difficult and rules out the existence of quantum algorithms for unstructured collision detection problems. The graph isomorphism problem and various problems that arise in cryptography are examples of collision detection problems. Once again, it follows that quantum algorithms can only solve such problems if their structure can be exploited. It is a major open question in quantum computing whether the graph isomorphism problem is in BQP.

V Quantum proofs

There are many quantum complexity classes of interest beyond BQP. This section concerns one such class, which is a quantum computational analogue of NP. The class is known as QMA, short for *quantum Merlin–Arthur*, and is based on the notion of a *quantum proof*: a quantum state that plays the role of a certificate or witness to a quantum computer that functions as a verification procedure. Interest in both the class QMA and the general notion of quantum proofs is primarily based on the fundamental importance of *efficient verification* in computational complexity. The notion of a quantum proof was first proposed by Knill [71] and studied more formally by Kitaev (presented at a talk in 1999 [67] and later published in [68]).

V.1 Definition of QMA

The definition of QMA is inspired by the standard definition of NP included in Section II.1 of this article. This definition is of course equivalent to the other well-known definition of NP based on nondeterministic Turing machines, but is much better-suited to consideration in the quantum setting—for nondeterminism is arguably a non-physical notion that does not naturally extend to quantum computing. In the definition of NP from Section II.1, the machine M functions as a *verification procedure* that treats each possible string $y \in \Sigma^{p(|x|)}$ as a potential *proof* that $x \in A_{\text{yes}}$. The conditions on M are known as the *completeness* and *soundness* conditions, which derive their names from logic: completeness refers to the condition that true statements have proofs, while soundness refers to the condition that false statements do not.

To define QMA, the set of possible proofs is extended to include quantum states, which of course presumes that the verification procedure is quantum. As quantum computations are inherently probabilistic, a bounded probability of error is allowed in the completeness and soundness conditions. (This is why the class is called QMA rather than QNP, as it is really MA and not NP that is the classical analogue of QMA.)

QMA Let $A = (A_{\text{yes}}, A_{\text{no}})$ be a promise problem, let p be a polynomial-bounded function, and let $a, b : \mathbb{N} \rightarrow [0, 1]$ be functions. Then $A \in \text{QMA}_p(a, b)$ if and only if there exists a polynomial-time generated family of circuits $Q = \{Q_n : n \in \mathbb{N}\}$, where each circuit Q_n takes $n + p(n)$ input qubits and produces one output qubit, with the following properties:

1. *Completeness.* For all $x \in A_{\text{yes}}$, there exists a $p(|x|)$-qubit quantum state ρ such that $\Pr[Q \text{ accepts } (x, \rho)] \geq a(|x|)$.
2. *Soundness.* For all $x \in A_{\text{no}}$ and all $p(|x|)$-qubit quantum states ρ it holds that $\Pr[Q \text{ accepts } (x, \rho)] \leq b(|x|)$.

Also define $\text{QMA} = \bigcup_p \text{QMA}_p(2/3, 1/3)$, where the union is over all polynomial-bounded functions p.

V.2 Problems in QMA

Before discussing the general properties of QMA that are known, it is appropriate to mention some examples of problems in this class. Of course it is clear that thousands of interesting combinatorial problems are in QMA, as the containment $\text{NP} \subseteq \text{QMA}$ is trivial. What is more interesting is the identification of problems in QMA that are not known to be in NP, for these are examples that provide insight into the power of quantum proofs. There are presently just a handful of known problems in QMA that are not known to be in NP, but the list is growing.

The local Hamiltonian problem

Historically speaking, the first problem identified to be complete for QMA was the *local Hamiltonian problem* [67, 68], which can be seen as a quantum analogue of the MAX-k-SAT problem. Its proof of completeness can roughly be seen as a quantum analogue of the proof of the Cook–Levin Theorem [40, 75]. The proof has subsequently been strengthened to achieve better parameters [62].

Suppose that M is a Hermitian matrix whose rows and columns are indexed by strings of length n for some integer $n \geq 1$. Then M is said to be *k-local* if and only if it can be expressed as

$$M = P_\pi (A \otimes I) P_\pi^{-1}$$

for an arbitrary matrix A indexed by Σ^k, P_π a permutation matrix defined by

$$P_\pi |x_1 \cdots x_n\rangle = |x_{\pi(1)} \cdots x_{\pi(n)}\rangle$$

for some permutation $\pi \in S_n$, and I denoting the identity matrix indexed by Σ^{n-k}. In less formal terms, M is a matrix that arises from a "gate" on k qubits, but where the gate is described by a $2^k \times 2^k$ Hermitian matrix A rather than a unitary matrix. It is possible to express such a matrix compactly by specifying A along with the bit-positions on which A acts.

Intuitively, a k-local matrix assigns a real number (typically thought of as representing *energy*) to any quantum state on n qubits. This number depends only on the reduced state of the k qubits where M acts nontrivially, and can be thought of as a locally defined penalty on a given quantum state. Loosely speaking, the k-local Hamiltonian problem asks whether there exists a quantum state that can significantly avoid a collection of such penalties.

THE k-LOCAL HAMILTONIAN PROBLEM

Input: A collection $H_1, \ldots, H_m$ of k-local Hermitian matrices with entries indexed by strings of length n and satisfying $\|H_j\| \leq 1$ for $j = 1, \ldots, m$.

Yes: There exists an n-qubit quantum state $|\psi\rangle$ such that $\langle\psi|H_1 + \cdots + H_m|\psi\rangle \leq -1$.

No: For every n-qubit quantum state $|\psi\rangle$ it holds that $\langle\psi|H_1 + \cdots + H_m|\psi\rangle \geq 1$.

Theorem 8. *The 2-local Hamiltonian problem is complete for* QMA *with respect to Karp reductions.*

The completeness of this problem has been shown to be closely related to the universality of the so-called *adiabatic* model of quantum computation [12].

Other problems complete for QMA

Aside from the local Hamiltonian problem, there are other promise problems known to be complete for QMA, including the following:

1. Several Variants of the local Hamiltonian problem. For example, the 2-local Hamiltonian problem remains QMA-complete when the local Hamiltonians are restricted to nearest-neighbor interactions on a two-dimensional array of qubits [86]. The hardness of the local Hamiltonian problem with nearest-neighbor interactions on one-dimensional systems is known to be QMA-complete for 12 dimensional particles in place of qubits [9], but is open for smaller systems including qubits.

2. The *density matrix consistency problem*. Here, the input is a collection of density matrices representing the reduced states of a hypothetical n-qubit state. The input is a yes-instance of the problem if there exists a state of the n-qubit system that is consistent with the given reduced density matrices, and is a no-instance if every state of the n-qubit system has reduced states that have significant disagreement from one or more of the input density matrices. This problem is known to be complete for QMA with respect to Cook reductions [76], but is not known to be complete for Karp reductions. A different problem with similar properties was considered in [77].

3. The *quantum clique problem*. Beigi and Shor [23] have proved that a promise version of the following problem is QMA-complete for any constant $k \geq 2$. Given a quantum operation Φ, are there k different inputs to Φ that are perfectly distinguishable after passing through Φ? They name this the quantum clique problem because there is a close connection between computing the zero-error capacity of a classical channel and computing the size of the largest clique in the complement of a graph representing the channel; and this is a quantum variant of that problem.

4. Several problems about quantum circuits. All of the problems mentioned above are proved to be QMA-hard through reductions from the local Hamiltonian problem. Other problems concerning properties of quantum circuits can be proved QMA-complete by more direct means, particularly when the problem itself concerns the existence of a quantum state that causes an input circuit to exhibit a certain behavior. An example in this category is the *non-identity check* problem [60] that asks whether a given unitary circuit implements an operation that is close to some scalar multiple of the identity.

The group non-membership problem

Another example of a problem in QMA that is not known to be in NP is the *group non-membership problem* [17, 101]. This problem is quite different from the QMA-complete problems mentioned above in two main respects. First, the problem corresponds to a language, meaning that the promise is vacuous: every possible input can be classified as a yes-instance or no-instance of the problem. (In all of the above problems it is not possible to do this without invalidating the known proofs that the problems are in QMA.) Second, the problem is not known to be complete for QMA; and indeed it would be surprising if QMA were shown to have a complete problem having a vacuous promise.

In the group non-membership problem the input is a subgroup H of some finite group G, along with an element $g \in G$. The yes-instances of the problem are those for which $g \notin H$, while the no-instances are those for which $g \in H$.

THE GROUP NON-MEMBERSHIP PROBLEM

Input: Group elements $h_1, \ldots, h_k$ and g from some finite group G. Let $H = \langle h_1, \ldots, h_k \rangle$ be the subgroup generated by $h_1, \ldots, h_k$.

Yes: $g \notin H$.

No: $g \in H$.

Like most group-theoretic computational problems, there are many specific variants of the group non-membership problem that differ in the way that group elements are represented. For example, group elements could be represented by permutations in cycle notation, invertible matrices over a finite field, or any number of other possibilities. The difficulty of the problem clearly varies depending the representation. The framework of *black-box groups*, put forth by Babai and Szemerédi [21], simplifies this issue by assuming that group elements are represented by meaningless labels that can only be multiplied or inverted by means of a black box, or *group oracle*. Any algorithm or protocol that solves a problem in this framework can then be adapted to any specific group provided that an efficient implementation of the group operations exists that can replace the group oracle.

Theorem 9. *The group non-membership problem is in* QMA *for every choice of a group oracle.*

V.3 Error reduction for QMA

The class QMA is robust with respect to error bounds, reflected by the completeness and soundness probabilities, in much the same way that the same is true of BQP. Two specific error reduction methods are presented in this section: *weak error reduction* and *strong error reduction*. The two methods differ with respect to the length of the quantum proof that is needed to obtain a given error bound, which is a unique consideration that arises when working with quantum proofs.

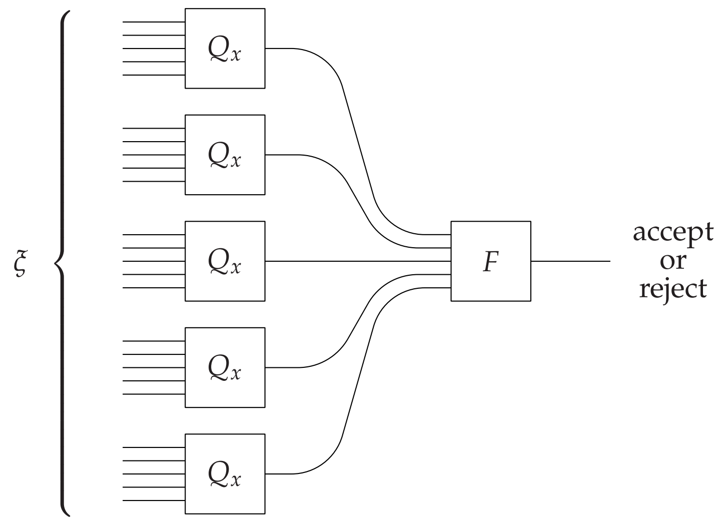


Figure 9: An illustration of the weak error reduction procedure for QMA. The circuits Q_x represent the original verification procedure, and the circuit labelled F outputs acceptance or rejection based on the frequency of accepts among its inputs (which can be adjusted depending on a and b). It cannot be assumed that the input state ξ takes the form of a product state $\rho \otimes \cdots \otimes \rho$; but it is not difficult to prove that there will always be at least one such state among the states maximizing the acceptance probability of the procedure.

Suppose that $A = (A_{\text{yes}}, A_{\text{no}})$ is a given promise problem for which a QMA-type quantum verification procedure exists. To describe both error reduction procedures, it is convenient to assume that the input to the problem is hard-coded into each circuit in the family representing the verification procedure. The verification procedure therefore takes the form $Q = \{Q_x : x \in \Sigma^*\}$, where each circuit Q_x takes a $p(|x|)$-qubit quantum state as input, for some polynomial-bounded function p representing the quantum proof length, and produces one output qubit. The completeness and soundness probabilities are assumed to be given by polynomial-time computable functions a and b as usual.

The natural approach to error reduction is repetition: for a given input x the verification circuit Q_x is evaluated multiple times, and a decision to accept or reject based on the frequency of accepts and rejects among the outcomes of the repetitions is made. Assuming $a(|x|)$ and $b(|x|)$ are not too close together, this results in a decrease in error that is exponential in the number of repetitions [68]. The problem that arises, however, is that each repetition of the verification procedure apparently destroys the quantum proof it verifies, which necessitates the composite verification procedure receiving $p(|x|)$ qubits for *each* repetition of the original procedure as illustrated in Figure 9. The form of error reduction implemented by this procedure is called *weak error reduction*, given that the length of the quantum proof must grow as the error decreases. (Of course one may view that it also has a strength, which is that it does not require a significant increase in circuit depth over the original procedure.)

The second error reduction procedure for QMA was described in [80]. Like weak error reduction it gives an exponential reduction in error for roughly a linear increase in circuit size, but has the advantage that it does not require any increase in the length of the quantum proof. This form of error reduction is called *strong error reduction* because of this advantage, which turns out to be quite handy in some situations. The procedure is illustrated in Figure 10. The following theorem follows from an analysis of this procedure.

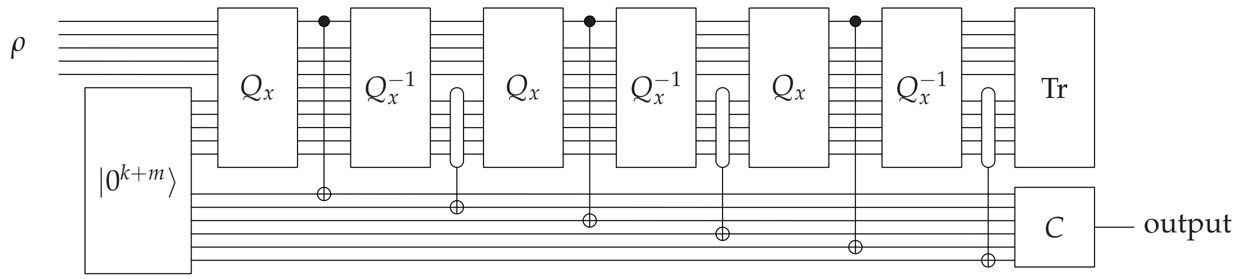


Figure 10: Illustration of the strong error reduction procedure for QMA. The circuit Q_x represents a unitary purification of a given QMA verification procedure on an input x, while the circuit C determines whether to accept or reject based on the number of alternations of its input qubits. The quantum proof is denoted by ρ.

Theorem 10. *Suppose that $a, b : \mathbb{N} \rightarrow [0,1]$ are polynomial-time computable functions and $q : \mathbb{N} \rightarrow \mathbb{N}$ is a polynomial-bounded function such that $a(n) - b(n) \geq 1/q(n)$ for all but finitely many $n \in \mathbb{N}$. Then for every choice of polynomial-bounded functions $p, r : \mathbb{N} \rightarrow \mathbb{N}$ such that $r(n) \geq 2$ for all but finitely many n, it holds that*

$$\mathrm{QMA}_p(a,b) = \mathrm{QMA}_p\left(1 - 2^{-r}, 2^{-r}\right).$$

V.4 Containment of QMA in PP

By means of strong error reduction for QMA, it can be proved that QMA is contained in PP as follows. Suppose that some promise problem $A = (A_{\text{yes}}, A_{\text{no}})$ is contained in QMA. Then by Theorem 10 it holds that

$$A \in \mathrm{QMA}_p\left(1 - 2^{-(p+2)}, 2^{-(p+2)}\right) \tag{2}$$

for some polynomial-bounded function p. What is important here is that the soundness probability is smaller than the reciprocal of the dimension of the space corresponding to the quantum proof (which is why strong error reduction is needed).

Now, consider an algorithm that does not receive any quantum proof, but instead just randomly guesses a quantum proof on p qubits and feeds this proof into a verification procedure having completeness and soundness probabilities consistent with the above inclusion (2). To be more precise, the quantum proof is substituted by the totally mixed state on p qubits. A simple analysis shows that this algorithm accepts every string $x \in A_{\text{yes}}$ with probability at least $2^{-(p(|x|)+1)}$ and accepts every string $x \in A_{\text{no}}$ with probability at most $2^{-(p(|x|)+2)}$. Both of these probabilities are exponentially small, but the separation between them is enough to establish that $A \in \mathrm{PQP} = \mathrm{PP}$.

V.5 Classical proofs for quantum verification procedures

One may also consider quantum verification procedures that receive classical proofs rather than quantum proofs. Aharonov and Naveh [11] defined a complexity class MQA accordingly.

MQA Let $A = (A_{\text{yes}}, A_{\text{no}})$ be a promise problem. Then $A \in$ MQA if and only if there exists a polynomial-bounded function p and a polynomial-time generated family of circuits $Q = \{Q_n : n \in \mathbb{N}\}$, where each circuit Q_n takes $n + p(n)$ input qubits and produces one output qubit, with the following properties. For all $x \in A_{\text{yes}}$, there exists a string $y \in \Sigma^{p(|x|)}$ such that $\Pr[Q \text{ accepts } (x, y)] \geq 2/3$; and for all $x \in A_{\text{no}}$ and all strings $y \in \Sigma^{p(|x|)}$ it holds that $\Pr[Q \text{ accepts } (x, y)] \leq 1/3$.

(This class was originally named QCMA, and is more commonly known by that name—but it is not too late to give this interesting class a better name.)

When considering the power of quantum proofs, it is the question of whether MQA is properly contained in QMA that arguably cuts to the heart of the issue. Aaronson and Kuperberg [5] studied this question, and based on a reasonable group-theoretic conjecture argued that the group non-membership problem is likely to be in MQA.

V.6 Are two quantum proofs better than one?

An unusual, yet intriguing question about quantum proofs was asked by Kobayashi, Matsumoto, and Yamakami [74]: *are two quantum proofs better than one?* The implicit setting in this question is similar to that of QMA, except that the verification procedure receives *two* quantum proofs that are *guaranteed to be unentangled*. The complexity class QMA(2) is defined to be the class of promise problems having such two-proof systems with completeness and soundness probabilities 2/3 and 1/3, respectively. (It is not known to what extent this type of proof system is robust with respect to error bounds, so it is conceivable that other reasonable choices of error bounds could give distinct complexity classes.) The class QMA(2), and its natural extension to more than two proofs, have subsequently been studied in [4, 32].

The restriction on entanglement that is present in the definition of QMA(2) may seem artificial from a physical perspective, as there is no obvious mechanism that would prevent two entities with otherwise unlimited computational power from sharing entanglement. Nevertheless, the question of the power of QMA(2) is interesting in that it turns around the familiar question in quantum computing about the computational power of entanglement. Rather than asking if computational power is limited by a constraint on entanglement, here the question is whether a constraint on entanglement *enhances* computational power. It is clear that two quantum proofs are no less powerful than one, as a verification procedure may simply ignore one of the proofs—which means that QMA $\subseteq$ QMA(2). Good upper bounds on QMA(2), however, are not known: the best upper bound presently known is QMA(2) $\subseteq$ NEXP, which follows easily from the fact that a nondeterministic exponential-time algorithm can guess explicit descriptions of the quantum proofs and then perform an exponential-time simulation of the verification procedure.

VI Quantum interactive proof systems

The notion of efficient proof verification is generalized by means of the *interactive proof system* model, which has fundamental importance in complexity theory and theoretical cryptography. Interactive proof systems, or *interactive proofs* for short, were first introduced by Babai [17, 20] and Goldwasser, Micali, and Rackoff [55, 56], and have led to remarkable discoveries in complexity such as the *PCP Theorem* [15, 16, 43]. In the most commonly studied variant of interactive proof systems, a polynomial-time *verifier* interacts with a computationally unbounded *prover* that tries to convince the verifier of the truth of some statement. The prover is not trustworthy, however,

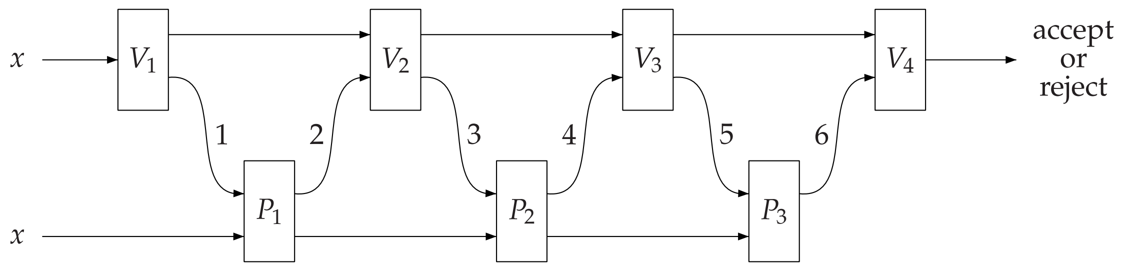


Figure 11: A quantum interactive proof system. There are six messages in this example, labelled $1, \ldots, 6$. (There may be polynomially many messages in general.) The arrows each represent a collection of qubits, rather than single qubits as in previous figures. The superscript n is omitted in the names of the prover and verifier circuits (which can safely be done when the input length n is determined by context).

and so the verifier must be specified in such a way that it is not convinced that false statements are true.

Quantum interactive proof systems [104, 69] are interactive proof systems in which the prover and verifier may exchange and process quantum information. This ability of both the prover and verifier to exchange and process quantum information endows quantum interactive proofs with interesting properties that distinguish them from classical interactive proofs, and illustrate unique features of quantum information.

VI.1 Definition of quantum interactive proofs and the class QIP

A quantum interactive proof system involves an interaction between a prover and verifier as suggested by Figure 11.

The verifier is described by a polynomial-time generated family

$$V = \left\{ V_j^n \,:\, n \in \mathbb{N},\ j \in \{1, \ldots, p(n)\} \right\}$$

of quantum circuits, for some polynomial-bounded function p. On an input string x having length n, the sequence of circuits $V_1^n, \ldots, V_{p(n)}^n$ determine the actions of the verifier over the course of the interaction, with the value $p(n)$ representing the number of *turns* the verifier takes. For instance, $p(n) = 4$ in the example depicted in Figure 11. The inputs and outputs of the verifier's circuits are divided into two categories: private *memory* qubits and *message* qubits. The message qubits are sent to, or received from, the prover, while the memory qubits are retained by the verifier as illustrated in the figure. It is always assumed that the verifier *receives* the last message; so if the number of messages is to be $m = m(n)$, then it must be the case that $p(n) = \lfloor m/2 \rfloor + 1$.

The prover is defined in a similar way to the verifier, although no computational assumptions are made: the prover is a family of arbitrary quantum operations

$$P = \left\{ P_j^n \,:\, n \in \mathbb{N},\ j \in \{1, \ldots, q(n)\} \right\}$$

that interface with a given verifier in the natural way (assuming the number and sizes of the message spaces are compatible). Again, this is as suggested by Figure 11.

Now, on a given input string x, the prover P and verifier V have an interaction by composing their circuits as described above, after which the verifier measures an output qubit to determine acceptance or rejection. In direct analogy to the classes NP, MA, and QMA, one defines quantum complexity classes based on the completeness and soundness properties of such interactions.

QIP Let $A = (A_{\text{yes}}, A_{\text{no}})$ be a promise problem, let m be a polynomial-bounded function, and let $a, b : \mathbb{N} \to [0,1]$ be polynomial-time computable functions. Then $A \in \text{QIP}(m, a, b)$ if and only if there exists an m-message quantum verifier V with the following properties:

1. *Completeness.* For all $x \in A_{\text{yes}}$, there exists a quantum prover P that causes V to accept x with probability at least $a(|x|)$.

2. *Soundness.* For all $x \in A_{\text{no}}$, every quantum prover P causes V to accept x with probability at most $b(|x|)$.

Also define $\text{QIP}(m) = \text{QIP}(m, 2/3, 1/3)$ for each polynomial-bounded function m and define $\text{QIP} = \bigcup_m \text{QIP}(m)$, where the union is over all polynomial-bounded functions m.

VI.2 Properties of quantum interactive proofs

Classical interactive proofs can trivially be simulated by quantum interactive proofs, and so the containment PSPACE $\subseteq$ QIP follows directly from IP = PSPACE [79, 92]. Unlike classical interactive proofs, quantum interactive proofs are not known to be simulatable in PSPACE. Simulation is possible in EXP through the use of semidefinite programming [69].

Theorem 11. QIP $\subseteq$ EXP.

Quantum interactive proofs, like ordinary classical interactive proofs, are quite robust with respect to the choice of completeness and soundness probabilities. In particular, the following facts hold [69, 58].

1. Every quantum interactive proof can be transformed into an equivalent quantum interactive proof with *perfect completeness*: the completeness probability is 1 and the soundness probability is bounded away from 1. The precise bound obtained for the soundness probability depends on the completeness and soundness probabilities of the original proof system. This transformation to a perfect-completeness proof system comes at the cost of one additional round (i.e., two messages) of communication.

2. Parallel repetition of quantum interactive proofs with perfect completeness gives an exponential reduction in soundness error. An exponential reduction in completeness and soundness error is also possible for quantum interactive proofs not having perfect completeness, but the procedure is slightly more complicated than for the perfect completeness case.

One of the major differences between quantum and classical interactive proof systems is that quantum interactive proofs can be *parallelized*: every quantum interactive proof system, possibly having polynomially many rounds of communication, can be transformed into an equivalent quantum interactive proof system with *three messages* [69]. This transformation comes at the cost of weakening the error bounds. However, it preserves perfect completeness, and the soundness error can subsequently be reduced by parallel repetition without increasing the number of messages beyond three. If classical interactive proof systems could be parallelized in this way, then it would follow that AM = PSPACE; a collapse that would surprise most complexity theorists and have major implications to the theory.

The following theorem summarizes the implications of the facts just discussed to the complexity classes $\mathrm{QIP}(m, a, b)$ defined above.

Theorem 12. *Let $a, b : \mathbb{N} \to [0, 1]$ be polynomial-time computable functions and let p be a polynomial-bounded function such that $a(n) - b(n) \geq \frac{1}{p(n)}$ for all but finitely many $n \in \mathbb{N}$. Also let m and r be polynomial-bounded functions. Then $\mathrm{QIP}(m, a, b) \subseteq \mathrm{QIP}(3, 1, 2^{-r})$.*

For a wide range of completeness and soundness probabilities this leaves just four complexity classes among those defined above: $\mathrm{QIP}(0) = \mathrm{BQP}$, $\mathrm{QIP}(1) = \mathrm{QMA}$, $\mathrm{QIP}(2)$, and $\mathrm{QIP}(3) = \mathrm{QIP}$.

The final property of quantum interactive proofs that will be discussed in this section is the existence of an interesting complete promise problem [90]. The problem is to determine whether the operations induced by two quantum circuits are significantly different, or are approximately the same, with respect to the same metric on quantum operations discussed in Section III.2.

THE QUANTUM CIRCUIT DISTINGUISHABILITY PROBLEM

Input:	Quantum circuits Q_0 and Q_1, both taking n input qubits and producing m output qubits.
Yes:	$\delta(Q_0, Q_1) \geq 2/3$.
No:	$\delta(Q_0, Q_1) \leq 1/3$.

Theorem 13. *The quantum circuit distinguishability problem is* QIP*-complete with respect to Karp reductions.*

VI.3 Zero-knowledge quantum interactive proofs

Interactive proof systems can sometimes be made to have a cryptographically motivated property known as the *zero-knowledge* property [56]. Informally speaking, an interactive proof system is zero-knowledge if a verifier "learns nothing" from an interaction with the prover on an input $x \in A_{\text{yes}}$, beyond the fact that it is indeed the case that $x \in A_{\text{yes}}$. This should hold even if the verifier deviates from the actions prescribed to it by the interactive proof being considered. At first this notion seems paradoxical, but nevertheless there are many interesting examples of such proof systems. For an introductory survey on zero-knowledge, see [98].

Several variants of zero-knowledge are often studied in the classical setting that differ in the particular way that the notion of "learning nothing" is formalized. This article will only consider *statistical* zero-knowledge, which is the variant of zero-knowledge that is most easily adapted to the quantum setting.

Suppose that $A = (A_{\text{yes}}, A_{\text{no}})$ is a promise problem, and (V, P) is a quantum interactive proof system for A. By this it is meant that V is the *honest verifier* and P is the *honest prover* that behave precisely as the proof system specifies. Whether or not this proof system possesses the zero-knowledge property depends on the characteristics of interactions between a *cheating verifier V'* with the honest prover P on inputs $x \in A_{\text{yes}}$. Figure 12 illustrates such an interaction, wherein it is viewed that the cheating verifier V' is using the interaction with P on input x to compute some quantum operation Φ_x. Informally speaking, the input to this operation represents the verifier's state of knowledge before the protocol is run, while the output represents the verifier's state of knowledge after.

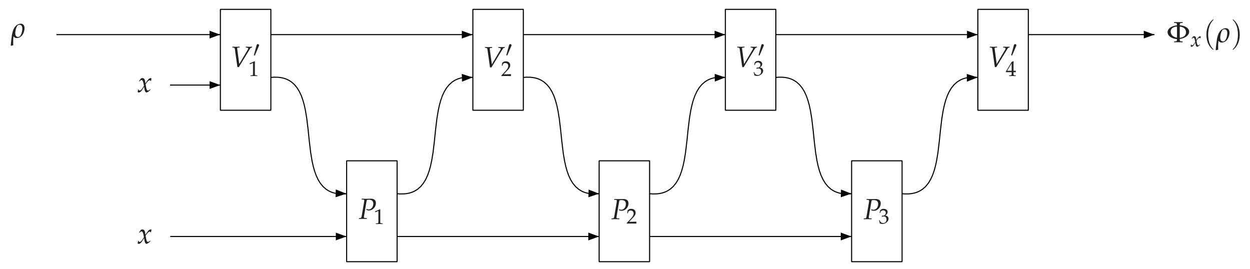


Figure 12: A cheating verifier V' performs a quantum operation Φ_x with the unintentional help of the honest prover.

Now, the proof system (V, P) is said to be *quantum statistical zero-knowledge* if, for any choice of a polynomial-time cheating verifier V', the quantum operation Φ_x can be efficiently approximated for all $x \in A_{\text{yes}}$. More precisely, the assumption that V' is described by polynomial-time generated quantum circuits must imply that there exists a polynomial-time generated family $\{Q_x : x \in \Sigma^*\}$ of quantum circuits for which $\delta(\Phi_x, Q_x)$ is negligible for every $x \in A_{\text{yes}}$. Intuitively this definition captures the notion of learning nothing—for anything that the cheating verifier could compute in polynomial time with the help of the prover could equally well have been computed in polynomial time without the prover's help.

The class of promise problems having statistical zero-knowledge quantum interactive proofs is denoted QSZK.

QSZK A promise problem $A = (A_{\text{yes}}, A_{\text{no}})$ is in QSZK if and only if it has a statistical zero-knowledge quantum interactive proof system.

Although it has not been proved, it is reasonable to conjecture that QSZK is properly contained in QIP; for the zero-knowledge property seems to be quite restrictive. Indeed, it was only recently established that there exist non-trivial quantum interactive proof systems that are statistical zero-knowledge [105]. The following facts [102, 105] are among those known about QSZK.

1. Statistical zero-knowledge quantum interactive proof systems can be parallelized to two messages. It follows that QSZK $\subseteq$ QIP(2).

2. The class QSZK is closed under complementation: a given promise problem A has a quantum statistical zero-knowledge proof if and only if the same is true for the problem obtained by exchanging the yes- and no-instances of A.

3. Statistical zero-knowledge quantum interactive proof systems can be simulated in polynomial space: QSZK $\subseteq$ PSPACE.

Classical analogues to the first and second facts in this list were shown first [91]. (The classical analogue to the third fact is SZK $\subseteq$ PSPACE, which follows trivially from IP $\subseteq$ PSPACE.) A key step toward proving the above properties (which is also similar to the classical case) is to establish that the following promise problem is complete for QSZK. The problem is a restricted version of the QIP-complete quantum circuit distinguishability problem.

THE QUANTUM STATE DISTINGUISHABILITY PROBLEM

Input:	Quantum circuits Q_0 and Q_1, both taking no input qubits and producing m output qubits. Let ρ_0 and ρ_1 be the density matrices corresponding to the outputs of these circuits.
Yes:	$\delta(\rho_0, \rho_1) \geq 2/3$.
No:	$\delta(\rho_0, \rho_1) \leq 1/3$.

Theorem 14. *The quantum state distinguishability problem is* QSZK*-complete with respect to Karp reductions.*

A quantum analogue of a different SZK-complete problem known as the *entropy difference problem* [54] has recently been shown to be complete for QSZK [26].

Kobayashi [72] has recently proved several interesting properties of quantum *computational* zero-knowledge proof systems.

VI.4 Multiple-prover quantum interactive proofs

Multiple-prover interactive proof systems are variants of interactive proofs where a verifier interacts with two or more provers that are not able to communicate with one another during the course of the interaction. Classical multiple-prover interactive proofs are extremely powerful: the class of promise problems having multiple-prover interactive proofs (denoted MIP) coincides with NEXP [19]. This is true even for two-prover interactive proofs wherein the verifier exchanges just one round of communication with each prover in parallel [47]. The key to the power of multiple-prover interactive proofs is the inability of the provers to communicate during the execution of the proof system. Similar to a detective interrogating two suspects in separate rooms in a police station, the verifier can ask questions of the provers that require strongly correlated answers, limiting the ability of cheating provers to convince the verifier of a false statement.

The identification of MIP with NEXP assumes a completely classical description of the provers. Quantum information, however, delivers a surprising twist for this model: even when the verifier is classical, an entangled quantum state shared between two provers can allow for non-classical correlations between their answers to the verifier's questions. This phenomenon is better known as a Bell-inequality violation [24] in the quantum physics literature. Indeed, there exist two-prover interactive proof systems that are sound against classical provers, but can be cheated by entangled quantum provers [37].

MIP*	A promise problem $A = (A_{\text{yes}}, A_{\text{no}})$ is in MIP* if and only if there exists a multiple-prover interactive proof system for A wherein the verifier is classical and the provers may share an arbitrary entangled state.

One may also consider fully quantum variants of multiple-prover interactive proofs, which were first studied by Kobayashi and Matsumoto [73].

QMIP	A promise problem $A = (A_{\text{yes}}, A_{\text{no}})$ is in QMIP if and only if there exists a multiple-prover quantum interactive proof system for A.

Various refinements on these classes have been studied, where parameters such as the number of provers, number of rounds of communication, completeness and soundness probabilities, and bounds on the amount of entanglement shared between provers are taken into account.

The results proved in both [37] and [73] support the claim that it is entanglement shared between provers that is the key issue in multiple-prover quantum interactive proofs. Both models are equivalent in power to MIP when provers are forbidden to share entanglement before the proof system is executed.

At the time of the writing of this article, research into these complexity classes and general properties of multiple-prover quantum interactive proofs is highly active and has led to several interesting results (such as [38, 65, 63, 64], among others). Despite this effort, little can be said at this time about the relationship among the above classes and other known complexity classes. For instance, only the trivial lower bounds PSPACE $\subseteq$ MIP* and QIP $\subseteq$ QMIP, and no good upper bounds, are known. It has not even been ruled out that non-computable languages could have multiple-prover quantum interactive proof systems. These difficulties seem to stem from two issues: (i) no bounds are known for the size of entangled states needed for provers to perform well in an interactive proof, and (ii) the possible correlations that can be induced with entanglement are not well-understood.

One highly restricted variant of MIP* that has been studied, along with its unentangled counterpart, is as follows.

$\oplus$MIP* A promise problem $A = (A_{\text{yes}}, A_{\text{no}})$ is in $\oplus$MIP* if and only if there exists a one-round two-prover interactive proof system for A wherein the provers each send a single bit to the verifier, and the verifier's decision to accept or reject is determined by the questions asked along with the XOR of these bits. The verifier is classical and the provers are quantum and share an arbitrary entangled state.

$\oplus$MIP This class is similar to $\oplus$MIP*, except that the provers may not share entanglement (and therefore can be assumed to be classical without loss of generality).

It holds that $\oplus$MIP $=$ NEXP for some choices of completeness and soundness probabilities [59, 25]. On the other hand, it has been proved [106] that $\oplus$MIP* $\subseteq$ QIP(2) and therefore $\oplus$MIP* $\subseteq$ EXP.

VI.5 Other variants of quantum interactive proofs

Other variants of quantum interactive proof systems have been studied, including *public-coin* quantum interactive proofs and quantum interactive proofs with *competing provers*.

Public-coin quantum interactive proofs

Public-coin interactive proof systems are a variant of interactive proofs wherein the verifier performs no computations until all messages with the prover have been exchanged. In place of the verifier's messages are sequences of random bits, visible to both the prover and verifier. Such interactive proof systems are typically called *Arthur–Merlin games* [17, 20], and the verifier and prover are called Arthur and Merlin, respectively, in this setting.

Quantum variants of such proof systems and their corresponding classes are easily defined. For instance, QAM is defined as follows [80].

QAM A promise problem $A = (A_{\text{yes}}, A_{\text{no}})$ is in QAM if and only if it has a quantum interactive proof system of the following restricted form. Arthur uniformly chooses some polynomial-bounded number of classical random bits, and sends a copy of these bits to Merlin. Merlin responds with a quantum state on a polynomial-bounded number of qubits. Arthur then performs a polynomial-time quantum computation on the input, the random bits, and the state sent by Merlin to determine acceptance or rejection.

It is clear that QAM $\subseteq$ QIP(2), but unlike the classical case [57] equality is not known in the quantum setting. It is straightforward to prove the upper bound QAM $\subseteq$ PSPACE, while containment QIP(2) $\subseteq$ PSPACE is not known.

The class QMAM may be defined through a similar analogy with classical Arthur–Merlin games. Here, Merlin sends the first message, Arthur responds with a selection of random bits, and then Merlin sends a second message.

QMAM A promise problem $A = (A_{\text{yes}}, A_{\text{no}})$ is in QMAM if and only if it has a quantum interactive proof system of the following restricted form. Merlin sends a polynomial-bounded number of qubits to Arthur. Without performing any computations, Arthur uniformly chooses some polynomial-bounded number of classical random bits, and sends a copy of these bits to Merlin. Merlin responds with a second collection of qubits. Arthur then performs a polynomial-time quantum computation on the input, the random bits, and the quantum information sent by Merlin in order to determine acceptance or rejection.

Even when Arthur is restricted to a single random bit, this class has the full power of QIP [80].

Theorem 15. QMAM = QIP.

Quantum interactive proofs with competing provers

Another variant of interactive proof systems that has been studied is one where a verifier interacts with two *competing* provers, sometimes called the *yes-prover* and the *no-prover*. Unlike ordinary two-prover interactive proof systems, it is now assumed that the provers have conflicting goals: the yes-prover wants to convince the verifier that a given input string is a yes-instance of the problem being considered, while the no-prover wants to convince the verifier that the input is a no-instance. The verifier is sometimes called the *referee* in this setting, given that interactive proof systems of this form are naturally modeled as competitive games between the two players. Two complexity classes based on interactive proofs with competing provers are the following.

RG A promise problem $A = (A_{\text{yes}}, A_{\text{no}})$ is in RG (short for *refereed games*) if and only if it has a classical interactive proof system with two competing provers. The completeness and soundness conditions for such a proof system are replaced by the following conditions:

1. For every $x \in A_{\text{yes}}$, there exists a yes-prover P_{yes} that convinces the referee to *accept* with probability at least 2/3, regardless of the strategy employed by the no-prover P_{no}.

2. For every $x \in A_{\text{no}}$, there exists a no-prover P_{no} that convinces the referee to *reject* with probability at least 2/3, regardless of the strategy employed by the yes-prover P_{yes}.

QRG A promise problem $A = (A_{\text{yes}}, A_{\text{no}})$ is in QRG (*quantum refereed games*) if and only if it has a quantum interactive proof system with two competing provers. The completeness and soundness conditions for such a proof system are analogous to RG.

Classical refereed games have the computational power of deterministic exponential time [46], and the same is true in the quantum setting [58]: RG = QRG = EXP. The containment EXP $\subseteq$ RG represents an application of the arithmetization technique, while QRG $\subseteq$ EXP exemplifies the power of semidefinite programming.

VII Other selected notions in quantum complexity

In this section of this article, a few other topics in quantum computational complexity theory are surveyed that do not fall under the headings of the previous sections. While incomplete, this selection should provide the reader with some sense for the topics that have been studied in quantum complexity.

VII.1 Quantum advice

Quantum advice is a formal abstraction that addresses this question: how powerful is quantum software? More precisely, let us suppose that some polynomial-time generated family of quantum circuits $Q = \{Q_n : n \in \mathbb{N}\}$ is given. Rather than assuming that each circuit Q_n takes n input qubits as for the class BQP, however, it is now assumed that Q_n takes $n + p(n)$ qubits for p some polynomial-bounded function: in addition to a given input $x \in \Sigma^n$, the circuit Q_n will take an *advice state* ρ_n on $p(n)$ qubits. This advice state may be viewed as pre-loaded quantum software for a quantum computer. The advice state may depend on the input length n, but not on the particular input $x \in \Sigma^n$. Similar to a quantum proof, the difficulty of preparing this state is ignored; but unlike a quantum proof the advice state is completely trusted. The quantum complexity class BQP/qpoly is now defined to be the class of promise problems that are solved by polynomial-time quantum algorithms with quantum advice in the natural way. The first published paper on quantum advice was [85], while the definitions and most of the results discussed in this section are due to Aaronson [2].

BQP/qpoly A promise problem $A = (A_{\text{yes}}, A_{\text{no}})$ is in BQP/qpoly if and only if there exists a polynomial-bounded function p, a collection of quantum states $\{\rho_n : n \in \mathbb{N}\}$ where each ρ_n is a $p(n)$-qubit state, and a polynomial-time generated family of quantum circuits $Q = \{Q_n : n \in \mathbb{N}\}$ with the following properties. For all $x \in A_{\text{yes}}$, Q accepts $(x, \rho_{|x|})$ with probability at least 2/3; and for all $x \in A_{\text{no}}$, Q accepts $(x, \rho_{|x|})$ with probability at most 1/3.

Similar to BQP without advice, it is straightforward to shown that the constants 2/3 and 1/3 in this definition can be replaced by a wide range of functions $a, b : \mathbb{N} \rightarrow [0, 1]$.

As the notation suggests, BQP/qpoly is a quantum analogue of the class P/poly. This analogy may be used as the basis for several relevant points about BQP/qpoly, quantum advice, and their relationship to classical complexity classes and notions.

1. As is well-known, P/poly may be defined either in a manner similar to the above definition for BQP/qpoly, or more simply as the class of promise problems solvable by polynomial-size Boolean circuit families with no uniformity restrictions. Based on similar ideas, the class BQP/qpoly does not change if the circuit family $\{Q_n : n \in \mathbb{N}\}$ is taken to be an arbitrary polynomial-size circuit family without uniformity constraints.

2. There is a good argument to be made that quantum advice is better viewed as a quantum analogue of *randomized advice* rather than *deterministic advice*. That is, BQP/qpoly can equally

well be viewed as a quantum analogue of the (suitably defined) complexity class BPP/rpoly. It happens to be the case, however, that BPP/rpoly = P/poly. (The situation is rather different for *logarithmic-length advice*, where randomized advice is strictly more powerful than ordinary deterministic advice.)

3. Any combination of quantum, randomized, or deterministic advice with quantum, randomized, or deterministic circuits can be considered. This leads to classes such as BQP/rpoly, BQP/poly, P/qpoly, P/rpoly, and so on. (The only reasonable interpretation of BPP/qpoly and P/qpoly is that classical circuits effectively measure quantum states in the standard basis the instant that they touch them.)

   At most three distinct classes among these possibilities arise: BQP/qpoly, BQP/poly, and P/poly. This is because BQP/rpoly = BQP/poly and

$$\text{BPP/qpoly} = \text{BPP/rpoly} = \text{BPP/poly} = \text{P/qpoly} = \text{P/rpoly} = \text{P/poly}.$$

   The principle behind these equalities is that nonuniformity is stronger than randomness [7].

The following theorem places an upper-bound on the power of polynomial-time quantum algorithms with quantum advice [2].

Theorem 16. BQP/qpoly $\subseteq$ PP/poly.

Although in all likelihood the class PP/poly is enormous, containing many interesting problems that one can have little hope of being able to solve in practice, the upper-bound represented by this theorem is far from obvious. The most important thing to notice is that the power of quantum advice (to a BQP machine) is simulated by deterministic advice (to a PP machine). This means that no matter how complex, a polynomial-size quantum advice state can never encode more information accessible to a polynomial-time quantum computer than a polynomial-length string can, albeit to an unbounded error probabilistic machine.

Quantum advice has also been considered for other quantum complexity classes such as QMA and QIP. For instance, the following bound is known on the power of QMA with quantum advice [3].

Theorem 17. QMA/qpoly $\subseteq$ PSPACE/poly.

Generally speaking, the study of both quantum and randomized advice is reasonably described as a sticky business when unbounded error machines or interactive protocols are involved. In these cases, complexity classes defined by both quantum and randomized advice can be highly non-intuitive and dependent on specific interpretations of models. For example, Raz [88] has shown that both QIP/qpoly and IP/rpoly contain all languages, provided that the advice is not accessible to the prover. Likewise, Aaronson [2] has observed that both PP/rpoly and PQP/qpoly contain all languages. These results are quite peculiar, given that significantly more powerful models can become strictly less powerful in the presence of quantum or randomized advice. For instance, even a Turing machine running in exponential space cannot decide all languages with bounded error given randomized advice.

VII.2 Space-bounded quantum computation

The quantum complexity classes that have been discussed thus far in this article are based on the abstraction that efficient quantum computations are those that can be performed in polynomial

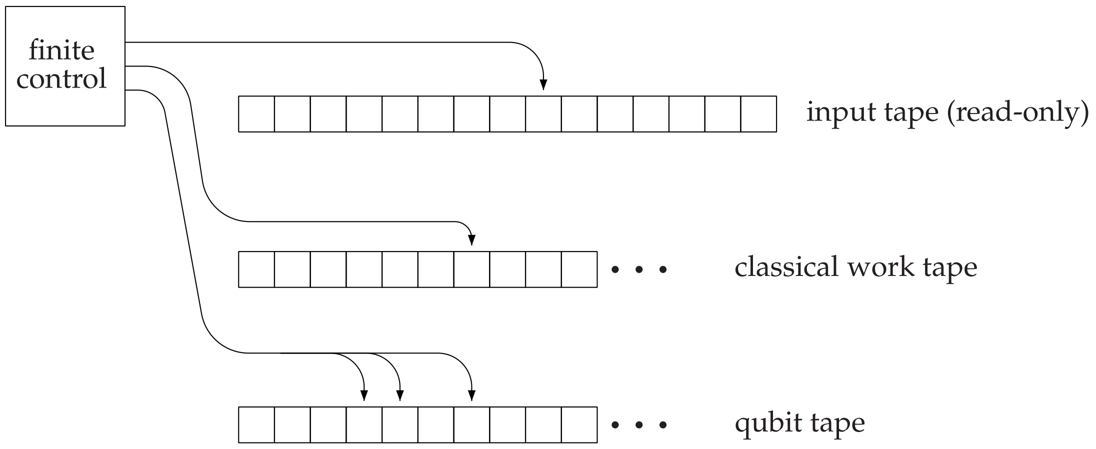


Figure 13: A quantum Turing machine. This variant of quantum Turing machine is classical with the exception of a quantum work tape, each square of which contains a qubit. Quantum operations and measurements can be performed on the qubits scanned by the quantum tape heads.

time. Quantum complexity classes may also be defined by bounds on space rather than time, but here a different computational model is required to reasonably compare with classical models of space-bounded computation. One simple choice of a suitable model is a hybrid between quantum circuits and classical Turing machines—and although this model is different from the variants of quantum Turing machines that were originally studied in the theory of quantum computing [41, 31], the term *quantum Turing machine* (QTM for short) is nevertheless appropriate. A different quantum Turing machine model that is suitable for studying time-space trade-offs has recently been proposed in [81].

Figure 13 illustrates a quantum Turing machine. A quantum Turing machine has a read-only classical input tape, a classical work tape, and a quantum tape consisting of an infinite sequence of qubits each initialized to the zero-state. Three tape heads scan the quantum tape, allowing quantum operations to be performed on the corresponding qubits. (It would be sufficient to have just two, but allowing three tape heads parallels the choice of a universal gate set that allows three-qubit operations.) A single step of a QTM's computation may involve ordinary moves by the classical parts of the machine and quantum operations on the quantum tape: Toffoli gates, Hadamard gates, phase-shift gates, or single-qubit measurements in the computational basis.

The running time of a quantum Turing machine is defined as for ordinary Turing machines. The space used by such a machine is the number of squares on the classical work tape plus the number of qubits on the quantum tape that are ever visited by one of the tape heads. Similar to the classical case, the input tape does not contribute to the space used by the machine because it is a read-only tape.

The following complexity classes are examples of classes that can be defined using this model.

BQL A promise problem $A = (A_{\text{yes}}, A_{\text{no}})$ is in BQL (bounded-error quantum logarithmic space) if and only if there exists a quantum Turing machine M running in polynomial time and logarithmic space that accepts every string $x \in A_{\text{yes}}$ with probability at least $2/3$ and accepts every string $x \in A_{\text{no}}$ with probability at most $1/3$.

PQL A promise problem $A = (A_{\text{yes}}, A_{\text{no}})$ is in PQL (unbounded-error quantum logarithmic space) if and only if there exists a quantum Turing machine M running in polynomial time and logarithmic space that accepts every string $x \in A_{\text{yes}}$ with probability strictly greater than $1/2$ and accepts every string $x \in A_{\text{no}}$ with probability at most $1/2$.

BQPSPACE A promise problem $A = (A_{\text{yes}}, A_{\text{no}})$ is in BQPSPACE (bounded-error quantum polynomial space) if and only if there exists a quantum Turing machine M running in polynomial space that accepts every string $x \in A_{\text{yes}}$ with probability at least $2/3$ and accepts every string $x \in A_{\text{no}}$ with probability at most $1/3$.

PQPSPACE A promise problem $A = (A_{\text{yes}}, A_{\text{no}})$ is in PQPSPACE (unbounded-error quantum polynomial space) if and only if there exists a quantum Turing machine M running in polynomial space that accepts every string $x \in A_{\text{yes}}$ with probability strictly greater than $1/2$ and accepts every string $x \in A_{\text{no}}$ with probability at most $1/2$.

Unlike polynomial-time computations, it is known that quantum information does not give a significant increase in computational power in the space-bounded case [100, 103].

Theorem 18. *The following relationships hold.*

1. BQL $\subseteq$ PQL $=$ PL.
2. BQPSPACE $=$ PQPSPACE $=$ PSPACE.

The key relationship in the above theorem, from the perspective of quantum complexity, is PQL $\subseteq$ PL, which can be shown using space-bounded counting complexity. In particular, the proof relies on a theory of GapL functions [13] that parallels the theory of GapP functions, and allows for a variety of matrix computations to be performed in PL.

The above theorem, together with the containment PL $\subseteq$ NC [34], implies that both BQL and PQL are contained in NC. An interpretation of this fact is that logarithmic-space quantum computations can be very efficiently simulated in parallel.

VII.3 Bounded-depth quantum circuits

The *depth* of a classical or quantum circuit is the maximum number of gates encountered on any path from an input bit or qubit to an output bit or qubit in the circuit. One may reasonably think of circuit depth as the parallel running time, or the number of time units needed to apply a circuit when operations may be parallelized in any way that respects the topology of the circuit.

The following complexity class, first defined by Moore and Nilsson [82], represent bounded-error quantum variants of the class NC.

QNC A promise problem $A = (A_{\text{yes}}, A_{\text{no}})$ is in QNC (bounded-error quantum NC) if and only if there exists a logarithmic-space generated family $\{Q_n : n \in \mathbb{N}\}$ of poly-logarithmic depth quantum circuits, where each circuit Q_n takes n input qubits and produces one output qubit, such that $\Pr[Q \text{ accepts } x] \geq 2/3$ for all $x \in A_{\text{yes}}$, and $\Pr[Q \text{ accepts } x] \leq 1/3$ for all $x \in A_{\text{no}}$.

Many other complexity classes based on bounded-depth quantum circuits have been studied as well. The survey of Bera, Green, and Homer [29] discusses several examples.

In the classical case there is a very close relationship between space-bounded and depth-bounded computation [33, 34]. This close relationship is based on two main ideas: the first is that space-bounded computations can be simulated efficiently by bounded-depth circuits using parallel algorithms for matrix computations, and the second is that bounded-depth Boolean circuits can be efficiently simulated by space-bounded computations via depth-first traversals of the circuit to be simulated.

For quantum computation this close relationship is not known to exist. One direction indeed holds, as was discussed in the previous subsection: space-bounded quantum computations can be efficiently simulated by depth-bounded circuits. The other direction, which is an efficient space-bounded simulation of bounded-depth quantum circuits, is not known to hold and is arguably quite unlikely. Informally speaking, bounded-depth quantum circuits are computationally powerful, whereas space-bounded quantum Turing machines are not. Three facts that support this claim are as follows.

1. Computing the acceptance probability for even constant-depth quantum circuits is as hard as computing acceptance probabilities for arbitrary polynomial-size quantum circuits [49].

2. Shor's factoring algorithm can be implemented by quantum circuits having logarithmic-depth, along with classical pre- and post-processing [39].

3. The quantum circuit distinguishability problem remains complete for QIP when restricted to logarithmic-depth quantum circuits [89].

It is reasonable to conjecture that QNC is incomparable with BPP and properly contained in BQP.

VIII Future directions

There are many future directions for research in quantum computational complexity theory. The following list suggests just a few of many possibilities.

1. The power of multiple-prover quantum interactive proofs, for both quantum and classical verifiers, is very poorly understood. In particular: (i) no interesting upper-bounds are known for either MIP^* or QMIP, and (ii) neither the containment $\text{NEXP} \subseteq \text{MIP}^*$ nor $\text{NEXP} \subseteq \text{QMIP}$ is known to hold.

2. The containment $\text{NP} \subseteq \text{BQP}$ would be a very powerful incentive to build a quantum computer, to say the least. While there is little reason to hope that this containment holds, there is at the same time little evidence against it aside from the fact that it fails relative to an oracle [28]. A better understanding of the relationship between BQP and NP, including possible consequences of one being contained in the other, is an interesting direction for further research in quantum complexity.

3. Along similar lines to the previous item, an understanding of the relationship between BQP and the polynomial-time hierarchy has remained elusive. It is not known whether BQP is contained in the polynomial-time hierarchy, whether there is an oracle relative to which this is false, or even whether there is an oracle relative to which BQP is not contained in AM.

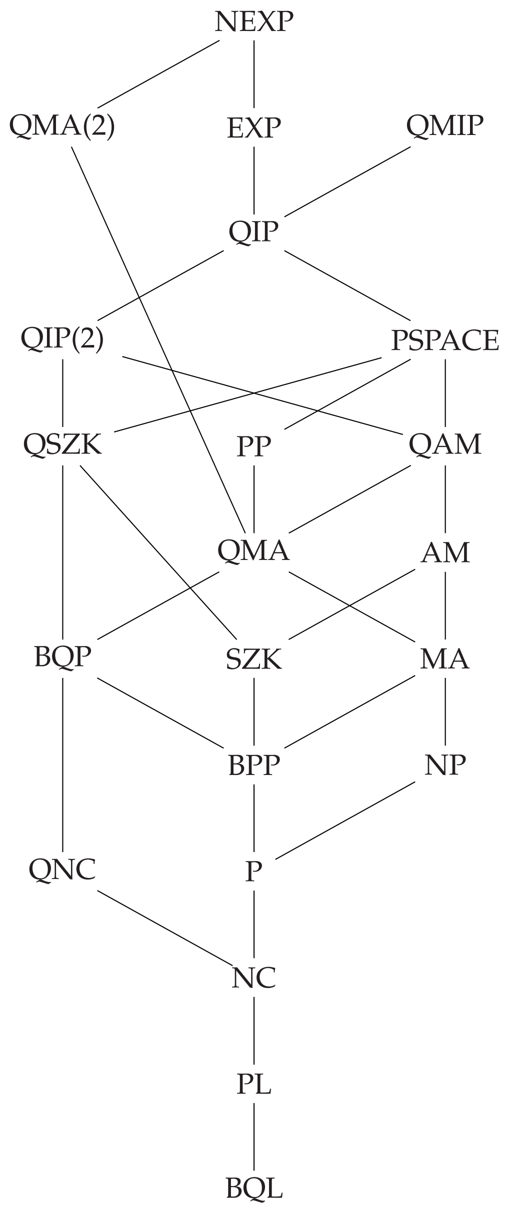


Figure 14: A diagram of inclusions among most of the complexity classes discussed in this article.

4. Interest in complexity classes is ultimately derived from the problems that they contain. An important future direction in quantum complexity theory is to prove the inclusion of interesting computational problems in the quantum complexity classes for which this is possible. Of the many problems that have been considered, one of the most perplexing from the perspective of quantum complexity is the graph isomorphism problem. It is a long-standing open problem whether the graph isomorphism problem is in BQP. A seemingly easier task than showing the inclusion of this problem in BQP is proving that its complement is contained in QMA. In other words, can Merlin prepare a quantum state that convinces Arthur that two graphs are not isomorphic?

References

[1] S. Aaronson. Quantum lower bound for the collision problem. In *Proceedings of the Thirty-Fourth Annual ACM Symposium on Theory of Computing*, 2002.

[2] S. Aaronson. Limitations of quantum advice and one-way communication. *Theory of Computing*, 1:1–28, 2005.

[3] S. Aaronson. QMA/qpoly is contained in PSPACE/poly: de-Merlinizing quantum protocols. In *Proceedings of the 21st Annual IEEE Conference on Computational Complexity*, pages 261–273, 2006.

[4] S. Aaronson, S. Beigi, A. Drucker, B. Fefferman, and P. Shor. The power of unentanglement. Available as arXiv.org e-print 0804.0802, 2008.

[5] S. Aaronson and G. Kuperberg. Quantum versus classical proofs and advice. *Theory of Computing*, 3:129–157, 2007.

[6] S. Aaronson and Y. Shi. Quantum lower bounds for the collision and the element distinctness problems. *Journal of the ACM*, 51(4):595–605, 2004.

[7] L. Adleman. Two theorems on random polynomial time. In *Proceeding of the 19th Annual IEEE Symposium on Foundations of Computer Science*, pages 75–83, 1978.

[8] L. Adleman, J. DeMarrais, and M. Huang. Quantum computability. *SIAM Journal on Computing*, 26(5):1524–1540, 1997.

[9] D. Aharonov, D. Gottesman, S. Irani, and J. Kempe. The power of quantum systems on a line. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*, pages 373–383, 2007.

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

[11] D. Aharonov and T. Naveh. Quantum NP – a survey. Available as arXiv.org e-Print quant-ph/0210077, 2002.

[12] D. Aharonov, W. van Dam, J. Kempe, Z. Landau, S. Lloyd, and O. Regev. Adiabatic quantum computation is equivalent to standard quantum computation. *SIAM Journal on Computing*, 37(1):166–194, 2007.

[13] E. Allender and M. Ogihara. Relationships among PL, #L, and the determinant. *RAIRO – Theoretical Informatics and Applications*, 30:1–21, 1996.

[14] S. Arora and B. Barak. *Complexity Theory: A Modern Approach (Preliminary web draft)*, 2006.

[15] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and the hardness of approximation problems. *Journal of the ACM*, 45(3):501–555, 1998.

[16] S. Arora and S. Safra. Probabilistic checking of proofs: a new characterization of NP. *Journal of the ACM*, 45(1):70–122, 1998.

[17] L. Babai. Trading group theory for randomness. In *Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing*, pages 421–429, 1985.

[18] L. Babai. Bounded round interactive proofs in finite groups. *SIAM Journal on Discrete Math*, 5(1):88–111, 1992.

[19] L. Babai, L. Fortnow, and C. Lund. Non-deterministic exponential time has two-prover interactive protocols. *Computational Complexity*, 1(1):3–40, 1991.

[20] L. Babai and S. Moran. Arthur-Merlin games: a randomized proof system, and a hierarchy of complexity classes. *Journal of Computer and System Sciences*, 36(2):254–276, 1988.

[21] L. Babai and E. Szemerédi. On the complexity of matrix group problems I. In *Proceedings of the 25th Annual IEEE Symposium on Foundations of Computer Science*, pages 229–240, 1984.

[22] R. Beigel, N. Reingold, and D. Spielman. PP is closed under intersection. *Journal of Computer and System Sciences*, 50(2):191–202, 1995.

[23] S. Beigi and P. Shor. On the complexity of computing zero-error and Holevo capacity of quantum channels. Available as arXiv.org e-Print 0709.2090, 2007.

[24] J. Bell. On the Einstein-Podolsky-Rosen paradox. *Physics*, 1(3):195–200, 1964.

[25] M. Bellare, O. Goldreich, and M. Sudan. Free bits, PCPs, and non-approximability – towards tight results. *SIAM Journal on Computing*, 27(3):804–915, 1998.

[26] A. Ben-Aroya and A. Ta-Shma. Quantum expanders and the quantum entropy difference problem. Available as arXiv.org e-print quant-ph/0702129, 2007.

[27] C. Bennett. Logical reversibility of computation. *IBM Journal of Research and Development*, 17:525–532, 1973.

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

[29] D. Bera, F. Green, and S. Homer. Small depth quantum circuits. *ACM SIGACT News*, 38(2):35–50, 2007.

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

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

[32] H. Blier and A. Tapp. All languages in NP have very short quantum proofs. Available as arXiv.org e-print 0709.0738, 2007.

[33] A. Borodin. On relating time and space to size and depth. *SIAM Journal on Computing*, 6:733–744, 1977.

[34] A. Borodin, S. Cook, and N. Pippenger. Parallel computation for well-endowed rings and space-bounded probabilistic machines. *Information and Control*, 58:113–136, 1983.

[35] G. Brassard. Quantum communication complexity. *Foundations of Physics*, 33(11):1593–1616, 2003.

[36] R. Cleve. An introduction to quantum complexity theory. In C. Macchiavello, G.M. Palma, and A. Zeilinger, editors, *Collected Papers on Quantum Computation and Quantum Information Theory*, pages 103–127. World Scientific, 2000.

[37] R. Cleve, P. Høyer, B. Toner, and J. Watrous. Consequences and limits of nonlocal strategies. In *Proceedings of the 19th Annual IEEE Conference on Computational Complexity*, pages 236–249, 2004.

[38] R. Cleve, W. Slofstra, F. Unger, and S. Upadhyay. Perfect parallel repetition theorem for quantum XOR proof systems. In *Proceedings of the 22nd Annual IEEE Conference on Computational Complexity*, pages 109–114, 2007.

[39] R. Cleve and J. Watrous. Fast parallel circuits for the quantum Fourier transform. In *Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science*, pages 526–536, 2000.

[40] S. Cook. The complexity of theorem proving procedures. In *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, pages 151–158, 1972.

[41] D. Deutsch. Quantum theory, the Church–Turing principle and the universal quantum computer. *Proceedings of the Royal Society of London*, A400:97–117, 1985.

[42] D. Deutsch. Quantum computational networks. *Proceedings of the Royal Society of London*, A425:73–90, 1989.

[43] I. Dinur. The PCP theorem by gap amplification. *Journal of the ACM*, 54(3), 2007.

[44] D.-Z. Du and K.-I. Ko. *Theory of Computational Complexity*. John Wiley & Sons, 2000.

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

[46] U. Feige and J. Kilian. Making games short. In *Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing*, pages 506–516, 1997.

[47] U. Feige and L. Lovász. Two-prover one-round proof systems: their power and their problems. In *Proceedings of the Twenty-Fourth Annual ACM Symposium on Theory of Computing*, pages 733–744, 1992.

[48] S. Fenner, L. Fortnow, and S. Kurtz. Gap-definable counting classes. *Journal of Computer and System Sciences*, 48:116–148, 1994.

[49] S. Fenner, F. Green S. Homer, and Y. Zhang. Bounds on the power of constant-depth quantum circuits. In *Proceedings of the 15th International Symposium on Fundamentals of Computation Theory*, volume 3623 of *Lecture Notes in Computer Science*, pages 44–55. Springer, 2005.

[50] R. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6/7):467–488, 1983.

[51] L. Fortnow. Counting complexity. In L. Hemaspaandra and A. Selman, editors, *Complexity Theory Retrospective II*, pages 81–107. Springer, 1997.

[52] L. Fortnow and J. Rogers. Complexity limitations on quantum computation. *Journal of Computer and System Sciences*, 59(2):240–252, 1999.

[53] O. Goldreich. On promise problems (a survey in memory of Shimon Even [1935–2004]). Electronic Colloquium on Computational Complexity, Report TR05-018, 2005.

[54] O. Goldreich and S. Vadhan. Comparing entropies in statistical zero-knowledge with applications to the structure of SZK. In *Proceedings of the 14th Annual IEEE Conference on Computational Complexity*, pages 54–73, 1999.

[55] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof systems. In *Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing*, pages 291–304, 1985.

[56] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1):186–208, 1989.

[57] S. Goldwasser and M. Sipser. Private coins versus public coins in interactive proof systems. In S. Micali, editor, *Randomness and Computation*, volume 5 of *Advances in Computing Research*, pages 73–90. JAI Press, 1989.

[58] G. Gutoski and J. Watrous. Toward a general theory of quantum games. In *Proceedings of the 39th ACM Symposium on Theory of Computing*, pages 565–574, 2007.

[59] J. Håstad. Some optimal inapproximability results. *Journal of the ACM*, 48(4):798–859, 2001.

[60] D. Janzing, P. Wocjan, and T. Beth. Non-identity-check is QMA-complete. *International Journal of Quantum Information*, 3(2):463–473, 2005.

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

[62] J. Kempe, A. Kitaev, and O. Regev. The complexity of the local Hamiltonian problem. *SIAM Journal on Computing*, 35(5):1070–1097, 2006.

[63] J. Kempe, H. Kobayashi, K. Matsumoto, B. Toner, and T. Vidick. Entangled games are hard to approximate. Available as arXiv.org e-Print 0704.2903, 2007.

[64] J. Kempe, H. Kobayashi, K. Matsumoto, and T. Vidick. Using entanglement in quantum multi-prover interactive proofs. In *Proceedings of the 23rd Annual Conference on Computational Complexity*, 2008. To appear.

[65] J. Kempe, O. Regev, and B. Toner. The unique games conjecture with entangled provers is false. Available as arXiv.org e-Print 0710.0655, 2007.

[66] A. Kitaev. Quantum computations: algorithms and error correction. *Russian Mathematical Surveys*, 52(6):1191–1249, 1997.

[67] A. Kitaev. “Quantum NP”. Talk at AQIP’99: Second Workshop on Algorithms in Quantum Information Processing, DePaul University, January 1999.

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

[69] A. Kitaev and J. Watrous. Parallelization, amplification, and exponential time simulation of quantum interactive proof system. In *Proceedings of the 32nd ACM Symposium on Theory of Computing*, pages 608–617, 2000.

[70] 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.

[71] E. Knill. Quantum randomness and nondeterminism. Technical Report LAUR-96-2186, Los Alamos National Laboratory, 1996. Available as arXiv.org e-Print quant-ph/9610012.

[72] H. Kobayashi. General properties of quantum zero-knowledge proofs. In *Proceedings of the Fifth IACR Theory of Cryptography Conference*, volume 4948 of *Lecture Notes in Computer Science*, pages 107–124. Springer, 2008.

[73] H. Kobayashi and K. Matsumoto. Quantum multi-prover interactive proof systems with limited prior entanglement. *Journal of Computer and System Sciences*, 66(3), 2003.

[74] H. Kobayashi, K. Matsumoto, and T. Yamakami. Quantum Merlin-Arthur proof systems: Are multiple Merlins more helpful to Arthur? In *Proceedings of the 14th Annual International Symposium on Algorithms and Computation*, 2003.

[75] L. Levin. Universal search problems (English translation). *Problems of Information Transmission*, 9(3):265–266, 1973.

[76] Y.-K. Liu. Consistency of local density matrices is QMA-complete. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, volume 4110 of *Lecture Notes in Computer Science*, pages 438–449. Springer, 2006.

[77] Y.-K. Liu, M. Christandl, and F. Verstraete. Quantum computational complexity of the n-representability problem: QMA complete. *Physical Review Letters*, 98(11):110503, 2007.

[78] S. Lloyd. Ultimate physical limits to computation. *Nature*, 406:1047–1054, 2000.

[79] C. Lund, L. Fortnow, H. Karloff, and N. Nisan. Algebraic methods for interactive proof systems. *Journal of the ACM*, 39(4):859–868, 1992.

[80] C. Marriott and J. Watrous. Quantum Arthur-Merlin games. *Computational Complexity*, 14(2):122–152, 2005.

[81] D. van Melkebeek and T. Watson. A quantum time-space lower bound for the counting hierarchy. Available as arXiv.org e-print 0712.2545, 2007.

[82] C. Moore and M. Nilsson. Parallel quantum computation and quantum codes. *SIAM Journal on Computing*, 31(3):799–815, 2002.

[83] G. Moore. Cramming more components onto integrated circuits. *Electronics*, 38(8), 1965.

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

[85] H. Nishimura and T. Yamakami. Polynomial time quantum computation with advice. *Information Processing Letters*, 90(4):195–204, 2004.

[86] R. Oliveira and B. Terhal. The complexity of quantum spin systems on a two-dimensional square lattice. Available as arXiv.org e-Print quant-ph/0504050, 2005.

[87] C. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

[88] R. Raz. Quantum information and the PCP theorem. In *46th Annual IEEE Symposium on Foundations of Computer Science*, pages 459–468, 2005.

[89] B. Rosgen. Distinguishing short quantum computations. In *Proceedings of the 25th International Symposium on Theoretical Aspects of Computer Science*, pages 597–608, 2008.

[90] B. Rosgen and J. Watrous. On the hardness of distinguishing mixed-state quantum computations. In *Proceedings of the 20th Annual Conference on Computational Complexity*, pages 344–354, 2005.

[91] A. Sahai and S. Vadhan. A complete promise problem for statistical zero-knowledge. *Journal of the ACM*, 50(2):196–249, 2003.

[92] A. Shamir. IP $=$ PSPACE. *Journal of the ACM*, 39(4):869–877, 1992.

[93] 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.

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

[95] W. F. Stinespring. Positive functions on C^*-algebras. *Proceedings of the American Mathematical Society*, 6(2):211–216, 1955.

[96] S. Toda. PP is as hard as the polynomial-time hierarchy. *SIAM Journal on Computing*, 20(5):865–887, 1991.

[97] T. Toffoli. Reversible computing. Technical Report MIT/LCS/TM-151, Laboratory for Computer Science, Massachusetts Institute of Technology, 1980.

[98] S. Vadhan. The complexity of zero knowledge. In *27th International Conference on Foundations of Software Technology and Theoretical Computer Science,*, volume 4855 of *Lecture Notes in Computer Science*, pages 52–70, 2007.

[99] L. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8:189–201, 1979.

[100] J. Watrous. Space-bounded quantum complexity. *Journal of Computer and System Sciences*, 59(2):281–326, 1999.

[101] J. Watrous. Succinct quantum proofs for properties of finite groups. In *Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science*, pages 537–546, 2000.

[102] J. Watrous. Limits on the power of quantum statistical zero-knowledge. In *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science*, pages 459–468, 2002.

[103] J. Watrous. On the complexity of simulating space-bounded quantum computations. *Computational Complexity*, 12:48–84, 2003.

[104] J. Watrous. PSPACE has constant-round quantum interactive proof systems. *Theoretical Computer Science*, 292(3):575–588, 2003.

[105] J. Watrous. Zero-knowledge against quantum attacks. In *Proceedings of the 38th ACM Symposium on Theory of Computing*, pages 296–305, 2006.

[106] S. Wehner. Entanglement in interactive proof systems with binary answers. In *Proceedings of the 23rd Annual Symposium on Theoretical Aspects of Computer Science*, volume 3884 of *Lecture Notes in Computer Science*, pages 162–171. Springer, 2006.

[107] R. de Wolf. Quantum communication and complexity. *Theoretical Computer Science*, 287(1):337–353, 2002.