Foundations and Trends® in
Theoretical Computer Science
Vol. 11, No. 1–2 (2015) 1–215

DOI: 10.1561/0400000068

Quantum Proofs

Thomas Vidick
California Institute of Technology
vidick@cms.caltech.edu

John Watrous
University of Waterloo
john.watrous@uwaterloo.ca

Contents

Abstract

Quantum information and computation provide a fascinating twist on the notion of proofs in computational complexity theory. For instance, one may consider a quantum computational analogue of the complexity class NP, known as QMA, in which a quantum state plays the role of a proof (also called a certificate or witness), and is checked by a polynomial-time quantum computation. For some problems, the fact that a quantum proof state could be a superposition over exponentially many classical states appears to offer computational advantages over classical proof strings. In the interactive proof system setting, one may consider a verifier and one or more provers that exchange and process quantum information rather than classical information during an interaction for a given input string, giving rise to quantum complexity classes such as QIP, QSZK, and QMIP* that represent natural quantum analogues of IP, SZK, and MIP. While quantum interactive proof systems inherit some properties from their classical counterparts, they also possess distinct and uniquely quantum features that lead to an interesting landscape of complexity classes based on variants of this model.

In this survey we provide an overview of many of the known results concerning quantum proofs, computational models based on this concept, and properties of the complexity classes they define. In particular, we discuss non-interactive proofs and the complexity class QMA, single-prover quantum interactive proof systems and the complexity class QIP, statistical zero-knowledge quantum interactive proof systems and the complexity class QSZK, and multiprover interactive proof systems and the complexity classes QMIP, QMIP*, and MIP*.

T. Vidick and J. Watrous. *Quantum Proofs.* Foundations and Trends® in Theoretical Computer Science, vol. 11, no. 1–2, pp. 1–215, 2015.
DOI: 10.1561/0400000068.

1

Introduction

The topic of this survey, *quantum interactive proof systems*, draws upon three different notions—quantum information, interaction, and proofs—whose combination forms a fascinating recipe best presented in the reverse order.

We begin with the notion of *proofs* in complexity theory. This notion has been central to complexity theory from its early beginnings, relating closely to the fundamental distinction between *efficient construction* and *efficient verification*. In greater detail, it has long been recognized that for some computational problems whose solutions may be difficult to obtain, it may nevertheless be possible to efficiently *verify* the correctness of a solution, given some additional information (representing a *proof*) that aids in this verification. The complexity class NP represents a formalization of this notion—it includes those decision problems for which positive instances can be efficiently verified given a suitable proof string (and for which negative instances are never incorrectly verified as positive ones).

The distinction between efficient construction and efficient verification appears, for instance, in work of Edmonds [55] from 1965 (although not in his more famous 1965 paper [56]), where he describes the *princi-*

ple of the absolute supervisor: a supervisor can ask his or her assistant to carry out a potentially lengthy search procedure for some computational problem (potentially "killing" the assistant with work!), and at the end of the day the assistant is expected to provide sufficient information so that his or her solution can be "verified with ease" by the supervisor.

The more modern terminology used to describe this situation is that of a *prover* and *verifier*: the prover represents the assistant, while the verifier represents the supervisor in Edmonds' story. With respect to this terminology, our sympathies are generally reversed: the verifier, faced with limitations on its computational abilities, simply wants to know whether or not a given input is a positive instance of a fixed decision problem, while the computationally unrestricted prover is untrustworthy and will try to convince the verifier that the input is a positive instance, irrespective of the truth.

The importance of what is now known as the P vs NP question, which essentially asks if there are indeed problems for which the efficient construction of a solution is impossible while an efficient verification is possible, was in fact implicitly noted some time prior to Edmonds' work—in a letter written to John von Neumann in the mid-1950s, Kurt Gödel observed the striking consequences that would result from an efficient solution to a certain problem in first-order logic that is now known to be NP-complete. The development of the theory of NP-completeness, by Cook [49], Levin [122], and Karp [105] in the early 1970s, placed the notion of proofs in computational complexity on a firm mathematical foundation.

Next, we add a second ingredient: *interaction*. The notion of an *interactive proof system* was introduced independently by Goldwasser, Micali, and Rackoff [71, 72] and Babai [19, 22] in the 1980s. Babai was following a similar line of thought that led to the introduction of P and NP: the identification of structural features that allow a fine classification of the difficulty of solving classes of computational problems (in this case, problems related to groups). Goldwasser, Micali, and Rackoff arrived at the notion from a different angle. They introduced a notion of "knowledge complexity" of an interactive proof (informally, the amount of information about a problem instance conveyed by the interaction beyond the problem's solution) and gave an example of a simple problem (testing quadratic residuosity) for which there existed a *zero-knowledge* interactive proof.

The simplest type of interactive proof system represents an interaction between a prover and verifier, which are similar characters to the ones introduced in the non-interactive setting above, except that now we imagine that they may engage in a discussion rather than the prover simply providing the verifier with information. In particular, the verifier may ask the prover questions and demand acceptable responses in order to be satisfied. As before, one views that the prover's aim is to convince the verifier that a given input string is a positive instance to a fixed decision problem (or, equivalently, that an input string possesses a fixed property of interest). The verifier's goal is to check the validity of the prover's argument, accepting only in the event that it is indeed convinced that the input string is a positive problem instance, and rejecting if not.

It turns out that the (classical) interactive proof system model only represents a departure from the non-interactive setting described above when the verifier makes use of *randomness*—in which case we must generally be satisfied with the verifier gathering overwhelming statistical evidence, but not having absolute certainty, in order to conclude that the prover's argument is valid. (When no randomness is used, the prover may as well attempt to convince the verifier to accept non-interactively by simply presenting a complete transcript of the conversation they would have had by interacting, which the verifier can efficiently check for validity by itself.) As in the non-interactive case, we also make the standard assumption that the prover's computational abilities are greater than the verifier's (or, at the very least, that the prover has access to information that the verifier lacks). The class IP is representative of the case in which the verifier is required to run in polynomial time and the prover is computationally unrestricted. The characterization IP=PSPACE [124, 150] cements the tight relationship between interactive proofs and computation, justifying its position as a fundamental concept in computational complexity theory.

Many variants of interactive proof systems have been considered that impose additional conditions on the interaction, place more stringent limits on the prover's abilities, or consider interactions between more diverse sets of parties, such as a verifier interacting with multiple cooperating or competing provers. Prominent examples include the class SZK of problems that have zero-knowledge interactive proofs and the class MIP of problems whose solution can be determined by a polynomial-time verifier interacting with multiple cooperating provers, restricted only in their inability to communicate with one other.

Finally, we finish off with a curious catalyst: *quantum information*. The Church–Turing thesis plays a foundational role in computer science by postulating that computability is model independent: whether based on the concept of a Turing machine, first-order logic, or any "purely mechanical process," the classes of functions whose values can be "effectively calculated" are identical. The development of quantum computing in the 1990s posed the first serious threat to this thesis. Impetus for the consideration of computational procedures based on the laws of quantum mechanics was provided by Shor's discovery of an efficient *quantum* algorithm for factoring [151, 152], a problem for which no efficient classical probabilistic algorithm is known. The study of the relation between P (or BPP) and BQP, the class of problems that can be decided in polynomial time by a quantum Turing machine, is among the most interesting and mysterious problems in modern complexity theory. The difficulty of this question prompts the introduction of "quantum analogues" of the most important classical complexity classes in an attempt to identify problems for which the consideration of quantum processes induces a strict separation.

One prominent example is the complexity class QMA of decision problems whose positive instances have *quantum proofs* that can be verified by an efficient quantum procedure. Aside from the fundamental problem of understanding the physical substrate of computation, the consideration of quantum mechanical states as proofs provides a fascinating window into some of the most subtle features of quantum physics. An essential way in which quantum states differ from their classical counterparts is in one's ability to recover information that is present in the mathematical description of the state. In quantum mechanics this ability is limited by the uncertainty principle—for example, both the momentum and position of an electron can be determined with high precision in principle, but there is a fundamental limit to the accuracy with which those two properties can be *simultaneously* determined. Thus, the study of QMA sheds light on the many areas of physics in which the properties of quantum states play an important role, from the theory of superconductors to that of black holes.

Stir vigorously, and you have a recipe for quantum interactive proofs. Beyond the class QMA already discussed, quantum interactive proofs reflect the richness of the classical model on which they are based, providing a powerful lens on the properties of quantum mechanics and quantum information. For example, single-prover quantum interactive proofs, corresponding to the class QIP, have the distinguishing property that they can be parallelized to three message interactions, and this property (unlikely to hold for classical interactive proofs) makes crucial use of the superposition principle of quantum mechanics. The no-cloning theorem plays an important role in the study of the class QSZK of problems having quantum zero-knowledge interactive proofs by hindering the construction of "simulators" essential to the study of classical zero-knowledge. By allowing multiple cooperating provers to share quantum entanglement, the class QMIP* provides a complexity-theoretic viewpoint on the nonlocal properties of entanglement.

Having set a rather ambitious stage for this survey, we proceed with a more concrete description of what is to come.

Chapter 2 introduces some preliminary material. While it is assumed that the reader will be familiar with the basics of complexity theory and quantum computing, we have made an effort to state and explain the facts that play an important role in the results to be discussed, directing the reader to standard textbooks for background material.

In Chapter 3 we begin with the consideration of the class QMA of languages that have efficiently verifiable quantum proofs. This class satisfies many of the desirable features of NP, such as strong error amplification procedures and a rich set of complete problems. It also has many variants restricting, or extending, the types of proofs allowed and the power of the verifier; a small but representative set of such variants is discussed in the chapter.

Chapter 4 considers single-prover quantum interactive proof systems. An important tool in the study of the associated class QIP is a semidefinite programming formulation of the verifier's maximum acceptance probability. We introduce this formulation and use it to establish a parallel repetition property of QIP as well as to give an essentially self-contained proof of the characterization QIP=PSPACE.

In Chapter 5 we consider the class QSZK of quantum zero-knowledge interactive proofs. One aspect in which these proof systems differ from their classical counterparts is the difficulty of extending the key techniques (such as rewinding) that are systematically used in the classical setting, and we describe known quantum analogues for such techniques.

The final chapter, Chapter 6, is devoted to quantum multi-prover interactive proofs. It will be seen that the consideration of entanglement between multiple provers leads to a failure of the most basic intuition on which the classical theory is built (most important of which are the technique of oracularization and the characterization MIP=NEXP). We describe ways to work around this failure by fighting fire with fire, devising techniques that make positive use of the provers' ability to share entanglement.

This survey is mainly intended for non-specialists having a basic background in complexity theory and quantum information. A typical reader may be a student or researcher in either area desiring to learn about the fundamentals of the (actively developing) theory of quantum interactive proofs. In most cases we have not included full proofs of the main results we present, but whenever possible we have either included detailed sketches of the key ideas behind the proofs, or have attempted to describe their most salient elements in simplified settings. Each chapter ends with notes that provide references for the results discussed in the chapter as well as a brief survey of related results and pointers to the literature.

2

Preliminary Notions

The purpose of this chapter is to summarize various notions, primarily concerning basic complexity theory and quantum computation, that we will rely upon in subsequent chapters and consider as requisite background material. The chapter is mainly intended to clarify our notation and terminology; readers unfamiliar with the notions summarized will likely find other sources, including textbooks and surveys focusing on this material, to better serve as a first introduction. Suggested references will be mentioned when appropriate.

2.1 Complexity theoretic notions

We assume the reader is familiar with standard classical complexity classes, such as NC, P, BPP, NP, AM, PSPACE, and NEXP, as well as the quantum complexity class BQP. The textbook of Arora and Barak [15] may be consulted for definitions and basic properties of these classes. All computational problems considered in this survey will be assumed to be encoded over the binary alphabet $\{0,1\}$, which is hereafter denoted Σ.

It is convenient for us to consider computational decision problems as *promise problems*, where input strings may be assumed to be drawn from some subset of all possible input strings. More formally, a promise problem is a pair $A = (A_{yes}, A_{no})$, where $A_{yes}, A_{no} \subseteq \Sigma^*$ are disjoint sets of strings. The strings contained in the sets A_{yes} and A_{no} are called the *yes-instances* and *no-instances* of the problem, and a correct answer to any such instance of the problem A requires that it be properly classified as a yes-instance or no-instance. All strings lying outside of $A_{yes} \cup A_{no}$ may be considered as "don't care" inputs, and no requirements whatsoever are placed on computations for such strings. All of the complexity classes mentioned above may be considered as classes of promise problems, as opposed to classes of languages (which are essentially promise problems for which $A_{yes} \cup A_{no} = \Sigma^*$).

Karp reductions (also called polynomial-time many-one reductions), as well as the notion of *completeness* of a problem for a complexity class, are defined for promise problems in the same way as for languages. More precisely, a promise problem $A = (A_{yes}, A_{no})$ is *Karp reducible* to a promise problem $B = (B_{yes}, B_{no})$ if there exists a polynomial-time computable function f that maps every string $x \in A_{yes}$ to $f(x) \in B_{yes}$, and every string $x \in A_{no}$ to $f(x) \in B_{no}$. In this case, the notation $A \leq_m B$ (where the "m" is short for "many-one") may be used to indicate this relationship. A promise problem A is said to be *complete* for a certain class C of promise problems if $A \in$ C and every promise problem in C is Karp reducible to A.

There are several occasions in which we speak of functions defined on the nonnegative integers $\mathbb{N} = \{0, 1, 2, \ldots\}$, taking either nonnegative integer values or real-number values. In particular, the following terminology will be used:

1. A function of the form $f : \mathbb{N} \rightarrow \mathbb{N}$ is said to be *polynomially bounded* if there exists a polynomial-time deterministic Turing machine that outputs $1^{f(n)}$ on input 1^n, for every $n \in \mathbb{N}$. We will use the notation poly to denote an arbitrary function that is polynomially bounded.
2. A function of the form $g : \mathbb{N} \rightarrow \mathbb{Q}$ is said to be *polynomial-time computable* if there exists a polynomial-time deterministic Turing machine that outputs $g(n)$, expressed as a ratio of integers written in binary notation, on input 1^n, for each $n \in \mathbb{N}$.

2.2 Quantum states, channels, and measurements

When discussing quantum interactive proof systems, and quantum computations more generally, it is useful to make use of some basic concepts of the theory of quantum information. Readers unfamiliar with quantum information and computation are referred to the books of Nielsen and Chuang [131] and Kaye, Laflamme, and Mosca [106].

2.2.1 Linear algebra notation

We use calligraphic letters $\mathcal{X}, \mathcal{Y}, \mathcal{W}$ to denote Hilbert spaces. All Hilbert spaces considered in this survey are finite-dimensional and will usually correspond to systems comprised of zero or more qubits (in which case their dimension is a power of 2). It is assumed that an orthonormal *standard basis* has been fixed for each such space $\mathcal{X}$.

In the typical case in which $\mathcal{X}$ is the Hilbert space corresponding to n qubits, the standard basis is expressed using Dirac notation as

$$\{|x\rangle \,:\, x \in \Sigma^n\}, \tag{2.1}$$

where $|x\rangle$ denotes the column vector with a 1 in the entry indexed by x and zero for all other entries. An arbitrary vector $|\psi\rangle$ in such a space may be expressed as

$$|\psi\rangle = \sum_{x\in\Sigma^n} \alpha_x |x\rangle \tag{2.2}$$

for some choice of complex coefficients $\{\alpha_x \,:\, x \in \Sigma^n\}$. The *conjugate-transpose* of this vector is given by

$$\langle\psi| = \sum_{x\in\Sigma^n} \overline{\alpha_x}\langle x|, \tag{2.3}$$

where $\langle x|$ denotes the row vector (as opposed to a column vector) with a 1 in the entry indexed by x and zero for all other entries. The inner product of two vectors, $|\phi\rangle$ and $|\psi\rangle$, is written $\langle\phi|\psi\rangle$, and the Euclidean norm of $|\psi\rangle$ is defined as $\| |\psi\rangle \| = \langle\psi|\psi\rangle^{1/2}$.

Given two Hilbert spaces $\mathcal{X}$ and $\mathcal{Y}$, the space of all linear mappings (or *operators*) from $\mathcal{X}$ to $\mathcal{Y}$ is denoted $\mathrm{L}(\mathcal{X}, \mathcal{Y})$, and in the case that $\mathcal{X} = \mathcal{Y}$ the shorthand $\mathrm{L}(\mathcal{X})$ is used in place of $\mathrm{L}(\mathcal{X}, \mathcal{X})$. The identity element of $\mathrm{L}(\mathcal{X})$ is written $\mathbb{1}_{\mathcal{X}}$, and the trace of an operator $X \in \mathrm{L}(\mathcal{X})$ is defined as

$$\operatorname{Tr}(X) = \sum_{x} \langle x|X|x\rangle, \tag{2.4}$$

where the sum is taken over the standard basis elements of $\mathcal{X}$. (Any other orthonormal basis would yield the same value.) With respect to the standard bases of $\mathcal{X}$ and $\mathcal{Y}$, operators in the set $\mathrm{L}(\mathcal{X}, \mathcal{Y})$ may be identified with matrices in the usual way, with the (x, y) entry of the matrix corresponding to an operator A being given by $\langle x|A|y\rangle$.

The *adjoint*, or *conjugate transpose*, of an operator $A \in \mathrm{L}(\mathcal{X}, \mathcal{Y})$ is the operator $A^* \in \mathrm{L}(\mathcal{Y}, \mathcal{Y})$ defined by the condition

$$\langle y|A^*|x\rangle = \overline{\langle x|A|y\rangle} \tag{2.5}$$

for every choice of standard basis states $|x\rangle \in \mathcal{X}$ and $|y\rangle \in \mathcal{Y}$. An inner product is defined on the space $\mathrm{L}(\mathcal{X}, \mathcal{Y})$ as

$$\langle A, B\rangle = \operatorname{Tr}(A^* B) \tag{2.6}$$

for all $A, B \in \mathrm{L}(\mathcal{X}, \mathcal{Y})$.

The following set of operators will often be mentioned throughout this survey:

1. An operator $U \in \mathrm{L}(\mathcal{X})$ is *unitary* if $U^*U = \mathbb{1}_{\mathcal{X}}$, which is an equivalent condition to $UU^* = \mathbb{1}_{\mathcal{X}}$. The set of all such operators is denoted $\mathrm{U}(\mathcal{X})$. More generally, an operator $A \in \mathrm{L}(\mathcal{X}, \mathcal{Y})$ is an *isometry* if $A^*A = \mathbb{1}_{\mathcal{X}}$, and the set of all such operators is denoted $\mathrm{U}(\mathcal{X}, \mathcal{Y})$.
2. An operator $H \in \mathrm{L}(\mathcal{X})$ is *Hermitian* if $H = H^*$. The set of all such operators is denoted $\mathrm{Herm}(\mathcal{X})$.
3. An operator $P \in \mathrm{L}(\mathcal{X})$ is *positive semidefinite* if it is Hermitian and has only nonnegative eigenvalues. The set of all such operators is denoted $\mathrm{Pos}(\mathcal{X})$.
4. An operator $\Pi \in \mathrm{L}(\mathcal{X})$ is a *projection operator* if it is Hermitian and satisfies $\Pi^2 = \Pi$. The set of all such operators is denoted $\mathrm{Proj}(\mathcal{X})$.
5. An operator $\rho \in \mathrm{L}(\mathcal{X})$ is a *density operator* if it is positive semidefinite and satisfies $\operatorname{Tr}(\rho) = 1$. The set of all such operators is denoted $\mathrm{D}(\mathcal{X})$.

2.2.2 States and registers

A *quantum state* is represented by a density operator $\rho \in \mathrm{D}(\mathcal{X})$, for some Hilbert space $\mathcal{X}$ that has been associated with the physical system whose state is being described. A state is *pure* if it is represented by a density operator of the form $\rho = |\psi\rangle\langle\psi|$, for $|\psi\rangle$ a unit vector in $\mathcal{X}$. Equivalently, a state is pure if its associated density operator $\rho \in \mathrm{D}(\mathcal{X})$ is an extreme point of the (convex) set $\mathrm{D}(\mathcal{X})$. A state is said to be *mixed* if it is not pure. When a unit vector $|\psi\rangle$ is referred to as being a state of a system, it is to be understood that one is speaking of the pure state $|\psi\rangle\langle\psi|$.

It is convenient to refer to physical systems that store quantum information as *registers*. Names such as X, Y, and Z, and other capital letters written in a *sans serif* font, are commonly used for this purpose. With a given register X, one associates a Hilbert space $\mathcal{X}$, so the set of possible quantum states of X coincides with $\mathrm{D}(\mathcal{X})$. As a general convention, we use the same letter in different fonts to refer to a register and its associated Hilbert space.

Pairs or k-tuples of registers, such as (X, Y) or $(\mathsf{X}_1, \ldots, \mathsf{X}_k)$, are often considered, and may themselves be treated as single registers. The Hilbert space associated with such a compound register is obtained by taking the tensor product of the Hilbert spaces associated with the individual registers. For example, if $\mathsf{Z} = (\mathsf{X}, \mathsf{Y})$, then the Hilbert space $\mathcal{Z}$ corresponding to Z is given by $\mathcal{Z} = \mathcal{X} \otimes \mathcal{Y}$, for $\mathcal{X}$ and $\mathcal{Y}$ being the Hilbert spaces associated with X and Y, respectively. The standard basis of $\mathcal{Z}$ in this case is obtained by tensoring the elements of the standard bases of $\mathcal{X}$ and $\mathcal{Y}$:

$$\big\{ |x\rangle|y\rangle \,:\, x \in \Sigma^n,\, y \in \Sigma^m \big\}, \tag{2.7}$$

assuming here that X is an n-qubit register and Y is an m-qubit register. (In general, a juxtaposition of vectors, such as $|\phi\rangle|\psi\rangle$ for $|\phi\rangle \in \mathcal{X}$ and $|\psi\rangle \in \mathcal{Y}$, denotes a tensor product: $|\phi\rangle|\psi\rangle = |\phi\rangle \otimes |\psi\rangle$.)

Every nonzero vector $|\gamma\rangle \in \mathcal{X} \otimes \mathcal{Y}$ of a bipartite tensor product space has a decomposition

$$|\gamma\rangle = \sum_{i=1}^{r} \alpha_i |\phi_i\rangle|\psi_i\rangle \tag{2.8}$$

for $\alpha_1, \ldots, \alpha_r$ being positive real numbers and $|\phi_1\rangle, \ldots, |\phi_r\rangle \in \mathcal{X}$ and $|\psi_1\rangle, \ldots, |\psi_r\rangle \in \mathcal{Y}$ being vectors for which both of the collections $\{|\phi_1\rangle, \ldots, |\phi_r\rangle\}$ and $\{|\psi_1\rangle, \ldots, |\psi_r\rangle\}$ are orthonormal. This decomposition, which is called the *Schmidt decomposition*, is closely related to the singular value decomposition of matrices. The values $\alpha_1, \ldots, \alpha_r$ are the *Schmidt coefficients* of $|\gamma\rangle$, and are uniquely determined by $|\gamma\rangle$. A pure state corresponding to a unit vector $|\gamma\rangle \in \mathcal{X} \otimes \mathcal{Y}$ is called a *product* state if it takes the form $|\gamma\rangle = |\psi\rangle|\phi\rangle$ for vectors $|\psi\rangle \in \mathcal{X}$ and $|\phi\rangle \in \mathcal{Y}$, which is equivalent to $|\gamma\rangle$ having a single (nonzero) Schmidt coefficient $\alpha_1 = 1$.

A mixed state $\rho \in \mathrm{D}(\mathcal{X} \otimes \mathcal{Y})$ is *separable* if it has a decomposition

$$\rho = \sum_{i=1}^{N} p_i \, \sigma_i \otimes \tau_i \tag{2.9}$$

for $\sigma_1, \ldots, \sigma_N \in \mathrm{D}(\mathcal{X})$ and $\tau_1, \ldots, \tau_N \in \mathrm{D}(\mathcal{Y})$, and $(p_1, \ldots, p_N)$ being a vector of probabilities. A state that is not separable is called *entangled*. A pure state is separable if and only if it is a product state, but the mixed-state case is more complicated: it is an NP-hard problem to decide if a given density operator is separable [79].

If $\rho \in \mathrm{D}(\mathcal{X} \otimes \mathcal{Y})$ is a state of a pair of registers (X, Y), then the *reduced state* of X is obtained by taking the *partial trace* over $\mathcal{Y}$:

$$\mathrm{Tr}_{\mathcal{Y}}(\rho) = \sum_{y} (\mathbb{1}_{\mathcal{X}} \otimes \langle y|) \rho (\mathbb{1}_{\mathcal{X}} \otimes |y\rangle), \tag{2.10}$$

where the sum ranges over the elements of the standard basis of $\mathcal{Y}$.

Given a state $\rho \in \mathrm{D}(\mathcal{X})$ of a register X, a *purification* of ρ is any pure state of a pair of registers (X, Y) whose reduced state on X is ρ. That is, such a state is represented by a unit vector $|\gamma\rangle \in \mathcal{X} \otimes \mathcal{Y}$ such that

$$\rho = \mathrm{Tr}_{\mathcal{Y}}(|\gamma\rangle\langle\gamma|). \tag{2.11}$$

Given a spectral decomposition

$$\rho = \sum_{i=1}^{r} \lambda_i |\psi_i\rangle\langle\psi_i| \tag{2.12}$$

with each $\lambda_i > 0$, one may obtain a purification

$$|\gamma\rangle = \sum_{i} \sqrt{\lambda_i} |\psi_i\rangle|\phi_i\rangle \tag{2.13}$$

provided $\mathcal{Y}$ has dimension at least the rank of ρ, allowing for the existence of an orthonormal collection $\{|\phi_i\rangle\}$. The following theorem concerning purifications has fundamental importance.

Theorem 2.1 (Unitary equivalence of purifications)**.** Let $\rho \in \mathrm{D}(\mathcal{X})$ and suppose $|\psi\rangle, |\phi\rangle \in \mathcal{X} \otimes \mathcal{Y}$ satisfy

$$\mathrm{Tr}_{\mathcal{Y}}(|\psi\rangle\langle\psi|) = \rho = \mathrm{Tr}_{\mathcal{Y}}(|\phi\rangle\langle\phi|). \tag{2.14}$$

There exists a unitary operator $U \in \mathrm{U}(\mathcal{Y})$ such that $|\phi\rangle = (\mathbb{1}_{\mathcal{X}} \otimes U)|\psi\rangle$.

2.2.3 Channels and measurements

Quantum channels describe discrete-time changes in the states of registers that, in an idealized sense, may be considered physically implementable. Given registers X and Y having associated Hilbert spaces $\mathcal{X}$ and $\mathcal{Y}$, the set of all quantum channels transforming states of X into states of Y, denoted $\mathrm{C}(\mathcal{X}, \mathcal{Y})$, can be characterized as the set of all linear maps of the form

$$\Phi : \mathrm{L}(\mathcal{X}) \rightarrow \mathrm{L}(\mathcal{Y}) \tag{2.15}$$

that are *completely positive* and *trace-preserving*. An equivalent way to describe the two conditions of being completely positive and trace-preserving is to require that, for every finite-dimensional Hilbert space $\mathcal{Z}$, it holds that

$$(\Phi \otimes \mathbb{1}_{\mathrm{L}(\mathcal{Z})})(\rho) \in \mathrm{D}(\mathcal{Y} \otimes \mathcal{Z}) \tag{2.16}$$

for every density operator $\rho \in \mathrm{D}(\mathcal{X} \otimes \mathcal{Z})$. (Here, the mapping $\mathbb{1}_{\mathrm{L}(\mathcal{Z})}$ denotes the identity mapping on $\mathrm{L}(\mathcal{Z})$.)

A channel $\Phi \in \mathrm{C}(\mathcal{X}, \mathcal{Y})$ should be understood as representing a physical transformation of register X into register Y. That is, if the state of X is given by $\rho \in \mathrm{D}(\mathcal{X})$ and the channel $\Phi \in \mathrm{C}(\mathcal{X}, \mathcal{Y})$ is performed, the register X is transformed into the register Y, whose state is then $\Phi(\rho)$. The two registers X and Y never simultaneously co-exist in this situation, so it is not meaningful to consider their joint state. Of course, nothing prevents one from taking Y to be equal to X, and in this situation it is natural to view that the channel has simply changed the state of X, as opposed to transforming X into a new register.

A convenient representation of quantum channels is the *Stinespring representation.* Given an arbitrary channel $\Phi \in \mathrm{C}(\mathcal{X}, \mathcal{Y})$ there always exists a Hilbert space $\mathcal{Z}$, which can be chosen to have dimension at most the product of the dimensions of $\mathcal{X}$ and $\mathcal{Y}$, along with a linear operator $A \in \mathrm{L}(\mathcal{X}, \mathcal{Y} \otimes \mathcal{Z})$, such that

1. $\Phi(X) = \mathrm{Tr}_{\mathcal{Z}}(AXA^*)$ for every $X \in \mathrm{L}(\mathcal{X})$, and
2. $A^*A = \mathbb{1}_{\mathcal{X}}$ (i.e., A is an isometry).

The fact that every channel can be represented in this way is a consequence of a theorem known as *Stinespring's dilation theorem.*

It is instructive to consider the case in which a channel Φ transforms an n-qubit register X into an m-qubit register Y. In this case, the Hilbert spaces corresponding to these registers are such that $\mathcal{X}$ has dimension 2^n and $\mathcal{Y}$ has dimension 2^m; and when considering a Stinespring representation of Φ, one may take Z to be an $(n+m)$-qubit register, so that $\mathcal{Z}$ has dimension 2^{n+m}. One finds that there must exist a Stinespring representation

$$\Phi(X) = \mathrm{Tr}_{\mathcal{Z}}(AXA^*), \tag{2.17}$$

for A being an isometry of the form $A \in \mathrm{L}(\mathcal{X}, \mathcal{Y} \otimes \mathcal{Z})$, as is illustrated in Figure 2.1. One may further observe that it must be possible to express any isometry A of the form $A \in \mathrm{L}(\mathcal{X}, \mathcal{Y} \otimes \mathcal{Z})$ as

$$A = U(\mathbb{1}_{\mathcal{X}} \otimes |0^{2m}\rangle) \tag{2.18}$$

for U being a unitary operator acting on $n + 2m$ qubits. It is therefore possible to implement an arbitrary channel transforming n qubits to m qubits by means of a unitary operation on $n + 2m$ qubits, as is also illustrated in Figure 2.1.

An analogous fact to the unitary equivalence of purifications holds for Stinespring representations, as the following theorem states.

Theorem 2.2. Let $\Phi \in \mathrm{C}(\mathcal{X}, \mathcal{Y})$ and let $A, B \in \mathrm{L}(\mathcal{X}, \mathcal{Y} \otimes \mathcal{Z})$ be isometries such that

$$\mathrm{Tr}_{\mathcal{Z}}(AXA^*) = \Phi(X) = \mathrm{Tr}_{\mathcal{Z}}(BXB^*) \tag{2.19}$$

for all $X \in \mathrm{L}(\mathcal{X})$. There exists a unitary operator $U \in \mathrm{U}(\mathcal{Z})$ such that $B = (\mathbb{1}_{\mathcal{Y}} \otimes U)A$.

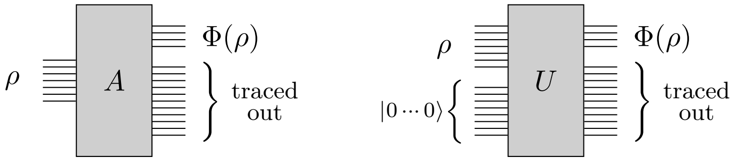


Figure 2.1: Given a quantum channel Φ transforming an n qubit register X into an m qubit register Y, one may always find an isometry A, transforming pure states of X to pure states of (Y, Z), for Z being an $n+m$ qubit register, so that $\Phi(\rho) = \operatorname{Tr}_{\mathcal{Z}}(A\rho A^*)$ for every state ρ of X. The isometry A can be realized as a unitary operation on $n+2m$ qubits, with all but the first n qubits being initialized to the $|0\rangle$ state.

A channel Φ is said to be a *unitary channel* if there exists a unitary operator U such that

$$\Phi(X) = UXU^* \tag{2.20}$$

for all $X \in \mathrm{L}(\mathcal{X})$, and Φ is said to be an *isometric channel* if there exists a linear isometry A such that

$$\Phi(X) = AXA^* \tag{2.21}$$

for all $X \in \mathrm{L}(\mathcal{X})$.

When considering channels and related notions, it is sometimes convenient to consider linear maps of the form

$$\Phi : \mathrm{L}(\mathcal{X}) \rightarrow \mathrm{L}(\mathcal{Y}) \tag{2.22}$$

that might be neither completely positive nor trace-preserving. The notation $\mathrm{T}(\mathcal{X}, \mathcal{Y})$ is used to denote the set of all linear maps of the form (2.22). For example, if $\Phi_0, \Phi_1 \in \mathrm{C}(\mathcal{X}, \mathcal{Y})$ are channels, it may be useful to consider the linear map $\Phi = \Phi_0 - \Phi_1 \in \mathrm{T}(\mathcal{X}, \mathcal{Y})$ as a way of representing the difference between the two channels. The *adjoint* of a map $\Phi \in \mathrm{T}(\mathcal{X}, \mathcal{Y})$ is the uniquely defined map $\Phi^* \in \mathrm{T}(\mathcal{Y}, \mathcal{X})$ satisfying

$$\langle Y, \Phi(X) \rangle = \langle \Phi^*(Y), X \rangle \tag{2.23}$$

for all $X \in \mathrm{L}(\mathcal{X})$ and $Y \in \mathrm{L}(\mathcal{Y})$.

A *measurement* is a process through which classical information is obtained from a register in a quantum state. For the purposes of this survey it will be sufficient to consider measurements that are defined as the following special cases of quantum channels:

1. A *standard-basis measurement* of a register X is described by the so-called *completely phase-damping* channel $\Lambda \in \mathrm{C}(\mathcal{X})$, which is defined as
$$\Lambda(\rho) = \sum_{x} \langle x|\rho|x\rangle |x\rangle\langle x|, \tag{2.24}$$
for the sum ranging over the standard basis elements of $\mathcal{X}$. When X is measured in this way, its resulting state is represented by a diagonal density operator, which is naturally associated with a classical, probabilistic state. That is, measuring ρ yields each outcome x with probability $\langle x|\rho|x\rangle$.

2. A general measurement of a register X can always be described as the composition of a channel Φ transforming X into Y, followed by a standard-basis measurement of Y. A composition of a channel and a standard-basis measurement of this sort can always be written as
$$(\Lambda\Phi)(X) = \sum_{y} \langle P_y, X\rangle |y\rangle\langle y|, \tag{2.25}$$
for $\{P_y\}$ being a collection of positive semidefinite *measurement operators* satisfying
$$\sum_{y} P_y = \mathbb{1}_{\mathcal{X}}. \tag{2.26}$$

2.2.4 Distance measures on quantum states and channels

The space $\mathrm{L}(\mathcal{X}, \mathcal{Y})$ is equipped with the *operator norm* (or *spectral norm*), derived from the Euclidean norm on $\mathcal{X}$ and $\mathcal{Y}$ as
$$\| X \| = \max\{ \| X|\psi\rangle \| \,:\, |\psi\rangle \in \mathcal{X},\ \| |\psi\rangle \| \leq 1 \}. \tag{2.27}$$

We will also make use of the *trace norm*, defined as
$$\| X \|_1 = \operatorname{Tr}\Bigl((XX^*)^{1/2} \Bigr). \tag{2.28}$$

Equivalently, $\| X \|_1$ is equal to the sum of the singular values of X. The operator norm and trace norm are dual to one another, meaning that the following relationships hold:

$$\begin{aligned} \|X\| &= \max\bigl\{|\langle Y, X\rangle| \,:\, Y \in \mathrm{L}(\mathcal{X},\mathcal{Y}),\ \|Y\|_1 \leq 1\bigr\},\\ \|X\|_1 &= \max\bigl\{|\langle Y, X\rangle| \,:\, Y \in \mathrm{L}(\mathcal{X},\mathcal{Y}),\ \|Y\| \leq 1\bigr\}. \end{aligned} \tag{2.29}$$

It is sometimes convenient to make use of the fact that, for any operator $X \in \mathrm{L}(\mathcal{X})$, it holds that

$$\|X\|_1 = \max_{U \in \mathrm{U}(\mathcal{X})} |\langle U, X\rangle|, \tag{2.30}$$

and furthermore if X is Hermitian this maximization may be restricted to operators U that are both unitary and Hermitian.

The most standard notion of distance between two quantum states ρ and σ is the *trace distance*

$$\frac{1}{2}\|\rho - \sigma\|_1. \tag{2.31}$$

(The factor $\frac{1}{2}$ ensures that the distance between two states lies in the interval $[0, 1]$.) A theorem called the *Holevo–Helstrom theorem* implies that

$$\begin{aligned} \max\Bigl\{\frac{1}{2}\langle P_0, \rho_0\rangle + \frac{1}{2}\langle P_1, \rho_1\rangle \,:\, P_0, P_1 \geq 0,\ P_0 + P_1 = \mathbb{1}_{\mathcal{X}}\Bigr\}&\\ = \frac{1}{2} + \frac{1}{4}\|\rho_0 - \rho_1\|_1,& \end{aligned} \tag{2.32}$$

which has the interpretation that the trace distance expresses the maximum bias with which any measurement can correctly distinguish between two states ρ_0 and ρ_1 given with equal probability.

It is sometimes convenient to refer to one state as being an *ε-approximation* to another when the trace-distance between the states is bounded from above by ε, as the following definition makes precise.

Definition 2.1. Let ρ and σ be states on the same space. It is said that σ is an *ε-approximation* to ρ if

$$\frac{1}{2}\|\rho - \sigma\|_1 \leq \varepsilon. \tag{2.33}$$

Another important measure of distance between quantum states is the *fidelity*, defined for density operators $\rho, \sigma \in \mathrm{D}(\mathcal{X})$ as

$$\mathrm{F}(\rho, \sigma) = \mathrm{Tr}\Bigl(\sqrt{\sqrt{\rho}\,\sigma\sqrt{\rho}}\Bigr) = \bigl\|\sqrt{\rho}\sqrt{\sigma}\bigr\|_1. \tag{2.34}$$

(The second expression makes it apparent that $\mathrm{F}(\rho,\sigma) = \mathrm{F}(\sigma,\rho)$, as $\sqrt{\rho}\sqrt{\sigma}$ and $\sqrt{\sigma}\sqrt{\rho} = (\sqrt{\rho}\sqrt{\sigma})^*$ must share the same singular values.) When $\sigma = |\psi\rangle\langle\psi|$ is a pure state, the expression simplifies to

$$\mathrm{F}(\rho, |\psi\rangle\langle\psi|) = \sqrt{\langle\psi|\rho|\psi\rangle}. \tag{2.35}$$

The following theorem gives an alternative characterization of the fidelity.

Theorem 2.3 (Uhlmann's theorem)**.** Let $\rho, \sigma \in \mathrm{D}(\mathcal{X})$ be density operators. It holds that

$$\mathrm{F}(\rho,\sigma) = \max_{|\psi\rangle,|\phi\rangle} |\langle\psi|\phi\rangle|, \tag{2.36}$$

where the maximization is over all purifications $|\psi\rangle$ and $|\phi\rangle$ of ρ and σ, respectively.

The fidelity is related to the trace distance by the *Fuchs–van de Graaf inequalities*:

$$1 - \mathrm{F}(\rho,\sigma) \leq \frac{1}{2}\|\rho - \sigma\|_1 \leq \sqrt{1 - \mathrm{F}(\rho,\sigma)^2}, \tag{2.37}$$

for all density operators ρ and σ.

There is a notion of distance between quantum channels that is analogous to the trace distance between quantum states. This notion of distance is defined by the *diamond norm* (also known as the *completely bounded trace norm*) of linear maps of the form $\Phi \in \mathrm{T}(\mathcal{X},\mathcal{Y})$. This norm is defined as

$$\|\Phi\|_\diamond = \max\bigl\{\bigl\|(\Phi \otimes \mathbb{1}_{\mathrm{L}(\mathcal{W})})(X)\bigr\|_1 : X \in \mathrm{L}(\mathcal{X}\otimes\mathcal{W}), \|X\|_1 \leq 1\bigr\}, \tag{2.38}$$

where $\mathcal{W}$ is any Hilbert space having dimension at least as large as $\mathcal{X}$. (Changing the dimension of $\mathcal{W}$ does not change the value of the norm, so long as $\dim(\mathcal{W}) \geq \dim(\mathcal{X})$.) An analogous theorem to the Holevo–Helstrom theorem establishes that the *diamond norm distance*

$$\frac{1}{2}\|\Phi_0 - \Phi_1\|_\diamond \tag{2.39}$$

between two channels $\Phi_0, \Phi_1 \in \mathrm{C}(\mathcal{X},\mathcal{Y})$ describes the maximum bias with which a physical process (consisting of an arbitrary state preparation, followed by a channel evaluation, followed by a measurement) can distinguish between Φ_0 and Φ_1 given with equal probability.

As for states, it is sometimes convenient to refer to one channel as being an *ε-approximation* to another when the diamond norm distance between the two channels is bounded from above by ε.

Definition 2.2. Let Φ and Ψ be channels sharing the same input spaces and the same output spaces. It is said that Φ is an *ε-approximation* to Ψ if

$$\frac{1}{2}\big\| \Psi - \Phi \big\|_{\diamond} \leq \varepsilon. \tag{2.40}$$

The following alternate characterization of the diamond norm will prove useful: if $\Phi \in \mathrm{T}(\mathcal{X}, \mathcal{Y})$ is a map specified by

$$\Phi(X) = \mathrm{Tr}_{\mathcal{Z}}(A_0 X A_1^*) \tag{2.41}$$

for all $X \in \mathrm{L}(\mathcal{X})$, for operators $A_0, A_1 \in \mathrm{L}(\mathcal{X}, \mathcal{Y} \otimes \mathcal{Z})$, then

$$\| \Phi \|_{\diamond} = \max_{\rho_0, \rho_1 \in \mathrm{D}(\mathcal{X})} \mathrm{F}\left(\Psi_0(\rho_0), \Psi_1(\rho_1)\right), \tag{2.42}$$

where

$$\Psi_0(X) = \mathrm{Tr}_{\mathcal{Y}}(A_0 X A_0^*) \quad \text{and} \quad \Psi_1(X) = \mathrm{Tr}_{\mathcal{Y}}(A_1 X A_1^*). \tag{2.43}$$

2.3 Quantum circuits

The primary model of computation used throughout this survey is the *quantum circuit* model. A quantum circuit is an acyclic network of *quantum gates* connected by *wires*. The quantum gates represent quantum channels while the wires represent qubits on which the channels act. In general, we allow the quantum channels implemented by the gates of a quantum circuit to be potentially non-unitary, as first suggested by Aharonov, Kitaev, and Nisan [9]. This general variant of the quantum circuit model has a fairly straightforward connection to the more commonly used model of unitary quantum circuits, by virtue of the Stinespring representation of channels, as will be discussed shortly.

An example of a quantum circuit having three input qubits and two output qubits is pictured in Figure 2.2. 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 a quantum channel from n qubits to m

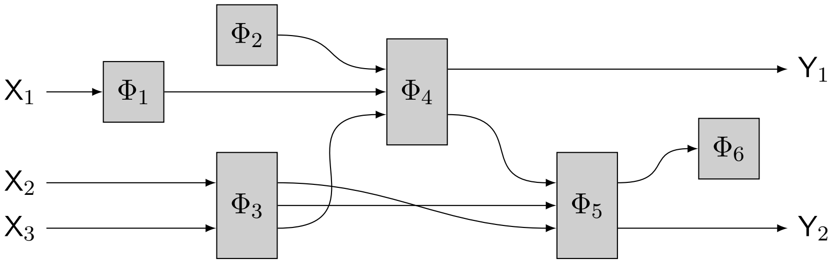


Figure 2.2: An example of a quantum circuit. The input qubits are labelled $\mathsf{X}_1, \mathsf{X}_2, \mathsf{X}_3$, the output qubits are labelled Y_1 and Y_2, and the gates are labelled by (hypothetical) quantum channels $\Phi_1, \ldots, \Phi_6$.

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 input and output qubits.

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. For the remainder of this survey, quantum circuits may be assumed to be composed of gates from the following list (representing a standard choice for a gate set):

1. *Toffoli gates.* A Toffoli gate is a three-qubit unitary gate Φ_T identified with the unitary transformation

$$T : |a\rangle|b\rangle|c\rangle \mapsto |a\rangle|b\rangle|c \oplus ab\rangle. \tag{2.44}$$

2. *Hadamard gates.* A Hadamard gate is a single-qubit unitary gate Φ_H identified with the unitary transformation

$$H : |a\rangle \mapsto \frac{1}{\sqrt{2}}|0\rangle + \frac{(-1)^a}{\sqrt{2}}|1\rangle. \tag{2.45}$$

3. *Phase-shift gates.* A Phase-shift gate is a single-qubit unitary gate Φ_P identified with the unitary transformation

$$P : |a\rangle \mapsto i^a|a\rangle. \tag{2.46}$$

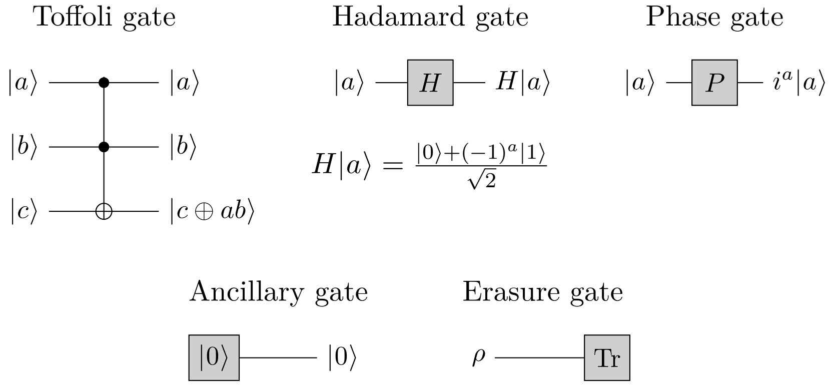


Figure 2.3: Commonly used notation for denoting gates from the universal gate set.

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

We note that it is not essential that one chooses this particular set of gates, and we will not often refer specifically to these gates in this survey—but it is convenient nevertheless to assume that reversible computations and Hadamard gates can be performed without error. Figure 2.3 illustrates the notation that is commonly used to describe these gates within quantum circuits.

The above gate set is *universal* in a strong sense: every quantum channel mapping qubits to qubits can be approximated to within any desired degree of accuracy by some quantum circuit composed of gates from this set. The following theorem expresses this fact in more precise terms.

Theorem 2.4 (Universality Theorem)**.** Let Φ be any quantum channel from n qubits to m qubits. For every $\varepsilon > 0$ there exists a quantum circuit Q with n input qubits and m output qubits that implements a ε-approximation to Φ. Moreover, for a fixed choice of n and m, the circuit Q may be taken to satisfy $\text{size}(Q) = \text{poly}(\log(1/\varepsilon))$.

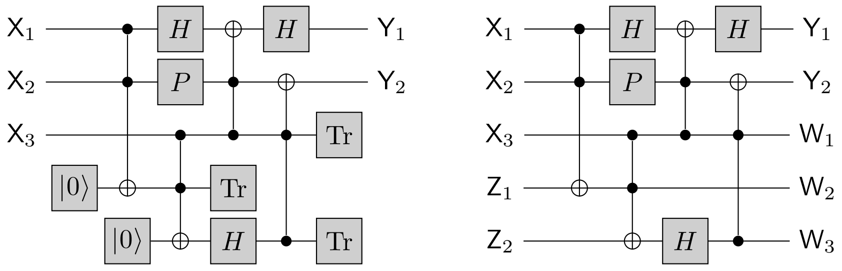


Figure 2.4: A general quantum circuit (left) and its unitary purification (right).

A *unitary quantum circuit* is a quantum circuit in which all of the gates correspond to unitary quantum channels, so that the channel associated to the entire circuit is therefore unitary as well. 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 equivalence between the unitary and general models of quantum circuits is made straightforward by the universal gate set described above. Suppose Q is a quantum circuit taking input qubits $(\mathsf{X}_1, \ldots, \mathsf{X}_n)$ and producing output qubits $(\mathsf{Y}_1, \ldots, \mathsf{Y}_m)$, and assume there are j ancillary gates and k erasure gates among the gates of Q. A new quantum circuit R may then be formed by removing the ancillary and erasure gates, and to account for the removal of these gates the circuit R takes j additional input qubits $(\mathsf{Z}_1, \ldots, \mathsf{Z}_j)$ and produces k additional output qubits $(\mathsf{W}_1, \ldots, \mathsf{W}_k)$. Figure 2.4 illustrates this simple process. The circuit R is said to be a *unitary purification* of Q. It holds that R is equivalent to Q, provided the qubits $(\mathsf{Z}_1, \ldots, \mathsf{Z}_j)$ are each initially set to the $|0\rangle$ state and the qubits $(\mathsf{W}_1, \ldots, \mathsf{W}_k)$ are ignored after the circuit is run. Despite the simplicity of this process, it is often useful to refer to unitary purifications of general quantum circuits obtained in this way.

Along similar lines, one may consider *isometric quantum circuits*, which are quantum circuits composed of unitary gates and ancillary gates, but no erasure gates. Such circuits implement isometric quantum channels. By performing the process described above, but only for the erasure gates of a general quantum circuit and not the ancillary gates, one obtains an isometric extension of a general quantum circuit.

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

3

Non-Interactive Quantum Proofs

The traditional notion of a proof in mathematics does not require an interaction to take place between an individual proving a theorem and one who verifies the proof (beyond the obvious requirement that the individual verifying the proof has come into possession of it). The same is true in a complexity-theoretic setting, in which proofs are typically abstracted as strings of symbols to be checked by computationally efficient verification procedures. The standard definition of NP in terms of polynomial-length proofs (or *certificates*), checked by polynomial-time deterministic computations, is representative of this traditional notion.

One natural way to generalize this notion to the quantum setting is to allow a proof to be a *quantum state* rather than a classical string of symbols. Such a state is to be verified by a computationally efficient procedure, as in the classical setting, but in this case the verification procedure will be an efficient *quantum* computation. The most natural and well-studied complexity class to be defined through this notion is QMA, which stands for "quantum Merlin–Arthur." Rather than being a direct quantum variant of NP, the class QMA is more accurately described as being a quantum computational analogue of the complexity class MA, which is essentially NP with a bounded-error polynomial-time probabilistic verifier rather than a deterministic one.

The study of the class QMA provides a complexity-theoretic perspective on the properties of quantum states and their relative power compared to classical states, when seen as untrusted advice to be given to a quantum or classical verifier respectively. The following points are among those to be discussed in this chapter:

1. In Section 3.1.2 the *group non-membership* problem is shown to be included in QMA. This problem illustrates the potential advantages of quantum over classical proofs.
2. Two procedures for error reduction—*parallel* error reduction and *witness-preserving* error reduction— for QMA are presented in Section 3.2. One consequence of witness-preserving error reduction is the inclusion QMA $\subseteq$ PP, providing an upper-bound on QMA in terms of a well-studied classical complexity class.
3. A few complete problems for QMA are introduced in Section 3.3. Some of these problems, such as the *local Hamiltonian problem*, illustrate the relevance of QMA for natural problems that arise in the study of quantum systems in physics or chemistry.
4. A selection of variants of QMA is presented, including ones with additional promises on the proof, such as it being *trusted*, *classical*, or *separable* with respect to some fixed partition. The study of these variants probes some of the essential features of the class QMA. Some of these variants are shown to be equivalent to QMA, while for others there is evidence that they may differ from QMA.

3.1 Definitions of quantum verification procedures and QMA

This section of the chapter is primarily concerned with the definition of the class QMA, along with some of its most basic mathematical aspects, such as its relationship to measurements and quantum circuits. A computational problem known as the *group non-membership problem* is shown to be contained in QMA, providing a simple example of how quantum proofs may be useful in a computational setting.

3.1.1 Definition of QMA

With the intuitive picture of a quantum proof that has been suggested above in mind, we wish to formalize the notion of an efficient quantum verification procedure that takes as input a quantum state, playing the role of a proof, and outputs a single binary value, indicating acceptance or rejection of the proof. A natural way to model such a procedure is as a quantum circuit that takes k qubits as input, where k denotes the length of the proof, and produces a single output qubit. Rather than stipulating that this output qubit must represent a classical state, indicating whether acceptance or rejection has occurred, we will simply assume that the qubit is to be measured with respect to the computational basis after being output by the circuit. The outcome of this measurement will indicate whether the proof has been accepted or rejected (with 1 indicating acceptance, 0 indicating rejection).

Suppose, somewhat more generally, that Φ is an arbitrary channel having k input qubits and 1 output qubit, and consider the following scenario. An individual (the *prover*) aims to provide Φ with an input state ρ on k qubits maximizing the probability that a standard basis measurement performed on the output qubit of Φ produces the outcome 1. In effect, the channel Φ followed by a standard basis measurement of its output qubit describes a general binary-valued measurement on the k qubits input to Φ. Indeed, it is straightforward to specify the measurement operators $P_0, P_1 \in \text{Pos}\big((\mathbb{C}^2)^{\otimes k}\big)$ associated with such a measurement, which are

$$P_0 = \Phi^*(|0\rangle\langle 0|) \quad \text{and} \quad P_1 = \Phi^*(|1\rangle\langle 1|), \tag{3.1}$$

where Φ^* denotes the adjoint mapping to Φ.

With this scenario in mind, we define the *value*[1] of Φ as

$$\omega(\Phi) = \max_{\rho} \langle 1|\Phi(\rho)|1\rangle, \tag{3.2}$$

where the maximum is over all density operators ρ on k qubits. In the case that Φ is the channel described by a circuit functioning as a

[1] The term *value* is not a standard term in this particular setting—but we use it nevertheless, as it is analogous to the standard usage of this term in the context of other models to be considered in subsequent chapters of this survey.

verification procedure, the value of Φ is the maximum probability with which a quantum proof may lead this procedure to accept. The value coincides with the largest eigenvalue of the measurement operator P_1 defined above, as a short calculation reveals:

$$\omega(\Phi) = \max_{\rho} \langle 1 | \Phi(\rho) | 1 \rangle = \max_{\rho} \langle \rho, \Phi^*(|1\rangle\langle 1|) \rangle = \lambda_1(P_1). \tag{3.3}$$

(In general, we write $\lambda_1(H)$, $\lambda_2(H)$, …, $\lambda_n(H)$ to denote the eigenvalues of a Hermitian operator H, sorted from largest to smallest, so $\lambda_1(P_1)$ denotes the largest eigenvalue of P_1.) A prover wishing to maximize the probability of obtaining the outcome 1 may as well take ρ to be a pure state $\rho = |\psi\rangle\langle\psi|$ for $|\psi\rangle$ being any unit eigenvector corresponding to the largest eigenvalue of P_1.

With the definition of the value of a channel in hand, we may define the class QMA as follows:

Definition 3.1. A promise problem $A = (A_{\text{yes}}, A_{\text{no}})$ is contained in the complexity class $\text{QMA}_{a,b}$ if there exists a polynomial-time computable function V possessing the following properties:

1. For every string $x \in A_{\text{yes}} \cup A_{\text{no}}$, one has that $V(x)$ is an encoding of a quantum circuit implementing a channel Φ_x having 1 output qubit.
2. *Completeness.* For every string $x \in A_{\text{yes}}$, the value of the channel Φ_x satisfies $\omega(\Phi_x) \geq a$.
3. *Soundness.* For every string $x \in A_{\text{no}}$, the value of the channel Φ_x satisfies $\omega(\Phi_x) \leq b$.

In this definition, $a, b \in [0,1]$ may be constant values or functions of the length of the input string x. When they are omitted, it is to be assumed that they are $a = 2/3$ and $b = 1/3$:

$$\text{QMA} = \text{QMA}_{\frac{2}{3},\frac{1}{3}}. \tag{3.4}$$

As usual, the bounds 2/3 and 1/3 on the maximum probability of the verifier outputting 1 are taken as being representative of statistically distinguishable experiments. Methods for reducing errors in quantum verification procedures are discussed in the section following this one,

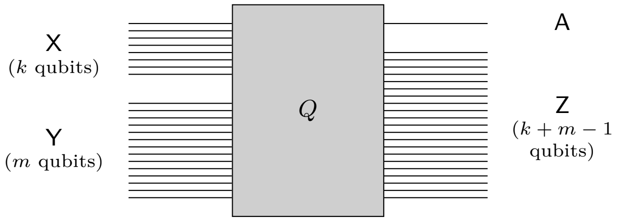


Figure 3.1: A unitary circuit implementing a verification procedure.

and these methods will allow the completeness and soundness bounds a and b to be taken as any functions exponentially close to 1 and 0 respectively. It is not known whether the completeness parameter can always be taken to equal 1 without changing the complexity class that results; a one-sided variant of QMA, denoted

$$\mathrm{QMA}_1 = \mathrm{QMA}_{1,\frac{1}{3}}, \tag{3.5}$$

clearly satisfies $\mathrm{QMA}_1 \subseteq \mathrm{QMA}$, but the two classes are not known to be equal.

We will sometimes identify the circuit encoding $V(x)$ with the channel that it implements in a self-explanatory way, writing $\omega(V(x))$ to mean $\omega(\Phi_x)$ for Φ_x being the channel implemented by $V(x)$. Along similar lines, we may write $\omega(V)$ to refer to the function whose value is $\omega(V(x)) = \omega(\Phi_x)$ for each input string x.

It will be instructive and useful later to consider the actions of a unitary circuit Q that purifies a circuit implementation of a channel Φ as above. As illustrated in Figure 3.1, such a circuit will take as input two registers: a k-qubit register X, which initially contains the state ρ representing the quantum proof, along with an m-qubit register Y initially containing the pure state $|0^m\rangle$, which represents the so-called *ancillary qubits* used by the circuit Q. (As explained in Section 2.3 the value of m will be at most linear in the number of gates required by the circuit implementation of Φ.) The output qubit of Φ will be named A, and the remaining $k+m-1$ qubits output by Q comprise a register Z.

3.1.2 Example: group non-membership

We will now describe an example of a computational problem, known as the *group non-membership problem*, that illustrates one potential way in which quantum proofs and verification procedures may gain advantages over classical proofs and verification procedures.[2] The group non-membership problem is perhaps most naturally described within the *black-box group* setting [23]. Here, one considers that there is an underlying finite group G_n that has been specified for each positive integer n, and elements of G_n are encoded as binary strings of length n (so that it must necessarily hold that $|G_n| \leq 2^n$). Not every string needs to encode a group element, but we will make the assumption (which is not always in place in the black-box group setting) that each group element has a unique binary string encoding. A *group oracle* is made available to perform the two group operations at unit cost, and in the quantum setting one assumes that the group oracle functions reversibly. For example, the group oracle may operate in the following way:

$$|g\rangle|h\rangle|b\rangle \mapsto \begin{cases} |hg\rangle|h\rangle|b\rangle & \text{if } b = 0 \\ |h^{-1}g\rangle|h\rangle|b\rangle & \text{if } b = 1, \end{cases} \tag{3.6}$$

assuming that $b \in \{0, 1\}$ and $g, h \in G_n$ are identified with their n-bit encodings. (We may assume that the group oracle acts as the identity operator when given a string that does not encode a group element.) When we consider the group non-membership problem below, it is to be assumed that group elements are given as n-bit strings and the group operations are determined by a fixed group oracle.

Positive results in the setting of black-box groups generally imply analogous positive results in concrete settings in which the group oracle can be implemented algorithmically. For the particular case at hand, the fact that the group non-membership problem is in QMA for black-box groups implies that it is also in QMA for concrete realizations of groups for which the unique encoding assumption is met and for which

[2] Although we do not go into the details, from the description in this section it will be evident that the group non-membership problem can be used to provide an *oracle separation* between the complexity classes MA and QMA.

the group operations can be implemented efficiently. Matrix groups over finite fields represent a fairly general class of examples in this category.

The group non-membership problem is as follows:

Group non-membership (GNM)

Input: Group elements $g_1, \ldots, g_m \in G_n$ and $a \in G_n$ (for some choice of n).

Yes: $a \notin \langle g_1, \ldots, g_m \rangle$.

No: $a \in \langle g_1, \ldots, g_m \rangle$.

Here, the notation $\langle g_1, \ldots, g_m \rangle$ means the subgroup of G_n generated by the elements $g_1, \ldots, g_m$.

Some might argue that the *subgroup non-membership problem* would be a more fitting name than the *group non-membership problem*, as the problem concerns membership in the subgroup $\langle g_1, \ldots, g_m \rangle$ rather than membership in the group G. It is, however, reasonable to view that $\langle g_1, \ldots, g_m \rangle$ is the group of interest in this problem, while G is a supergroup that happens to contain $g_1, \ldots, g_m$ and a.

Before discussing quantum proofs and verification procedures for this problem, it is fitting to mention what is known in the classical setting. It is known how to design an efficient classical verification procedure for the complementary problem to GNM, in which the yes-instances are those with $a \in \langle g_1, \ldots, g_m \rangle$ and the no-instances are those with $a \notin \langle g_1, \ldots, g_m \rangle$. Intuitively speaking, a short classical proof that $a \in \langle g_1, \ldots, g_m \rangle$ may consist of a list of instructions for obtaining a from $g_1, \ldots, g_m$ through the group operations. (One cannot simply give a sequence of elements selected from the set $\{g_1, \ldots, g_m, g_1^{-1}, \ldots, g_m^{-1}\}$ whose product is a, because such a list might need to be as long as the size of G_n itself—but a so-called *straight-line program* can be used instead. The *reachability lemma* of Babai and Szemerédi [23] guarantees that a short straight-line program to generate a from $g_1, \ldots, g_m$ must always exist when $a \in \langle g_1, \ldots, g_m \rangle$.) For some groups, including permutation groups, there exist efficient classical verification procedures for the GNM problem, but in the black-box group setting one can prove that no efficient classical verification procedure exists.

Using quantum proofs, however, the solution becomes elementary.

A quantum proof certifying that a given element a is not contained in the subgroup $K = \langle g_1, \ldots, g_m \rangle$, for a quantum verification procedure to be described shortly, is the state

$$|K\rangle = \frac{1}{\sqrt{|K|}} \sum_{g \in K} |g\rangle, \tag{3.7}$$

i.e., a uniform superposition over the elements in K. This state is independent of a, and will function correctly as a proof that $a \notin K$ for all such choices of a.

Now, if one truly had a copy of the state $|K\rangle$, it would not be difficult to test membership in K with bounded, one-sided error. If it is the case that $h \in K$ for some group element h, then the state

$$|hK\rangle = \frac{1}{\sqrt{|K|}} \sum_{g \in K} |hg\rangle \tag{3.8}$$

satisfies $|hK\rangle = |K\rangle$. On the other hand, if $h \notin K$, then $|hK\rangle \perp |K\rangle$. The following test, which we call the *controlled-unitary test*,[3] can be used to distinguish between the two cases:

<u>Controlled-unitary test</u>

Given: An n-qubit state $|\psi\rangle$ and a quantum circuit specifying an n-qubit unitary U.

Outcome: A classical bit that is 0 with probability

$$p = \frac{1 + \Re(\langle \psi | U | \psi \rangle)}{2}$$

and 1 with probability $1 - p$.

Procedure: See Figure 3.2.

The circuit described in Figure 3.2 implements the controlled-unitary test. The measurement illustrated in the figure is a standard basis measurement.

[3] We are not aware of a standard name for this test, and have selected a name for the sake of convenience. Irrespective of the name, it is a very commonly used test in quantum algorithms and complexity [45], and can be viewed as a low-precision form of the *eigenvalue estimation* procedure associated with Shor's algorithms for factoring and computing discrete logarithms.

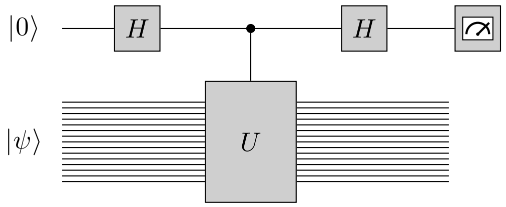


Figure 3.2: Circuit implementing the controlled-unitary test on state $|\psi\rangle$ and unitary U.

In the circuit, the unitary U is controlled by the top qubit, so that the action of the entire controlled-unitary gate is as follows:

$$|0\rangle|\psi\rangle \mapsto |0\rangle|\psi\rangle \quad \text{and} \quad |1\rangle|\psi\rangle \mapsto |1\rangle U|\psi\rangle. \tag{3.9}$$

After applying the second Hadamard gate, the state of the $n+1$ qubits is

$$\frac{|0\rangle+|1\rangle}{2} \otimes |\psi\rangle + \frac{|0\rangle-|1\rangle}{2} \otimes U|\psi\rangle, \tag{3.10}$$

and measuring the top qubit with respect to the computational basis produces the outcome 0 with probability

$$\frac{1}{4}\big\||\psi\rangle + U|\psi\rangle\big\|^2 = \frac{1+\Re(\langle\psi|U|\psi\rangle)}{2}. \tag{3.11}$$

In the setting of the group non-membership problem, the controlled-unitary test will be applied to the unitary operation M_a that multiplies (on the left) by a:

$$M_a|g\rangle = |ag\rangle. \tag{3.12}$$

This is a reversible (and therefore unitary) operation that can be implemented efficiently using the group oracle (or under the assumption that the group operations can be computed efficiently). By executing the controlled-unitary test on the state $|\psi\rangle = |K\rangle$ and the unitary $U = M_a$, the verifier will obtain a bit that is 0 with certainty if $a \in K$, and is uniformly distributed if $a \notin K$. If the test is run, leading to the outcome 1, then it must therefore be the case that $a \notin K$, and if $a \notin K$ then this event will happen with probability $1/2$.

There is, of course, a problem with treating the procedure just described as a verification procedure for the GNM problem, which is that one cannot trust that a given quantum proof really is the state $|K\rangle$. One could, for instance, substitute the state $|1\rangle$ (where 1 denotes the identity element in the group) for $|K\rangle$, which would lead to the incorrect conclusion that all non-identity elements of K are not contained in K.

One solution to this problem is to first choose a collection of elements $h_1, \ldots, h_N$ from K, and to sequentially run the verification procedure on $h_1, \ldots, h_N$ taken in place of a. In addition to the measurement outcome, the membership test outputs the qubits that initially contained the proof state, and after each test these qubits are supplied as the proof state to the next test. Naturally, if an element $h \in K$ is selected, one would expect that running the membership test on h would reveal that h is indeed contained in K—so if the membership test were to reveal that h is not contained in K, then the proof state must have been invalid and can be rejected. To see that this reasoning is valid not only for the first test, but for each of the tests in sequence, from (3.10) one may observe that conditioned on the test outputting 0 (which indicates a positive test for membership) for a particular choice of $h \in G_n$ and a proof state $|\psi\rangle$, the new proof state output by the test is

$$|\psi\rangle + M_h|\psi\rangle \quad \text{(normalized)}. \tag{3.13}$$

Because $M_h|K\rangle = |hK\rangle = |K\rangle$ holds when $h \in K$, the proof state $|K\rangle$ is unchanged by any such test.

Now suppose that the membership test is run on a given proof state $|\psi\rangle$ for a sequence of group elements $h_1, \ldots, h_N \in K$. Under the assumption that every one of the membership tests is consistent with the fact that $h_1, \ldots, h_N \in K$, the resulting proof state becomes

$$\sum_{k_1,\ldots,k_N \in \{0,1\}} M_{h_1}^{k_1} M_{h_2}^{k_2} \cdots M_{h_N}^{k_N} |\psi\rangle \quad \text{(normalized)}. \tag{3.14}$$

Regardless of whether or not the original state $|\psi\rangle$ was close to $|K\rangle$, the state above must be nearly invariant under left-multiplication by elements of K, provided that $h_1, \ldots, h_N$ were chosen well. In particular, if they represent a so-called *ε-uniform Erdős–Rényi generating sequence* for K, which means that the distribution of group elements

$$h_1{}^{k_1} h_2{}^{k_2} \cdots h_N{}^{k_N} \tag{3.15}$$

is ε-close to uniform over K for $(k_1, \ldots, k_N) \in \{0,1\}^N$ chosen uniformly at random, then the above state will function in approximately the same way as $|K\rangle$ with respect to the membership test described above. There is a classical randomized procedure due to Babai [20] that produces such a sequence with high probability.

As a remark, it is important to realize that the ability to uniformly sample from K is not known to allow one to efficiently prepare the state $|K\rangle$. By performing the sampling in superposition it is possible to prepare a state

$$|K'\rangle = \frac{1}{2^{R/2}} \sum_{r \in \{0,1\}^R} |r\rangle |h(r)\rangle, \tag{3.16}$$

where r denotes the randomness used by the sampling procedure and $h(r)$ denotes the sampled group element. In order to obtain a good approximation to $|K\rangle$ it would be necessary to "erase," or uncompute, the string r based on $h(r)$, and this may not be possible (for instance if $r \to h(r)$ is not one-to-one).

The final procedure is described in Figure 3.3. One of the error-reduction procedures to be described in the next section can be applied to this procedure to yield error bounded by 1/3, or even exponentially small error if desired.

3.2 Error reduction

In a classical setting, error reduction for polynomial-time bounded-error verification procedures can be handled in a straightforward way: the verification procedure is independently run multiple times on a given proof string, and is accepted if and only if the number of acceptances obtained by the individual runs meets or exceeds some suitably chosen threshold value. With respect to the analysis of such a method, no significant new challenges arise as compared with the standard analysis of error reduction for bounded-error algorithms.

1. Input n-bit encodings of group elements $g_1, \ldots, g_m, a \in G_n$ and a proof state contained in an k-qubit register X. Set ε to be a small positive constant (such as $\varepsilon = 1/16$).
2. Randomly select elements $h_1, \ldots, h_N$ so that, with probability at least $1 - \varepsilon$, the sequence $h_1, \ldots, h_N$ is an ϵ-uniform Erdős–Rényi generating sequence for $K = \langle g_1, \ldots, g_m \rangle$.
3. For each $j = 1, \ldots, N$, perform the controlled-unitary test described in Figure 3.2 on the state contained in X and the unitary M_{h_j}. If any of these tests results in the outcome 1, indicating non-membership, then reject.
4. Run the membership test for a on X and accept if the outcome is 1 (indicating non-membership), reject if the outcome is 0.

Figure 3.3: Verification procedure for the group non-membership problem

In the quantum setting, this strategy does not work—running a verification procedure on a quantum proof will generally change it, so the original proof may no longer be available after the first verification. For instance, if the measurement that is performed is a binary projective measurement $\{\Pi_0, \Pi_1\}$, then the post-measurement state is either $\Pi_0|\psi\rangle$ or $\Pi_1|\psi\rangle$ (properly normalized), so that repeating the measurement will result in the same outcome with certainty.

Two solutions to this problem are known. One solution is to request multiple, independent copies of the original proof, one for each run of the verification procedure. This requires an analysis to verify that no advantage may be found in correlating the registers that are supposed to contain these independent proof copies. Another solution, which has the advantage that it leads to a reduction in error without an increase in proof size, involves repeatedly running a unitary quantum circuit implementation of the verification procedure forward and backward in a manner reminiscent of Grover's quantum search algorithm [78]. The two methods are described in the subsections that follow.

3.2.1 Parallel error reduction

Assume that a verification procedure is given that takes as input a k-qubit register X and outputs a single qubit, which is measured with respect to the standard basis after being output. We will refer to the verification procedure as V, with the understanding that V refers to the actions of a verifier on some fixed input string that will not be named explicitly. The first strategy for error reduction for QMA operates as follows, for T and t being positive integers satisfying $t \leq T$ to be selected later.

1. Receive registers $\mathsf{X}_1, \ldots, \mathsf{X}_T$, each comprising k qubits.
2. Run V independently on each of the registers $\mathsf{X}_1, \ldots, \mathsf{X}_T$, and let $a_1, \ldots, a_T \in \{0, 1\}$ be the resulting binary-valued measurement outcomes.
3. Accept (i.e., output 1) if and only if $a_1 + \cdots + a_T \geq t$.

This strategy is illustrated in Figure 3.4, for $T = 5$, and where the circuit labelled F denotes the classical threshold value computation for some choice of t (which is not specified in the figure).

With the original verification procedure V, one can associate a binary-valued measurement, as was described in the previous section. Such a measurement may be represented by two $2^k \times 2^k$ dimensional positive semidefinite operators P_0 and P_1 satisfying $P_0 + P_1 = \mathbb{1}$. Along similar lines, one can associate a binary-valued measurement with the procedure described above—in this case described by two $2^{kT} \times 2^{kT}$ positive semidefinite operators Q_0 and Q_1 as follows:

$$\begin{aligned} Q_0 &= \sum_{\substack{a_1,\ldots,a_T \in \{0,1\} \\ a_1+\cdots+a_T < t}} P_{a_1} \otimes \cdots \otimes P_{a_T}, \\ Q_1 &= \sum_{\substack{a_1,\ldots,a_T \in \{0,1\} \\ a_1+\cdots+a_T \geq t}} P_{a_1} \otimes \cdots \otimes P_{a_T}. \end{aligned} \tag{3.17}$$

The spectra of these operators can be analyzed without difficulty due to the fact that P_0 and P_1 necessarily commute: $P_1 = \mathbb{1} - P_0$, and therefore

$$P_0 P_1 = P_0(\mathbb{1} - P_0) = P_0 - P_0^2 = (\mathbb{1} - P_0)P_0 = P_1 P_0. \tag{3.18}$$

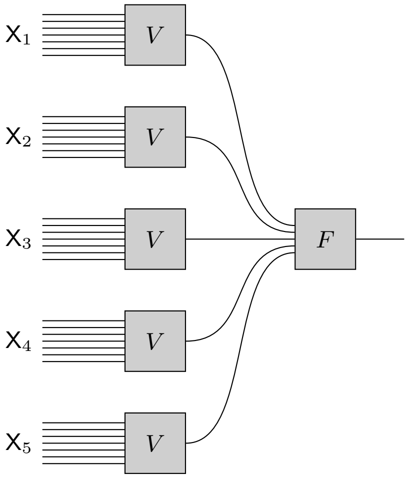


Figure 3.4: Parallel error reduction

In particular, supposing that $\{|\psi_1\rangle, \ldots, |\psi_K\rangle\}$ is an orthonormal basis of eigenvectors of P_1 having corresponding eigenvalues

$$\lambda_1(P_1) \geq \cdots \geq \lambda_K(P_1), \tag{3.19}$$

where $K = 2^k$, one has that an orthonormal basis of eigenvectors of Q_1 is obtained by tensoring these eigenvectors of P_1 (which are also eigenvectors of P_0) in all possible combinations:

$$\bigl\{|\psi_{j_1}\rangle \cdots |\psi_{j_T}\rangle \,:\, 1 \leq j_1, \ldots, j_T \leq K\bigr\}. \tag{3.20}$$

The value of the procedure that has been obtained is equal to the largest eigenvalue of Q_1, which is

$$\lambda_1(Q_1) = \sum_{\substack{a_1,\ldots,a_T \in \{0,1\} \\ a_1+\cdots+a_T \geq t}} \langle\psi_1|P_{a_1}|\psi_1\rangle \cdots \langle\psi_1|P_{a_T}|\psi_1\rangle. \tag{3.21}$$

One now sees that an optimal choice of a proof for the procedure is one in which each of the registers $\mathsf{X}_1, \ldots, \mathsf{X}_T$ is independently prepared in the optimal proof state $|\psi_1\rangle$ for the original verification procedure V.

At this point, a suitable selection of T and t provides for an exponential reduction in error, based on standard bounds on the tails of binomial distributions. For instance, if a given verifier V has completeness and soundness probability bounds a and b, respectively, before error reduction, then choosing

$$T = \frac{r}{(a-b)^2} \quad \text{and} \quad t = \left\lceil \frac{a+b}{2} T \right\rceil \tag{3.22}$$

results in a new verifier V' having completeness and soundness probabilities exponentially close to 1 and 0, respectively, with respect to a chosen error parameter r.

3.2.2 Witness-preserving error reduction

The second method for error reduction of QMA is sequential, but has the advantage that no increase in proof size is required as the error is reduced. Before describing the method in precise terms, it will be helpful to first discuss the intuition behind it.

Suppose that a verification procedure V, taking as input a k-qubit register X and outputting a single qubit, is given. As described in the previous section, one can consider a unitary circuit implementation Q of V, which is a unitary procedure taking as input two registers: the k-qubit proof register X along with an m-qubit ancillary register Y. The output qubits of Q are split between a single qubit register A, which corresponds to the output qubit of V, along with an $(m+k-1)$-qubit register Z (which could be viewed as a "garbage" register, although it will not be treated as garbage by the error reduction procedure).

Now, suppose that a pure state $|\psi\rangle$ on k qubits has been selected as a quantum proof, and Q is run on the input state $|\psi\rangle|0^m\rangle$. The resulting state may be expressed as

$$Q|\psi\rangle|0^m\rangle = \sqrt{p_0(\psi)}\,|0\rangle|\phi_0(\psi)\rangle + \sqrt{p_1(\psi)}\,|1\rangle|\phi_1(\psi)\rangle, \tag{3.23}$$

where the numbers $p_0(\psi)$ and $p_1(\psi)$ represent the probabilities for a measurement of the register A with respect to the standard basis to give the outcomes 0 and 1, respectively. Measuring the output qubit gives a single sample, 0 or 1, from a Bernoulli distribution $(p_0(\psi), p_1(\psi))$ that one would ideally like to sample multiple times. It is natural to ask if the original proof state $|\psi\rangle$ can be recovered, so as to allow for more samples, and perhaps the first thing one would be inclined to do to try to recover $|\psi\rangle$ is to run Q in reverse. This yields one of the two states

$$Q^*|0\rangle|\phi_0(\psi)\rangle \quad \text{or} \quad Q^*|1\rangle|\phi_1(\psi)\rangle, \tag{3.24}$$

depending on whether the outcome of the first measurement was 0 or 1.

It is not clear that the states (3.24) allow for a reconstruction of $|\psi\rangle$, or if they are useful at all for that matter—but under a simple assumption on the original quantum proof state $|\psi\rangle$, a recovery of $|\psi\rangle$ will generally be possible. The assumption is that $|\psi\rangle$ is a common *eigenvector* of the two measurement operators P_0 and $P_1 = \mathbb{1} - P_0$ corresponding to the binary-valued measurement implemented by V. Expanding on (3.1), these measurement operators may be expressed explicitly in terms of Q as follows:

$$\begin{aligned} P_0 &= (\mathbb{1} \otimes \langle 0^m|)Q^*(|0\rangle\langle 0| \otimes \mathbb{1})Q(\mathbb{1} \otimes |0^m\rangle), \\ P_1 &= (\mathbb{1} \otimes \langle 0^m|)Q^*(|1\rangle\langle 1| \otimes \mathbb{1})Q(\mathbb{1} \otimes |0^m\rangle). \end{aligned} \tag{3.25}$$

It may be noted that the assumption that $|\psi\rangle$ is an eigenvector of these operators is not a significant restriction; a choice of $|\psi\rangle$ that maximizes $p_1(\psi)$ will necessarily be an eigenvector of these operators, as discussed previously.

Now, to see why the condition that $|\psi\rangle$ is an eigenvector of these operators is relevant, one may consider the states (3.24); as they can be analyzed similarly, we will consider $Q^*|0\rangle|\phi_0(\psi)\rangle$. Imagine that the m-qubit ancillary register Y for this state is measured with respect to the standard basis, and let us focus on the case in which the measurement outcome is the all-zero string 0^m. Before normalization, the state remaining in the register X is

$$\begin{aligned} &(\mathbb{1} \otimes \langle 0^m|)Q^*|0\rangle|\phi_0(\psi)\rangle \\ &\qquad = \frac{1}{\sqrt{p_0(\psi)}}(\mathbb{1} \otimes \langle 0^m|)Q^*(|0\rangle\langle 0| \otimes \mathbb{1})Q|\psi\rangle|0^m\rangle \\ &\qquad = \frac{1}{\sqrt{p_0(\psi)}}P_0|\psi\rangle \\ &\qquad = \sqrt{p_0(\psi)}|\psi\rangle. \end{aligned} \tag{3.26}$$

Based on this calculation, one concludes that the all-zero measurement outcome occurs with probability $p_0(\psi)$, and conditioned on this outcome the original proof state $|\psi\rangle$ is available in the register X to be tested again. For the state $Q^*|1\rangle|\phi_1(\psi)\rangle$, the probability of obtaining the all-zero measurement outcome 0^m is $p_1(\psi)$, and again the original proof state $|\psi\rangle$ is available in the register X to be tested again.

Of course, obtaining the all-zero measurement outcome is fortuitous, and the possibility of obtaining a different measurement outcome would seem to be a potential problem. Nevertheless, the indication that there is a possibility to obtain further samples is encouraging. As it turns out, the potential difficulty represented by the possibility to not obtain the all-zero measurement outcome can be overcome by making a different choice of the measurement on the ancillary register. Rather than measuring with respect to the standard basis, we will measure with respect to a binary-valued projective measurement having measurement operators

$$\Delta_0 = |0^m\rangle\langle 0^m| \quad \text{and} \quad \Delta_1 = \mathbb{1} - |0^m\rangle\langle 0^m|. \tag{3.27}$$

Let us describe the actual error reduction procedure in precise terms before proceeding further with the discussion. The procedure is described in Figure 3.5, where $1 \leq t \leq T$ are two arbitrary integers and it is to be assumed that the register X initially contains the proof state $|\psi\rangle$ and Y is initialized to the state $|0^m\rangle$. The procedure is illustrated in Figure 3.6 for the case $T = 3$. The box labelled R represents the classical computation performed in step 2, and it is to be assumed that all of the qubits aside from those included in X are initialized to the $|0\rangle$ state before the circuit in the figure is executed.

To analyze the procedure, we can extend the analysis that has been started above: we have determined, under the assumption that $|\psi\rangle$ is an eigenvector of the measurement operators P_0 and P_1, that

$$Q^*|0\rangle|\phi_0(\psi)\rangle = \sqrt{p_0(\psi)}|\psi\rangle|0^m\rangle + \sqrt{p_1(\psi)}|\gamma(\psi)\rangle, \tag{3.30}$$

where $|\gamma(\psi)\rangle$ is defined as

$$|\gamma(\psi)\rangle = \frac{1}{\sqrt{p_1(\psi)}}(\mathbb{1} \otimes \Delta_1)Q^*|0\rangle|\phi_0(\psi)\rangle. \tag{3.31}$$

(We will ignore the possibility that either of $p_0(\psi)$ or $p_1(\psi)$ is zero, which can be handled as a simple special case.) Because Q is unitary,

1. Repeat the following for each $j = 1, \ldots, T$:
   a. Apply the unitary circuit Q to (X, Y), obtaining (A, Z).
   b. Perform a standard basis measurement on A, letting $a_j \in \{0, 1\}$ denote the outcome.
   c. Apply the unitary circuit Q^* to (A, Z), obtaining (X, Y).
   d. Perform the projective measurement $\{\Delta_0, \Delta_1\}$ on Y, where

$$\Delta_0 = |0^m\rangle\langle 0^m| \quad \text{and} \quad \Delta_1 = \mathbb{1} - |0^m\rangle\langle 0^m|, \tag{3.28}$$

   letting $b_j \in \{0, 1\}$ denote the outcome.

2. Define $c_1, \ldots, c_{2T} \in \{0, 1\}$ as follows:

$$\begin{aligned} c_1 &= a_1, \\ c_{2j} &= a_j \oplus b_j \quad (\text{for } j = 1, \ldots, T) \\ c_{2j-1} &= a_j \oplus b_{j-1} \quad (\text{for } j = 2, \ldots, T). \end{aligned} \tag{3.29}$$

   Accept if $c_1 + \cdots + c_{2T} \geq 2t$, reject otherwise.

Figure 3.5: Witness-preserving error reduction procedure for QMA.

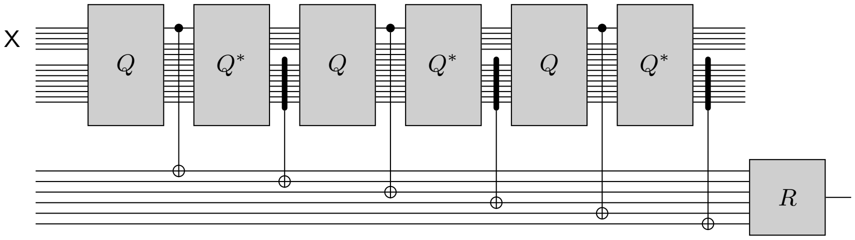


Figure 3.6: Witness-preserving error reduction

we also have

$$Q^*\Big(\sqrt{p_0(\psi)}|0\rangle|\phi_0(\psi)\rangle + \sqrt{p_1(\psi)}|1\rangle|\phi_1(\psi)\rangle\Big) = |\psi\rangle|0^m\rangle, \tag{3.32}$$

which provides us with enough information to determine the state of the registers in the procedure after every measurement, conditioned on every value. In particular, we have

$$\begin{aligned} Q^*|0\rangle|\phi_0(\psi)\rangle &= \sqrt{p_0(\psi)}|\psi\rangle|0^m\rangle + \sqrt{p_1(\psi)}|\gamma(\psi)\rangle, \\ Q^*|1\rangle|\phi_1(\psi)\rangle &= \sqrt{p_1(\psi)}|\psi\rangle|0^m\rangle - \sqrt{p_0(\psi)}|\gamma(\psi)\rangle, \end{aligned} \tag{3.33}$$

and

$$\begin{aligned} Q|\psi\rangle|0^m\rangle &= \sqrt{p_0(\psi)}|0\rangle|\phi_0(\psi)\rangle + \sqrt{p_1(\psi)}|1\rangle|\phi_1(\psi)\rangle, \\ Q|\gamma(\psi)\rangle &= \sqrt{p_1(\psi)}|0\rangle|\phi_0(\psi)\rangle - \sqrt{p_0(\psi)}|1\rangle|\phi_1(\psi)\rangle. \end{aligned} \tag{3.34}$$

From these equations it follows that

$$\begin{aligned} &\Pr(b_j = 0|a_j = 0) = p_0(\psi), \quad \Pr(a_{j+1} = 0|b_j = 0) = p_0(\psi), \\ &\Pr(b_j = 1|a_j = 0) = p_1(\psi), \quad \Pr(a_{j+1} = 1|b_j = 0) = p_1(\psi), \\ &\Pr(b_j = 0|a_j = 1) = p_1(\psi), \quad \Pr(a_{j+1} = 0|b_j = 1) = p_1(\psi), \\ &\Pr(b_j = 1|a_j = 1) = p_0(\psi), \quad \Pr(a_{j+1} = 1|b_j = 1) = p_0(\psi), \end{aligned} \tag{3.35}$$

and therefore

$$\Pr(c_j = 0) = p_0(\psi) \quad \text{and} \quad \Pr(c_j = 1) = p_1(\psi) \tag{3.36}$$

for every $j = 1, \ldots, 2T$. Figure 3.7 illustrates the transition probabilities expressed by the equations (3.35). A similar choice of T and t to the parallel error reduction procedure yields an exponential reduction of error.

One must still consider the behavior of the witness-preserving error reduction procedure when the given proof state $|\psi\rangle$ is not an eigenvector of the original measurement operators P_0 and P_1, but this is easily done and we will omit the details. Intuitively speaking, the procedure operates independently on each eigenvector, so that an arbitrary proof state must always behave as if it were a random mixture of pure state

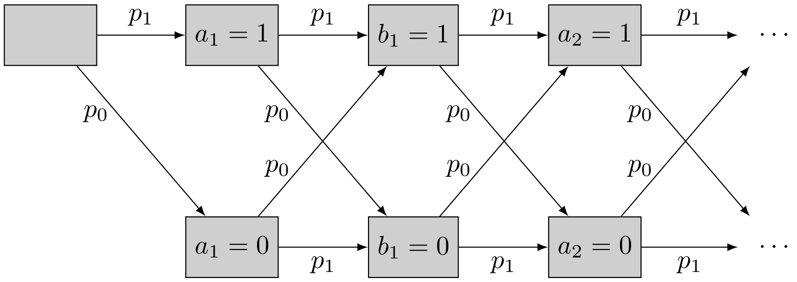


Figure 3.7: Transition probabilities between different measurement outcomes in the witness-preserving error reduction procedure.

eigenvectors. Alternatively, one can argue that the measurement operators resulting from the witness-preserving error reduction procedure share a common set of eigenvectors with the original measurement operators P_0 and P_1, so there is no loss of generality in considering the behavior of the procedure on one of these eigenvectors.

3.2.3 QMA $\subseteq$ PP

Consider a problem $A = (A_{\text{yes}}, A_{\text{no}})$ in QMA and consider a verifier for A for which the completeness and soundness probabilities have been amplified to $a \geq 3/4$ and $b \leq 2^{-(k+2)}$ respectively, where k is the number of qubits of the proof. That this is possible follows from the witness-preserving error reduction procedure described in the previous section. For a fixed input x, the verifier's maximum probability of producing the outcome 1 can be expressed as the largest eigenvalue of the k-qubit measurement operator P_1 defined in (3.1). Thus

$$\begin{aligned} x \in A_{\text{yes}} &\implies \operatorname{Tr}(P_1) \geq \lambda_1(P_1) \geq \frac{3}{4}, \\ x \in A_{\text{no}} &\implies \operatorname{Tr}(P_1) \leq 2^k \lambda_1(P_1) \leq \frac{1}{4}. \end{aligned} \tag{3.37}$$

It follows that this problem can be decided by an unbounded-error quantum polynomial-time procedure as follows. The procedure performs the same measurement as the QMA verifier but replaces the witness by the completely mixed state on k qubits, created, for instance, as half of k EPR pairs.[4] This procedure will output 1 (i.e., accept) with probability at least $\frac{3}{4}2^{-k}$ in case $x \in A_{\text{yes}}$, and accept with probability at most $\frac{1}{4}2^{-k}$ in case $x \in A_{\text{no}}$.

The above already shows QMA $\subseteq$ PQP, the class of problems that have unbounded-error quantum polynomial-time algorithms. The inclusion can be re-stated as QMA $\subseteq$ PP, the class of problems that have unbounded-error classical randomized polynomial-time algorithms, because the equality PQP = PP holds in general. (Indeed, the two best-known proofs of the containment BQP $\subseteq$ PP, due to Adleman, DeMarrais, and Huang [4] and Fortnow and Rogers [63], do not rely on the assumption of bounded error for BQP.)

3.3 Complete promise problems

The class QMA has an interesting collection of complete promise problems having connections to a range of problems motivated by quantum information theory, condensed-matter physics, and quantum chemistry. While it would be premature to compare the QMA-complete promise problems to the rich collection of known NP-complete problems, either with respect to the number of such problems or their broad relevance within science and engineering, there is strong and active interest in the notion of QMA-completeness, and the list of known QMA-complete problems is growing steadily. In this section we present just a couple of examples of QMA-complete problems, referring the reader to the chapter notes for pointers to further work on the subject.

Not surprisingly, it is possible to translate the definition of QMA directly into a complete promise problem. The following problem statement represents one way of doing this:

(a, b)-Quantum Circuit Satisfiability ((a, b)-QCS)

Input: A quantum circuit specifying a channel Φ with k input qubits and 1 output qubit.

Yes: There exists a k-qubit state ρ such that $\langle 1|\Phi(\rho)|1\rangle \geq a$.

No: For all k-qubit states ρ, $\langle 1|\Phi(\rho)|1\rangle \leq b$.

[4] An EPR pair is a pair of qubits in the joint state $(|00\rangle + |11\rangle)/\sqrt{2}$.

It follows immediately from the definition of QMA and the error amplification procedure discussed in the previous section that (a, b)-QCS is complete for QMA for any a and b satisfying

$$2^{-\operatorname{poly}(n)} \leq b < a \leq 1 - 2^{-\operatorname{poly}(n)} \quad \text{and} \quad b - a \geq \operatorname{poly}^{-1}(n). \tag{3.38}$$

Along similar lines, $(1, b)$-QCS is QMA_1-complete for any b satisfying

$$2^{-\operatorname{poly}(n)} \leq b \leq 1 - \operatorname{poly}^{-1}(n). \tag{3.39}$$

Several of the classes of quantum interactive proofs to be discussed in later chapters have complete problems of the following form: "Given quantum channels Φ_0 and Φ_1, determine whether or not Φ_0 and Φ_1 are close." Problems of increasing complexity are obtained by varying the precise way in which closeness of the channels is measured, as well as the type of channels that are considered. The following problem is QMA-complete for the same range of parameters (a, b) as quantum circuit satisfiability.

(a, b)-Non-Identity Check ((a, b)-NIC)

Input: A unitary channel $\Phi : \rho \rightarrow U\rho U^*$ implemented by a quantum circuit on k qubits.

Yes: $\frac{1}{2}\|\Phi - \mathbb{1}\|_\diamond \geq a$.

No: $\frac{1}{2}\|\Phi - \mathbb{1}\|_\diamond \leq b$.

The proof that this problem lies in QMA for any $b - a > \operatorname{poly}^{-1}(n)$ follows by using the quantum phase estimation procedure, and QMA-completeness can be shown by a simple reduction from (a, b)-QCS [101].[5]

The problems described in the subsections that follow represent complete problems for QMA that are, in some sense, more interesting than QCS—the problems are interesting and well-motivated in their own right, and they are not simply rephrasings of the definition of QMA.

[5] The problem considered in [101] is specified in a slightly different form, but essentially the same reduction can be used to show completeness of (a, b)-NIC.

3.3.1 The local Hamiltonian problem

The local Hamiltonian problem, introduced by Kitaev in the late 1990s, was the first problem to be shown to be complete for QMA. Intuitively speaking, the proof that this problem is complete for QMA is similar in spirit to the classical proof of the Cook–Levin theorem (but different at a technical level for reasons to be discussed below).

The local Hamiltonian problem captures the notion of a quantum constraint satisfaction problem, using the language of quantum many-body systems in condensed-matter physics. In this setting, a Hamiltonian H is a Hermitian operator representing the total energy of a physical system. A typical Hamiltonian in classical mechanics is $H = p^2/2m$, where $p = mv$ is the momentum: this Hamiltonian characterizes the kinetic energy of the system. The eigenvectors of H represent possible states of the system, and the associated (real) eigenvalues specify the energy. The problem of determining the smallest eigenvalue is of particular interest, as it represents the energy of the equilibrium state at zero temperature. The smallest eigenvalue is also called the *ground state energy*, and the associated eigenvector the *ground state*.

Given a Hamiltonian H acting on the Hilbert space corresponding to n qubits, it is said that H is *k-local* if it admits a decomposition

$$H = \sum_{i=1}^{m} H_i, \tag{3.40}$$

where each H_i can be written as a tensor product of an operator $\tilde{H}_i$, acting on a subset $S_i \subseteq \{1, \dots, n\}$ of at most $|S_i| \leq k$ qubits, with the identity acting on the remaining qubits. Many physical systems can be characterized by local Hamiltonians: the locality reflects the fact that the system's energy only depends (at least to some approximation) on local interactions, such as particle-field interactions (which are 1-local) and pairwise particle-particle interactions (which are 2-local). While a general Hamiltonian on n qubits may require a number of bits exponential in n to be fully specified, a k-local Hamiltonian can be described using $\mathrm{poly}(n, m, 2^k)$ bits by listing the local terms $\tilde{H}_1, \dots, \tilde{H}_m$ along with a specification of the subsets $S_1, \dots, S_m$ of qubits they act upon.

(k, a, b)-Local Hamiltonian ((k, a, b)-LH)

Input: A k-local Hamiltonian $H = \sum_{i=1}^{m} H_i$ acting on n qubits. For every $i \in \{1, \dots, m\}$, H_i acts nontrivially on at most k qubits and satisfies $0 \leq H_i \leq \mathbb{1}$.

Yes: There exists a pure state $|\psi\rangle$ on n qubits such that $\frac{1}{m}\langle\psi|H|\psi\rangle \leq 1 - a$.

No: For all pure states $|\psi\rangle$ on n qubits, $\frac{1}{m}\langle\psi|H|\psi\rangle \geq 1 - b$.

The requirement that each Hamiltonian term satisfies $0 \leq H_i \leq \mathbb{1}$ provides a convenient normalization of the problem. The case of non-positive H_i can always be reduced to $H_i \geq 0$ by adding a constant shift.

The problem (k, a, b)-LH is complete for QMA for any $k \geq 2$ and for a and b such that $b < a \leq 1 - \text{poly}^{-1}(n)$ and $a - b = \Theta(\text{poly}^{-1}(n))$. We write k-LH when it is understood that a, b are set to such values.

A QMA verifier for an instance of (k, a, b)-LH can be constructed as follows: the verifier selects one of the m local terms H_i uniformly at random, performs the measurement $\{\mathbb{1} - H_i, H_i\}$ on the n-qubit quantum proof, and outputs 1 if and only if the first outcome is obtained. In the case of a yes-instance, setting the proof to be a state $|\psi\rangle$ such that $\frac{1}{m}\langle\psi|H|\psi\rangle \leq 1 - a$ leads to acceptance with probability at least a. In the case of a no-instance, no proof can lead to the verifier outputting 1 with probability larger than b. Under the assumption that $a - b > \text{poly}^{-1}(n)$, the difference between the two probabilities can be amplified to a constant, showing that the problem is in QMA.

Establishing completeness of k-LH for QMA requires more work. The original proof, due to Kitaev [113], shows this for $k = 5$, with subsequent improvements bringing the locality down to 2-local terms.

To see the difficulty, consider first the analogous task in the classical setting. In order to reduce from an instance of circuit-SAT to a local constraint satisfaction problem, it is customary to introduce auxiliary variables associated with each wire in the circuit, as well as local constraints that enforce that the variables corresponding to the input and output wires for any gate in the circuit are related as required by the gate. A direct translation of this reduction does not work in the case of a circuit acting on quantum states, even for the most trivial of circuits. This is due in part to the fact that the equality of two pure quantum states cannot be checked locally. For example, the states $\frac{1}{\sqrt{2}}(|0\cdots 0\rangle + |1\cdots 1\rangle)$ and $\frac{1}{\sqrt{2}}(|0\cdots 0\rangle - |1\cdots 1\rangle)$ are orthogonal, but have the same reduced density operator as soon as just one of the qubits is discarded.

It is therefore necessary to consider a different reduction. Suppose we are given a quantum circuit Q with T gates acting on n input qubits. The idea is to require the proof $|\psi\rangle$ to be a uniform superposition of "snapshot states" $|\psi_i\rangle$, where for $i = 0, \dots, T$, $|\psi_i\rangle$ is the state of all qubits in the circuit after the i-th gate has been applied. More precisely, the expected quantum proof has the form

$$|\psi\rangle = \frac{1}{\sqrt{T+1}} \sum_{i=0}^{T} |i\rangle|\psi_i\rangle, \tag{3.41}$$

where the first register is called the *clock* register. The state $|\psi\rangle$ is sometimes called the *history state* of the computation.

To check that an arbitrary state is of the form (3.41), it is possible to define a local Hamiltonian term H_i for each gate in the circuit, acting only on the clock register and the qubits of $|\psi\rangle$ on which the gate operates, such that $\langle\psi|H_i|\psi\rangle = 0$ if and only if $|\psi_{i+1}\rangle = U_i|\psi_i\rangle$, where U_i is the local unitary implemented by the i-th gate in Q. In addition, there should be terms to enforce that $|\psi_0\rangle$ is initialized correctly (each ancilla qubit is set to $|0\rangle$), and $|\psi_T\rangle$ is an accepting state (the output qubit is in state $|1\rangle$). An analysis of this construction shows QMA-completeness of (k, a, b)-LH for $k = O(\log n)$, the size of the clock register plus the locality of any gate in the circuit, and a, b inverse polynomial in T. A different implementation of the clock, relying on a unary, rather than binary, encoding, can be used to devise a 5-local Hamiltonian. Different techniques, such as ones based on the use of perturbation theory, can be used to reduce the locality to 2.

It is not known whether error amplification can be performed for k-LH: the natural amplification procedure would replace H by H^t for a sufficiently large positive integer t, which would lead to a corresponding increase in the parameter k. The *Quantum PCP Conjecture* posits that the problem (k, a, b)-LH remains QMA-complete for parameters

a and b that are separated by a constant. Aside from being a natural quantum analogue of the classical PCP theorem, this conjecture has implications for the study of entanglement in low-energy eigenstates of local Hamiltonians, and it has attracted interest from theoretical computer scientists and condensed matter theorists alike.

3.3.2 Quantum 3-SAT

The special case of 3-LH where all of the local terms H_i are rank-one projections is called *quantum* 3-*SAT*. Each local term defines a 1-dimensional subspace of invalid configurations within the 8-dimensional subspace on which H_i acts non-trivially.

b-Quantum 3-SAT (b-Q3SAT)

Input: A 3-local Hamiltonian $H = \sum_{i=1}^{m} P_i$ on n qubits, where each P_i is a rank-one projection acting non-trivially on at most 3 qubits.

Yes: There exists an n-qubit state $|\psi\rangle$ such that $\langle\psi|H|\psi\rangle = 0$.

No: For all states $|\psi\rangle$ on n qubits, $\frac{1}{m}\langle\psi|H|\psi\rangle \geq 1 - b$.

For any $b = 1 - \Theta(\text{poly}^{-1}(n))$ the problem b-Quantum 3-SAT is complete for QMA_1. Containment in QMA_1 follows as for k-LH, except that in order to guarantee completeness 1 it is important that the measurement $\{\mathbb{1} - P_i, P_i\}$ can be implemented perfectly by the QMA verifier. Depending on how the operator P_i is specified, and which universal gate set is allowed for the verifier's circuit, this may not be the case. For our purposes it is sufficient to point out that the required compatibility can always be ensured by a careful choice of definitions.

The proof that b-Q3SAT is QMA_1-hard follows the same general outline as sketched previously for 3-LH, but additional difficulties arise from the requirement of perfect completeness—highly nontrivial modifications to the construction of the clock register are required for the proof.

3.3.3 Consistency of Density Operators

We end this section with a problem indicative of the diversity of problems that have been shown complete for QMA.

(k, b)-Consistency of Density Operators

Input: Density operators $\sigma_1, \ldots, \sigma_m$ on at most k qubits each, together with subsets $S_1, \ldots, S_m \subseteq \{1, \ldots, n\}$ such that $|S_i| \leq k$ for each i.

Yes: There exists an n-qubit density operator ρ such that $\operatorname{Tr}_{\{1,\ldots,n\}\backslash S_i}(\rho) = \sigma_i$ for every $i \in \{1, \ldots, m\}$.

No: For every n-qubit density operator ρ, it holds that

$$\left\| \operatorname{Tr}_{\{1,\ldots,n\}\backslash S_i}(\rho) - \sigma_i \right\|_1 \geq b$$

for at least one choice of $i \in \{1, \ldots, m\}$.

This problem is known to be QMA-complete with respect to Cook reductions (i.e., polynomial-time Turing reductions). Whether or not the problem is also QMA-complete with respect to Karp reductions is an interesting open problem.

3.4 Variations on QMA

In this section we discuss a few noteworthy variants of QMA. In each case we consider changes to the definition of QMA that either enhance the power of the verifier or restrict the allowable set of quantum proof states it may receive. In some cases these changes are superficial, leading to an equivalent definition of the class QMA, and in other cases it appears that the changes have a substantial effect. The following variants will be discussed:

1. *Super-verifiers.* Super-verifiers are granted the ability to estimate the probability $\langle 1|V(\rho)|1\rangle$ with good accuracy, for any chosen QMA verifier V, and for ρ being the proof received from the prover. This definition leads to a class QMA_+, which is equal to QMA—a fact that is sometimes useful for proving problems to be contained in QMA.

2. *Subset-state proofs.* Here one restricts the completeness condition of QMA so that a valid quantum proof on yes-inputs to the problem must be a uniform superposition over a subset of computational basis states, disallowing any phase differences between the coefficients of these basis states. This leads to a class SQMA, which is also equal to QMA.
3. *Trusted advice.* The notion of quantum advice is similar in spirit to quantum proofs, except that the advice state can be *trusted*—but it must also be the same state for all input strings of a given length. This leads to the class BQP/qpoly. This is a non-uniform class and is therefore different from QMA, but the precise relationship of these classes is not clear. It is known, however, that BQP/qpoly $\subseteq$ QMA/poly; trusted quantum advice can be simulated by trusted classical advice together with an untrusted quantum proof.
4. *Unentangled quantum proofs.* Here the verifier is granted the promise that the quantum proof splits into two unentangled parts, $|\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle$, with respect to a fixed bipartition of the qubits comprising the proof. This leads to the class QMA(2). While it is an open question whether or not QMA(2) and QMA are equal, some evidence exists to suggest that QMA(2) is larger than QMA.
5. *Classical proofs.* Finally we consider the case in which a classical proof is supplied to a quantum verifier, which leads to the class QCMA. There are arguments both in favor of and against this class being strictly smaller than QMA.

3.4.1 Variations equivalent to QMA

By definition, a promise problem is in QMA if yes-inputs have quantum proofs that convince the verifier V to accept with high probability, whereas for no-inputs there is no such witness. Thus, it is assumed that yes- and no-inputs can be distinguished by the *maximum* probability $\omega(V)$ with which the verifier can be made to output 1 over all possible input states.

In this section two independent modifications to this scenario are considered that lead to alternative ways of defining QMA. In the first modification, it is assumed that a prover aims not necessarily to maximize the probability that the verifier accepts, but to obtain a certain target probability $p = \langle 1|V(\rho)|1\rangle$ for a given verifier V. In the second modification, the proof is restricted to the subset-state form suggested above for yes-instances of the problem. The fact that these modifications yield equivalent definitions of QMA has a positive interpretation, in the sense that they show that QMA verifiers are more powerful than immediately apparent. It may also be helpful to rely on either promise when reasoning about the class QMA.

Arbitrary probabilities

The error reduction procedure for QMA described in Section 3.2.2 shows that, for a given verifier V, it is possible to design a verifier V' that uses V to obtain an accurate estimate of the maximum probability $\omega(V)$ of V to output 1, given an optimal proof state for V. This procedure crucially relies on $\omega(V)$ being defined as the maximum probability for V to accept. (More generally, a similar process would work for any eigenvalue of the measurement operators (3.1) associated with the verifier, given a corresponding eigenvector for that eigenvalue.) Suppose instead that a new verifier V' is granted the power to estimate, to within inverse polynomial precision, the probability $\langle 1|V(\rho)|1\rangle$ for a verifier V to accept an arbitrary state ρ provided as the proof, for V being any QMA verifier selected by V'. Thus, V' can decide, for any target probability p, whether there exists a state ρ such that $\langle 1|V(\rho)|1\rangle \approx p$. It is natural to question whether this ability allow V' to decide problems beyond those contained in QMA.

More formally, one defines a *super-verifier* as a triple (V, p, η) consisting of a polynomial-time mapping V from strings $x \in \Sigma^*$ to QMA verifiers $V(x)$, along with polynomial-time computable functions $p : \Sigma^* \to [0, 1]$ and $\eta : \mathbb{N} \to (0, 1]$ for which η is larger than the inverse of some polynomially bounded function.

Definition 3.2. A promise problem $(A_{\text{yes}}, A_{\text{no}})$ is contained in QMA_+ if there exists a polynomial q and a super-verifier (V, p, η) such that the following conditions hold:

1. For every $x \in A_{\text{yes}}$, there exists a state ρ such that
$$|\langle 1|V(x)(\rho)|1\rangle - p(x)| \leq \eta(|x|). \tag{3.42}$$

2. For every $x \in A_{\text{no}}$, and for every state ρ,
$$|\langle 1|V(x)(\rho)|1\rangle - p(|x|)| \geq \eta(|x|) + 1/q(|x|). \tag{3.43}$$

It is evident that QMA $\subseteq$ QMA$_+$: given a QMA$_{\frac{3}{4},\frac{1}{4}}$ verifier V we can define an equivalent super-verifier $(V, 3/4, 1/4)$. More interesting is that the reverse inclusion also holds, so that
$$\text{QMA}_+ = \text{QMA}. \tag{3.44}$$
To see that this is so, let (V, p, η) be a given super-verifier. For some sufficiently large (but polynomially bounded) number k, consider a verifier V' that expects k copies of a quantum proof for V, measures each copy independently according to the binary-valued measurement associated with V, and accepts if and only if the fraction r of outcomes 1 obtained satisfies $|r - p| \leq \eta + 1/(2q)$. Provided that k is chosen as a sufficiently large multiple of q, it follows from a Chernoff-type bound that every input $x \in A_{\text{yes}}$ has a proof that is accepted by V' with probability exponentially close to 1. To establish soundness, suppose that the QMA$_+$ super-verifier V is such that the inequality $|\langle 1|V(\rho)|1\rangle - p| \geq \eta + 1/q$ holds for all states ρ. Let σ be an arbitrary witness for V' and define a state ρ by taking the average of the k reduced density operators of σ on each of the registers on which V' executes the circuit specified by V. From this definition it follows that $\langle 1|V(\rho)|1\rangle$ coincides with the expectation of r, the fraction of acceptances that V' witnesses when making the k successive measurements of V on the corresponding registers of σ. There are two cases:

(i) $\langle 1|V(\rho)|1\rangle \leq p - \eta - 1/q$. By Markov's inequality the probability that $r \geq p - \eta - 1/(2q)$ is at most $1 - 1/(2q)$.

(ii) $\langle 1|V(\rho)|1\rangle \geq p + \eta + 1/q$. In this case Markov's inequality applied to $1 - r$ shows that the probability that $r \leq p + \eta + 1/(2q)$ is at most $1 - 1/(2q)$.

In both cases, V' rejects with probability at least $1/(2q)$. Thus, the soundness probability is bounded away from 1 by an inverse polynomial, and the gap between the completeness and soundness probabilities is large enough that it can be amplified using the methods described in Section 3.2.

Subset-state witnesses

Given an integer n and a nonempty set $S \subseteq \Sigma^n$, define the n-qubit *subset state* $|S\rangle$ as

$$|S\rangle = \frac{1}{\sqrt{|S|}} \sum_{x \in S} |x\rangle. \tag{3.45}$$

A promise problem $(A_{\text{yes}}, A_{\text{no}})$ is said to lie in the class SQMA if for every $x \in A_{\text{yes}}$ there is a state of the form (3.45) that convinces the verifier to accept with probability at least $2/3$, while for $x \in A_{\text{no}}$ no state (of any form) will convince the verifier to accept with probability more than $1/3$.

With this definition it is clear that SQMA $\subseteq$ QMA, because the witness is restricted to have a special form only in the case of a yes-instance. It is perhaps surprising that the equality SQMA $=$ QMA holds. The main observation required to show this is that subset states are sufficiently dense in the set of all states.

More precisely, it can be shown that for any n-qubit unit vector $|\psi\rangle$, there exists a subset $S \subseteq \Sigma^n$ such that $|\langle\psi|S\rangle| = \Omega(n^{-1/2})$. To see that this overlap is sufficient to conclude that SQMA $=$ QMA, recall that the error reduction procedure described in Section 3.2.1 shows that any problem in QMA has a verifier V with completeness and soundness parameters exponentially close to 1 and 0 respectively. In particular the soundness error can be made smaller than any polynomial in the number of qubits of the witness. Whenever there exists a witness $|\psi\rangle$ accepted by V with probability exponentially close to 1, the subset state with maximal overlap on $|\psi\rangle$ will convince the verifier to accept with inverse polynomial probability. Thus, for any problem in QMA, it is possible to construct a SQMA verifier with completeness and soundness parameters separated by an inverse polynomial. The completeness can be amplified to at least $2/3$ by performing parallel error reduction, which preserves the property that there exists a good witness that has the form of a subset state. A similar argument can be made for any restriction on the proof that forces it to belong to a set that remains dense enough in the unit sphere.

The fact just described demonstrates that the strength of quantum proofs does not lie in the possibility to use signed, or complex, amplitudes. Rather, the strength appears to lie in the ability to use superpositions in various ways, such as in the case of the group non-membership problem discussed in Section 3.1.2.

3.4.2 Quantum advice

The definition of the class QMA specifies that the quantum proof provided by the prover to the verifier is *untrusted*: the prover is assumed to always attempt to maximize its chances of convincing the verifier to accept, requiring the verifier to carefully check the information provided by the prover. One may envision a less paranoid situation in which the prover is *trusted*, and always attempts to convince the verifier to make the right decision: accept yes-inputs and reject no-inputs. Without any further restrictions, such a prover would immediately allow the verifier to decide all problems, as a single bit of advice suffices to inform the verifier of whether $x \in A_{\text{yes}}$ or $x \in A_{\text{no}}$.

In the setting of *computational advice*, the following restriction is considered: the quantum state provided by the prover is trusted, but is only allowed to depend on the input length $n = |x|$; so that the same advice is to be provided for all inputs of the same length. In greater detail, the class BQP/qpoly is defined as follows:

Definition 3.3. A promise problem $A = (A_{\text{yes}}, A_{\text{no}})$ is contained in BQP/qpoly if there exists polynomial-time computable functions V and p possessing the following properties:

1. For every string $x \in A_{\text{yes}} \cup A_{\text{no}}$, $V(x)$ is an encoding of a quantum circuit implementing a channel Φ_x with $p(|x|)$ input qubits and 1 output qubit.
2. For every integer n there exists a state ρ_n on $p(n)$ qubits such that the following two conditions hold:

(a) *Completeness.* If $x \in \Sigma^n \cap A_{\text{yes}}$ then $\langle 1|\Phi_x(\rho_n)|1\rangle \geq a$.

(b) *Soundness.* If $x \in \Sigma^n \cap A_{\text{no}}$ then $\langle 1|\Phi_x(\rho_n)|1\rangle \leq b$.

In the case of classical advice, the restriction that an advice string can only depend on the input length immediately rules out the sort of strategy suggested above for a prover to trivially help the verifier in deciding arbitrary problems, as there are exponentially many inputs of a given length and only polynomially many bits provided as advice. In the quantum setting, the situation is not quite as clear, but quantum information-theoretic arguments (namely Holevo's theorem and Nayak's bound) similarly rule out the possibility that a quantum state on polynomially many qubits could encode the answers to an exponential number of problem instances in a way that would be accessible by a valid quantum measurement.

Although it is therefore evident that there are limitations on the power of quantum advice, it is not at all obvious how one can obtain interesting complexity-theoretic upper bounds on the power of quantum advice. One striking upper bound that is known directly relates quantum advice with quantum proofs. It is the containment

$$\text{BQP/qpoly} \subseteq \text{QMA/poly}, \tag{3.46}$$

which demonstrates that a trusted quantum state is no more useful to a polynomial-time quantum verifier than an untrusted quantum state, complemented with a trusted classical advice string of polynomial length. The class QMA/poly is defined in a similar way to BQP/qpoly, where in addition to the untrusted quantum proof from the QMA prover the verifier receives polynomially many classical bits of advice that are only allowed to depend on the input length n.

The proof of the inclusion (3.46) is rather involved. Given a verifier V for a problem in BQP/qpoly, a QMA/poly verifier V' that decides the same problem is constructed. To accomplish this task, the polynomial number of bits of classical advice specify a polynomial-size quantum circuit enabling V' to verify that the untrusted witness $|\psi\rangle$ matches, "for all practical purposes," the trusted advice state ρ_n expected by V. A simple counting argument shows that no polynomial-size circuit could certify closeness of $|\psi\rangle$ to an arbitrary state ρ_n in trace distance to within any reasonable accuracy, but such a strong guarantee is not necessary. The key observation is that it is sufficient to guarantee that $|\psi\rangle$ reproduces approximately the same statistics as ρ_n, not with respect to the outcome of any measurement (which would lead to an approximation in trace distance), but only with respect to polynomial-size quantum circuits of the form that can be executed by V.

There are still exponentially many such circuits, and the fact that such a verification procedure can be specified using only polynomially many bits, and implemented efficiently by V', constitutes most of the work in establishing (3.46). The proof provides a method, based on a tool called the *majority-certificates lemma* [2], to achieve this. Given any state ρ_n, the lemma specifies that there exists a polynomial number of tests, each of which can be specified by a polynomial-size quantum circuit, that the verifier V' can perform on the untrusted $|\psi\rangle$ such that, provided $|\psi\rangle$ passes all tests, it is guaranteed that $|\psi\rangle$ will also approximately reproduce the same outcome as ρ_n with respect to all polynomial-size quantum circuits.

3.4.3 Two unentangled proofs

Are two proofs more useful than one? Unless one imposes very strict length requirements the answer to this question for the case of classical proofs is uninspiring: two classical proof strings of a given length are equivalent to a single proof string of twice that length. In the case of quantum proofs, however, the situation is more subtle. The question is studied by introducing the class QMA(t).

Definition 3.4. Let $t : \mathbb{N} \to \mathbb{N}$ be a polynomially bounded function. A promise problem $A = (A_{\text{yes}}, A_{\text{no}})$ is contained in $\text{QMA}_{a,b}(t)$ if there exists a polynomial-time computable function V possessing the following properties:

1. For every string $x \in A_{\text{yes}} \cup A_{\text{no}}$, $V(x)$ is an encoding of a quantum circuit implementing a channel Φ_x having $t \cdot k$ input qubits and 1 output qubit, for some choice of k and for $t = t(|x|)$.

2. *Completeness.* If $x \in A_{\text{yes}}$, then there exist t states $|\psi_1\rangle, \ldots, |\psi_t\rangle$, on k qubits each, such that
$$\langle 1|\Phi_x(|\psi_1\rangle\langle\psi_1| \otimes \cdots \otimes |\psi_t\rangle\langle\psi_t|)|1\rangle \geq a. \tag{3.47}$$

3. *Soundness.* If $x \in A_{\text{no}}$, then for all choices of k-qubit states $|\psi_1\rangle, \ldots, |\psi_t\rangle$, it holds that
$$\langle 1|\Phi_x(|\psi_1\rangle\langle\psi_1| \otimes \cdots \otimes |\psi_t\rangle\langle\psi_t|)|1\rangle \leq b. \tag{3.48}$$

It is known that, for any polynomially bounded number of witnesses $t \geq 2$, the equality $\text{QMA}(t) = \text{QMA}(2)$ holds. Furthermore, a strong error reduction is possible:

$$\text{QMA}_{a,b}(2) = \text{QMA}_{1-2^{-p(n)},2^{-p(n)}}(2) \tag{3.49}$$

for every polynomial p, provided

$$a(n) - b(n) \geq \frac{1}{q(n)} \tag{3.50}$$

for some polynomial q. Both equalities require a corresponding increase in the witness length, by a factor t for the first transformation and $O(p \cdot q)$ for the second.

The proofs rely on the following *product test* that attempts to determine whether a state is unentangled across t registers, by being given access to *two* unentangled copies of the state.

<u>Product test</u>

Input: Pure states $|\phi_1\rangle$, $|\phi_2\rangle$ on t registers of k qubits each.

Procedure: Perform the SWAP test on each of the t pairs of k-qubit registers. Accept if and only if all tests succeed.

Guarantee: (i) If $|\phi_1\rangle = |\phi_2\rangle = |\psi_1\rangle \otimes \cdots \otimes |\psi_t\rangle$, then the test always accepts.
(ii) If the test accepts with probability $1-\varepsilon$ then there exist states $|\psi_1\rangle, \ldots, |\psi_t\rangle$ such that

$$\min\left\{|\langle\phi_1|\psi_1, \ldots, \psi_t\rangle|^2, |\langle\phi_2|\psi_1, \ldots, \psi_t\rangle|^2\right\} = 1-O(\varepsilon).$$

The SWAP test is a special case of the controlled-unitary test described in Figure 3.2:

SWAP test

Given: Pure states $|\psi\rangle$, $|\varphi\rangle$ on k qubits each.

Outcome: A classical bit that is 0 with probability

$$p = \frac{1 + |\langle\psi|\varphi\rangle|^2}{2}$$

and 1 with probability $1 - p$.

Procedure: Perform the controlled-unitary test using the $2k$-qubit state $|\psi\rangle|\varphi\rangle$ as input, and the $2k$-qubit unitary S that permutes its two sets of k input qubits.

If S denotes the unitary that implements the permutation used in the SWAP test, then it holds that

$$\Re(\langle\varphi,\psi|S|\varphi,\psi\rangle) = |\langle\varphi|\psi\rangle|^2, \tag{3.51}$$

and the analysis of the SWAP test follows immediately from that of the controlled-unitary test.

The product test requires *two* copies of the state to be tested, and it allows for a reduction of the number of required unentangled proof states from any polynomial t to 2. It is open if two proofs are more powerful than one, but there is some evidence pointing in the direction of a positive answer.

As discussed in Section 3.2.3, it is known that QMA $\subseteq$ PP, but the best upper bound known on QMA(2) is the trivial bound of NEXP obtained by guessing exponential-size vectors for the two witnesses. The problem of devising better upper bounds on QMA(2) directly relates to that of optimizing over the set

$$\mathrm{SEP} = \mathrm{Conv}\{\rho \otimes \sigma,\ \rho, \sigma \in \mathrm{D}(\mathbb{C}^d)\} \tag{3.52}$$

of separable states. Although this set is convex, there is no efficient membership oracle known. In fact, deciding weak membership is known to be NP-hard for precision up to inverse polynomial in d.

Perhaps the strongest evidence known that suggests QMA(2) may be a strictly larger class than QMA is the following: there is an efficient QMA(2) verifier V for the satisfiability of 3-SAT formulas on n variables, with completeness and soundness parameters separated by a constant, given access to two unentangled quantum proofs of $O(\sqrt{n}\,\mathrm{polylog}(n))$ qubits each. The existence of such a procedure with a single quantum proof of the same size (or even twice the size) would imply 3-SAT $\in$ DTIME$(\exp(\sqrt{n}\,\mathrm{polylog}(n)))$, thereby violating the exponential time hypothesis. Thus, the ability to receive unentangled witnesses can at least provide a quadratic improvement on the minimum witness length, which may be seen as evidence in favor of QMA $\neq$ QMA(2).

3.4.4 Classical certificates

The class QMA differs from NP and MA in two important ways: the verifier is able to apply a quantum circuit, and the proof that is provided to the verifier may be a quantum state. Of course, a quantum proof requires a quantum verifier and it does not make sense to consider the latter without the former—but it is possible to ask about the power of a *quantum* verifier given access to a *classical* proof. This question can be studied by introducing the class QCMA, defined as QMA except the proof is restricted to be a classical polynomial-length string.

It holds that MA $\subseteq$ QCMA $\subseteq$ QMA, and none of these containments is known to be strict. One may conjecture that QMA properly contains QCMA, but little evidence to support this conjecture is known.

A quantum oracle separation

One piece of evidence to suggest that QCMA is properly contained in QMA is a *quantum oracle separation* as follows. Black-box access to a unitary transformation U is made available, and it is promised that either (i) there exists a state $|\psi\rangle$ such that $U|\psi\rangle = -|\psi\rangle$, and $U|\phi\rangle = |\phi\rangle$ for all $|\phi\rangle \perp |\psi\rangle$, or (ii) it holds that $U|\phi\rangle = |\phi\rangle$ for all states $|\phi\rangle$. The yes-instances are those for which the property (i) holds.

As is to be expected, one can prove that this problem is contained in QMA by taking $|\psi\rangle$ to be given as a quantum proof. It is possible to prove that this problem is not contained in QCMA under the assumption that only black-box access to U is permitted. More precisely, at least $\Omega(2^{k/2}/\sqrt{m})$ queries to U are required to decide between the two possibilities, given a classical proof of length m. Intuitively speaking, the reason why this is so is that the state $|\psi\rangle$ could be any k-qubit state, so the best strategy for a classical prover is to fix a net over the space of all such states, and to provide the verifier with a classical description of the element of the net that is closest to $|\psi\rangle$. Using a measure-theoretic argument, it is possible to show that for any partition of the set of all k-qubit pure states into at most 2^m regions, there will exist a region S with the property that, for every state $|\phi\rangle$, a state $|\psi\rangle$ chosen uniformly from S will have expected overlap

$$\mathrm{E}_{|\psi\rangle\in S}|\langle\phi|\psi\rangle|^2 = O(m2^{-k}) \tag{3.53}$$

with $|\phi\rangle$. This means that, if the m-bit classical proof is interpreted as the description of such a region, in the worst case the proof will only let the verifier reconstruct a state $|\phi\rangle$ whose overlap with a randomly selected state $|\psi\rangle$ satisfies $|\langle\phi|\psi\rangle|^2 = O(m2^{-k})$. Using this state as a starting point, and implementing a procedure based on amplitude amplification, the quantum verifier can find $|\psi\rangle$ using the gate U as a black box with $\Omega(2^{k/2}/\sqrt{m})$ queries, which can be shown to be optimal.

It is reasonable to conjecture that the problem suggested above is contained in QMA, but not QCMA, when the unitary U is specified as a quantum circuit, rather being given as a black box—but naturally the proof suggested above is not sufficient to establish that this is so, as it does not rule out the possibility that an analysis of a quantum circuit's structure could lead to the problem being contained in QCMA.

Verifying Group Non-Membership using a classical witness

In Section 3.1.2 we introduced the group non-membership (GNM) problem as a promise problem having a natural QMA verification procedure. As GNM is not known to be QMA-complete, it is a natural target problem to put in QCMA.

The honest witness for GNM has a specific form (3.7), which is the uniform superposition over all elements in a subgroup. As discussed in

Section 3.4.1, however, *every* language in QMA has a verifier for which there is a witness that has a similar "subset state" form. Thus, the form of the GNM witness (3.7) is not directly indicative of a problem that should be easier than QMA-complete problems.

Nevertheless, it can be shown that with the help of a classical polynomial-size witness, a quantum verifier can decide any instance of GNM using only a polynomial number of queries to the group oracle. The catch is that this verifier may require an exponential amount of "side" computation—operations that do not involve the group G in question. The idea is to use a classical witness to specify a certain "model group" Γ, as well as an injective homomorphism $f : G \to \Gamma$. Both can be specified with a polynomial number of bits using an appropriate set of generators for G and Γ. The verifier can compute the image in Γ of G, the subgroup H, and x, and verify that $f(x) \notin f(H)$ using only polynomially many operations in G (to decompose x and generators of H on the generators of G provided by the classical witness) and exponentially many operations in Γ (to verify $f(x) \notin f(H)$).

The difficulty of this approach is to verify that the witness has the correct form, i.e., that the map f is (close to) an injective homomorphism. Checking that f is close to a homomorphism can be done efficiently in randomized polynomial time by verifying the identity $f(g_1 g_2) = f(g_1)f(g_2)$ for sufficiently many random pairs of group elements (g_1, g_2). Thus, the main step is to check injectivity of f. But this is an instance of the hidden subgroup problem (HSP) and can be solved by a quantum circuit making polynomially many group operations (and possibly exponentially many classical operations not involving G).

3.5 Chapter notes

Quantum proofs were evidently first discussed by Knill [115], and formalized as a complexity class called BQNP by Kitaev around 1999 (and later published in [113]). The name QMA first appears in [166], where it was also shown that group non-membership is in QMA. Parallel error reduction is analyzed in [113], and the procedure for witness-preserving error reduction is due to [125], where the containment QMA $\subseteq$ PP was proved but attributed to unpublished work of Kitaev and Watrous.

The local Hamiltonian problem was introduced by Kitaev in his original work on QMA and shown to be QMA-complete for $k = 5$. This was improved to $k = 3$ in [110] and $k = 2$ in [107]. Physical motivations suggest the consideration of restricted families of interaction graphs and types of local Hamiltonians. For instance, [135] proved QMA-hardness for instances whose interaction graph is restricted to a two-dimensional grid, and this is extended to 2-local Hamiltonians on a line in [8]. The paper [51] established a classification of restricted classes of 2-local Hamiltonians in terms of their hardness, providing a quantum analogue of Schaefer's dichotomy theorem for Boolean constraint satisfaction problems. The Quantum PCP conjecture was first formulated in [76, 5]. For further background on the conjecture we refer to the survey [6].

Quantum k-SAT was introduced in [39], where it was shown that the problem is in P for $k = 2$ and QMA_1-complete for $k \geq 4$; QMA_1-completeness for $k = 3$ is due to [74]. The Consistency of Density Operators problem was shown to be QMA-complete with respect to Cook reductions in [123]. We refer to the survey [34] for a more extensive list of QMA-complete problems.

The class SQMA was introduced in [77], where an analogue for QMA(2) was also introduced, and the equality SQMA(2) = QMA(2) was proved along the same lines as SQMA = QMA. The class QMA_+ was introduced in [10], where it was used to show that $\mathrm{coGapSVP}_{\sqrt{n}}$, a gapped version of the shortest vector problem in lattices, lies in $\mathrm{QMA}_+ = \mathrm{QMA}$. At the time it was not known if this problem was contained in NP, but the authors later "de-quantized" their result to obtain this fact [10]. The class BQP/qpoly was first considered in [132], and the inclusion BQP/qpoly $\subseteq$ QMA/poly was proved in [2]. The class QMA(t) for $t \geq 2$ was introduced in [120]. The product test, its analysis, and the equality QMA(t) = QMA(2) for $t \geq 2$ are due to [88]. The existence of a protocol for verifying a 3-SAT formula on n variables using unentangled proofs totalling $O(\sqrt{n}\,\mathrm{poly}\log n)$ qubits was first shown in [1]. A different protocol, this time for 3-coloring [32], is analyzed in [43] where a trade-off between the number of proofs used and the is given. NP-hardness of the weak membership problem for the set of separable states is shown in [79] for exponential accuracies, and improved to inverse polynomial accuracies in [66]. A consequence of the product test from [88] is that weak membership for constant accuracy cannot be decided in polynomial time unless 3-SAT$\in$ DTIME$(\exp(\sqrt{n}\,\mathrm{poly}\log n))$. Better upper bounds than NEXP are known on QMA(2) when additional restrictions on the verifier are imposed [37].

The class QCMA was defined in [3], where a quantum oracle separation with QMA is proved. Few complete problems for QCMA are known; an interesting example is the Ground State Connectivity (GSCON) problem considered in [68].

A number of additional variations of the class QMA have been considered. The class UQMA, or unique QMA, corresponds to those problems for which in the yes-case there is a one-dimensional subspace of witnesses that convince the verifier to accept with high probability, while any state in the orthogonal subspace will lead to a success probability that is smaller by at least a fixed inverse polynomial. The class FewQMA is defined similarly, replacing the one-dimensional subspace of convincing witnesses by a subspace having dimension at most polynomial in the input size. These classes, which are analogues of variants of NP considered by Valiant and Vazirani [160], were introduced in [7]. It is known that FewQMA = UQMA [97], but it is still open whether these classes equal QMA.

The class DQMA, introduced in [12] by analogy with a similar extension of NP called DP [138], consists of all those problems whose difference is in QMA: $(A_{\mathrm{yes}}, A_{\mathrm{no}}) \in$ DQMA if there exists $(B_{\mathrm{yes}}, B_{\mathrm{no}})$ and $(C_{\mathrm{yes}}, C_{\mathrm{no}}) \in$ QMA such that $x \in A_{\mathrm{yes}}$ implies $x \in B_{\mathrm{yes}} \cap C_{\mathrm{no}}$ and $x \in A_{\mathrm{no}}$ implies $x \in B_{\mathrm{no}} \cup C_{\mathrm{yes}}$, as well as $x \in (B_{\mathrm{yes}} \cup B_{\mathrm{no}}) \cap (C_{\mathrm{yes}} \cup C_{\mathrm{no}})$. Complete problems for DQMA that do not appear to lie in QMA are given in [12]; these include the problem of deciding whether the ground state energy of a local Hamiltonian lies in a certain interval, or is outside of that interval.

Quantum analogues of classes higher in the polynomial hierarchy such as Σ_2^p and Π_2^p are introduced in [67], where complete problems for these classes are given.

4

Single-Prover Quantum Interactive Proofs

This chapter introduces a quantum computational analogue of the most standard interactive proof system model, in which a verifier interacts with a single prover, and surveys several known results concerning this model. A few highlights of the results to be discussed in the chapter are as follows:

1. Quantum interactive proof systems can be *parallelized* to three-turn interactive proof systems having strong error bounds. More precisely, any promise problem having a bounded-error, polynomial-turn, single-prover quantum interactive proof system must also have a three-turn, single-prover quantum interactive proof system with perfect completeness and exponentially small soundness error.
2. The problem of optimizing the probability for a verifier in a single-prover quantum interactive proof system to accept a given string can be represented as a semidefinite program in a fairly simple and direct way. This representation provides a useful tool for reasoning about single-prover quantum interactive proof systems.
3. The class of promise problems having bounded-error, single-prover quantum interactive proof systems coincides with PSPACE.

4.1 Definitions of quantum interactive proof systems

In the classical setting, interactive proof systems have historically been defined through variants of the probabilistic Turing machine model, modified in such a way as to allow for interactions with an external entity (such as another Turing machine). Quantum computation, on the other hand, is more conveniently modeled by quantum circuits, as was suggested in Chapter 2, and for this reason our definitions of quantum interactive proof systems will be based on circuits rather than Turing machines.

The precise definitions of quantum interactive proof systems that we adopt in this survey are essentially the same as ones considered in prior work on the subject, although we will place a somewhat greater emphasis on the fixed-size interactions that are induced by interactive proof systems on fixed input strings. The notion of an *interactive game*, to be introduced shortly, is intended to be an abstraction of this sort of interaction.

4.1.1 Interactive games

The first part of the definition of quantum interactive proof systems involves the introduction of the general notion of an *interactive game*. Intuitively speaking, interactive games are abstractions of fixed-size interactions, in which there is no notion of a shared input string to the participants. Although the focus of the current chapter will be on quantum interactions, the concept of an interactive game is not inherently quantum—the basic concept can be adapted to the classical setting is a straightforward manner.

The notion of an interactive game is, in fact, sufficiently general that one may adapt it to formulate definitions of other interactive proof system variants (such as interactive proof systems with multiple provers, either cooperating or competing, and zero-knowledge interactive proofs), as well as cryptographic interactions such as coin-flipping. Indeed, it will likely be quite evident from the discussion that follows that the notion of an interactive game can be generalized in numerous ways, allowing for three or more participants, outputs for any subset

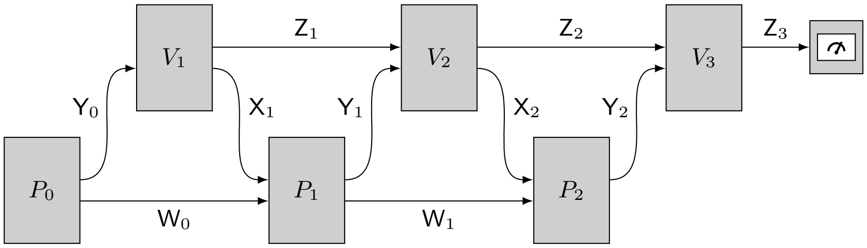


Figure 4.1: A five-turn interactive game involving a prover and a verifier. The prover's actions are represented by the boxes labeled P_0, P_1, and P_2, while the verifier's actions are represented by the boxes labeled V_1, V_2, and V_3. It is a five-turn interactive game because five registers are exchanged during the interaction that is illustrated: in the first turn the prover sends the register Y_0 to the verifier, in the second turn the verifier sends X_1 to the prover, and so on. The registers Z_1 and Z_2 represent the verifier's memory registers, while W_0 and W_1 represent the prover's memory registers. The rightmost box represents a measurement that produces a classical output bit.

of the participants, and so on. Although we will consider some such adaptations and generalizations in other parts of this survey, we will not attempt to emphasize the generality of the notion at this stage; our focus here will be limited to interactive games that are representative of interactions between two entities, playing the roles of prover and verifier.

With this focus in mind, an interactive game describes a situation in which two participants, a *prover* and a *verifier*, exchange fixed-size quantum registers for a fixed number of steps. At the end of the interaction, the verifier produces a single classical bit as output. Figure 4.1 illustrates an interactive game of this sort, in the particular case in which five register exchanges (or *turns*) occur during the interaction.[1] The actions performed by a prover and verifier at each step of an interactive game must be valid physical operations, at least in the idealized sense that is modeled by the theory of quantum information, and must therefore be described by quantum channels.

[1] One may alternatively count *messages* rather than *turns*. In the single-prover setting, these notions are equivalent, but in the multi-prover setting (to be considered in Chapter 6) it is convenient to consider that a turn may involve a collection of messages being either sent or received in parallel by a single entity.

When it is convenient, we will refer to an *m-turn verifier*, an *m-turn prover*, or an *m-turn interactive game* to indicate that m register exchanges between the verifier and prover take place, for the verifier, prover, or interactive game being considered. A prover and verifier in an interactive game must naturally be *compatible*, in the sense that they agree on both the number and timing of the register exchanges and on the sizes of these registers. Hereafter, we will take it as an implicit assumption that such an agreement is in place, as there is little to be said about interactions between a prover and verifier that are not compatible.

The pattern represented by the labeling of the registers and transformations in Figure 4.1 will be mimicked throughout this survey to the extent that it is possible. In general, the verifier's actions will be described by channels $V_1, \ldots, V_n$, with each channel V_k transforming the register pair $(\mathsf{Z}_{k-1}, \mathsf{Y}_{k-1})$ to $(\mathsf{Z}_k, \mathsf{X}_k)$, as suggested by Figure 4.2, while the prover's actions are described by channels $P_0, \ldots, P_{n-1}$ (in case the number of turns is odd) or $P_1, \ldots, P_{n-1}$ (in case the number of turns is even), with each channel P_k transforming $(\mathsf{X}_k, \mathsf{W}_{k-1})$ to $(\mathsf{Y}_k, \mathsf{W}_k)$, as suggested by Figure 4.3. It will be possible to avoid the need to handle special cases at the beginning and end of interactions by adopting the convention that "absent" registers, such as Z_0, X_n, X_0, and W_{-1}, and possibly Y_0, are identified with trivial registers comprising zero qubits.[2]

Our primary interest is in the situation in which the specification of a verifier is fixed, and it is to be viewed that one's goal is to optimize the actions of the prover so as to cause the verifier to produce the output 1 (representing *acceptance* in the terminology of interactive proofs). Generalizing the notation and terminology used in the previous chapter, we write $\omega(V)$ to denote the *value* of a given verifier V in an interactive game, which is defined as the maximum probability with which a prover (compatible with V) can cause V to output 1.[3]

[2] Registers having zero qubits are legitimate quantum systems that have a single classical state, and a corresponding Hilbert space equal to $\mathbb{C}$. We do not consider that the transmission of trivial registers contributes to the number of turns in an interaction.

[3] Formally speaking, the value of a verifier V is more naturally defined as the *supremum* probability with which a compatible prover can cause V to output 1, as

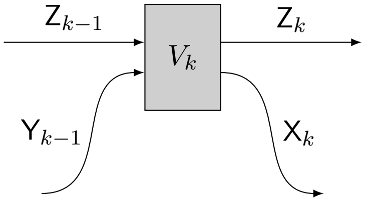


Figure 4.2: A general verifier action.

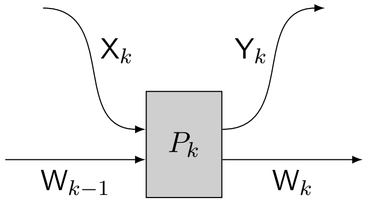


Figure 4.3: A general prover action.

Example 4.1. The following simple example is intended to illustrate the basic concept of an interactive game. Define a two-turn verifier V as follows:

1. The verifier's first action represents the creation of a maximally entangled state

$$|\psi\rangle = \frac{1}{\sqrt{2}}|0\rangle|0\rangle + \frac{1}{\sqrt{2}}|1\rangle|1\rangle \tag{4.1}$$

   of a pair of single-qubit registers $(\mathsf{Z}_1, \mathsf{X}_1)$. The qubit X_1 is sent to the prover.

2. The verifier's second action represents a binary-valued measurement performed on the pair $(\mathsf{Z}_1, \mathsf{Y}_1)$, for Y_1 being a single-qubit register received from the prover. The measurement operator corresponding to the outcome 1 is defined as $\Pi_1 = |\phi\rangle\langle\phi|$ for

$$|\phi\rangle = \cos(\pi/8)|0\rangle|0\rangle + \sin(\pi/8)|1\rangle|1\rangle, \tag{4.2}$$

   while the measurement operator corresponding to the outcome 0 is $\Pi_0 = \mathbb{1} - |\phi\rangle\langle\phi|$.

The optimal probability with which a prover can cause this verifier to output 1 is equal to

$$\omega(V) = \cos^2(\pi/8) \approx 0.85. \tag{4.3}$$

it is not immediate that this supremum value is achieved by a single prover. The supremum *is* achieved, however, so one is justified in considering the value as the maximum probability with which a prover can cause V to output 1.

It is easy to see that this probability is achievable: a prover may simply return the register X_1 to the verifier, renaming it Y_1 but otherwise leaving it unchanged, which causes the verifier to output 1 with probability

$$|\langle\phi|\psi\rangle|^2 = \left|\frac{\cos(\pi/8)+\sin(\pi/8)}{\sqrt{2}}\right|^2 = \cos^2(\pi/8). \tag{4.4}$$

The fact that this probability is optimal follows from the fact that the probability with which the verifier accepts, for any choice of a prover, is given by

$$\operatorname{Tr}(\Pi_1\rho) = \operatorname{F}(|\phi\rangle\langle\phi|,\rho)^2 \tag{4.5}$$

for some two-qubit state ρ whose first qubit, when viewed in isolation, is completely mixed. By the fact that the fidelity is non-decreasing under partial tracing, one finds that the probability of acceptance is at most

$$\operatorname{F}\left(\begin{pmatrix}1/2 & 0\\ 0 & 1/2\end{pmatrix}, \begin{pmatrix}\cos^2(\pi/8) & 0\\ 0 & \sin^2(\pi/8)\end{pmatrix}\right)^2 = \cos^2(\pi/8), \tag{4.6}$$

which establishes the optimality of this acceptance probability.

This example illustrates an important theme in the analysis of quantum interactive games, which is that a prover's possible actions exactly correspond to those transformations that leave the reduced state of the verifier's register unchanged (under the assumption that the joint state of the prover and verifier is pure). The analysis made here will re-appear in the proof of the perfect completeness property in Section 4.2.1, and the idea is the key to the formulation of the value of an interactive game as the optimum of a semidefinite program to be described in Section 4.3.

In the previous example, we have not specified the verifier's actions as quantum channels, at least in a formal sense. It is, however, possible to do this. In particular, the verifier's first action corresponds to a channel V_1 that takes no input (or, equivalently, takes a pair of trivial registers $(\mathsf{Z}_0, \mathsf{Y}_0)$ as input) and outputs the state $|\psi\rangle\langle\psi|$ contained in the pair $(\mathsf{Z}_1, \mathsf{X}_1)$. More formally speaking, this channel corresponds to the mapping

$$V_1(\alpha) = \alpha|\psi\rangle\langle\psi| \tag{4.7}$$

for all $\alpha \in \mathbb{C}$, which is a completely positive and trace-preserving map. The second action may be expressed as a channel as well, in this case transforming the pair of registers $(\mathsf{Z}_1, \mathsf{Y}_1)$ into a single-qubit register Z_2 (or, equivalently, into a pair $(\mathsf{Z}_2, \mathsf{X}_2)$ where X_2 is trivial) in the manner described by the mapping

$$V_2(X) = \langle \Pi_0, X \rangle |0\rangle\langle 0| + \langle \Pi_1, X \rangle |1\rangle\langle 1| \tag{4.8}$$

for all $X \in \mathrm{L}(\mathcal{Z}_1 \otimes \mathcal{Y}_1)$.

For other interactive games described in this chapter, we will generally omit the sorts of details that have been given in the previous paragraph—it is usually a routine exercise to fill in such details.

4.1.2 Descriptions and encodings of interactive games

There are two natural ways in which one may describe either or both of the participants in an interactive game: one is by *quantum circuits*, and the other is by *explicit matrix representations* of the participants' actions.

1. *Quantum circuit representations.* An m-turn verifier is determined by an n-tuple $V = (V_1, \ldots, V_n)$, for $n = \lfloor m/2+1 \rfloor$, where each V_k is a quantum channel transforming a pair of registers $(\mathsf{Z}_{k-1}, \mathsf{Y}_{k-1})$ to a pair of registers $(\mathsf{Z}_k, \mathsf{X}_k)$. A quantum circuit description of such a verifier is simply an n-tuple of quantum circuits, each implementing one of these channels, along with a specification of which input and output qubits of each circuit are to be associated with the two registers in each pair. Each individual circuit may be encoded following the general principles outlined in Section 2.3. Provers can be described in an analogous manner (although we are typically not concerned with the efficiency of provers, making circuit descriptions of them generally less useful).
2. *Explicit matrix representations.* As above, an m-turn verifier is determined by an n-tuple of quantum channels $V = (V_1, \ldots, V_n)$, for $n = \lfloor m/2+1 \rfloor$. Along with a specification of which input and output qubits of each channel are to be associated with the two registers

in each register pair, one may describe each channel V_k by an explicit matrix representation (such as a Stinespring representation). Again, provers may be represented in an analogous way.

4.1.3 Quantum interactive proof systems

Having defined interactive games, we are now prepared to define various complexity classes based on the concept of quantum interactive proof systems. Intuitively speaking, we view a quantum interactive proof system to be the specification of a quantum interactive game for each possible input string to the problem being considered.

As is typical for interactive proof system models, we will constrain verifiers in interactive game representations of quantum interactive proof systems to be computationally bounded. To be more precise, we require the verifier's actions to be represented by quantum circuits whose descriptions can be generated in polynomial time from the problem input. The fact that the prover is not computationally bounded is manifested in the requirement that the maximum acceptance probability of a given verifier places no computational restrictions on the prover's actions.

Definition 4.1. A promise problem $A = (A_{\text{yes}}, A_{\text{no}})$ is contained in the complexity class $\text{QIP}_{a,b}(m)$ if there exists a polynomial-time computable function V that possesses the following properties:

1. For every string $x \in A_{\text{yes}} \cup A_{\text{no}}$, one has that $V(x)$ is an encoding of a quantum circuit description of an m-turn verifier in an interactive game.
2. *Completeness.* For every string $x \in A_{\text{yes}}$, it holds that $\omega(V(x)) \geq a$.
3. *Soundness.* For every string $x \in A_{\text{no}}$, it holds that $\omega(V(x)) \leq b$.

In this definition, one may take m, a, and b to be constants or functions of the length of x. When a and b are omitted, it is to be understood that $a = 2/3$ and $b = 1/3$, so that

$$\text{QIP}(m) = \text{QIP}_{2/3,1/3}(m). \tag{4.9}$$

We also write QIP, without specifying a number of turns, to refer to the class of promise problems A for which there exists a polynomially bounded function m such that $A \in \mathrm{QIP}(m)$. Given that a polynomial-time computable function V representing a verifier would not be capable of outputting an n-tuple of quantum circuit descriptions with n being super-polynomial in $|x|$, this is equivalent to placing no restrictions on the number of turns.

It should be noted that any classical verifier in an interactive game can be viewed as a restricted type of quantum verifier. It is not difficult to prove that quantum prover strategies cannot gain an advantage over optimal classical prover strategies against classical verifiers, and based on this observation one may verify that

$$\mathrm{IP}_{a,b}(m) \subseteq \mathrm{QIP}_{a,b}(m) \tag{4.10}$$

for all choices of a, b, and m.[4]

4.1.4 Purifications of interactive games

Interactive games are, in some situations, easier to analyze when the joint state of all of the co-existing registers at each instant is a pure state, as opposed to being an arbitrary mixed state. Indeed, for some of the proof techniques we will use later in this chapter, this assumption of purity is essential.

Fortunately, there is no generality lost in restricting one's attention to interactive games with this property. This follows from the fact that each individual channel performed by either of the participants in an interactive game may be purified in the manner described in Section 2.2.3, so that each action is represented by a linear isometry. The additional output qubits produced by this process must be considered private memory qubits for whichever player performs that particular channel. Assuming that these additional qubits are not touched again during the interactive game (so that subsequent actions of the player act trivially on these qubits), the effect is identical to the original channel. This process is illustrated in Figure 4.4. In the situation in which the efficiency of a verifier is concerned, this transformation is done gate-by-gate rather than turn-by-turn, as discussed in Section 2.3.

[4] The situation is not nearly so simple in the multi-prover setting, as will be discussed in Chapter 6.

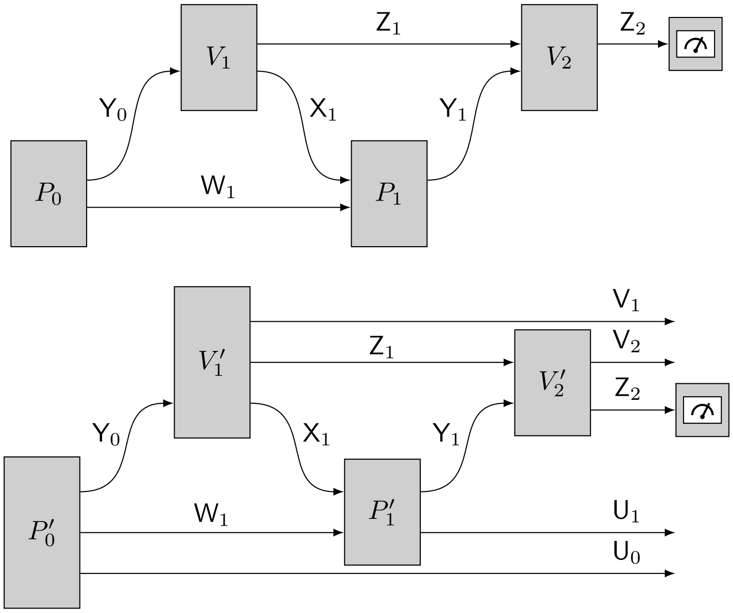


Figure 4.4: A three-message interactive game and its purification.

Naturally, it may also be assumed that the actions performed by each participant correspond to unitary transformations, as opposed to transformations described by linear isometries, provided that sufficiently many ancillary input qubits are provided to each of these unitary tranformations. Again, the initialized qubits must be understood to be included in a given participant's private memory, so that the other participant may not tamper with them to influence the output of the game.

4.2 Perfect completeness and parallelization

Two basic facts concerning single-prover interactive proof systems will be discussed in the present section. These facts may be proved through direct constructions, which efficiently transform quantum interactive games in ways that allow one to conclude that the facts hold.

The first construction establishes the relation

$$\mathrm{QIP}_{a,b}(m) \subseteq \mathrm{QIP}_{1,c}(m+2), \tag{4.11}$$

for any number of turns m, and for c being bounded away from 1 as a function of the gap $a - b$ between the completeness and soundness parameters. This relation implies that quantum interactive proof systems do not lose any power when restricted to having *perfect completeness*, as long as one is willing to increase the number of turns by two.

The second construction establishes the relation

$$\mathrm{QIP}_{1,c}(m) \subseteq \mathrm{QIP}_{1,d}(3) \tag{4.12}$$

for any polynomially bounded function m, for d being bounded away from 1 when the same is true of c. This implies that quantum interactive proof systems can be *parallelized* to a high degree, which is an important property that distinguishes them from classical interactive proof systems.

4.2.1 Perfect completeness

Given a quantum circuit description of a verifier V in a quantum interactive game, as well as a target threshold α for its value $\omega(V)$, it is possible to efficiently construct a new verifier V' in such a way that the following properties are in place:

1. If V is an m-turn verifier, then V' is an $(m+2)$-turn verifier.
2. If it is the case that $\omega(V) \geq \alpha$, then $\omega(V') = 1$.
3. If it is the case that $\omega(V) < \alpha$, then $\omega(V') < 1 - (\alpha - \omega(V))^2$.

We make the assumption that α is a dyadic rational in the construction, which is to be explained shortly. The construction has the following implication to quantum interactive proof system classes.

Theorem 4.1. For any choice of polynomial-time computable functions $a, b : \mathbb{N} \to (0,1)$ and $m : \mathbb{N} \to \mathbb{N}$ for which $a(n) < b(n)$ and $m(n) \geq 1$ for every $n \in \mathbb{N}$, it holds that

$$\mathrm{QIP}_{a,b}(m) \subseteq \mathrm{QIP}_{1,c}(m+2) \tag{4.13}$$

for

$$c = 1 - \frac{1}{2}(a - b)^2. \tag{4.14}$$

Thus, quantum interactive proof systems with *perfect completeness* are at least as powerful as those having a nonzero completeness error, provided that one allows the quantum interactive proof system with perfect completeness to have two additional turns and a somewhat larger soundness error. (As we will soon see, a reduction in the number of turns and in the soundness error is possible through the use of other methods, so these are not major concessions to make for the property of perfect completeness.)

To describe the essential idea behind the construction, it will be convenient to first consider the case in which $\alpha = 1/2$, delaying the discussion of how one may handle other values of α until later. It will be assumed that the given verifier V is in a purified form, so that its actions are described by an n-tuple of isometries $(V_1, \ldots, V_n)$. Moreover, it will be assumed that if $\omega(V) \geq 1/2$, then there exists a prover P that is capable of making V output 1 with probability *exactly* $1/2$. If these assumptions were not in place, it would be straightforward to preprocess the description of the given verifier to ensure that these assumptions are met; the process of purifying quantum channels has already been discussed, and the second assumption can be imposed by wiring the verifier with an additional qubit that allows a prover to force the output 0 if it chooses, effectively throwing the game with any desired probability. Finally, it will be assumed, without loss of generality, that at the end of the interaction, the first of the verifier's qubits is considered to be the output qubit, which is measured with respect to the computational basis to produce the verifier's output bit.

With these assumptions in place, the construction of the $(m+2)$-turn verifier V' from V is as follows:

1. The verifier V' behaves precisely as V does for m turns, up to but not including the final measurement of the output qubit of V.
2. The verifier V' then makes a *pseudo-copy* of the output qubit of V by performing the isometry defined by

$$|00\rangle\langle 0| + |11\rangle\langle 1| \tag{4.15}$$

on this output qubit. In turn number $m+1$ of the interaction, V' sends all of its qubits to the prover, aside from one of the two qubits produced by the pseudo-copy operation.

3. In the final turn, the verifier V' receives a single qubit. It then measures the two qubits it holds (one from the pseudo-copy operation and the other received from the prover) against the two-qubit state
$$|\gamma\rangle = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle \tag{4.16}$$
(i.e., with respect to a measurement having measurement operators $|\gamma\rangle\langle\gamma|$ and $\mathbb{1} - |\gamma\rangle\langle\gamma|$). If the measurement outcome is consistent with this target state, it outputs 1, and otherwise it outputs 0.

Consider the case in which there exists a prover P that causes V to accept with probability exactly $1/2$. It may be assumed that P has been purified, as the purification process has no effect on the probability of acceptance. As the acceptance probability of V is $1/2$, the final state of the interactive game, immediately before a measurement of the verifier's output qubit takes place, must take the following form:
$$\frac{1}{\sqrt{2}}|0\rangle|\phi_0\rangle + \frac{1}{\sqrt{2}}|1\rangle|\phi_1\rangle. \tag{4.17}$$
Here, the first qubit represents the verifier's output qubit, and the vectors $|\phi_0\rangle$ and $|\phi_1\rangle$ represent all of the other qubits, including all of the verifier's private qubits aside from the output qubit and all of the prover's private qubits. It need not be the case that $|\phi_0\rangle$ and $|\phi_1\rangle$ are orthogonal, but they are necessarily unit vectors.

One may now define a new prover P' that causes V' to output 1 with certainty. The following description of P' will achieve this goal:

1. The new prover P' behaves precisely as P does for m turns.
2. On turn number $m+1$, the new prover P' receives a collection of qubits from V', and is expected to return a single qubit on turn number $m+2$. The required transformation for P' is as follows:
$$|0\rangle|\phi_0\rangle \mapsto |0\rangle|\psi\rangle \qquad \text{and} \qquad |1\rangle|\phi_1\rangle \mapsto |1\rangle|\psi\rangle, \tag{4.18}$$

where $|\psi\rangle$ is any fixed unit vector, and where $|\phi_0\rangle$ and $|\phi_1\rangle$ are as above. The first qubit, which came from the verifier's pseudo-copy operation, is the qubit to be returned on the last turn. The sets

$$\{|0\rangle|\phi_0\rangle, |1\rangle|\phi_1\rangle\} \qquad \text{and} \qquad \{|0\rangle|\psi\rangle, |1\rangle|\psi\rangle\} \tag{4.19}$$

are both orthonormal sets, for any choice of a unit vector $|\psi\rangle$, so under the assumption that $|\psi\rangle$ represents the same number of qubits as $|\phi_0\rangle$ and $|\phi_1\rangle$, one can extend the transformation (4.18) to a unitary operation.

For P' being defined in this way, the final state of the interaction immediately before the measurement V' performs on its last step is

$$\left(\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle\right)|\psi\rangle, \tag{4.20}$$

which causes V' to output 1 with certainty.

It remains to consider the case in which the maximum acceptance probability of V is less than 1/2. Let us assume, more precisely, that V' interacts with a prover P' that, after m turns, would have led V to accept with probability $1/2 - \varepsilon$ for some choice of $\varepsilon > 0$. When the verifier V' creates the pseudo-copy of what would have been the output qubit of V and sends everything to P' aside from one of the qubits resulting from the pseudo-copy, it is evident that the reduced state of this single qubit is given by

$$\rho = \begin{pmatrix} \frac{1}{2} + \varepsilon & 0 \\ 0 & \frac{1}{2} - \varepsilon \end{pmatrix}. \tag{4.21}$$

Regardless of the actions of P', the acceptance probability of V' must be given by $\mathrm{F}(|\gamma\rangle\langle\gamma|, \sigma)^2 = \langle\gamma|\sigma|\gamma\rangle$, for σ being a density operator that represents the state of the two qubits held by the verifier at the beginning of the last turn of the protocol. Because the prover's actions are unitary it must be the case that σ extends ρ. As the fidelity is monotonically increasing under partial tracing, it is not possible that the quantity above exceeds

$$\mathrm{F}\left(\begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix}, \begin{pmatrix} \frac{1}{2} + \varepsilon & 0 \\ 0 & \frac{1}{2} - \varepsilon \end{pmatrix}\right)^2 = \frac{1}{2} + \frac{1}{2}\sqrt{1 - 4\varepsilon^2} < 1 - \varepsilon^2. \tag{4.22}$$

The required properties of V', as they relate to V, have therefore been verified.

In the case in which a different value of α is to be considered, the construction of V' from V is identical aside from the substitution of

$$|\gamma\rangle = \sqrt{1-\alpha}|00\rangle + \sqrt{\alpha}|11\rangle \tag{4.23}$$

in place of (4.16). The analysis is the same, except that one must obtain an upper bound on the value

$$\mathrm{F}\left(\begin{pmatrix} 1-\alpha & 0 \\ 0 & \alpha \end{pmatrix}, \begin{pmatrix} (1-\alpha)+\varepsilon & 0 \\ 0 & \alpha-\varepsilon \end{pmatrix}\right)^2. \tag{4.24}$$

Writing $\beta = \alpha - \varepsilon$, one may bound the value (4.24) using the arithmetic-geometric mean inequality as follows:

$$\begin{aligned}
&\mathrm{F}\left(\begin{pmatrix} 1-\alpha & 0 \\ 0 & \alpha \end{pmatrix}, \begin{pmatrix} 1-\beta & 0 \\ 0 & \beta \end{pmatrix}\right)^2 \\
&\qquad = \left(\sqrt{\alpha\beta} + \sqrt{(1-\alpha)(1-\beta)}\right)^2 \\
&\qquad = \alpha\beta + (1-\alpha)(1-\beta) + 2\sqrt{\alpha\beta(1-\alpha)(1-\beta)} \\
&\qquad \leq \alpha\beta + (1-\alpha)(1-\beta) + \alpha(1-\alpha) + \beta(1-\beta) \\
&\qquad = 1 - (\alpha-\beta)^2 \\
&\qquad = 1 - \varepsilon^2.
\end{aligned} \tag{4.25}$$

We may now connect the construction described above to the statement of Theorem 4.1 in a fairly straightforward way. For a given promise problem $A \in \mathrm{QIP}_{a,b}(m)$, we have a polynomial-time computable function V witnessing this inclusion. The functions (or constants) a and b are polynomial-time computable, so on a given input string x of length n, one may compute a dyadic rational number α satisfying the inequalities

$$\frac{3a+b}{4} \leq \alpha \leq a. \tag{4.26}$$

Intuitively speaking, what this is doing is to truncate a binary representation of a to obtain α, taking sufficiently many bits to leave a reasonably large gap between α and b. The requirement that α is a dyadic rational guarantees that one may perform a measurement against a state $|\gamma\rangle$ of the form (4.23) without error, using the gates from the universal gate set described in Section 2.3. (If a different set of gates were selected, a suitable choice of α could be substituted to allow for an error-free computation in this step, provided that the gate set is a reasonable one.) Finally, one may take V' to be the function that outputs a description of the verifier derived from V from the construction above, for the choice of α that has just been specified. This is a polynomial-time computable function witnessing the inclusion $A \in \mathrm{QIP}_{1,c}(m+2)$ for

$$c \leq 1 - \left(\frac{3(a-b)}{4}\right)^2 \leq 1 - \frac{1}{2}(a-b)^2, \tag{4.27}$$

as required.

4.2.2 Parallelization to three turns

One of the most striking complexity-theoretic properties of quantum interactive proof systems, at least insofar as they compare with classical interactive proof systems, is that they may be *parallelized* to a constant number of turns without diminishing their computational power. To be more precise, one has the following theorem.

Theorem 4.2. $\mathrm{QIP} = \mathrm{QIP}(3)$.

That is, any promise problem having a polynomial-turn quantum interactive proof system necessarily has a three-turn quantum interactive proof system as well. It is an open question whether an analogous statement holds classically, but if such a statement were true it would imply the collapse of the polynomial-time hierarchy [19, 73]—and so it is viewed by many as being unlikely, and is closely connected to the most central open problems of computational complexity.

There are essentially three steps required to prove Theorem 4.2, only one of which is directly concerned with the parallelization process itself. The first step involves the transformation of a given quantum interactive proof system to one having perfect completeness, as was discussed in the previous subsection; the second step is the parallelization step, which will be discussed in the present subsection; and the final step is concerned with error reduction, which will be discussed in the section following this one. Each of these steps may be represented by an efficient transformation applied to a quantum circuit description of a verifier in an interactive game, and by combining them in the most natural way the relationship QIP = QIP(3) is obtained.

For the remainder of the present subsection, we will consider an efficient transformation that operates as follows. It is assumed that an m-turn verifier V in a quantum interactive game is given, where m may be arbitrary. From this verifier V, a new verifier V' is constructed that has the following properties:

1. V' is a three-turn verifier.
2. If it is the case that $\omega(V) = 1$, then $\omega(V') = 1$ as well.
3. If it is the case that $\omega(V) \leq 1 - \varepsilon$, then $\omega(V') \leq 1 - \varepsilon/m^2$.

There are, in fact, multiple constructions known to parallelize quantum interactive proof systems in this way. We will describe a particularly simple construction of Kempe, Kobayashi, Matsumoto, and Vidick [109], which is well-suited to the presentation of quantum interactive proof systems through the interactive games framework that has been adopted in this survey. The essential idea of the construction is to iteratively transform a verifier in a quantum interactive game into a new verifier having roughly half as many turns, using a cut-and-choose style argument. Each iteration will result in at most a constant-factor increase in the size of the verifier descriptions, so the transformation may be iterated logarithmically many times to reduce the number of turns to a constant. The method cannot be used to reduce the number of turns below three.

With such an iterative approach in mind, suppose that V is an m-turn verifier, for m taking the form

$$m = 2^{r+1} + 1 \tag{4.28}$$

for some positive integer r. For cases in which m does not take this form, one may simply add dummy turns that have no influence on the output of V. (In general, the addition of such dummy turns will slightly less than double the number of turns, and does not need to be iterated—it is only done once at the beginning of the iterative process.)

Under the assumption that $m = 2^{r+1} + 1$, actions of V are specified by an n-tuple $(V_1, \ldots, V_n)$ for $n = (m+1)/2 = 2^r + 1$. It will be assumed that each V_k is a unitary operator of the form

$$V_k \in \mathrm{U}(\mathcal{Z}_{k-1} \otimes \mathcal{Y}_{k-1}, \mathcal{Z}_k \otimes \mathcal{X}_k); \tag{4.29}$$

if this is not the case, then the purification procedure described in Section 4.1.4 may be applied. It will also be assumed that every one of the registers $\mathsf{X}_1, \mathsf{Y}_1, \ldots, \mathsf{X}_{n-1}, \mathsf{Y}_{n-1}$ comprises exactly the same number of qubits, which is a constraint that is easily met by adding dummy qubits to registers as needed. Now consider the following verifier that is derived from $(V_1, \ldots, V_n)$.

1. Receive the pair of registers $(\mathsf{Z}_{t-1}, \mathsf{Y}_{t-1})$ from the prover, where $t = 2^{r-1} + 1$.
2. Choose a bit $a \in \{0, 1\}$ uniformly at random. If $a = 0$, the original interactive game will be run forward in time, while if $a = 1$, the original interactive game will be run backward in time. In either case, the bit a is concatenated to the first message to be sent by the verifier to the prover (so that the prover knows which direction in time the game will be run).

   Forward $(a = 0)$: Operate precisely at the original verifier V operates, as if the register Y_{t-1} has just been received from the prover. The messages exchanged in the remainder of the interaction therefore correspond to the registers X_t, Y_t, $\ldots$, X_{n-1}, Y_{n-1}. The acceptance condition for V' is the same as that of V.

   Backward $(a = 1)$: Send Y_{t-1} back to the prover. Each subsequent action of V' is the inverse of an action of V, and the actions are taken in the reverse order. In the turn immediately after Y_{t-1} is sent back to the prover, the verifier V' expects to receive X_{t-1}, it applies V_{t-1}^{-1} to $(\mathsf{Z}_{t-1}, \mathsf{X}_{t-1})$, obtaining $(\mathsf{Z}_{t-2}, \mathsf{Y}_{t-2})$, and sends Y_{t-2} to the prover. This pattern continues until the verifier receives the register X_1. The overall sequence of messages exchanged in this case therefore corresponds to $\mathsf{Y}_{t-1}, \mathsf{X}_{t-1}, \ldots, \mathsf{Y}_1, \mathsf{X}_1$. The verifier applies

V_1^{-1} to the pair $(\mathsf{Z}_1, \mathsf{X}_1)$, obtaining $(\mathsf{Z}_0, \mathsf{Y}_0)$, and outputs 1 (accept) if and only if a measurement of each qubit of Z_0 in the standard basis yields 0.

In the case that the random bit a is equal to 0, the total number of turns in the protocol is $2(n-t)+1$ while the number of turns is $2(t-1)+1$ in case $a=1$. One has $2(n-t)+1 = 2^r+1 = 2(t-1)+1$, and therefore V' is a (2^r+1)-turn verifier.

It remains to consider the relationship between $\omega(V)$ and $\omega(V')$. It is evident that, if $\omega(V)=1$, then $\omega(V')=1$ as well, for if there exists a prover P that causes V to output 1 with certainty, then one may obtain a prover P' causing V' to output 1 with certainty by adapting a unitary purification of P in the most straightforward way. That is, P' initially prepares the registers $(\mathsf{Z}_{t-1}, \mathsf{Y}_{t-1}, \mathsf{W}_{t-1})$ in the pure state in which they would have been, had the unitary purification of P interacted with V up to this point in the game; and then P' runs the unitary purification of P forward or backward appropriately.

In the case that $\omega(V)$ is smaller than 1, we may obtain an upper bound on $\omega(V')$ by focusing on the possible states of the register Z_{t-1}, over all possible choices of a prover interacting with V. To be clear, we are considering the possible states of Z_{t-1} viewed in isolation, which will generally be mixed states; Y_{t-1} and W_{t-1} are to be viewed as having been traced out. Let us, in particular, consider two sets of states $\mathcal{C}_0, \mathcal{C}_1 \subseteq \mathrm{D}(\mathcal{Z}_{t-1})$ of the register Z_{t-1}. The set $\mathcal{C}_0$ represents all possible states of this register that could be reached by some prover interacting with V, while $\mathcal{C}_1$ represents all possible states of this register that could, under the actions of some possibly different prover, lead to V outputting 1 with certainty. A fairly direct application of Uhlmann's theorem (Theorem 2.3) reveals the expression

$$\omega(V) = \max\Bigl\{\mathrm{F}(\sigma_0, \sigma_1)^2 \;:\; \sigma_0 \in \mathcal{C}_0,\; \sigma_1 \in \mathcal{C}_1\Bigr\}. \tag{4.30}$$

Now, a prover interacting with V' must make an initial choice for the state of the register Z_{t-1}, and a similar reasoning reveals that the value $\omega(V')$ of V' is given by the expression

$$\max\biggl\{\frac{\mathrm{F}(\sigma_0, \rho)^2 + \mathrm{F}(\sigma_1, \rho)^2}{2} \;:\; \sigma_0 \in \mathcal{C}_0,\; \sigma_1 \in \mathcal{C}_1,\; \rho \in \mathrm{D}(\mathcal{Z}_{t-1})\biggr\}. \tag{4.31}$$

Maximizing over ρ, one obtains

$$\begin{aligned}\omega(V') &= \frac{1}{2} + \frac{1}{2}\max\Bigl\{\mathrm{F}(\sigma_0,\sigma_1) \,:\, \sigma_0 \in \mathcal{C}_0,\ \sigma_1 \in \mathcal{C}_1\Bigr\}\\ &= \frac{1}{2} + \frac{1}{2}\sqrt{\omega(V)}\end{aligned} \tag{4.32}$$

by a sum-of-squares relationship for the fidelity function,

$$\max_{\rho}\Bigl(\mathrm{F}(\sigma_0,\rho)^2 + \mathrm{F}(\sigma_1,\rho)^2\Bigr) = 1 + \mathrm{F}(\rho_0,\rho_1), \tag{4.33}$$

due to Spekkens and Rudolph [154]. If it is the case that $\omega(V) \leq 1-\varepsilon$ for some choice of $\varepsilon > 0$, then it follows that

$$\omega(V') \leq \frac{1}{2} + \frac{1}{2}\sqrt{1-\varepsilon} \leq 1 - \frac{\varepsilon}{4}. \tag{4.34}$$

When this method is applied iteratively r times, a three-turn verifier V' is obtained that satisfies

$$\omega(V') \leq 1 - \frac{\varepsilon}{4^r} \leq 1 - \frac{\varepsilon}{m^2}. \tag{4.35}$$

(The second inequality also accounts for the possibility that dummy turns were initially added to V.) As each iteration of the procedure described above results in at most a constant factor increase in the size of the description of the verifier, iterating it r times gives a polynomial-time procedure.

Theorem 4.3. For every polynomially bounded function m and every function $\varepsilon : \mathbb{N} \rightarrow [0,1]$, it holds that

$$\mathrm{QIP}_{1,1-\varepsilon}(m) \subseteq \mathrm{QIP}_{1,1-\delta}(3) \tag{4.36}$$

for $\delta = \varepsilon/m^2$.

The transformation from V to V' described above can be applied to a three-turn verifier V. In the interactive game that results, the verifier V' will again be a three-turn verifier, but it will also have the interesting property that the only message it sends to the prover is the single random bit a. The prover sends the register Z_1 to V' as its first message, and the register Y_1 (in case $a = 0$) or X_1 (in case $a = 1$) is sent in the third turn. The result is that any problem in QIP also has a three-turn *public-coin* quantum interactive proof system:

$$\text{QIP} = \text{QMAM}, \tag{4.37}$$

where QMAM is the class of promise problems having three-turn public-coin quantum interactive proofs. We will return to this transformation in the context of multi-prover interactive games in Chapter 6.

4.3 SDPs for interactive games and parallel repetition

This section explains how the optimization over all prover strategies in an interactive game may be expressed as a semidefinite program. When a generic semidefinite programming algorithm (such as the ellipsoid algorithm) is applied to such a semidefinite programming formulation, the relation

$$\text{QIP} \subseteq \text{EXP} \tag{4.38}$$

is easily obtained. (The semidefinite programs obtained from a given problem $A \in$ QIP are of size exponential in the input to A, so a polynomial-time algorithm for solving semidefinite programs gives an exponential time algorithm for A.) The relation (4.38) can be improved to

$$\text{QIP} = \text{PSPACE}, \tag{4.39}$$

as will be explained in the next section—but the underlying ideas behind this result are closely connected with the semidefinite programming formulation to be discussed below.

A different fact that emerges from this semidefinite programming formulation is that single-prover quantum interactive proof systems having perfect completeness possess the property of perfect parallel repetition. This is proved through semidefinite programming duality and will be described later in this section.

Semidefinite programs for optimizing over prover strategies

Consider an arbitrary verifier in an interactive game having any number of turns. One may consider an optimization over all possible prover strategies against this verifier—we take the probability that the verifier outputs 1 as the objective function to be maximized, so that the optimal value is $\omega(V)$.

There are two distinct formulations of this optimization problem as a semidefinite program that are known. We will focus on just one of these formulations, in which the variables of a semidefinite program represent states of the various registers of the interactive game at different moments in time. (The other formulation uses the variables of a semidefinite program to represent the prover's actions through the use of the Choi representation of channels.)

Assume hereafter that an m-turn verifier V has been fixed. It will be assumed that this verifier has been purified, as discussed previously, so that V is described by an n-tuple $(V_1, \ldots, V_n)$ of linear isometries, for $n = \lfloor m/2 + 1 \rfloor$, and where each V_k takes the form

$$V_k \in \mathrm{U}(\mathcal{Z}_{k-1} \otimes \mathcal{Y}_{k-1}, \mathcal{Z}_k \otimes \mathcal{X}_k). \tag{4.40}$$

In accordance with our default assumption, the registers Z_0 and X_n are taken to be trivial—the verifier starts with no memory at the beginning of the interaction and sends no message to the prover immediately before making its final decision—so that $\mathcal{Z}_0 = \mathbb{C}$ and $\mathcal{X}_n = \mathbb{C}$. It is irrelevant to the present discussion whether the number of turns m is even or odd; in the interest of generality, one may assume that an arbitrary prover P that interacts with V is described by channels $(P_0, \ldots, P_n)$, with Y_0 being a trivial register in case the number of turns happens to be even.

Now consider the possible states of the registers

$$\mathsf{Z}_1, \ldots, \mathsf{Z}_n, \quad \mathsf{Y}_0, \ldots, \mathsf{Y}_{n-1}, \quad \text{and} \quad \mathsf{X}_1, \ldots, \mathsf{X}_{n-1}, \tag{4.41}$$

taken in groups of one or two for which the registers are co-existing, in an interaction between V and an arbitrary prover P. For instance, in Figure 4.5, one may consider the states of Y_0, $(\mathsf{Z}_1, \mathsf{X}_1)$, $(\mathsf{Z}_1, \mathsf{Y}_1)$, $(\mathsf{Z}_2, \mathsf{X}_2)$, $(\mathsf{Z}_2, \mathsf{Y}_2)$, and Z_3 in isolation. Let us choose names to represent these possible states as follows: the states immediately prior to verifier actions will correspond to density operators

$$\sigma_0 \in \mathrm{D}(\mathcal{Y}_0),\ \sigma_1 \in \mathrm{D}(\mathcal{Z}_1 \otimes \mathcal{Y}_1),\ \ldots,\ \sigma_{n-1} \in \mathrm{D}(\mathcal{Z}_{n-1} \otimes \mathcal{Y}_{n-1}), \tag{4.42}$$

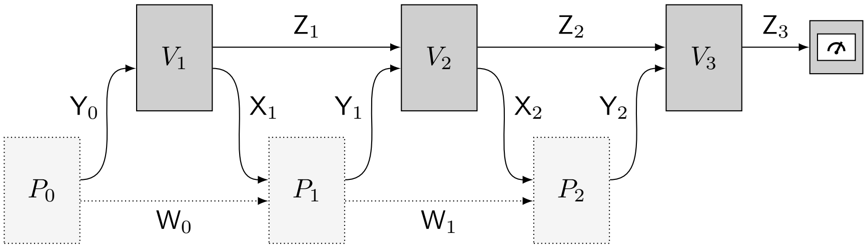


Figure 4.5: For a fixed verifier in an interactive game, one may consider optimizing over all prover strategies against that verifier. The parts of the figure represented by dotted rectangles and arrows could be optimized in this example.

representing states of Y_0, $(\mathsf{Z}_1, \mathsf{Y}_1)$, ..., $(\mathsf{Z}_{n-1}, \mathsf{Y}_{n-1})$, while the states immediately after verifier actions will correspond to density operators

$$\rho_1 \in \mathrm{D}(\mathcal{Z}_1 \otimes \mathcal{X}_1), \ldots, \rho_{n-1} \in \mathrm{D}(\mathcal{Z}_{n-1} \otimes \mathcal{X}_{n-1}), \rho_n \in \mathrm{D}(\mathcal{Z}_n), \quad (4.43)$$

representing states of $(\mathsf{Z}_1, \mathsf{X}_1)$, ..., $(\mathsf{Z}_{n-1}, \mathsf{X}_{n-1})$, Z_n.

If it is the case that the states (4.42) and (4.43) truly arise from an interaction between V and a legitimate prover P, then these states must evidently obey certain simple constraints. There are two essential types of constraints, which are as follows:

1. For every choice of $k \in \{1, \ldots, n-1\}$, the state of the register Z_k, when it is viewed in isolation, must be the same with respect to both ρ_k and σ_k. This is so because a prover interacting with V cannot touch these registers. It must therefore hold that

$$\mathrm{Tr}_{\mathcal{X}_k}(\rho_k) = \mathrm{Tr}_{\mathcal{Y}_k}(\sigma_k). \quad (4.44)$$

2. For every choice of $k \in \{1, \ldots, n\}$, it must hold that

$$\rho_k = V_k \sigma_{k-1} V_k^*. \quad (4.45)$$

This is so because the transition from σ_k to ρ_k is completely determined by the verifier's action at the corresponding position in the interaction.

It is quite straightforward to see that these two types of constraints must necessarily hold when the states (4.42) and (4.43) arise from an interaction between V and some prover P. What is more remarkable is that these constraints are not only necessary but sufficient in this regard. That is, for any collection of states $\sigma_0, \ldots, \sigma_{n-1}$ and $\rho_1, \ldots, \rho_n$, having the forms (4.42) and (4.43) and satisfying the constraints (4.44) and (4.45) for all of the possible values of k indicated above, it must hold that there exists a prover P that causes these states to occur at their corresponding positions in an interaction with V.[5]

That the constraints above are indeed sufficient in the respect just described follows from the unitary equivalence of purifications (Theorem 2.1). Under the assumption that a prover holds a purification of the state of Z_k in registers $(\mathsf{X}_k, \mathsf{W}_{k-1})$, meaning that the state of $(\mathsf{Z}_k, \mathsf{X}_k, \mathsf{W}_{k-1})$ is pure, it is free to transform the state of these registers into any pure state of $(\mathsf{Z}_k, \mathsf{Y}_k, \mathsf{W}_k)$ whatsoever, provided that the state of Z_k, when viewed in isolation, does not change. In particular, the prover may transform these registers in such a way that the state of the pair $(\mathsf{Z}_k, \mathsf{Y}_k)$ has been transformed to σ_k, by virtue of the fact that the reduced state of Z_k is the same for both of these density operators. In doing this, the prover holds a purification of σ_k in the register W_k, and is ready to perform the transformation corresponding to the next step in the interaction.

It is now evident that a maximization of the probability for a prover to cause the verifier to output 1 can be represented as a semidefinite program. The probability that the verifier outputs 1 is given by a linear function $\langle \Pi, \rho_n \rangle$ of the state of Z_n, for Π being a measurement operator that corresponds to the verifier outputting 1. One could then formulate a semidefinite program that maximizes this value over all choices of density operators $\sigma_0, \ldots, \sigma_{n-1}, \rho_1, \ldots, \rho_n$ of the forms (4.42) and (4.43) satisfying the constraints described above; density operators must be positive semidefinite and trace 1, and all of the constraints described above are linear, which allows for such a semidefinite program.

In the interests of simplicity, one may omit the variables corresponding to the states $\rho_1, \ldots, \rho_n$, as these states are determined by $\sigma_0, \ldots, \sigma_{n-1}$. We obtain the semidefinite program whose primal form

[5] The reader is cautioned this statement is very much reliant on the assumption that V has been purified: $V_1, \ldots, V_n$ are isometries and not general channels.

$$
\begin{aligned}
\text{maximize:}\quad & \langle V_n^* \Pi V_n, \sigma_{n-1} \rangle \\
\text{subject to:}\quad & \operatorname{Tr}(\sigma_0) = 1, \\
& \operatorname{Tr}_{\mathcal{Y}_1}(\sigma_1) = \operatorname{Tr}_{\mathcal{X}_1}(V_1 \sigma_0 V_1^*), \\
& \qquad \vdots \\
& \operatorname{Tr}_{\mathcal{Y}_{n-1}}(\sigma_{n-1}) = \operatorname{Tr}_{\mathcal{X}_{n-1}}(V_{n-1} \sigma_{n-2} V_{n-1}^*), \\
& \sigma_0 \in \operatorname{Pos}(\mathcal{Y}_0) \\
& \sigma_1 \in \operatorname{Pos}(\mathcal{Z}_1 \otimes \mathcal{Y}_1), \\
& \qquad \vdots \\
& \sigma_{n-1} \in \operatorname{Pos}(\mathcal{Z}_{n-1} \otimes \mathcal{Y}_{n-1}).
\end{aligned}
$$

Figure 4.6: Primal form of a semidefinite program for computing $\omega(V)$.

is given in Figure 4.6. One may compute that the corresponding dual form of this semidefinite program is as given in Figure 4.7.

It is evident that strong duality holds for this semidefinite program; by choosing the dual variables $\lambda, Z_1, \ldots, Z_{n-1}$ to be suitably large scalar multiples of the identity, a strictly feasible dual solution is obtained, which leads to strong duality by Slater's theorem.

Parallel repetition

Consider the situation in which a prover plays two independent interactive games, as suggested by Figure 4.8. The property of *parallel repetition* concerns the optimal probability that a prover may win both games simultaneously, and how this optimal probability compares with the optimal probabilities with which the two games may be won individually. One may also consider the situation in which three or more games are played in parallel, but once the behavior is understood for two independent games, the general case will follow (either by induction or by generalizing the methodology in the most straightforward way).

$$
\begin{aligned}
\text{minimize:}\quad & \lambda \\
\text{subject to:}\quad & \lambda \mathbb{1}_{\mathcal{Y}_0} \geq V_1^*(Z_1 \otimes \mathbb{1}_{\mathcal{X}_1})V_1, \\
& Z_1 \otimes \mathbb{1}_{\mathcal{Y}_1} \geq V_2^*(Z_2 \otimes \mathbb{1}_{\mathcal{X}_2})V_2, \\
& \qquad \vdots \\
& Z_{n-2} \otimes \mathbb{1}_{\mathcal{Y}_{n-2}} \geq V_{n-1}^*(Z_{n-1} \otimes \mathbb{1}_{\mathcal{X}_{n-1}})V_{n-1}, \\
& Z_{n-1} \otimes \mathbb{1}_{\mathcal{Y}_{n-1}} \geq V_n^* \Pi V_n, \\
& \lambda \in \mathbb{R} \\
& Z_1 \in \mathrm{Herm}(\mathcal{Z}_1), \\
& \qquad \vdots \\
& Z_{n-1} \in \mathrm{Herm}(\mathcal{Z}_{n-1}).
\end{aligned}
$$

Figure 4.7: Dual form of a semidefinite program for computing $\omega(V)$.

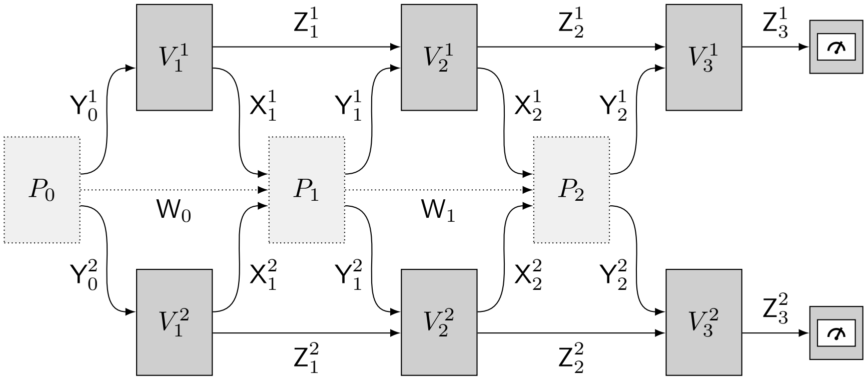


Figure 4.8: A prover P, described by channels P_0, P_1, and P_2 plays two interactive games simultaneously, one against a verifier described by (V_1^1, V_2^1, V_3^1) and the other described by (V_1^2, V_2^2, V_3^2). While the two verifiers behave independently, there is nothing that forces the prover P to treat the two games independently.

In more precise terms, suppose that two verifiers $V^1 = (V_1^1, \ldots, V_n^1)$ and $V^2 = (V_1^2, \ldots, V_n^2)$ for interactive games are given. It is assumed that the verifiers agree on the number of turns they exchange with a prover for simplicity—and if the two verifiers did not agree on the number of turns, there would be nothing lost in the discussion that follows by adding dummy turns to one of them in order to meet this condition. Let us write[6] $V^1 \otimes V^2$ to denote the verifier obtained by running V^1 and V^2 in parallel, and defining the output bit of the combined prover to be the AND of the output bits of V^1 and V^2.

It is evident that the optimal acceptance probability of the verifiers V^1, V^2, and $V^1 \otimes V^2$ satisfies

$$\omega(V^1 \otimes V^2) \geq \omega(V^1)\omega(V^2), \tag{4.46}$$

as a prover may achieve the acceptance probability $\omega(V^1)\omega(V^2)$ against $V^1 \otimes V^2$ simply by playing optimally and independently against V^1 and V^2. Given that a prover need not treat the two games independently, it is not obvious that this inequality can be replaced by an equality in general. This is, however, the case: for every choice of verifiers V^1 and V^2, one has

$$\omega(V^1 \otimes V^2) = \omega(V^1)\omega(V^2). \tag{4.47}$$

More generally, for $V^1 \otimes \cdots \otimes V^k$ being a combined verifier defined from any choice of verifiers $V^1, \ldots, V^k$ in the most natural way, one has

$$\omega(V^1 \otimes \cdots \otimes V^k) = \omega(V^1) \cdots \omega(V^k). \tag{4.48}$$

One way to prove that this is so is to use semidefinite programming duality. Consider the dual form of the semidefinite program for the optimal acceptance probabilities $\omega(V^1)$, $\omega(V^2)$, and $\omega(V^1 \otimes V^2)$. For any choice of dual-feasible solutions to the semidefinite programs representing $\omega(V^1)$ and $\omega(V^2)$, which may be denoted

$$(\lambda^1, Z_1^1, \ldots, Z_{n-1}^1) \quad \text{and} \quad (\lambda^2, Z_1^2, \ldots, Z_{n-1}^2) \tag{4.49}$$

[6] The notation $V^1 \otimes V^2$ is slightly abusive, as it is not immediate how one tensors two verifiers in a formal sense, but it is nevertheless a reasonably natural notation; each of the actions performed by this combined verifier corresponds to a tensor product of channels, and similarly for the measurement operator corresponding to acceptance.

(where superscripts represent indices, not exponents), one obtains a dual-feasible solution to the semidefinite program for $\omega(V^1 \otimes V^2)$ by taking

$$\lambda = \lambda^1 \lambda^2, \; Z_1 = Z_1^1 \otimes Z_1^2, \; \dots, \; Z_{n-1} = Z_{n-1}^1 \otimes Z_{n-1}^2. \tag{4.50}$$

The dual-feasibility of the solution $(\lambda, Z_1, \dots, Z_{n-1})$ defined in this way follows from the observation that each of the operators $Z_1^1, \dots, Z_{n-1}^1$ and $Z_1^2, \dots, Z_{n-1}^2$ must be positive semidefinite, together with the operator inequality

$$Q_1 \otimes Q_2 \geq R_1 \otimes R_2, \tag{4.51}$$

which holds provided that

$$Q_1 \geq R_1 \geq 0 \quad \text{and} \quad Q_2 \geq R_2 \geq 0. \tag{4.52}$$

The dual objective value achieved by $(\lambda, Z_1, \dots, Z_{n-1})$ is precisely $\lambda^1 \lambda^2$, and by optimizing over all dual feasible solutions (4.49) and considering that strong duality holds, one obtains

$$\omega(V^1 \otimes \cdots \otimes V^k) = \omega(V^1) \cdots \omega(V^k). \tag{4.53}$$

Based on the fact just described, the following theorem may be obtained.

Theorem 4.4. Let $a, b : \mathbb{N} \to [0,1]$ and $m : \mathbb{N} \to \mathbb{N}$ be functions. For every choice of a positive, polynomially bounded function $p : \mathbb{N} \to \mathbb{N}$, it holds that

$$\mathrm{QIP}_{a,b}(m) \subseteq \mathrm{QIP}_{a^p,b^p}(m). \tag{4.54}$$

In particular, for every choice of positive, polynomially bounded functions r and q, one has

$$\mathrm{QIP}_{1,1-1/r}(m) \subseteq \mathrm{QIP}_{1,2^{-q}}(m). \tag{4.55}$$

Corollary 4.5. It holds that $\mathrm{QIP} = \mathrm{QIP}_{1,2^{-r}}(3)$ for every positive, polynomially bounded function $r : \mathbb{N} \to \mathbb{N}$.

4.4 QIP = PSPACE

The final section of the chapter concerns the proof of the following theorem.

Theorem 4.6. QIP = PSPACE.

As a result of this theorem, together with the well-known result IP = PSPACE of Shamir [150], based partly on the work of Lund, Fortnow, Karloff, and Nisan [124], one finds that single-prover quantum interactive proof systems have precisely the same computational power as single-prover classical interactive proof systems.

Indeed, because IP = PSPACE and IP $\subseteq$ QIP, one of the containments required to prove Theorem 4.6 follows immediately, namely PSPACE $\subseteq$ QIP. The main focus of the present section is on the reverse containment, which is QIP $\subseteq$ PSPACE.

Theorem 4.6 was first proved by Jain, Ji, Upadhyay, and Watrous [96]. The proof we present below makes use of a simplification due to Wu [171]. The overall structure of the two proofs are the same, but at a technical level Wu's formulation has a significant advantage, in that it replaces a more complicated multiplicative weights update algorithm for a semidefinite program with a simpler one for a min-max problem.

4.4.1 Reduction of QIP to a min-max value computation

The first step in the proof that QIP = PSPACE concerns the relationship between interactive games and a certain type of min-max problem to be solved by a polynomial-space algorithm. By Corollary 4.5, which implies that QIP = $\text{QIP}_{1,\delta}(3)$ for any choice of a constant $\delta \in (0,1)$, we may restrict our attention to three-turn interactive games.

Consider an arbitrary three-turn verifier, given by its quantum circuit encoding. Through the purification process described earlier, one may efficiently process the description of such a verifier's circuits to obtain a unitary circuit description of a verifier $V = (V_1, V_2)$ as suggested by Figure 4.9. Here, it is assumed that Z_0 comprises all of the ancillary qubits needed by the verifier's computations, while Z_2 represents a single qubit, which is the output qubit of the interactive game

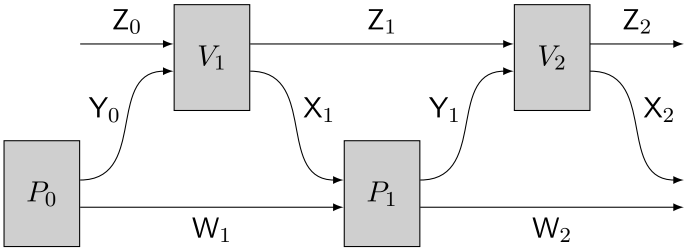


Figure 4.9: A three-turn interactive game with a unitary verifier.

to be measured with respect to the standard basis. Ordinarily there is no need for the register X_2 in a three-turn interactive game, and in the present case this register need not to be interpreted as a fourth message register—it is simply a register representing all of the qubits held by the verifier, aside from the output qubit, at the end of the game. Although this register has no influence on the outcome of an interaction between V and a prover P, it will play an important role in the reduction that follows.

Next, define two channels, $\Phi_1 \in \mathrm{C}(\mathcal{Y}_0, \mathcal{Z}_1)$ and $\Phi_2 \in \mathrm{C}(\mathcal{X}_2, \mathcal{Z}_1)$, as

$$\begin{aligned} \Phi_1(\rho_1) &= \operatorname{Tr}_{\mathcal{X}_1}\bigl(V_1(|0\cdots 0\rangle\langle 0\cdots 0| \otimes \rho_1)V_1^*\bigr) \\ \Phi_2(\rho_2) &= \operatorname{Tr}_{\mathcal{Y}_1}\bigl(V_2^*(|1\rangle\langle 1| \otimes \rho_2)V_2\bigr) \end{aligned} \tag{4.56}$$

for every $\rho_1 \in \mathrm{D}(\mathcal{Y}_0)$ and $\rho_2 \in \mathrm{D}(\mathcal{X}_2)$. The action of these channels is illustrated in Figure 4.10. As was already encountered in (4.30), the maximum acceptance probability of V is given by

$$\omega(V) = \max_{\rho_1,\rho_2} \mathrm{F}(\Phi_1(\rho_1), \Phi_2(\rho_2))^2, \tag{4.57}$$

where the maximum is over all states $\rho_1 \in \mathrm{D}(\mathcal{Y}_0)$ and $\rho_2 \in \mathrm{D}(\mathcal{X}_2)$. Finally, define channels $\Psi_1, \Psi_2 \in \mathrm{C}(\mathcal{Y}_0 \otimes \mathcal{X}_2, \mathcal{Z}_1)$ as

$$\Psi_1 = \Phi_1 \otimes \operatorname{Tr} \quad \text{and} \quad \Psi_2 = \operatorname{Tr} \otimes \Phi_2, \tag{4.58}$$

where the traces are defined on $\mathcal{X}_2$ and $\mathcal{Y}_0$, respectively.

Now, for $\Xi = \Psi_1 - \Psi_2$ being the difference between these channels, one may consider the min-max quantity

$$\eta = \min_{\rho \in \mathrm{D}(\mathcal{Y}_0 \otimes \mathcal{X}_2)} \max_{\Pi \in \mathrm{Proj}(\mathcal{Z}_1)} \langle \Pi, \Xi(\rho) \rangle. \tag{4.59}$$

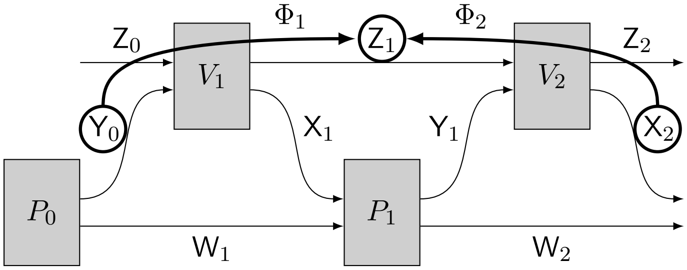


Figure 4.10: The action of the channels Φ_1 and Φ_2.

There are two relevant cases to consider.

Case 1: $\omega(V) = 1$. In this case, one may set $\rho = \rho_1 \otimes \rho_2$, for $\rho_1 \in \mathrm{D}(\mathcal{Y}_0)$ and $\rho_2 \in \mathrm{D}(\mathcal{X}_2)$ maximizing the expression on the right-hand side of (4.57). As $\mathrm{F}(\Phi_1(\rho_1), \Phi_2(\rho_2)) = 1$, it follows that $\Phi_1(\rho_1) = \Phi_2(\rho_2)$, so $\Xi(\rho) = 0$, and therefore $\eta = 0$.

Case 2: $\omega(V) \leq \delta$ for $\delta \in (0,1)$. In this case, for every choice of $\rho \in \mathrm{D}(\mathcal{Y}_0 \otimes \mathcal{X}_2)$ one has

$$\|\Xi(\rho)\|_1 = \big\|\Phi_1(\rho_1) - \Phi_2(\rho_2)\big\|_1 \geq 2 - 2\,\mathrm{F}\big(\Phi_1(\rho_1), \Phi_2(\rho_2)\big), \tag{4.60}$$

for $\rho_1 = \mathrm{Tr}_{\mathcal{X}_2}(\rho)$ and $\rho_2 = \mathrm{Tr}_{\mathcal{Y}_0}(\rho)$. The fidelity $\mathrm{F}\big(\Phi_1(\rho_1), \Phi_2(\rho_2)\big)$ can be no larger than $\sqrt{\delta}$ by (4.57), so that

$$\|\Xi(\rho)\|_1 \geq 2 - 2\sqrt{\delta}, \tag{4.61}$$

and thus, for an optimal choice of Π, one has $\langle \Pi, \Xi(\rho)\rangle \geq 1 - \sqrt{\delta}$. Therefore, one has $\eta \geq 1 - \sqrt{\delta}$ in this case.

An implication of the case analysis performed above may be stated as a lemma as follows.

Lemma 4.7. Let V be the verifier in a three-turn interactive game, and let η be the min-max quantity defined from V in (4.59). The following implications hold:

$$\big[\omega(V) = 1\big] \Rightarrow \big[\eta = 0\big] \quad \text{and} \quad \left[\omega(V) \leq \frac{1}{4}\right] \Rightarrow \left[\eta \geq \frac{1}{2}\right] \tag{4.62}$$

4.4.2 Matrix multiplicative weights update method

At the heart of the proof that QIP $\subseteq$ PSPACE is an algorithmic method known as the *matrix multiplicative weights update method.* There are a variety of known algorithms that make use of this method, which is well-suited to certain specialized forms of convex optimization problems. This subsection will present one algorithm in this family that will allow the containment QIP $\subseteq$ PSPACE to be proved, based on the min-max problem described in the previous subsection. The specific formulation of this algorithm as a PSPACE algorithm will be discussed in the next subsection.

The input to the problem is an explicit matrix description of a mapping $\Xi = \Psi_1 - \Psi_2$, which is assumed to be the difference of two channels of the form $\Psi_1, \Psi_2 \in \mathrm{C}(\mathbb{C}^N, \mathbb{C}^M)$. The goal of the algorithm will be to approximate the min-max value

$$\eta = \min_{\rho} \max_{P} \langle P, \Xi(\rho) \rangle, \tag{4.63}$$

where the minimum is over all density operators $\rho \in \mathrm{D}(\mathbb{C}^N)$ and the maximum is over all measurement operators $P \in \mathrm{Pos}(\mathbb{C}^M)$ (meaning that P satisfies $0 \leq P \leq \mathbb{1}$). As the function

$$(\rho, P) \mapsto \langle P, \Xi(\rho) \rangle \tag{4.64}$$

is bilinear and the minimum and maximum are over convex and compact sets, one may freely change the order of the minimum and maximum by Sion's min-max theorem. It is convenient to observe that

$$\eta = \min_{\rho} \max_{\Pi} \langle \Pi, \Xi(\rho) \rangle, \tag{4.65}$$

where the maximum is over all projection operators $\Pi \in \mathrm{Proj}(\mathbb{C}^M)$ (as it was in the min-max problem described in the previous subsection); when the min-max expression is viewed as a game and ρ is played first, there is nothing lost in restricting the maximization to the extreme points of the set of measurement operators (i.e., the projection operators). Along similar lines, if P were played first, one could then choose a minimizing ρ among the pure states in $\mathrm{D}(\mathbb{C}^N)$.

The accuracy of the algorithm is determined by the setting of an accuracy parameter $\varepsilon > 0$, which is left as an indeterminate value for

1. Set $X_1 = \mathbb{1}_N$ and $T = \lceil 2\ln(N)/\varepsilon^2 \rceil$.
2. For each $t = 1, \ldots, T$, let

$$\rho_t = \frac{X_t}{\operatorname{Tr}(X_t)}, \tag{4.66}$$

   let $\Pi_t \in \operatorname{Proj}(\mathbb{C}^M)$ be the projection operator corresponding to the positive eigenspace of $\Xi(\rho_t)$, and let

$$X_{t+1} = \exp\bigl(-\varepsilon\, \Xi^*(\Pi_1 + \cdots + \Pi_t)\bigr). \tag{4.67}$$

3. Output

$$\frac{1}{T}\sum_{t=1}^{T} \bigl\langle \Pi_t, \Xi(\rho_t) \bigr\rangle. \tag{4.68}$$

Figure 4.11: Matrix multiplicative weights update algorithm for approximating the min-max value η defined in (4.63).

the time being; a choice of ε that is appropriate to the application of the algorithm in proving QIP = PSPACE is made in the subsection following this one. (To obtain a cleaner expression on the accuracy of the algorithm, it is convenient to make the assumption that $\varepsilon < 1/4$.) The algorithm is described in Figure 4.11.

To analyze this algorithm, one may begin with the following technical lemma. We omit the proof, which is based on elementary and routine calculus and matrix theory.

Lemma 4.8. Let N be a positive integer, let $H \in \operatorname{Herm}(\mathbb{C}^N)$ be a Hermitian operator satisfying $\lVert H \rVert \leq 1$, and let $\rho \in \mathrm{D}(\mathbb{C}^N)$ be a density operator. For every positive real number $\varepsilon > 0$, it holds that

$$\bigl\langle \rho, \exp(-\varepsilon H) \bigr\rangle \leq \exp\bigl(-\varepsilon \exp(-2\varepsilon)\langle \rho, H\rangle\bigr) \exp\bigl(2\varepsilon \sinh(2\varepsilon)\bigr). \tag{4.69}$$

The next lemma forms the mathematical backbone of the algorithm analysis.

Lemma 4.9. Let T and N be positive integers, let $H_1, \ldots, H_T \in \operatorname{Herm}(\mathbb{C}^N)$ be Hermitian operators satisfying $\lVert H_t \rVert \leq 1$ for each

$t \in \{1, \ldots, T\}$, and let $\varepsilon > 0$. Define

$$X_1 = \mathbb{1}, \quad X_{t+1} = \exp\bigl(-\varepsilon(H_1 + \cdots + H_t)\bigr), \quad \text{and} \quad \rho_t = \frac{X_t}{\operatorname{Tr}(X_t)} \tag{4.70}$$

for each $t \in \{1, \ldots, T\}$. It holds that

$$\begin{aligned} &\lambda_{\min}(H_1 + \cdots + H_T) \\ &\qquad \geq \exp(-2\varepsilon) \sum_{t=1}^{T} \langle \rho_t, H_t \rangle - \frac{\ln(N)}{\varepsilon} - 2T \sinh(2\varepsilon). \end{aligned} \tag{4.71}$$

Proof. For each $t \in \{1, \ldots, T\}$, one has

$$\operatorname{Tr}(X_{t+1}) \leq \operatorname{Tr}\bigl(X_t \exp(-\varepsilon H_t)\bigr) = \operatorname{Tr}(X_t)\bigl\langle \rho_t, \exp(-\varepsilon H_t)\bigr\rangle \tag{4.72}$$

by a matrix inequality known as the *Golden–Thompson inequality*, which states that $\operatorname{Tr}(\exp(A+B)) \leq \operatorname{Tr}(\exp(A)\exp(B))$ for every choice of Hermitian operators A and B. By applying this inequality repeatedly, and noting that $\operatorname{Tr}(X_1) = N$, one finds that

$$\operatorname{Tr}(X_{T+1}) \leq N \prod_{t=1}^{T} \bigl\langle \rho_t, \exp(-\varepsilon H_t)\bigr\rangle. \tag{4.73}$$

Because the trace of a positive semidefinite operator is at least as large as its largest eigenvalue, it follows that

$$\operatorname{Tr}(X_{T+1}) \geq \lambda_{\max}(X_{T+1}) = \exp\bigl(-\varepsilon \lambda_{\min}(H_1 + \cdots + H_T)\bigr). \tag{4.74}$$

The required bound (4.71) is obtained by combining (4.73) and (4.74) with Lemma 4.8. □

Now let us compare the output value of the algorithm to the min-max quantity η. It is evident that

$$\eta \leq \frac{1}{T} \sum_{t=1}^{T} \langle \Pi_t, \Xi(\rho_t) \rangle; \tag{4.75}$$

as each Π_t is selected by the algorithm so that the quantity $\langle \Pi_t, \Xi(\rho_t) \rangle$ is maximized, it follows that $\eta \leq \langle \Pi_t, \Xi(\rho_t) \rangle$ for each $t \in \{1, \ldots, T\}$. An upper bound on the output value is obtained from Lemma 4.9 by setting $H_t = \Xi^*(\Pi_t)$ for each $t \in \{1, \ldots, T\}$. One has that $\|H_t\| \leq 1$

by virtue of the fact that Ξ is a difference between two channels, and so it follows that

$$\begin{aligned}\frac{1}{T}\sum_{t=1}^{T}\langle \Pi_t, \Xi(\rho_t)\rangle \leq \exp(2\varepsilon)\,\lambda_{\min}\biggl(\Xi^*\Bigl(\frac{\Pi_1+\cdots+\Pi_T}{T}\Bigr)\biggr)&\\ +\frac{\ln(N)\exp(2\varepsilon)}{\varepsilon T}+\bigl(\exp(4\varepsilon)-1\bigr).&\end{aligned} \tag{4.76}$$

One may observe that

$$\begin{aligned}&\lambda_{\min}\biggl(\Xi^*\Bigl(\frac{\Pi_1+\cdots+\Pi_T}{T}\Bigr)\biggr)\\ &\quad=\min_{\rho}\biggl\langle \Xi(\rho), \frac{\Pi_1+\cdots+\Pi_T}{T}\biggr\rangle \leq \eta,\end{aligned} \tag{4.77}$$

and therefore

$$\frac{1}{T}\sum_{t=1}^{T}\langle \Pi_t, \Xi(\rho_t)\rangle \leq \exp(2\varepsilon)\eta+\frac{\varepsilon\exp(2\varepsilon)}{2}+\bigl(\exp(4\varepsilon)-1\bigr). \tag{4.78}$$

For any choice of $\delta \leq 1$, it holds that $\exp(\delta)-1 \leq 2\delta$, and by combining this bound with the observation that $\eta \leq 1$, one obtains

$$\frac{1}{T}\sum_{t=1}^{T}\langle \Pi_t, \Xi(\rho_t)\rangle \leq \eta+16\varepsilon. \tag{4.79}$$

4.4.3 PSPACE and bounded-depth circuits

The final step in the proof that QIP = PSPACE is to analyze the complexity of an algorithm that has been suggested by the previous two subsections. In more explicit terms, it is to be assumed that $A = (A_{\text{yes}}, A_{\text{no}})$ is an arbitrary promise problem in QIP, and our goal is to prove $A \in$ PSPACE. There are two primary steps in the following algorithm, which will soon be shown to be implementable as a PSPACE algorithm for A.

1. It holds that QIP $=$ $\text{QIP}_{1,1/4}(3)$, and therefore, for a given input string x, there exists a corresponding three-turn verifier V such that

$$x \in A_{\text{yes}} \Rightarrow \omega(V) = 1 \quad \text{and} \quad x \in A_{\text{no}} \Rightarrow \omega(V) \leq \frac{1}{4}. \tag{4.80}$$

   Compute an explicit description of the mapping Ξ corresponding to V, as described in Section 4.4.1.

2. Run the matrix multiplicative weights update algorithm described in Figure 4.11 for approximating the min-max value η associated with Ξ, with sufficient precision to distinguish the cases $\eta = 0$ and $\eta \geq 1/2$ (e.g., $\varepsilon = 1/64$). *Accept* if $\eta = 0$ and *reject* if $\eta \geq 1/2$.

Of course, an explicit description of the mapping Ξ will generally have size exponential in $|x|$, so one will not obtain a PSPACE algorithm by applying the two steps above in the most straightforward way. Instead, the steps are to be implemented by bounded-depth Boolean circuits, and the implementation of the resulting algorithm as a PSPACE algorithm will follow from a circuit complexity result due to Borodin [35].

In greater detail, consider the complexity class NC, which we will take to include all functions, including predicates representing decision problems, computable by logarithmic-space uniform Boolean circuit families of polylogarithmic depth. The requirement that such a family is logarithmic-space uniform implies that its circuits are polynomial in size, and therefore represent polynomial-time computations.

We also consider a "scaled-up" variant of NC, to be denoted NC(*poly*), that consists of all functions computable by *polynomial-space uniform* families of Boolean circuits having *polynomial-depth.* (The notation $\text{NC}(2^{poly})$ has also previously been used for this class [36].) A family of circuits meeting these requirements could potentially have exponential size, and therefore does not necessarily represent an efficient computation. However, the polynomial bound on the depth of these circuits does represent a significant computational restriction. In particular, restricting our attention to decision problems, we have

$$\text{NC}(poly) = \text{PSPACE}, \tag{4.81}$$

which is the result of Borodin [35] suggested above. An appeal of this reformulation is that it allows one to make use of known parallel algorithms for performing various computational tasks when designing PSPACE algorithms.

Given that NC(*poly*) = PSPACE, we may consider that the task at hand is to prove $A \in \mathrm{NC}(\mathit{poly})$. When doing this we will make use of a property of NC and NC(*poly*), which is that functions in these classes compose well. Specifically, if F is a function in NC(*poly*) and G is a function in NC, then the composition $G \circ F$ is also in NC(*poly*). This follows from the most straightforward way of composing the families of circuits that compute F and G. Consequently, if it is proved that the first step of the algorithm above can be implemented as an NC(*poly*) computation, and the second step can be implemented as an NC computation, then their composition represents an NC(*poly*) computation.

Implementing the first step of the algorithm above as an NC(*poly*) computation turns out to be straightforward, through the use of elementary facts about quantum computations and bounded-depth circuits. For a given input string x, there are at most polynomially many quantum gates in the circuit description of the verifier V on this input, and they may be listed in the order they are to be applied in polynomial time. Expanding this list of gates into their explicit matrix representations yields a polynomial-length list of exponential-size matrices, and producing such a list is possible using an NC(*poly*) computation. From such a list of matrices, one may obtain an explicit description of Ξ through elementary matrix operations performed on the explicit matrix representations of the gates. As sums, differences, and iterated products of matrices can be implemented as NC computations, the fact that step 1 can be implemented as an NC(*poly*) computation follows.

The implementation of the second step of the algorithm as an NC computation is more involved, but the key observation underlying such an implementation is that the main loop in step 2 of the algorithm described in Figure 4.11 runs for a number of iterations that is logarithmic in N. Based on this observation, the fact that the entire algorithm can be implemented as an NC computation follows, provided that (i) each individual step of the algorithm can be implemented as an NC computation, and (ii) the storage requirements of the algorithm grow at most linearly with each iteration. By fixing ahead of time the number of bits of precision to which each matrix entry is stored, the second requirement is evidently met. The first requirement follows from these observations:

1. Elementary matrix computations can be performed in NC.
2. Matrix exponentials can be approximated with high accuracy in NC (simply by truncating the power series representation of the exponential function to polynomially many terms).
3. Positive eigenspace computations for Hermitian operators can be approximated with high accuracy using the fact that polynomial root approximation is in NC [28, 130, 31].

It turns out to be somewhat tedious, although conceptually quite straightforward, to verify that the approximations suggested above can be performed in NC with sufficient precision to yield a correct answer to the problem of determining whether $\eta = 0$ or $\eta \geq 1/2$. Using Lemma 4.7 this finishes the proof of Theorem 4.6. Interested readers may find further details in [96].

4.4.4 A complete problem for QIP

As a simple byproduct of one part of the analysis described above in Section 4.4.1, one finds that there is an interesting complete promise problem for the class QIP, called the *quantum circuit distinguishability* problem.

Given that QIP = PSPACE, it holds that every PSPACE-complete problem is QIP-complete (and by the same reasoning the quantum circuit distinguishability problem to be described shortly is also PSPACE-complete). It is useful nevertheless to highlight this particular problem for two reasons. One reason is that the quantum circuit distinguishability problem is a fairly natural problem (within the setting of quantum information and computation) having a more direct connection to the quantum interactive proof system model than other known PSPACE-complete problems. Indeed, the problem was known to be complete for

QIP for quite some time before it was known that QIP = PSPACE. The second reason is that variants of this problem can, in some cases, be shown to be complete for other complexity classes based on quantum interactive proof systems. One example, the Non-Identity Check problem, was already encountered in the previous chapter. Another example, concerning zero-knowledge quantum interactive proof systems, will play an important role in the chapter following this one.

The quantum circuit distinguishability problem is parameterized by two real numbers, α and β, satisfying $0 \leq \beta < \alpha \leq 1$. One may take α and β to be fixed constants or functions of the input length. The statement of the problem is as follows.

(α, β)-quantum circuit distinguishability ((α, β)-QCD)

Input: Quantum circuits Q_0 and Q_1, agreeing on both the number of input qubits they take and on the number of output qubits they produce.

Yes: $\frac{1}{2}\|Q_0 - Q_1\|_{\diamond} \geq \alpha$.

No: $\frac{1}{2}\|Q_0 - Q_1\|_{\diamond} \leq \beta$.

The fact that the quantum circuit distinguishability problem is in QIP (for a wide range of choices of α and β) may be shown directly through a common sort of interactive proof system reminiscent of a "blind taste-test." In essence, the verifier chooses a bit $a \in \{0,1\}$ uniformly at random, and agrees to perform for the prover one evaluation of the circuit Q_a, for whichever value of a was randomly selected. The verifier accepts if and only if the prover successfully identifies the value of a after the circuit evaluation is performed on an input of the prover's choice and the output qubits are returned to the prover. The maximum acceptance probability is precisely

$$\frac{1}{2} + \frac{1}{4}\|Q_0 - Q_1\|_{\diamond}, \tag{4.82}$$

yielding a quantum interactive proof system for the quantum circuit distinguishability problem with completeness and soundness bounds $(1+\alpha)/2$ and $(1+\beta)/2$, respectively.

One way to prove that the quantum circuit distinguishability problem is complete for QIP is to first observe that the following *close circuit images* problem is QIP-complete.

(α, β)-close circuit images ((α, β)-CCI)

Input: Quantum circuits R_0 and R_1, agreeing on both the number of input qubits they take and on the number of output qubits they produce.

Yes: $\mathrm{F}(R_0(\rho_0), R_1(\rho_1))^2 \geq \alpha$ for some choice of input states ρ_0 and ρ_1.

No: $\mathrm{F}(R_0(\rho_0), R_1(\rho_1))^2 \leq \beta$ for all choices of input states ρ_0 and ρ_1.

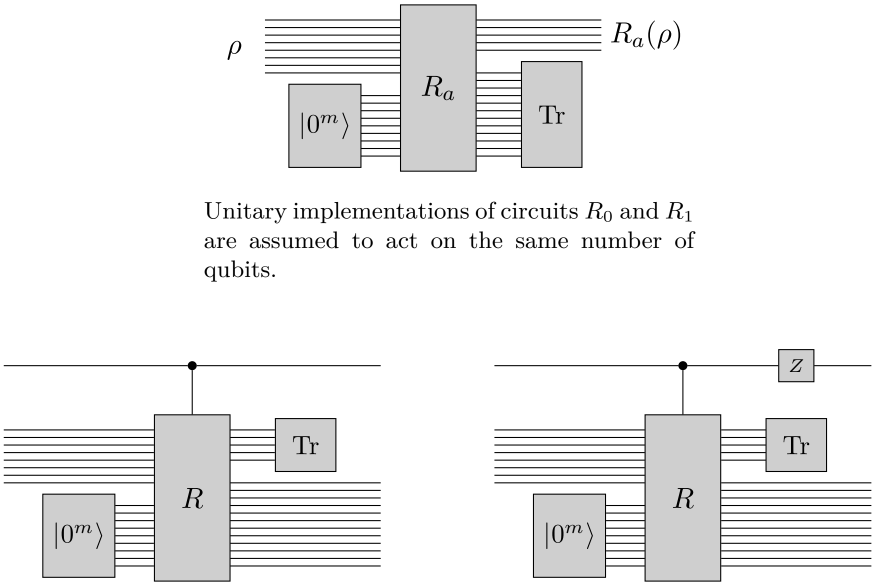
It follows from the analysis in Section 4.4.1 that this problem is QIP-complete for any choice of constants α and β satisfying $0 < \beta < \alpha \leq 1$. It is also complete for α and β being functions of the input length, provided that α and β can be computed in polynomial time, that α and β are separated by an inverse polynomial gap, and that β is at least inverse exponentially large.

Now, to prove that the quantum circuit distinguishability problem is QIP-complete, it suffices to exhibit a reduction to it from the close images problem. The reduction itself is quite simple, and is illustrated in Figure 4.12. From an instance of the close images problem, one first creates a controlled-unitary implementation of the two input circuits R_0 and R_1. The circuits Q_0 and Q_1, representing an instance of the quantum circuit distinguishability problem, take this control qubit as both an input and output, but reverse the roles of the output qubits and "garbage" qubits of both R_0 and R_1. The circuits Q_0 and Q_1 differ only by a

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{4.83}$$

operation being applied to the control qubit for one of the two circuits.

Intuitively speaking, Q_0 and Q_1 differ significantly precisely when R_0 and R_1 can be forced to output similar states—for this is the only



Unitary implementations of circuits R_0 and R_1 are assumed to act on the same number of qubits.

The circuit Q_0.

The circuit Q_1.

Figure 4.12: The circuit construction used to prove the QIP-completeness of the quantum circuit distinguishability problem. The circuits Q_0 and Q_1 are constructed from a controlled-unitary circuit implementation of R_0/R_1. The circuits Q_0 and Q_1 differ only in the Z gate on the control qubit.

way to produce a “coherent” superposition of $|0\rangle$ and $|1\rangle$ on the control qubit, allowing the Z gate to act nontrivially. An analysis reveals that

$$\frac{1}{2}\big\|Q_0 - Q_1\big\|_{\diamond} = \max_{\rho_0,\rho_1} \mathrm{F}(R_0(\rho_0), R_1(\rho_1)), \tag{4.84}$$

where the maximum is over all input states to R_0 and R_1 (and where the exponent 2 has intentionally been omitted on the right-hand side expression), which directly translates to the required completeness and soundness conditions being met.

4.5 Chapter notes

Quantum interactive proof systems were first proposed and studied in [165], where it was proved that PSPACE $\subseteq$ QIP(3). (A journal version of this paper appeared later as [168].) This result was subsumed shortly after by the results of Kitaev and Watrous [114], who proved the perfect completeness, parallelization to three turns, and parallel repetition results described in Sections 4.2 and 4.3, as well as the upper-bound QIP $\subseteq$ EXP through semidefinite programming. The parallelization proof presented in this chapter is due to Kempe, Kobayashi, Matsumoto, and Vidick [109], while the parallel repetition proof presented was inspired by the work of Mittal and Szegedy [128] on product properties of semidefinite programs. It is an open question whether the parallelization results could be extended to interactive game with only two turns, and aside from the obvious inclusions QMA $\subseteq$ QIP(2) $\subseteq$ QIP(3) few results are known concerning QIP(2).

The public-coin variant of quantum interactive proof systems suggested in the text, where Arthur (the verifier) generates random coinflips and Merlin (the prover) sends quantum information to Arthur, alternating in turns, and finally Arthur performs a measurement on all of the quantum information sent by Merlin, was considered by Marriott and Watrous [125] and shown to be equivalent in power to the ordinary quantum interactive proof system model.

The equality QIP = PSPACE was proved by Jain, Ji, Upadhyay, and Watrous [96], following related but weaker results of Jain and Watrous [100] and Jain, Upadhyay, and Watrous [99]. In particular, the paper of Jain, Upadhyay, and Watrous proved the containment $\text{QIP}(2) \subseteq \text{PSPACE}$. As stated in the text, the proof of $\text{QIP} = \text{PSPACE}$ presented in this chapter makes use of simplifications due to Wu [171]. The specific formulation of the matrix multiplicative weights update method upon which the proof that $\text{QIP} = \text{PSPACE}$ relies was discovered independently by Warmuth and Kuzmin [164] and Arora and Kale [16]. Readers interested in learning more about the matrix multiplicative weights update method, as well as a bit of its history, are referred to Kale's PhD thesis [104]. Quantum interactive proof systems with unbounded error were considered in [94] and shown to be equivalent in power to EXP, which suggests (unless $\text{PSPACE} = \text{EXP}$) that a nonnegligible separation between the completeness and soundness probability bounds in the definition of QIP is essential to the proof of $\text{QIP} \subseteq \text{PSPACE}$. (In contrast, the containment $\text{IP} \subseteq \text{PSPACE}$ does not rely on a similar assumption.)

The quantum circuit distinguishability problem was proved to be complete for QIP by Rosgen and Watrous [147], and the proof of this fact that has been summarized in this chapter makes use of a simplification due to Kobayashi (through a personal communication). Further work due to Rosgen on the computational hardness of distinguishing between restricted classes of quantum channels appears in [144, 145, 146]. Another example of a complete promise problem for QIP relating to quantum channels appears in [90]. Other problems of a similar nature are known to be complete for the class QSZK, which is discussed in the chapter following this one.

Several interesting facts are known to hold concerning *competing prover* quantum interactive proof systems, where a *yes-prover* and *no-prover* compete for the verifier's decision in the most natural way. Proof systems of this sort were first considered by Gutoski and Watrous [83] and studied further in [80, 81, 84, 85]. In particular, [84] proved that the class QRG (short for *quantum refereed games*) representing problems for which a bounded-error competing-prover quantum interactive proof system exists coincides with EXP; while [85] extended the machinery used to prove $\text{QIP} = \text{PSPACE}$ to obtain the result $\text{QRG}(2) = \text{PSPACE}$, exactly characterizing the complexity of two-turn competing-prover quantum interactive proof systems.

Quantum interactive proof systems in which only logarithmically many qubits are exchanged between the prover and verifier were considered in [25] and shown to have only the power of BQP, thereby offering no significant computational advantages over quantum computers that do not interact with a prover.

5

Quantum Zero-Knowledge

Some interactive proof systems possess the property of being *zero-knowledge*, which means that the verifier can learn nothing (or almost nothing) from an interaction with the prover, beyond the validity of the statement being proved. Although it might initially seem paradoxical, or perhaps impossible to fulfill, there are many interesting interactive proof systems that possess this property.

Perhaps the most well-known example is the Goldreich–Micali–Wigderson graph isomorphism proof system described in Figure 5.1. When one considers this proof system, it is intuitively clear that a verifier can learn nothing from an interaction with the prover on a positive instance of the problem, even if it "cheats" by deviating from the proof system—for all it sees during an execution of the proof system is an isomorphism between one of two input graphs and a randomly chosen permutation of that graph. To prove in a more formal sense that this proof system is zero-knowledge, one defines a polynomial-time *simulator* that is capable of mimicking anything that a *cheating verifier* could compute by means of an interaction with the prover (under the assumption that the input graphs are isomorphic). Because the simulator runs in polynomial time and (by assumption) does not interact with the

The input is a pair (G_0, G_1) of simple, undirected n-vertex graphs. It is assumed that the prover knows a permutation $\sigma \in S_n$, for S_n denoting the symmetric group on indices $\{1, \ldots, n\}$, that satisfies $\sigma(G_1) = G_0$ if G_0 and G_1 are isomorphic.

Prover's step 1: Choose a permutation $\pi \in S_n$ uniformly at random and send the graph $H = \pi(G_0)$ to the verifier.

Verifier's step 1: Choose $a \in \{0, 1\}$ uniformly at random and send a to the prover. (Implicitly, the verifier is challenging the prover to exhibit an isomorphism between G_a and H.)

Prover's step 2: Set $\tau = \pi\sigma^a$ and send τ to the verifier. (If it is the case that $\sigma(G_1) = G_0$, then it must hold that $\tau(G_a) = H$.)

Verifier's step 2: Accept if $\tau(G_a) = H$, reject otherwise.

Figure 5.1: The Goldreich–Micali–Wigderson graph isomorphism proof system.

prover, one interprets that no "knowledge" about the input graphs is revealed by the prover—whatever a cheating verifier could have learned from the interaction could equally well have been computed efficiently without the prover's help.

In the classical setting, the construction of such a simulator is fairly straightforward in the case of the Goldreich–Micali–Wigderson graph isomorphism proof system: it randomly guesses the cheating verifier's challenge $a \in \{0, 1\}$, computes the graph H that satisfies $\tau(H) = G_a$ for a random permutation τ, and hopes the cheating verifier issues the challenge a when sent the graph H. If it does, the simulator can output the prover's correct response, which is τ. Otherwise the simulation has failed, but in this case one can simply "rewind" the simulator and try again, repeating the process with a new choice of a random guess a, a random permutation τ, and so on. An analysis reveals that the simulator's guess must agree with the cheating verifier's challenge with probability $1/2$ (assuming a uniform selection of a and τ), and conditioned on a correct guess the output of the simulation will be in perfect agreement with the cheating verifier/prover interaction.

The quantum setting brings a new challenge to the subject of zero-knowledge, as compared with the classical setting. The algorithmic capabilities of quantum attackers (including the ability to efficiently factor and compute discrete logarithms using Shor's algorithms) represents one aspect of this challenge. More problematic from an analytic viewpoint is the failure of basic classical techniques used in the study of zero-knowledge, such as the *rewinding* technique suggested above, to carry over directly to the quantum setting. (In particular, if a cheating verifier begins its attack holding a potentially useful quantum state, it could be that an unsuccessful simulation will have irreparably damaged that state, ruling out a straightforward rewinding approach like the one suggested above.) Indeed, for some time even the simplest examples of classical zero-knowledge proof systems, such as the Goldreich–Micali–Wigderson graph isomorphism proof system, were not known to be zero-knowledge against quantum attacks, leading some to question whether the zero-knowledge property could ever be established in the quantum setting (aside from trivial cases in which the zero-knowledge property holds vacuously).

The present chapter surveys some of the definitions and known facts concerning quantum zero-knowledge proof systems. Our focus will be on a variant of zero-knowledge known as *statistical* zero-knowledge, which demands an information-theoretic security condition as a part of its definition. The other fundamental variant is *computational* zero-knowledge, which relaxes the security condition in a natural complexity-theoretic way. While the notion of quantum computational zero-knowledge is certainly well-motivated (see, for instance, [117] for work in this direction) it is reasonable to claim that most of the uniquely quantum aspects of zero-knowledge that are currently known are well-represented by the study of statistical zero-knowledge.[1]

[1] There would also be a further overhead required to develop the requisite definitions and facts for a proper discussion of quantum computational zero-knowledge, so limiting our attention to quantum statistical zero-knowledge also serves to simplify the chapter in this respect.

Two highlights of the results to be described in this chapter are as follows:

1. A fact known as the *quantum rewinding lemma* may be used to prove that some simple interactive proof systems are zero-knowledge against general quantum verifier attacks. This is illustrated for the Goldreich–Micali–Wigderson graph isomorphism proof system introduced above.
2. It is proved that a promise problem called the *close quantum states* problem is complete for the class QSZK of problems having quantum statistical zero-knowledge proof systems. The completeness of this problem allows one to conclude various facts, including the fact that every quantum statistical zero-knowledge proof system can be parallelized to just two turns (so that QSZK $\subseteq$ QIP(2)).

5.1 Definitions of zero-knowledge

As suggested above, the two most fundamental variants of zero-knowledge are *statistical* zero-knowledge and *computational* zero-knowledge. In addition, there are two specific formulations of the zero-knowledge property in both cases: *general verifier* zero-knowledge and *honest verifier* zero-knowledge. Intuitively speaking, the distinction concerns how hard the verifier, now thought of as an "adversary" to the proof system, might try to extract knowledge from a prover.

General verifier zero-knowledge is the more cryptographically meaningful of the two formulations, as one is required to prove security against arbitrary attacks (made by computationally restricted adversaries) in this case. Honest verifier zero-knowledge, on the other hand, only requires that a verifier (or someone looking over the verifier's shoulder) learns nothing by interacting with a prover, assuming that the verifier does not deviate from a fixed specification of how it is supposed to behave.

The motivation for considering honest verifier zero-knowledge is that it is generally much easier to reason about. It may be viewed as a relaxation of the general verifier definition, so any limitation that one proves on honest-verifier zero-knowledge proof systems must also

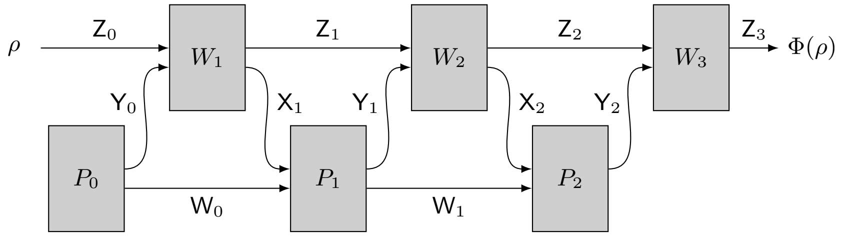


Figure 5.2: A cheating verifier $W = (W_1, \ldots, W_n)$ implements a channel Φ by means of an interaction with a prover $P = (P_0, \ldots, P_{n-1})$.

hold for general-verifier zero-knowledge proof systems. In the particular case of statistical zero-knowledge (both in the classical and quantum settings), it turns out that the honest verifier and general verifier definitions lead to identical complexity classes.

5.1.1 General verifier zero-knowledge

Suppose that $P = (P_0, \ldots, P_{n-1})$ is a prover in an interactive game. Under ordinary circumstances, it is to be expected that a fixed verifier V engages in an interaction with P and outputs a single binary value, indicating acceptance or rejection. In the context of zero-knowledge, one considers not just the interaction of such a verifier V with P, but of other entities (to be called *cheating verifiers*, as was done at the beginning of the chapter) that aim to extract "knowledge" from P. Figure 5.2 illustrates a hypothetical cheating verifier W interacting with a prover $P = (P_0, P_1, P_2)$. The cheating verifier W does not necessarily output 0 or 1 like an ordinary verifier, but instead aims to implement some channel Φ by means of an interaction with P.

There is not a formal definition of "knowledge" in this situation, and the term is placed in quotes to emphasize this point. Rather, one considers the complexity-theoretic aspects of the class of channels Φ that can be implemented through an interaction with P. Intuitively speaking, if every such channel can be efficiently implemented *without* making use of an interaction with P, then it is to be interpreted that no "knowledge" was revealed by P—for whatever the cheating verifier might have learned from P could equally well have been computed efficiently without interacting with P. Although it is not necessary to use this terminology in the definitions to follow, the term *simulator* is typically used to refer to a hypothetical entity that implements or approximates the channel implemented by a cheating verifier/prover interaction in this way.

The following definitions aim to formalize the notions just suggested. The first definition concerns the collection of channels that may be implemented through an interaction with a given prover.

Definition 5.1. Let $A \subseteq \Sigma^*$ be a set of binary strings, let r and s be polynomially bounded functions, and let $P(x)$ be a prover in an interactive game for each $x \in A$. Suppose further that

$$\{\Phi_x \,:\, x \in A\} \tag{5.1}$$

is a collection of channels, where Φ_x transforms $r(|x|)$ qubits to $s(|x|)$ qubits, for each $x \in A$. The collection (5.1) is *efficiently implementable through an interaction with P* if there exists a polynomial-time computable function W possessing the following properties for each $x \in A$:

1. $W(x)$ is an encoding of the quantum circuit description of a cheating verifier that is compatible with $P(x)$.
2. In an interaction with $P(x)$, the cheating verifier encoded by $W(x)$ takes $r(|x|)$ input qubits, produces $s(|x|)$ output qubits, and implements the channel Φ_x through its interaction with $P(x)$.

The next definition formalizes what it means for one collection of channels to *efficiently approximate* another.

Definition 5.2. Let $A \subseteq \Sigma^*$ be a set of binary strings, let r and s be polynomially bounded functions, and let $\varepsilon : \mathbb{N} \rightarrow [0, 1]$ be a function. Suppose further that

$$\{\Phi_x \,:\, x \in A\} \tag{5.2}$$

is a collection of channels, where Φ_x transforms $r(|x|)$ qubits to $s(|x|)$ qubits for each $x \in A$. The collection (5.2) is *efficiently ε-approximable* if there exists a polynomial-time computable function Q possessing the following properties for each $x \in A$:

1. $Q(x)$ is an encoding of a quantum circuit transforming $r(|x|)$ qubits to $s(|x|)$ qubits.
2. The circuit with encoding $Q(x)$ implements an $\varepsilon(|x|)$-approximation to the channel Φ_x.

The property of a given prover being *quantum statistical zero-knowledge* on a particular set of input strings may now be defined. Intuitively speaking, a prover has this property for a set of inputs if every channel that can be implemented through an interaction with it on this set of inputs can also be efficiently approximated without interacting with it.

Definition 5.3. Let $A \subseteq \Sigma^*$ be a set of binary strings and let $P(x)$ be a prover in an interactive game for each $x \in A$. It is said that P is a *quantum statistical zero-knowledge* prover on the set A if, for every collection of channels

$$\{\Phi_x \,:\, x \in A\} \tag{5.3}$$

that is efficiently implementable through an interaction with P, one has that the collection (5.3) is also efficiently ε-approximable for some choice of a negligible[2] function $\varepsilon : \mathbb{N} \to [0,1]$.

Definition 5.4. A promise problem $A = (A_{\text{yes}}, A_{\text{no}})$ is contained in the complexity class $\text{QSZK}_{a,b}(m)$ if there exists a polynomial-time computable function V and a function P that possess the following properties:

1. For every string $x \in A_{\text{yes}} \cup A_{\text{no}}$, one has that $V(x)$ is an encoding of a quantum circuit description of an m-message verifier in an interactive game, and $P(x)$ is a prover that is compatible with $V(x)$.
2. *Completeness.* For every string $x \in A_{\text{yes}}$, the interaction between $V(x)$ and $P(x)$ causes $V(x)$ to output 1 with probability at least a.
3. *Soundness.* For every string $x \in A_{\text{no}}$, the value of the verifier $V(x)$ satisfies $\omega(V(x)) \leq b$.

[2] A function $\varepsilon : \mathbb{N} \to [0,1]$ is typically defined to be *negligible* if $\varepsilon(n) = o(n^{-c})$ for every positive constant $c > 0$. An alternative definition that serves equally well for the purposes of this survey is that ε is negligible if and only if $\varepsilon(n) = O\big(2^{-n^c}\big)$ for some choice of a positive constant $c > 0$.

4. *Zero-knowledge.* P is a quantum statistical zero-knowledge prover on the set A_{yes}.

We will make use of shorthand conventions when referring to complexity classes of the form $\text{QSZK}_{a,b}(m)$ that are similar to those used when referring to $\text{QIP}_{a,b}(m)$. In particular, it is to be understood that $a = 2/3$ and $b = 1/3$ when these subscripts are omitted, while the omission of m indicates a union over all polynomially bounded functions m.

A few remarks concerning the previous definition are in order. One is that the containment

$$\text{QSZK}_{a,b}(m) \subseteq \text{QIP}_{a,b}(m) \tag{5.4}$$

is immediate, for every choice of m, a, and b; the verifier V whose existence is required for a given promise problem A to be included in $\text{QSZK}_{a,b}(m)$ fulfills, without any modifications being necessary, the requirements for A to be contained in $\text{QIP}_{a,b}(m)$. The additional requirements for a promise problem to be included in $\text{QSZK}_{a,b}(m)$, as compared with $\text{QIP}_{a,b}(m)$, therefore concern the prover P referred to in the definition. First, it must be the case that this prover P fulfills the requirements of the completeness condition for V, and second, it must not be possible for any cheating verifier to extract "knowledge" from $P(x)$, for any yes-input string $x \in A_{\text{yes}}$.

It is natural to ask why the zero-knowledge condition is required to hold only on the set A_{yes}, and not also on A_{no}, for instance. Practically speaking, extending the zero-knowledge requirement from the set A_{yes} to all of $A_{\text{yes}} \cup A_{\text{no}}$ would immediately lead to the collapse $\text{QSZK}_{a,b}(m) = \text{BQP}$ (assuming a and b are representative of reasonable, bounded-error probability bounds). Indeed, even approximating the channel implemented by the *honest* verifier V when interacting with P, which has no input qubits and one output qubit, would provide a BQP algorithm for A. Requiring the zero-knowledge condition to hold for all input strings is therefore too strong of a requirement to be interesting. A more philosophical reason for requiring the zero-knowledge condition only on A_{yes} is that there is no "knowledge" to be gained from a prover P on an input string $x \in A_{\text{no}}$. (Alternatively, one may view that it is not the role of cryptography to protect the dishonest against one another.)

Another natural question regarding the definition above is why one considers the possible *channels* that can be implemented through an interaction with a given prover, as opposed (for instance) to the states that may be prepared through an interaction with a prover. Again, there is a practical reason and a philosophical reason. Practically speaking, the definition above that is based on channels rather than state preparations is more robust, possessing various closure properties that one would hope for such a concept to fulfill. Philosophically speaking, the definition based on channels captures the idea that a cheating verifier cannot *increase* its knowledge through an interaction with a given prover, while an analogous definition based on state preparations would represent the idea that a verifier that knows nothing cannot learn something through an interaction with a prover. The channel-based definition is the more satisfying definition from a cryptographic point of view for both reasons.

As suggested above, it is possible to formulate other natural variants of zero-knowledge, such as *quantum computational zero-knowledge*, by modifying the definitions of quantum statistical zero-knowledge. In particular, it is only the notion of one channel approximating another (Definition 2.2) that needs to be modified in a substantive way to lead to the notion of quantum computational zero-knowledge. Rather than requiring two channels to be close with respect to the diamond norm distance, which represents the impossibility for the two channels to be distinguished in an information-theoretic sense, quantum computational zero-knowledge requires only that two channels cannot be *efficiently distinguished* (by polynomial-size quantum circuit families).

5.1.2 Honest verifier zero-knowledge

The zero-knowledge property of a prover P on a collection of yes-inputs, for a given promise problem $A = (A_{\text{yes}}, A_{\text{no}})$, is sometimes difficult to establish—one must prove the existence of an efficient ε-approximation (i.e., a simulator) for *every* collection of channels that can be efficiently implemented through an interaction with P. A simpler and less restrictive notion of zero-knowledge is known as *honest-verifier* zero-

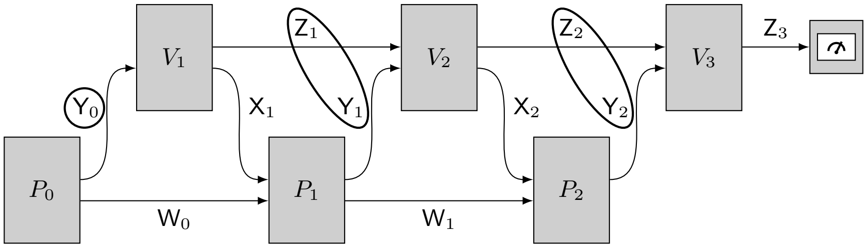


Figure 5.3: The verifier's *view* of an interaction is represented by the sequence of states corresponding to the registers it hold immediately prior to its actions. In the 5-message case, the view corresponds to the states of the registers Y_0, $(\mathsf{Z}_1, \mathsf{Y}_1)$, and $(\mathsf{Z}_2, \mathsf{Y}_2)$.

knowledge. To define this notion, one must first define what is meant by the *view* of a given verifier in an interactive game.

Definition 5.5. Suppose $V = (V_1, \ldots, V_n)$ is a verifier in an interactive game and $P = (P_0, \ldots, P_{n-1})$ is a prover compatible with V. The *view* of V when interacting with P is the state

$$\text{view}(V, P) = \rho_0 \otimes \rho_1 \otimes \cdots \otimes \rho_{n-1}, \tag{5.5}$$

where

$$\rho_0 \in \mathrm{D}(\mathcal{Y}_0),\; \rho_1 \in \mathrm{D}(\mathcal{Z}_1 \otimes \mathcal{Y}_1),\; \ldots,\; \rho_{n-1} \in \mathrm{D}(\mathcal{Z}_{n-1} \otimes \mathcal{Y}_{n-1}) \tag{5.6}$$

represent the states of the registers Y_0, $(\mathsf{Z}_1, \mathsf{Y}_1)$, $\ldots$, $(\mathsf{Z}_{n-1}, \mathsf{Y}_{n-1})$ at their corresponding times in the interaction between V and P.

Figure 5.3 illustrates the state representing the view of a 5-message verifier interacting with a prover.

It must be noted that the state $\rho_0 \otimes \rho_1 \otimes \cdots \otimes \rho_{n-1}$ representing the view of a verifier interacting with a prover does not itself represent a state of a collection of registers that ever co-exist at any moment. Rather, one should view this state as a mathematical abstraction—it is not possible to faithfully represent a transcript of a quantum interaction by a single quantum state, but the state $\rho_0 \otimes \rho_1 \otimes \cdots \otimes \rho_{n-1}$ is "good enough" to serve the needs of a study of the zero-knowledge property.

Next, we require a definition that formalizes what it means for a set of states to be efficiently approximable. This definition is analogous to Definition 5.2, but for states rather than channels.

Definition 5.6. Let $A \subseteq \Sigma^*$ be a set of binary strings, let r be a polynomially bounded function, and let $\varepsilon : \mathbb{N} \rightarrow [0,1]$ be a function. Suppose further that

$$\{\rho_x : x \in A\} \tag{5.7}$$

is a collection of states on $r(|x|)$ qubits for each $x \in A$. The collection (5.7) is *efficiently ε-approximable* if there exists a polynomial-time computable function Q possessing the following properties for each $x \in A$:

1. $Q(x)$ is an encoding of a quantum circuit taking no input qubits and outputting a state on $r(|x|)$ qubits.
2. The state output by the circuit with encoding $Q(x)$ is an $\varepsilon(|x|)$-approximation to the state ρ_x.

The honest-verifier zero-knowledge property is concerned only with one specific prover/verifier pair (for each yes-input to the problem under consideration)—it states that the view of the verifier when interacting with the prover must be close to a state that could be efficiently prepared by a quantum circuit (without an interaction with a prover). One critical requirement of the definition that must be highlighted is that the verifier with respect to which it is defined must be represented by a collection of isometric circuits, and not by general quantum circuits. The reason for this requirement will be discussed shortly, but first we will state the definition itself.

Definition 5.7 (Honest-verifier quantum statistical zero-knowledge)**.** A promise problem $A = (A_{\text{yes}}, A_{\text{no}})$ is contained in the complexity class $\text{HVQSZK}_{a,b}(m)$ if there exists a polynomial-time computable function V and a function P that possess the following properties:

1. For every string $x \in A_{\text{yes}} \cup A_{\text{no}}$, one has that $V(x)$ is an encoding of an isometric quantum circuit description of an m-message verifier in an interactive game, and $P(x)$ is a prover compatible with $V(x)$.
2. *Completeness.* For every string $x \in A_{\text{yes}}$, the interaction between $V(x)$ and $P(x)$ causes $V(x)$ to output 1 with probability at least a.

3. *Soundness.* For every string $x \in A_{\text{no}}$, it holds that $\omega(V(x)) \leq b$.
4. *Zero-knowledge.* The collection of states

$$\bigl\{\text{view}(V(x), P(x)) \,:\, x \in A_{\text{yes}}\bigr\} \tag{5.8}$$

is efficiently ε-approximable for some choice of a negligible function $\varepsilon : \mathbb{N} \rightarrow [0,1]$.

It is appropriate to make a few comments regarding the above definition of honest-verifier quantum statistical zero-knowledge. One may reasonably view the definition to be representative of an easily checked constraint on a prover/verifier pair that is inspired by the zero-knowledge condition, and chosen so that a general zero-knowledge prover/verifier pair is clearly honest-verifier zero-knowledge as well. It is therefore evident that QSZK is contained in HVQSZK. On the other hand, while the constraint represented by the definition of HVQSZK is intended to be strong enough to allow nontrivial bounds on the power of quantum statistical zero-knowledge interactive proofs to be obtained, it should not be viewed that the definition is necessarily cryptographically well-motivated or satisfying. As it turns out, it is indeed the case that HVQSZK and QSZK coincide, but this fact should perhaps be seen as good fortune rather than something that should be expected.[3]

An important aspect of the above definition already alluded to is the requirement that the verifier V in the definition is constrained to be isometric, meaning that each of the channels it performs must correspond to a linear isometry. By making this requirement of the verifier, one is potentially making the zero-knowledge condition harder to satisfy, and this potential of added difficulty must simply be accepted as an aspect of the definition. The utility of this assumption is that a complementary relationship between what the verifier sees and what the prover controls naturally emerges.

[3] It should be said, however, that it was not *unexpected* that this equality would hold, as the analogous fact was known to hold in the classical setting prior to the formulation of the definition [70].

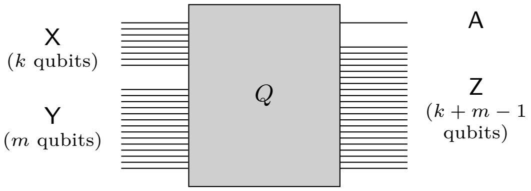


Figure 5.4: A unitary circuit Q for the quantum rewinding lemma.

5.2 Quantum rewinding

This section discusses a fact known as the *quantum rewinding lemma*, which allows for some interesting quantum interactive proof systems to be proved to possess the (general verifier) zero-knowledge property. The quantum rewinding lemma has a close connection to the strong error reduction procedure for QMA, which was discussed in Section 3.2.

5.2.1 Exact quantum rewinding lemma

When introducing the quantum rewinding lemma, it is helpful to begin with an exact version, which conveys the most relevant ideas of the lemma while avoiding complications present in an approximate version.

Consider a unitary quantum circuit Q acting on $k+m$ qubits, for positive integers k and m. It is to be assumed that the input qubits comprise two registers: an k qubit register X and an m qubit register Y. The first qubit output by Q will be considered as a single-qubit register A, while the remaining $k+m-1$ qubits form a register Z. Figure 5.4 provides an illustration of such a circuit.

The situation that the quantum rewinding lemma concerns is that Q acts on a pure state of the form $|\psi\rangle|0^m\rangle$, resulting in a state

$$Q|\psi\rangle|0^m\rangle = \sqrt{p(\psi)}|0\rangle|\phi_0(\psi)\rangle + \sqrt{1-p(\psi)}|1\rangle|\phi_1(\psi)\rangle \tag{5.9}$$

for some real number $p(\psi) \in [0,1]$ and $(k+m-1)$-qubit unit vectors $|\phi_0(\psi)\rangle$ and $|\phi_1(\psi)\rangle$. If one measures the qubit A with respect to the standard basis, the result will be 0 with probability $p(\psi)$, and conditioned on this outcome the state of Z becomes $|\phi_0(\psi)\rangle$. Otherwise, the measurement outcome is 1, and the state of Z becomes $|\phi_1(\psi)\rangle$. It is to be viewed that one desires to obtain the state $|\phi_0(\psi)\rangle$. Evidently, if Q is performed on $|\psi\rangle|0^m\rangle$, the register A is measured with respect to the standard basis, and the outcome 0 is obtained, then this goal is achieved—but this only happens with probability $p(\psi)$, and one may hope to improve on this probability of success.

In general, there is little that can be done to successfully recover the state $|\phi_0(\psi)\rangle$ with probability greater than $p(\psi)$, under the assumption that a single copy of the input state $|\psi\rangle$ is made available. What the quantum rewinding lemma provides is a condition under which it is possible to efficiently recover the state $|\phi_0(\psi)\rangle$ with high probability, from a single copy of $|\psi\rangle$. The condition is that the probability $p(\psi)$ is nonzero and independent of $|\psi\rangle$. This is, in some sense, quite intuitive—if the probability $p(\psi)$ is independent of $|\psi\rangle$, then a measurement of A reveals absolutely no information about $|\psi\rangle$, so one might expect the quantum information represented by $|\psi\rangle$ to be somehow contained in Z. One can therefore hope to recover $|\psi\rangle$, and therefore have another try at obtaining $|\phi_0(\psi)\rangle$ by running Q and measuring A. The proof of the quantum rewinding lemma explains precisely how this may be done.

Lemma 5.1 (Exact quantum rewinding lemma). Let Q be a unitary quantum circuit acting on $k+m$ qubits, and suppose that there exists a real number $p \in (0,1)$ for which the following statement holds: for every choice of an k-qubit unit vector $|\psi\rangle$, there exist $(k+m-1)$-qubit unit vectors $|\phi_0(\psi)\rangle$ and $|\phi_1(\psi)\rangle$ such that

$$Q|\psi\rangle|0^m\rangle = \sqrt{p}\,|0\rangle|\phi_0(\psi)\rangle + \sqrt{1-p}\,|1\rangle|\phi_1(\psi)\rangle. \tag{5.10}$$

For every $\varepsilon \in (0,1/2)$, there exists a quantum circuit R taking k input qubits, outputting $k+m-1$ qubits, having size

$$\operatorname{size}(R) = O\left(\frac{\log(1/\varepsilon)\operatorname{size}(Q)}{p(1-p)}\right), \tag{5.11}$$

and satisfying

$$\big\|\,|\phi_0(\psi)\rangle\langle\phi_0(\psi)| - R(|\psi\rangle\langle\psi|)\,\big\|_1 < \varepsilon \tag{5.12}$$

for every k-qubit unit vector $|\psi\rangle$. Moreover, a description of the circuit R can be generated from a description of Q in polynomial time.

```
Initial conditions:
    The register $\mathsf{X}$ contains a $k$-qubit quantum input $|\psi\rangle$.
    The register $\mathsf{Y}$ is initialized to the state $|0^m\rangle$.
The procedure:
    Apply the circuit $Q$ to the pair $(\mathsf{X}, \mathsf{Y})$ obtaining $(\mathsf{A}, \mathsf{Z})$.
    Repeat $\lceil \log(1/\varepsilon)/(p(1-p)) \rceil$ times:
        Measure $\mathsf{A}$ with respect to the standard basis.
        If the outcome of the measurement is 1, do the following:
            Apply $Q^{-1}$ to $(\mathsf{A}, \mathsf{Z})$, obtaining $(\mathsf{X}, \mathsf{Y})$.
            Apply the unitary operation $U = 2|0^m\rangle\langle 0^m| - \mathbb{1}$ to $\mathsf{Y}$.
            Apply $Q$ to the pair $(\mathsf{X}, \mathsf{Y})$ obtaining $(\mathsf{A}, \mathsf{Z})$.
    Output the register $\mathsf{Y}$.
```

Figure 5.5: The quantum rewinding procedure.

Proof. Consider the procedure described in Figure 5.5, and let R be a quantum circuit implementing this procedure. It is evident from the description of the procedure that one may design R so that (5.11) holds, and moreover that a description of R can be generated from a description of Q in polynomial time. It remains to consider the output of R.

The first step of the analysis is to consider the two operators

$$\begin{aligned} P_0 &= (\mathbb{1} \otimes \langle 0^m|)Q^*(|0\rangle\langle 0| \otimes \mathbb{1})Q(\mathbb{1} \otimes |0^m\rangle), \\ P_1 &= (\mathbb{1} \otimes \langle 0^m|)Q^*(|1\rangle\langle 1| \otimes \mathbb{1})Q(\mathbb{1} \otimes |0^m\rangle), \end{aligned} \tag{5.13}$$

where Q is being regarded as a unitary operator in these equations. Because P_0 and P_1 are positive semidefinite and satisfy $P_0 + P_1 = \mathbb{1}$, they describe a measurement $\{P_0, P_1\}$. This is the operator description of the measurement that is applied to X when Y is initialized to $|0^m\rangle$, Q is applied to (X, Y), yielding (A, Z), and A is measured with respect to the standard basis. By the assumptions of the lemma, one has

$$\langle\psi|P_0|\psi\rangle = p \quad \text{and} \quad \langle\psi|P_1|\psi\rangle = 1 - p \tag{5.14}$$

for every choice of $|\psi\rangle$, which implies that

$$P_0 = p\mathbb{1} \quad \text{and} \quad P_1 = (1-p)\mathbb{1}. \tag{5.15}$$

This is a consequence of the fact that an operator X acting on some complex vector space is uniquely determined by the function $|\psi\rangle \mapsto \langle\psi|X|\psi\rangle$, defined over the unit sphere of that space.

Now, suppose $|\psi\rangle$ is a particular k-qubit unit vector, and the procedure described in Figure 5.5 is run when X is in the state $|\psi\rangle$ and Y is initialized to the state $|0^m\rangle$. The first step of the procedure is the application of Q, which leaves (A,Z) in the state

$$Q|\psi\rangle|0^m\rangle = \sqrt{p}\,|0\rangle|\phi_0(\psi)\rangle + \sqrt{1-p}\,|1\rangle|\phi_1(\psi)\rangle. \tag{5.16}$$

The loop is then iterated, beginning with a measurement of A with respect to the standard basis. If the measurement outcome is 0, the state of Z becomes $|\phi_0(\psi)\rangle$. An inspection of the procedure reveals that once the measurement outcome 0 is obtained, no further changes are made to Z, so the state $|\phi_0(\psi)\rangle$ will indeed be the output of the procedure, conditioned on this first measurement resulting in 0.

If the measurement result is 1, the state of the pair (A,Z) becomes $|1\rangle|\phi_1(\psi)\rangle$, and the conditional statement is performed. This results in (A,Z) being left in the state

$$\begin{aligned} &Q(\mathbb{1}\otimes U)Q^*|1\rangle|\phi_1(\psi)\rangle \\ &\quad = -|1\rangle|\phi_1(\psi)\rangle + 2Q(\mathbb{1}\otimes|0^m\rangle\langle 0^m|)Q^*|1\rangle|\phi_1(\psi)\rangle. \end{aligned} \tag{5.17}$$

This expression can be simplified by making use of the equations (5.15). In particular, one has

$$|1\rangle|\phi_1(\psi)\rangle = \frac{1}{\sqrt{1-p}}(|1\rangle\langle 1|\otimes\mathbb{1})Q(\mathbb{1}\otimes|0^m\rangle)|\psi\rangle, \tag{5.18}$$

and therefore

$$\begin{aligned} &Q(\mathbb{1}\otimes|0^m\rangle\langle 0^m|)Q^*|1\rangle|\phi_1(\psi)\rangle \\ &\quad = \frac{1}{\sqrt{1-p}}Q(\mathbb{1}\otimes|0^m\rangle\langle 0^m|)Q^*(|1\rangle\langle 1|\otimes\mathbb{1})Q(\mathbb{1}\otimes|0^m\rangle)|\psi\rangle \\ &\quad = \frac{1}{\sqrt{1-p}}Q(\mathbb{1}\otimes|0^m\rangle)P_1|\psi\rangle \\ &\quad = \sqrt{1-p}\,Q|\psi\rangle|0^m\rangle \\ &\quad = \sqrt{p(1-p)}|0\rangle|\phi_0(\psi)\rangle + (1-p)|1\rangle|\phi_1(\psi)\rangle, \end{aligned} \tag{5.19}$$

which implies that the state of (A, Z) immediately after the conditional statement becomes

$$\begin{aligned} &Q(\mathbb{1} \otimes U)Q^* |1\rangle|\phi_1(\psi)\rangle \\ &\quad = 2\sqrt{p(1-p)}|0\rangle|\phi_0(\psi)\rangle + (1-2p)|1\rangle|\phi_1(\psi)\rangle. \end{aligned} \tag{5.20}$$

On subsequent iterations of the loop, the analysis is the same. The measurement of A now gives the outcome 0 with probability $4p(1-p)$, and conditioned on this outcome the register Z is left in state $|\phi_0(\psi)\rangle$ until the end of the procedure. If the measurement outcome is 1, the state of (A, Z) reverts back to $|1\rangle|\phi_1(\psi)\rangle$, and the conditional statement is applied precisely as before.

As a result of this analysis, one obtains the following closed-form expression for the output of the procedure:

$$R(|\psi\rangle\langle\psi|) = (1-\delta)|\phi_0(\psi)\rangle\langle\phi_0(\psi)| + \delta|\phi_1(\psi)\rangle\langle\phi_1(\psi)|, \tag{5.21}$$

where

$$\delta = (1-p)(1-2p)^{2T} \quad \text{and} \quad T = \left\lceil \frac{\log(1/\varepsilon)}{p(1-p)} \right\rceil. \tag{5.22}$$

It holds that $\delta < \varepsilon/2$, from which the bound (5.12) follows. □

5.2.2 Example: graph isomorphism

We will now illustrate the use of the quantum rewinding lemma to prove that the Goldreich–Micali–Wigderson proof system for graph isomorphism (described at the beginning of the chapter in Figure 5.1) has the zero-knowledge property against cheating quantum verifiers.

Let (G_0, G_1) be a pair of isomorphic graphs, and P the prover described by the Goldreich–Micali–Wigderson graph isomorphism proof system on the input (G_0, G_1). Suppose $W = (W_1, W_2)$ is a (computationally efficient) cheating verifier that implements some channel Φ by means of an interaction with P. To prove that the proof system is zero-knowledge against quantum attacks, it must be shown that every channel Φ that can be implemented in this way can be efficiently implemented without making use of an interaction with P.

Note that, by the assumption that the operations performed by the cheating verifier W are efficiently implementable, it suffices to focus on the state contained in the registers $(\mathsf{Z}_1, \mathsf{Y}_1)$ after the prover's second message—for if the state obtained by the cheating verifier W at this point can be efficiently computed, then by simply applying the channel W_2 to this state, one obtains Φ.

There is also no loss of generality in assuming that the cheating verifier is such that its first action is given by an isometry

$$W_1 \in \mathrm{U}(\mathcal{Z}_0 \otimes \mathcal{Y}_0, \mathcal{Z}_1 \otimes \mathcal{X}_1). \tag{5.23}$$

It is convenient to express this isometry as

$$W_1 = A_0 \otimes |0\rangle + A_1 \otimes |1\rangle \tag{5.24}$$

for operators $A_0, A_1 \in \mathrm{L}(\mathcal{Z}_0 \otimes \mathcal{Y}_0, \mathcal{Z}_1)$, which is possible because $\mathcal{X}_1$ corresponds to a single qubit (which will be seen as a single bit response by the honest, classical prover in the Goldreich–Micali–Wigderson proof system). One may verify that the channel implemented by such a cheating verifier when interacting with the honest prover is given by

$$\Phi(\rho) = \frac{1}{n!} \sum_{\substack{\pi \in S_n \\ a \in \{0,1\}}} A_a\big(\rho \otimes |\pi(G_0)\rangle\langle\pi(G_0)|\big) A_a^* \otimes |\pi\sigma^a\rangle\langle\pi\sigma^a|. \tag{5.25}$$

It is not immediately clear that the channel (5.25) can be implemented efficiently because the permutation σ may not be efficiently computable from (G_0, G_1) (unless the graph isomorphism problem is in BQP). We will need to make use of the quantum rewinding lemma to prove that it can be efficiently implemented.

We note first that it is possible to efficiently implement a unitary computation Q transforming $|\psi\rangle|0^m\rangle$, for a suitable choice of m, to the state

$$Q|\psi\rangle|0^m\rangle = \frac{1}{\sqrt{2}}|0\rangle|\phi_0(\psi)\rangle + \frac{1}{\sqrt{2}}|1\rangle|\phi_1(\psi)\rangle \tag{5.26}$$

for

$$\begin{aligned} |\phi_0(\psi)\rangle &= \frac{1}{\sqrt{n!}} \sum_{\tau \in S_n} \sum_{b \in \{0,1\}} A_b(|\psi\rangle|\tau(G_b)\rangle)|b\rangle|\tau\rangle, \\ |\phi_1(\psi)\rangle &= \frac{1}{\sqrt{n!}} \sum_{\tau \in S_n} \sum_{b \in \{0,1\}} A_{1-b}(|\psi\rangle|\tau(G_b)\rangle)|b\rangle|\tau\rangle. \end{aligned} \tag{5.27}$$

For instance, one may create the state

$$\frac{1}{\sqrt{2n!}} \sum_{\tau \in S_n} \sum_{b \in \{0,1\}} |\tau(G_b)\rangle |b\rangle |\tau\rangle, \tag{5.28}$$

apply W_1 to the first register of this state together with $|\psi\rangle$, and then finally apply a controlled-NOT gate (and a permutation of the ordering of the qubits) to obtain (5.26).

Given that the above transformation Q can be performed efficiently, one may recover $|\phi_0(\psi)\rangle$ using the quantum rewinding lemma. By post-processing this state (by measuring the last register with respect to the standard basis and discarding the qubit corresponding to b), the outcome $\Phi\big(|\psi\rangle\langle\psi|\big)$ is obtained. As this is so for every state $|\psi\rangle$, the channel Φ has been implemented efficiently. This implies that the prover P is a statistical zero-knowledge prover on inputs (G_0, G_1) for which G_0 and G_1 are isomorphic.

It may be noted that there is a fairly direct correspondence between the classical argument (which was briefly summarized at the beginning of the chapter) for proving that the Goldreich–Micali–Wigderson graph isomorphism proof system is zero-knowledge against classical cheating verifiers and the quantum case just discussed. The state (5.28) is effectively a purified form of a guess for the messages exchanged between the prover and verifier, and the first qubit of (5.26) indicates whether this guess was correct: if this qubit is in the 0 state, the guess was correct and the correct output can be produced, while if this qubit is in the 1 state, then rewinding takes place, giving the simulation another chance for success.

5.2.3 Approximate quantum rewinding lemma

For an arbitrary quantum circuit Q acting on $k+m$ qubits, as described in the previous section, one may always write

$$Q|\psi\rangle|0^m\rangle = \sqrt{p(\psi)}|0\rangle|\phi_0(\psi)\rangle + \sqrt{1-p(\psi)}|1\rangle|\phi_1(\psi)\rangle \tag{5.29}$$

for any given pure state $|\psi\rangle$, for some real number $p(\psi) \in [0,1]$ and $(k+m-1)$-qubit unit vectors $|\phi_0(\psi)\rangle$ and $|\phi_1(\psi)\rangle$. The exact quantum rewinding lemma will generally not be applicable to such a circuit, as

$p(\psi)$ will generally depend on ψ. Although one cannot hope that an approximate version of the quantum rewinding lemma should hold in cases where $p(\psi)$ varies significantly as $|\psi\rangle$ ranges over all n-qubit unit vectors, it is reasonable to expect that small variations in $p(\psi)$ can be tolerated. The following lemma states that this is indeed possible, provided that one accepts a small loss in the procedure's accuracy.

Lemma 5.2 (Approximate quantum rewinding lemma). Let Q be a unitary quantum circuit acting on $k+m$ qubits, and write

$$Q|\psi\rangle|0^m\rangle = \sqrt{p(\psi)}\,|0\rangle|\phi_0(\psi)\rangle + \sqrt{1-p(\psi)}\,|1\rangle|\phi_1(\psi)\rangle \tag{5.30}$$

for each k-qubit unit vector $|\psi\rangle$. Assume that there exist real numbers $\delta, q \in (0,1)$ such that $|p(\psi) - q| < \delta$ for every k-qubit unit vector $|\psi\rangle$. For every $\varepsilon \in (0, 1/2)$, there exists a quantum circuit R taking k input qubits, outputting $k+m-1$ qubits, and satisfying $\operatorname{size}(R) = O(T \operatorname{size}(Q))$ and

$$\big\|\,|\phi_0(\psi)\rangle\langle\phi_0(\psi)| - R(|\psi\rangle\langle\psi|)\,\big\|_1 < \varepsilon + (4T+2)\sqrt{2\delta} \tag{5.31}$$

for every k-qubit unit vector $|\psi\rangle$, for

$$T = \left\lceil \frac{\log(1/\varepsilon)}{q(1-q)} \right\rceil. \tag{5.32}$$

Moreover, a description of the circuit R can be generated from a description of Q in time polynomial in $\operatorname{size}(Q)$ and T.

The principle behind the proof of this approximate version of the quantum rewinding lemma is simple: one runs precisely the same rewinding procedure as in the exact case, imagining that the action of Q is given by a unitary operator U satisfying

$$U|\psi\rangle|0^m\rangle = \sqrt{q}\,|0\rangle|\phi_0(\psi)\rangle + \sqrt{1-q}\,|1\rangle|\phi_1(\psi)\rangle \tag{5.33}$$

for all $|\psi\rangle$. The analysis from the proof of the exact quantum rewinding lemma reveals what the output of the procedure would be if Q acted in this idealized way. Finally, it is proved that there must exist a unitary operator acting as above and satisfying $\|Q - U\| \leq \sqrt{2\delta}$. The bound in the statement of the approximate quantum rewinding lemma is then recovered from the fact that errors accumulate at most additively in compositions of quantum circuits, which may be proved through iterative applications of the triangle inequality.

5.3 Quantum statistical zero knowledge

A collection of interesting properties of the complexity class QSZK are known to hold. For instance, QSZK is closed under complementation, contained in QIP(2), and equal to HVQSZK. Moreover, a complete promise problem is known for QSZK that is both simple and fundamental from the viewpoint of quantum computation. Each of these facts has a classical analogue that is also known to hold, and the techniques for proving the quantum variants of these facts borrow heavily from known proofs of their classical analogues.

5.3.1 The close quantum states problem

Much of the discussion of the complexity class QSZK to follow in the present subsection is centered on the following promise problem, called the (α, β)-*close quantum states problem*. This is the problem mentioned above that will soon be proved complete for QSZK.

(α, β)-close quantum states ((α, β)-CQS)

Input: Quantum circuits Q_0 and Q_1 taking no input and outputting quantum states ρ_0 and ρ_1 on the same number of qubits.

Yes: $\frac{1}{2}\|\rho_0 - \rho_1\|_1 < \alpha$.

No: $\frac{1}{2}\|\rho_0 - \rho_1\|_1 > \beta$.

Formally speaking, each choice of a pair (α, β) defines a different promise problem (α, β)-CQS. One may consider the situation in which α and β are constants or functions; and if they are functions, it should be understood that they are functions of the number n representing the size of the input description of the pair (Q_0, Q_1).

It is not surprising that the complexity of the (α, β)-CQS problem may depend on the choice of α and β. As it turns out, there is a wide range of choices for these values that lead to polynomial-time equivalent problems.

Theorem 5.3 (Sahai–Vadhan)**.** Let p and q be polynomially bounded functions and let α and β be polynomial-time computable functions satisfying

$$0 < \alpha(n) \leq \beta(n)^2 - \frac{1}{q(n)} \tag{5.34}$$

for all sufficiently large positive integers n. It holds that

$$(\alpha, \beta)\text{-CQS} \leq_m (\delta, 1-\delta)\text{-CQS} \tag{5.35}$$

for $\delta(n) = 2^{-p(n)}$.

This theorem may be attributed to Sahai and Vadhan [148], who proved a classical analogue to this fact—but the extension to quantum states is direct, requiring nothing more than a verification that the techniques and bounds indeed extend to quantum states [167].

There are two main observations required to prove this theorem. The first observation is that, for any choice of quantum states ρ_0 and ρ_1 (on the same number of qubits) and a positive integer m, and for states

$$\begin{aligned} \xi_0 &= \frac{1}{2^{m-1}} \sum_{\substack{a_1,\ldots,a_m \in \{0,1\} \\ a_1+\cdots+a_m \equiv 0 \pmod 2}} \rho_{a_1} \otimes \cdots \otimes \rho_{a_m}, \\ \xi_1 &= \frac{1}{2^{m-1}} \sum_{\substack{a_1,\ldots,a_m \in \{0,1\} \\ a_1+\cdots+a_m \equiv 1 \pmod 2}} \rho_{a_1} \otimes \cdots \otimes \rho_{a_m}, \end{aligned} \tag{5.36}$$

one has

$$\frac{1}{2}\|\xi_0 - \xi_1\|_1 = \left(\frac{1}{2}\|\rho_0 - \rho_1\|_1\right)^m. \tag{5.37}$$

The second observation is that, again for any choice of quantum states ρ_0 and ρ_1 and a positive integer m, one has

$$1 - \exp\!\left(-\frac{m\|\rho_0 - \rho_1\|_1^2}{8}\right) \leq \frac{1}{2}\left\|\rho_0^{\otimes m} - \rho_1^{\otimes m}\right\|_1 \leq \frac{m}{2}\|\rho_0 - \rho_1\|_1. \tag{5.38}$$

These two facts suggest constructions that may be applied to two circuits Q_0 and Q_1 given as input to the close quantum states problem.

The first construction produces states (corresponding to ξ_0 and ξ_1) that are generally much closer to one another than the states ρ_0 and ρ_1 produced by Q_0 and Q_1—but the rate at which ξ_0 and ξ_1 approach one another is much faster (as a function of m) in the case that ρ_0 and ρ_1 are close. The second construction simply repeats a state many times, causing the resulting states to move away from one another, this time more rapidly when the original states are farther apart. By alternating the two constructions in a suitable fashion (first the first construction, then the second, then the first again), the input circuits are transformed so as to produce output states that are either very close or very far, depending on the initial closeness of the states. The requirement $\alpha(n) \leq \beta(n)^2 - 1/q(n)$ is needed to guarantee that the correct behavior results from these constructions.

5.3.2 HVQSZK-completeness of the close quantum states problem

There are two main steps required to prove that the close quantum states problem is complete for QSZK, for any choice of α and β for which Theorem 5.3 holds. The first step is to prove that this problem is complete for HVQSZK, and the second is to prove that the problem itself is contained in QSZK. Because QSZK is closed under Karp reductions and QSZK $\subseteq$ HVQSZK, the QSZK-completeness of the problem follows—and in the process one finds that QSZK = HVQSZK. The fact that the close quantum states problem is contained in QSZK will make use of the quantum rewinding lemma, and is discussed in the subsection following this one. The present section concerns the fact that this problem is complete for HVQSZK.

The HVQSZK-completeness of the close quantum states problem can be proved in multiple ways. Here we will sketch a proof that is somewhat different from the original proof [167], although the essential ideas are similar.

Suppose that A is any promise problem for which $A \in$ HVQSZK. There must therefore exist a prover $P(x)$ and verifier $V(x)$ satisfying the conditions of Definition 5.7, for $a = 2/3$, $b = 1/3$, and m being a polynomially bounded function. By considering the three transformations associated with the perfect completeness, parallelization, and

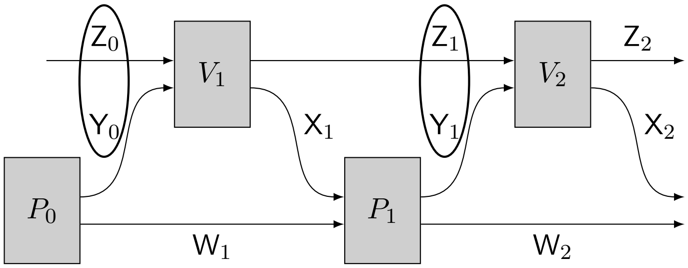


Figure 5.6: A unitary verifier's view in a three-message interactive game.

error reduction by parallel repetition constructions discussed in the previous chapter, one may transform both the verifier V and the prover P into a new, three-message prover/verifier pair (P', V') for A having perfect completeness and exponentially small soundness error. Moreover, it may be assumed that the new verifier V' is an isometric verifier, and that the honest-verifier zero-knowledge condition is still in place for the pair (P', V'). The fact that these assumptions are justified follows from a consideration of each of the three constructions: in each case, the most natural and direct way of defining the actions of the honest prover and the corresponding view of the verifier maintains the honest-verifier zero-knowledge condition.

Given a pair (P', V') as just described, one may output an instance of the close quantum states problem for each input $x \in A_{\text{yes}} \cup A_{\text{no}}$ in the following way. First, one may compute circuits R_0 and R_1 that take no input qubits and output states of the registers $(\mathsf{Z}_0, \mathsf{Y}_0)$ and $(\mathsf{Z}_1, \mathsf{Y}_1)$, respectively, with the property that these states are close to the true states of these registers when V' interacts with P' (as illustrated in Figure 5.6), assuming the input is a yes-input for A. This is possible by the honest-verifier statistical zero-knowledge property of (P', V'). The circuits Q_0 and Q_1, representing the instance of the close quantum states problem being produced, are derived from R_0 and R_1 in the following way:

1. Q_0 is the circuit obtained by first running R_0 to obtain a state of $(\mathsf{Z}_0, \mathsf{Y}_0)$, replacing each qubit of Z_0 by an initialized qubit in the

$|0\rangle$ state, applying V_1 to the pair $(\mathsf{Z}_0, \mathsf{Y}_0)$ to obtain $(\mathsf{Z}_1, \mathsf{X}_1)$, and then outputting just the register Z_1.

2. Q_1 is the circuit obtained by first running R_1 to obtain a state of $(\mathsf{Z}_1, \mathsf{Y}_1)$, applying V_2 to obtain a state of $(\mathsf{Z}_2, \mathsf{X}_2)$, replacing the single-qubit register Z_2 with a qubit in the $|1\rangle$ state, applying V_2^{-1} to $(\mathsf{Z}_2, \mathsf{X}_2)$ to obtain $(\mathsf{Z}_1, \mathsf{Y}_1)$, and then outputting just the register Z_1.

If P' causes V' to accept with probability 1, then the states output by Q_0 and Q_1 will necessarily be close, by virtue of the honest-verifier statistical zero-knowledge property: on yes-instances of the problem A, the circuits R_0 and R_1 output close approximations of the true states of the registers $(\mathsf{Z}_0, \mathsf{Y}_0)$ and $(\mathsf{Z}_1, \mathsf{Y}_1)$, which leads to Q_0 and Q_1 outputting nearly identical states. On the other hand, in the case that $\omega(V')$ is small, the states output by Q_0 and Q_1 must necessarily be far apart, irrespective of the zero-knowledge properties of (P', V'), as follows from a similar argument to the one described in the previous chapter concerning the close circuit images problem.

5.3.3 Close quantum states in QSZK

We will now sketch an argument demonstrating that the close quantum states problem is contained in QSZK. By Theorem 5.3, it suffices to consider the problem $(\delta, 1-\delta)$-CQS for $\delta(n)$ being exponentially small.

For a given input (Q_0, Q_1) to the $(\delta, 1-\delta)$-CQS problem, consider the interactive game described in Figure 5.7. It is evident that this game is both complete and sound as an interactive game. In particular, if Q_0 and Q_1 output close quantum states, the verifier will accept with probability exponentially close to 1 when interacting with the honest prover described in the figure. On the other hand, if Q_0 and Q_1 output quantum states that are almost perfectly distinguishable, there cannot exist a state of X that is consistent with both of the pure states output by R_0 and R_1 when run on the all-zero input state—and an analysis reveals that the verifier rejects with probability exponentially close to 1/2 in this case.

It remains to consider the zero-knowledge property, which may

The input is a pair (Q_0, Q_1) of quantum circuits that take no input and output states ρ_0 and ρ_1 of a k-qubit register X. By purifying Q_0 and Q_1, and padding one of the circuits with extra qubits if necessary, unitary quantum circuits R_0 and R_1 acting on registers (X, Y), for Y being an m-qubit register, are obtained. Initializing (X, Y) to $|0^k\rangle|0^m\rangle$, applying R_a, and tracing out Y leaves X in the state ρ_a.

Prover's step 1: Initialize (X, Y) to $|0^k\rangle|0^m\rangle$, apply R_0 to the pair (X, Y), and send X to the verifier.

Verifier's step 1: Choose $a \in \{0, 1\}$ uniformly at random and send a to the prover.

Prover's step 2: If $a = 0$, send Y to the verifier. If $a = 1$, apply U to Y and then send Y to the verifier, where U is an m-qubit unitary satisfying

$$\big\|(\mathbb{1} \otimes U)R_0|0^{k+m}\rangle - R_1|0^{k+m}\rangle\big\|_1 \leq \varepsilon. \tag{5.39}$$

Verifier's step 2: Apply R_a^{-1} to (X, Y) and measure all of the qubits in these registers with respect to the standard basis. Accept if all measurement results are 0, reject otherwise.

Figure 5.7: Public-coin proof system for close quantum states.

be established using the (approximate) quantum rewinding lemma, through a similar methodology to the proof that the Goldreich–Micali–Wigderson graph isomorphism proof system is zero-knowledge against quantum attacks, as described in Section 5.2.2. As in that proof, it suffices to consider a cheating verifier whose first action is given by an isometry $W_1 \in \mathrm{U}(\mathcal{Z}_0 \otimes \mathcal{Y}_0, \mathcal{Z}_1 \otimes \mathcal{X}_1)$, expressed as

$$W_1 = A_0 \otimes |0\rangle + A_1 \otimes |1\rangle \tag{5.40}$$

for $A_0, A_1 \in \mathrm{L}(\mathcal{Z}_0 \otimes \mathcal{Y}_0, \mathcal{Z}_1)$. In the present case, the channel implemented by such a cheating verifier (disregarding the application of W_2)

through an interaction with the honest prover P is given by

$$\Phi(\rho) = \sum_{a\in\{0,1\}} \bigl(A_a \otimes \mathbb{1}_{\mathcal{Y}_1}\bigr)\bigl(\rho \otimes |\gamma_a\rangle\langle\gamma_a|\bigr)\bigl(A_a \otimes \mathbb{1}_{\mathcal{Y}_1}\bigr)^* \tag{5.41}$$

for

$$|\gamma_0\rangle = R_0|0^{k+m}\rangle \quad \text{and} \quad |\gamma_1\rangle = (\mathbb{1}\otimes U)R_0|0^{k+m}\rangle. \tag{5.42}$$

This time, it is the fact that U may not be efficiently implementable that represents the main obstacle to efficiently implementing Φ in the most obvious way.

It is possible, however, to efficiently implement a unitary quantum circuit Q that operates as follows: for a given pure state $|\psi\rangle$ and a suitably chosen value r, the circuit Q transforms $|\psi\rangle|0^r\rangle$ as follows:

$$Q|\psi\rangle|0^r\rangle = \frac{1}{\sqrt{2}}|0\rangle|\phi_0(\psi)\rangle + \frac{1}{\sqrt{2}}|1\rangle|\phi_1(\psi)\rangle \tag{5.43}$$

for

$$\begin{aligned} |\phi_0(\psi)\rangle &= \sum_{b\in\{0,1\}} (A_b\otimes\mathbb{1})|\psi\rangle R_b|0^{k+m}\rangle|b\rangle,\\ |\phi_1(\psi)\rangle &= \sum_{b\in\{0,1\}} (A_{1-b}\otimes\mathbb{1})|\psi\rangle R_b|0^{k+m}\rangle|b\rangle. \end{aligned} \tag{5.44}$$

The vectors $|\phi_0(\psi)\rangle$ and $|\phi_1(\psi)\rangle$ are not necessarily unit vectors in this case, but they are exponentially close to unit vectors under the assumption that Q_0 and Q_1 output states that are exponentially close to one another. Applying the approximate quantum rewinding lemma allows for the recovery of $|\phi_0(\psi)\rangle$ with high probability, which may then be processed to implement a close approximation to Φ.

Theorem 5.4. Let q be a polynomially bounded function and let α and β be polynomial-time computable functions satisfying

$$0 < \alpha(n) \leq \beta(n)^2 - \frac{1}{q(n)} \tag{5.45}$$

for all sufficiently large positive integers n. It holds that (α,β)-CQS $\in$ QSZK.

Corollary 5.5. QSZK = HVQSZK.

The input is a pair (Q_0, Q_1) of quantum circuits that take no input and output states ρ_0 and ρ_1 of an n-qubit register X.

Verifier's step 1: Choose $a \in \{0,1\}$ uniformly at random, prepare the state ρ_a produced by Q_a in a register X, and send X to the prover.

Prover's step 1: Perform an optimal measurement to distinguish the states ρ_0 and ρ_1. If the measurement indicates that the state is ρ_0, send $b = 0$ to the verifier, and otherwise send $b = 1$ to the verifier.

Verifier's step 2: Accept if $a = b$, reject otherwise.

Figure 5.8: A proof system for the complement of the close quantum states problem.

5.3.4 Closure of QSZK under complementation

Finally, we may observe that the class QSZK is closed under complementation, meaning that a promise problem $A = (A_{\text{yes}}, A_{\text{no}})$ is contained in QSZK if and only if the same is true of the promise problem $\overline{A} = (A_{\text{no}}, A_{\text{yes}})$. By virtue of the fact that QSZK = HVQSZK and the close quantum states problem is complete for QSZK, it suffices to prove that the complement of $(\delta, 1-\delta)$-CQS is contained in HVQSZK for $\delta(n) = 2^{-n}$.

To prove that this is so, one may consider the very simple interactive game described in Figure 5.8. The value of the verifier V for the input (Q_0, Q_1) described in the figure is given by the expression

$$\omega(V) = \frac{1}{2} + \frac{1}{4}\big\|\rho_0 - \rho_1\big\|_1, \tag{5.46}$$

and the prover P described in the figure achieves this optimal winning probability.

It remains to prove that, for a purified form of the verifier described in the figure, one has that the prover and verifier pair (P, V) possesses the honest-verifier zero-knowledge property on yes-inputs, which are those for which $\frac{1}{2}\|\rho_0 - \rho_1\|_1$ is exponentially close to 1 (i.e., ρ_0 and ρ_1 are almost perfectly distinguishable). The view of V in this case is simply the state of the pair $(\mathsf{Z}_1, \mathsf{Y}_1)$ immediately after the prover sends its only message in the proof system. It is nearly trivial to approximate this state efficiently—one may simply perform the computation represented by the verifier's first action V_1, then substitute the correct identification of b for the prover's message. Specifically, this may be done by discarding the register X_1, measuring the qubit corresponding to the verifier's random choice of $a \in \{0, 1\}$, and then setting Y_1 to contain this classical value. Because the honest prover correctly determines the value of a with a negligible error probability, this nearly trivial approximation to the verifier's view will deviate from the verifier's actual view on yes-inputs by a negligible quantity.

5.4 Chapter notes

The notion of zero-knowledge was proposed by Goldwasser, Micali, and Rackoff [71, 72], and has been investigated by many researchers in theoretical computer science and cryptography since then. The survey of Goldreich [69] may be consulted by readers interested in learning more about this topic of study in the classical setting. The topic of quantum zero-knowledge, as well as the problematic issue of classical techniques for proving interactive proof systems to possess the zero-knowledge property not carrying over to the quantum setting, was raised by van de Graaf in his PhD thesis [162].

Honest-verifier quantum statistical zero-knowledge was defined and studied in [167], wherein it was proved that HVQSZK is closed under complementation, contained in QIP(2), and has the complete promise problems mentioned in Section 5.3. As suggested in the main text, these facts have classical analogues—the paper of Sahai and Vadhan [148] proves several facts along these lines. Earlier papers, including ones of Fortnow [61], Aiello and Håstad [11], and Okamoto [134], proved related results on the complexity-theoretic aspects of statistical zero-knowledge proof systems. Kobayashi [116] proved several results of a similar nature for a non-interactive variant of quantum statistical zero-knowledge.

The quantum rewinding lemma was proved in [169], along with its application to the security of the Goldreich–Micali–Wigderson graph isomorphism proof system against quantum attacks, the equality of QSZK and HVQSZK, and to the security of a (computational) zero-knowledge proof system for graph 3-coloring (also due to Goldreich, Micali, and Wigderson). Hallgren, Kolla, Sen, and Zhang [86] extended these results to prove that a wide range of classical statistical zero-knowledge proof systems remain zero-knowledge against quantum attacks, and Kobayashi [117] proved several results of a similar nature concerning quantum computational zero-knowledge. The quantum rewinding lemma has been applied in a couple of other settings relating to quantum cryptography—see, for instance, Damgård and Lunemann [52] and Hallgren, Smith, and Song [87]. Unruh [159] has considered quantum proofs of knowledge, making use of the quantum rewinding lemma and a different rewinding technique to prove interesting results concerning this notion. Ambainis, Rosmanis, and Unruh [13] have proved limitations on the applicability of these rewinding techniques, which indeed appear to be rather limited when compared to their classical counterparts.

In addition to the close quantum states problem (and its complement), there are a few other promise problems known to be complete for the class QSZK. Ben-Aroya, Schwartz, and Ta-Shma [27] proved that a promise problem based on deciding which of two quantum circuits produces a state with greater von Neumann entropy is complete for QSZK, and Gutoski, Hayden, Milner, and Wilde [82] proved the QSZK-completeness of a promise problem relating to separability testing. Gutoski, Hayden, Milner, and Wilde also prove the completeness of other problems relating to separability testing for other complexity classes based on quantum proofs.

6

Multi-Prover Quantum Interactive Proofs

This chapter considers *multi-prover* interactive proof systems, an extension of the model of single-prover interactive proofs in which the verifier interacts simultaneously with two or more provers. Each prover is modeled as a separate participant in the game, and together the provers attempt to maximize the verifier's probability of eventually outputting 1, meaning that it accepts the interaction.

What makes these games particularly interesting is that the provers, which will always be assumed to cooperate in this chapter, are not permitted to exchange messages with each other—the only communication is between the verifier and each individual prover. This restriction empowers the verifier, allowing for games in which the prover's answers are checked against each other, akin to a detective attempting to confound a suspected team of robbers by submitting them to isolated interrogations and cross-checking their answers against one other.

The first scenario that will be considered is the case of a quantum verifier interacting with quantum provers that are restricted to applying local transformations and do not have any further means of coordinating their actions. After introducing the required definitions in Section 6.2.1, it will be shown that the class QMIP of problems that can be decided in this model is precisely equal to its classical counterpart MIP, in which all parties are classical: QMIP = MIP.

While the model evidently collapses to the single-prover model as soon as communication between the provers is allowed, it is interesting to consider provers having access to sources of correlations that do not require communication. For the case of classical provers, shared randomness may be considered, but it does not affect the computational power of the model—any shared random string used by the provers can always be replaced by a deterministic setting of the shared string that maximizes the probability with which the verifier accepts. For the case of quantum provers, quantum physics suggests that it may be beneficial to the provers to share a quantum state that is entangled across the registers associated with different provers. The study of Bell inequalities demonstrates that by performing local measurements on a shared entangled state (such as an EPR pair) the provers are able to generate correlations that, although they do not imply communication, are stronger than the correlations that can be generated by shared randomness alone. Thus, the use of entanglement may enhance the provers' ability to coordinate their answers, leading to a class QMIP* of problems having entangled-prover multi-prover interactive proof systems that is *a priori* distinct from QMIP.

Most of this chapter is concerned with results on the class QMIP*. We begin by considering the effect that the use of entanglement can have on the soundness of multi-prover interactive proof systems. In Section 6.2.2 it will be shown that the important classical technique of *oracularization* fails in the presence of entanglement, and in Section 6.2.3 we will see that entanglement leads to the collapse of a certain restricted class of proof systems, namely two-prover XOR interactive proofs.

Section 6.3 discusses structural results on QMIP*. Many of these results have the interesting peculiarity that they are only known to be achievable when honest provers make use of shared entanglement. These results include the parallelization of arbitrary multi-prover interactive proofs to ones having a single round of interaction, the transformation of multi-prover interactive proofs into ones possessing the property of perfect completeness, and the simulation of arbitrary multi-prover quantum interactive proofs by ones in which the verifier is classical, leading to the equality QMIP* = MIP*.

Section 6.4 is devoted to a proof that NEXP $\subseteq$ QMIP*. This shows that, in spite of the possible use of entanglement by the provers, the verifier in a quantum multi-prover interactive proof system has no less verification power than that of classical multi-prover interactive proofs, which is characterized as MIP = NEXP. The analysis will introduce a three-prover variant of oracularization and discuss its relation to a phenomenon known as the *monogamy of entanglement.*

In the concluding Section 6.5, we discuss two important topics in the study of quantum multi-prover interactive proof systems that remain largely unsettled. The first topic is the question of parallel error amplification, and the second is the problem of placing upper bounds on the class QMIP*.

6.1 Definitions of multi-prover interactive proof systems

As was suggested in Chapter 4, the interactive game model through which single-prover quantum interactive proof systems were defined may be extended in a straightforward way that allows a verifier to interact with multiple provers. A multi-prover interactive game is completely determined by the description of the verifier, and we will always assume that messages are sent synchronously in *turns*, consisting either of a set of messages from the verifier to each of the provers, or a set of messages from the provers to the verifier. A *round* is made of two turns, the first consisting of messages from the verifier to the provers and the second consisting of messages from the provers back to the verifier. Except when stated otherwise, for notational convenience we will usually assume that all interactive games have an integral number of rounds (and in particular the first turn consists of a set of messages from the verifier to the provers).

We will use the same labeling convention for the registers corresponding to different messages and private memories as in the single-prover case, introducing superscripts to distinguish registers associated

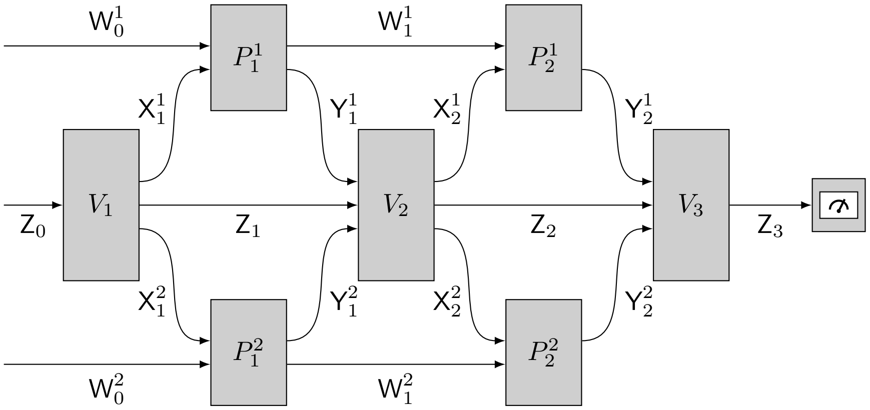


Figure 6.1: A four-turn interactive game between a verifier and two provers. The provers do not exchange any messages between themselves. Their initial private registers $(\mathsf{W}_0^1, \mathsf{W}_0^2)$ may be initialized in an arbitrary joint entangled state.

with distinct provers. For instance, X_1^2 denotes the register containing the first message sent by the verifier to the second prover and W_0^1 the register representing the first prover's private memory at the start of the game. Figure 6.1 provides an illustration of a four-turn (or two-round) interactive game between a verifier and two provers. As in the single-prover case, one may assume without loss of generality that all of the verifier and prover actions in a multi-prover interactive game are represented by isometric channels acting on pure states, or unitary channels provided that sufficiently large ancillary spaces are made available for each participant at the start of the game.

The initial state of the k provers' private registers $\mathsf{W}_0^1, \dots, \mathsf{W}_0^k$ will play a particularly important role in multi-prover interactive games. While it is always possible in the case of single-prover games to assume, without any loss of generality, that the prover's starting register W_0 is initialized to the all-zero standard basis state (or to ignore the existence of this register altogether), this is no longer the case for multiple provers. An alternative preparation of a single prover's starting register could always be incorporated into this prover's first action, but multiple provers might benefit from shared starting states (especially *entangled states*) that cannot be prepared locally. Provers whose private registers are initialized to a product state, which could be prepared locally and independently by each prover, will be referred to as *unentangled* provers. General provers, on the other hand, are permitted to start the game with the collection of registers $(\mathsf{W}_0^1, \ldots, \mathsf{W}_0^k)$ initialized in an arbitrary quantum state. Such provers will typically be called *entangled* provers, and the shared starting state will be referred to as their *prior shared entanglement*.

The following example demonstrates that the set of *entangled strategies*, or strategies that can be implemented by entangled provers having access to prior shared entanglement, is strictly larger than the set of *unentangled strategies* associated with provers restricted to initial product states.

Example 6.1 (Coherent state exchange game). Consider the following one-round two-prover verifier $V = (V_1, V_2)$. Following a well-established convention the two provers will be given the names *Alice* and *Bob*. In the game defined by this verifier, the registers Z_1, Y_1^1, and Y_1^2 are qubit registers while X_1^1 and X_1^2 are qutrit registers (having standard basis states $|0\rangle$, $|1\rangle$, and $|2\rangle$).

1. The verifier prepares the registers $(\mathsf{Z}_1, \mathsf{X}_1^1, \mathsf{X}_1^2)$ in the pure state

$$\frac{1}{\sqrt{2}}|0\rangle|00\rangle + \frac{1}{\sqrt{2}}|1\rangle|\phi\rangle, \tag{6.1}$$

for

$$|\phi\rangle = \frac{1}{\sqrt{2}}|11\rangle + \frac{1}{\sqrt{2}}|22\rangle. \tag{6.2}$$

It sends X_1^1 to Alice and X_1^2 to Bob.

2. Alice and Bob respond with the registers Y_1^1 and Y_1^2, respectively.

3. The verifier measures the registers $(\mathsf{Z}_1, \mathsf{Y}_1^1, \mathsf{Y}_1^2)$ using a binary projective measurement $\{\Pi_1, \mathbb{1} - \Pi_1\}$, with the outcome 1 being associated with the projector $\Pi_1 = |\gamma\rangle\langle\gamma|$, for

$$|\gamma\rangle = \frac{1}{\sqrt{2}}|0\rangle|00\rangle + \frac{1}{\sqrt{2}}|1\rangle|11\rangle. \tag{6.3}$$

Intuitively speaking, the provers Alice in Bob are aiming to transform $(\mathsf{X}_1^1, \mathsf{X}_1^2)$ into $(\mathsf{Y}_1^1, \mathsf{Y}_1^2)$ in such a way that (i) the state $|00\rangle$ is transformed to $|00\rangle$, (ii) the state $|\phi\rangle$ is transformed to $|11\rangle$, and (iii) the "superposition" between Z_1 being in the states $|0\rangle$ and $|1\rangle$ is not disturbed. This is challenging for them because $|\phi\rangle$ is entangled while $|00\rangle$ is not.

This game has the particularity that the provers' maximum probability of convincing the verifier to produce the outcome 1 increases with the dimension of their initial private registers W_0^1 and W_0^2. Informally speaking, this is so because the entanglement present in the state $|\phi\rangle$ can be "hidden" within a vast reservoir of entanglement in such a way that the "superposition" between Z_1 being in the states $|0\rangle$ and $|1\rangle$ is not disturbed. (The idea is essentially the reverse of the *embezzling* of entanglement phenomenon of van Dam and Hayden [161].) More quantitatively, as shown in [121], unentangled provers can achieve a success probability of at most 3/4 in this game, but optimal entangled provers sharing a state of local dimension d succeed with probability $1 - \Theta(\log^{-2} d)$, which tends to 1 as $d \to \infty$.

In particular, a simple strategy achieving a success probability that approaches 1 as the dimension of the provers' shared entangled state grows can be devised as follows. Suppose that Alice and Bob share the entangled state

$$\frac{1}{\sqrt{N}} \sum_{i=1}^{N} |00\rangle^{\otimes i} \otimes |\phi\rangle^{\otimes (N-i+1)} \tag{6.4}$$

for a very large value of N, where each copy of $|00\rangle$ and $|\phi\rangle$ represents the state of a pair of qutrits shared between Alice and Bob. Using this state as a resource, Alice and Bob can approximately convert $|00\rangle$ to $|\phi\rangle$ (or *vice versa*) by the unitary process which performs a cyclic rotation of the $(N+1)$ registers in their possession; the term *embezzlement* comes from the fact that this process will leave the entanglement almost unchanged (for large N). When used as a subroutine, this process allows Alice and Bob to win the game described above with probability approaching 1 as N goes to infinity.

Example 6.1 suggests the introduction of two distinct quantities to measure the maximum acceptance probability of the verifier in a multi-prover interactive game:

- The *unentangled value*, denoted $\omega(V)$, is the highest probability with which the verifier can be made to output 1 when interacting with provers whose private registers are all initialized to the all-zero product state $|0\cdots 0\rangle$.
- The *entangled value* $\omega^*(V)$ is defined as the supremum over all finite-dimensional Hilbert spaces $\mathcal{W}_0^1, \ldots, \mathcal{W}_0^k$, corresponding to the provers' initial private registers $\mathsf{W}_0^1, \ldots, \mathsf{W}_0^k$, and all initial pure states[1] $|\psi\rangle \in \mathcal{W}_0^1 \otimes \cdots \otimes \mathcal{W}_0^k$ of these registers, of the provers' maximum probability of causing the verifier to output 1.

These two values lead to potentially distinct classes of problems having multi-prover interactive proof systems: QMIP for the case of unentangled provers and QMIP* when the provers are allowed to share arbitrary entangled states.

Definition 6.1. A promise problem $A = (A_{\text{yes}}, A_{\text{no}})$ is contained in the complexity class $\text{QMIP}_{a,b}(k, m)$ if and only if there exists a polynomial-time computable function V that possesses the following properties:

1. For every string $x \in A_{\text{yes}} \cup A_{\text{no}}$, one has that $V(x)$ is an encoding of a quantum circuit description of an m-turn verifier in an interactive game with k provers.
2. *Completeness.* For every string $x \in A_{\text{yes}}$, it holds that $\omega(V(x)) \geq a$.
3. *Soundness.* For every string $x \in A_{\text{no}}$, it holds that $\omega(V(x)) \leq b$.

The complexity class $\text{QMIP}^*_{a,b}(k, m)$ is defined in the same way, except that the quantity $\omega^*(V(x))$ replaces $\omega(V(x))$.

Similar conventions to those in the single-prover setting will be used to refer to the classes above. For instance, we denote

$$\begin{aligned} \text{QMIP}(k, m) &= \text{QMIP}_{2/3,1/3}(k, m), \\ \text{QMIP}^*(k, m) &= \text{QMIP}^*_{2/3,1/3}(k, m). \end{aligned} \tag{6.5}$$

[1] Similar to the classical setting in which shared randomness does not affect the power of multiple provers, there is no increase in power for multiple quantum provers when their initial private registers are in a mixed quantum state, as compared with a pure state.

We let $\mathrm{QMIP}_{a,b}$ and $\mathrm{QMIP}^*_{a,b}$ denote the classes of promise problems A for which $A \in \mathrm{QMIP}_{a,b}(k,m)$ or $A \in \mathrm{QMIP}^*_{a,b}(k,m)$, respectively, for some choice of polynomially bounded functions k and m, and we denote $\mathrm{QMIP} = \mathrm{QMIP}_{2/3,1/3}$ and $\mathrm{QMIP}^* = \mathrm{QMIP}^*_{2/3,1/3}$.

In addition, the classes $\mathrm{MIP}_{a,b}(k,m)$ and $\mathrm{MIP}^*_{a,b}(k,m)$ are defined in an analogous way, except that the verifier is classical (specified by a classical Boolean circuit that may take a uniformly random bit string as an auxiliary input). All messages exchanged with the provers are restricted to being classical strings in this case.

The fact that both the completeness and soundness parameters of the classes $\mathrm{QMIP}_{a,b}(k,m)$ and $\mathrm{QMIP}^*_{a,b}(k,m)$ are defined with respect to ω and ω^*, respectively, makes their relationship non-obvious. The inequality $\omega^* \geq \omega$ always holds, but it can have countervailing effects. First, it implies that a proof system sound against unentangled provers may no longer be sound when the provers are allowed to share entanglement. Second, a proof system achieving a certain completeness parameter with entangled provers may not have the same property when the provers are restricted to unentangled strategies. Because both the soundness and completeness parameters are affected in possibly different ways, it is not clear in which cases the presence of a gap between the parameters (corresponding to the distinction between yes- and no-inputs) is preserved. This phenomenon will be discussed in greater detail in subsequent sections.

As in the single-prover setting, the choice of completeness and soundness parameters a, b does not affect the class of problems that lie in $\mathrm{QMIP}_{a,b}$ or $\mathrm{QMIP}^*_{a,b}$, so long as they are polynomially separated—any inverse polynomial separation between a and b can be amplified in a straightforward way, either by repeating the game sequentially or with different sets of provers. The following proposition states this fact in more precise terms.

Proposition 6.1. Let V be a verifier in a k-prover m-turn interactive game and let $a, b \in [0,1]$ be real numbers such that $a > b$. For every positive integer T, there exists a verifier V' in a k-prover, Tm-turn interactive game (or, alternatively, a verifier V' in a Tk-prover, m-turn interactive game) for which the implications

$$\begin{aligned} \omega(V) \geq a \;\Rightarrow\; \omega(V') &\geq 1 - \exp\Big(-\frac{(a-b)^2}{2}T\Big) \\ \omega(V) \leq b \;\Rightarrow\; \omega(V') &\leq \exp\Big(-\frac{(a-b)^2}{2}T\Big) \end{aligned} \tag{6.6}$$

hold. Furthermore, the same implications hold for ω^* (under the same transformation).

The fact that this procedure works as described when repeated in parallel with Tk entangled provers follows along the same lines as for sequential repetition, as it can always be considered that the interaction with each group of k provers is performed in sequence.

6.2 The importance of entanglement

The first part of this section, Section 6.2.1, is devoted to proof systems with multiple unentangled provers. It will be shown that quantum verifiers have exactly the same power as classical verifiers in this setting: QMIP = MIP = NEXP. The proof relies on the characterization MIP = NEXP, but is otherwise not difficult. Thus, in the absence of entanglement between the provers, quantum verifiers are neither less nor more powerful than their classical counterparts.

In the second part of this section, Section 6.2.2, it will be argued that the situation is markedly different in the presence of entangled provers. In particular, the technique of *oracularization*, which is central to establishing the soundness property of natural proof systems for NEXP-complete problems, is shown to fail for entangled provers in its most standard form.

Not only does entanglement allow provers to break the soundness of simple proof systems, but for certain restricted classes of verifiers it appears to be impossible (under commonly conjectured complexity-theoretic assumptions) to modify proof systems in such a way as to make them sound against entangled provers. This will be demonstrated in Section 6.2.3 for the special case of XOR proof systems, for which the associated class with unentangled provers, $\oplus$MIP, equals NEXP, but collapses to a subset of PSPACE when the provers are allowed to share prior entanglement.

6.2.1 Provers without prior shared entanglement: QMIP=NEXP

This section considers interactive proof systems based on games in which the provers do not share any prior entanglement. As will be shown, the class QMIP of promise problems that can be decided by such proof systems exactly coincides with the class MIP = NEXP of problems that can be decided by a classical verifier interacting with multiple unentangled provers. The situation in this case is therefore analogous to the single-prover setting, where the equality QIP = IP demonstrates that the ability to exchange quantum information does not affect the verification power of the verifier.

The proof that QMIP coincides with NEXP relies on two separate inclusions. The first inclusion is

$$\mathrm{QMIP} \subseteq \mathrm{NEXP}, \tag{6.7}$$

which follows from the existence of a nondeterministic exponential-time procedure for determining the unentangled value of a multi-prover interactive game with high accuracy. Second, the containment

$$\mathrm{MIP} \subseteq \mathrm{QMIP} \tag{6.8}$$

is easily seen to hold, as a quantum verifier can simulate a classical verifier in a straightforward way by systematically measuring the provers' messages in the standard basis. This leaves the value of the game unchanged, as unentangled provers gain no advantage from using quantum information against a classical verifier. The equality

$$\mathrm{QMIP} = \mathrm{NEXP} \tag{6.9}$$

follows by combining (6.7) and (6.8) together with the characterization MIP = NEXP, which is an important classical result to which we will return in Section 6.4. We are not aware of a direct proof of QMIP = MIP that does not rely on this characterization.

One consequence of the equality QMIP = MIP is that various results applying to classical multi-prover interactive proof systems immediately extend to their quantum unentangled counterparts. For instance, it is known that such proof systems can be given perfect completeness and exponentially small soundness error, and can be parallelized to a single round of interaction with just two provers.

Theorem 6.2. For every positive polynomially bounded function p it holds that

$$\text{QMIP} = \text{QMIP}_{1,2^{-p}}(2,2) = \text{NEXP}. \tag{6.10}$$

Assuming the known results on MIP just suggested (about which more will be said when we discuss their entangled-prover counterparts in Section 6.3), one therefore has that Theorem 6.2 follows from the inclusion (6.7).

With the goal of proving (6.7) in mind, consider the problem of certifying the provers' maximum acceptance probability in a given k-prover, m-turn interactive game. An arbitrary strategy for the provers can be specified by an explicit description of the j-th prover's isometry in the i-th round,

$$P_i^j \in \text{U}(\mathcal{W}_{i-1}^j \otimes \mathcal{X}_i^j, \mathcal{W}_i^j \otimes \mathcal{Y}_i^j), \tag{6.11}$$

for all $j = 1, \ldots, k$ and $i = 1, \ldots, \lceil m/2 \rceil$. Putting issues of precision aside, which can be handled by specifying rational approximations with exponential accuracy to the real and imaginary part of each complex matrix entry, the probability of the verifier outputting 1 in the corresponding interaction can be computed by performing the appropriate matrix operations.

The inclusion QMIP $\subseteq$ NEXP will therefore follow once it is proved that there exists an optimal prover strategy that can be specified by isometries of dimension at most exponential in the description size of the verifier. Because the message registers necessarily satisfy such a bound, it will suffice to bound the dimension of the private register W_i^j associated with the j-th prover's isometry in the i-th round. Such a bound can be obtained based on the following theorem (which represents a very minor extension of Theorem 2.2).

Theorem 6.3. Let $\mathcal{X}$, $\mathcal{Y}$, $\mathcal{V}$, and $\mathcal{W}$ be finite-dimensional Hilbert spaces with $\dim(\mathcal{W}) \geq \dim(\mathcal{V}) = \dim(\mathcal{X} \otimes \mathcal{Y})$, and let $A \in \text{U}(\mathcal{X}, \mathcal{W} \otimes \mathcal{Y})$ be

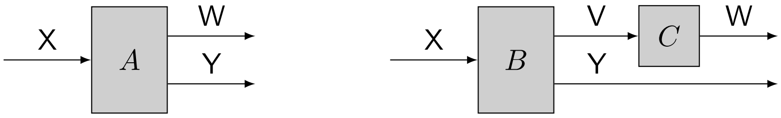


Figure 6.2: An isometry A transforming X to (W, Y) is necessarily equivalent to an isometry B transforming X to (V, Y), followed by an isometry C transforming V to W, assuming V has the same size as (X, Y) and W is at least this large.

an isometry. There exist isometries $B \in \mathrm{U}(\mathcal{X}, \mathcal{V} \otimes \mathcal{Y})$ and $C \in \mathrm{U}(\mathcal{V}, \mathcal{W})$ such that

$$A = (C \otimes \mathbb{1}_{\mathcal{Y}})B. \tag{6.12}$$

Figure 6.2 illustrates this theorem in the form of a picture suggestive of a circuit diagram.

Through the use of this theorem, one may replace a given prover P^j by an equivalent prover Q^j that substitutes a register V_i^j, which has dimension equal to the product of the dimensions of the message registers

$$\mathsf{X}_1^j, \mathsf{Y}_1^j, \ldots, \mathsf{X}_i^j, \mathsf{Y}_i^j, \tag{6.13}$$

for each register W_i^j used by P^j (which we assume has been specified by a collection of isometries $\{P_i^j\}$ as in (6.11)). The theorem is applied independently to each prover action, beginning with $i = 1$ and increasing to $i = \lceil m/2 \rceil$.

In particular, one starts with $i = 1$, and takes $\mathcal{X} = \mathcal{X}_1^j$, $\mathcal{Y} = \mathcal{Y}_1^j$, $\mathcal{W} = \mathcal{W}_1^j$, and $\mathcal{V}$ being a space with

$$\dim(\mathcal{V}) = \dim(\mathcal{X} \otimes \mathcal{Y}) = \dim(\mathcal{X}_1^j \otimes \mathcal{Y}_1^j), \tag{6.14}$$

corresponding to a new private memory register V_1^j that will replace the register W_1^j. The first isometry performed by the new prover Q^j is the isometry $Q_1^j \in \mathrm{U}(\mathcal{X}_1^j, \mathcal{V}_1^j \otimes \mathcal{Y}_1^j)$ that is represented by B in the theorem. The isometry C from the theorem is composed with P_2^j, and the process is repeated for $i = 2, \ldots, \lceil m/2 \rceil$. In general, one applies the theorem with $\mathcal{X} = \mathcal{V}_{i-1}^j \otimes \mathcal{X}_i^j$ (with $\mathcal{V}_0 = \mathbb{C}$), $\mathcal{Y} = \mathcal{Y}_i^j$, $\mathcal{W} = \mathcal{W}_i^j$, and $\mathcal{V}$ being a space with

$$\dim(\mathcal{V}) = \dim(\mathcal{X} \otimes \mathcal{Y}) = \dim(\mathcal{V}_{i-1}^j \otimes \mathcal{X}_i^j \otimes \mathcal{Y}_i^j), \tag{6.15}$$

corresponding to a new private memory register V_i^j that will replace the register W_i^j. The size of the memory register V_i^j obtained in this way therefore has the same size as the tuple of registers

$$(\mathsf{X}_1^j, \mathsf{Y}_1^j, \ldots, \mathsf{X}_i^j, \mathsf{Y}_i^j). \tag{6.16}$$

When the theorem is applied to each prover's final operation, the isometry C is simply discarded—as the verifier never touches the provers' private memory registers, nothing is lost in disregarding this isometry.

It is worth noting that, in contrast to the single-prover case, it is not known if an efficient optimization over strategies for the provers in a multi-prover interactive game is possible (given an explicit matrix description of a verifier). One cannot accomplish such an optimization by considering only the local properties of a sequence of reduced states of the verifier's private and message registers at each turn of the interactive game in a manner similar to the single-prover setting, as there is no known analogue of Theorem 2.1 on the unitary equivalence of purifications that would apply to the setting of multiple quantum provers.

For example, consider a setting in which three single-qubit registers $(\mathsf{X}_1, \mathsf{Z}, \mathsf{X}_2)$ are in the mixed state

$$|0\rangle\langle 0| \otimes \frac{\mathbb{1}}{2} \otimes |0\rangle\langle 0|. \tag{6.17}$$

Prover 1 is permitted to transform X_1 into Y_1 and prover 2 transforms X_2 into Y_2, where Y_1 and Y_2 are also single-qubit registers. One may ask if it is possible for the provers to transform the original state (6.17) into one of the three states

$$|\phi\rangle\langle\phi| \otimes |0\rangle\langle 0|, \quad |0\rangle\langle 0| \otimes |\phi\rangle\langle\phi|, \quad \text{or} \quad |\gamma\rangle\langle\gamma| \tag{6.18}$$

of $(\mathsf{Y}_1, \mathsf{Z}, \mathsf{Y}_2)$, where

$$\begin{aligned} |\phi\rangle &= \frac{1}{\sqrt{2}}|0\rangle|0\rangle + \frac{1}{\sqrt{2}}|1\rangle|1\rangle, \\ |\gamma\rangle &= \frac{1}{\sqrt{2}}|0\rangle|0\rangle|0\rangle + \frac{1}{\sqrt{2}}|1\rangle|1\rangle|1\rangle. \end{aligned} \tag{6.19}$$

All three transformations may or may not be possible, depending on the initial correlations among the registers X_1, Z, and X_2 and two additional registers W_1 and W_2 representing the memories of prover 1 and prover 2, respectively. For instance, if $(\mathsf{W}_1, \mathsf{Z})$ is initially in the pure state $|\phi\rangle$, then the provers are capable of transforming the original state (6.17) into the first state of (6.18), but neither the second nor the third. A transformation to either the second or third state is also possible assuming different initial states of $(\mathsf{W}_1, \mathsf{X}_1, \mathsf{Z}, \mathsf{X}_2, \mathsf{W}_2)$. The ability of the provers to perform a particular transformation is therefore not a function of the states in question, but also depends on the initial state of the provers' memories.

6.2.2 The failure of oracularization

The equality QMIP = MIP demonstrates that quantum multi-prover interactive proof systems with unentangled provers are no more powerful than their classical counterparts. As already discussed, allowing the use of prior shared entanglement for the provers can raise the value of an interactive game, affecting both the soundness and completeness parameters of a proof system. As a result, both inclusions on which the aforementioned equality are based,

$$\text{QMIP} \subseteq \text{NEXP} \quad \text{and} \quad \text{MIP} \subseteq \text{QMIP}, \tag{6.20}$$

may in principle fail for entangled provers.

The possible failure of the inclusion $\text{QMIP}^* \overset{?}{\subseteq} \text{NEXP}$ is directly related to the absence of an analogue of Theorem 6.3 for provers sharing prior entanglement of arbitrary dimension. This issue will be discussed in greater detail in Section 6.5.2.

The possible failure of the second inclusion, $\text{MIP} \overset{?}{\subseteq} \text{QMIP}^*$, is suggested by Example 6.1, which demonstrates that the entangled value can be much larger than the unentangled value. To investigate how entanglement may allow the provers to break the soundness property of a proof system, we study this effect in more detail in the context of so-called *oracularized games*. Oracularization is a technique frequently employed in the study of the class MIP—it allows for a reduction in both the numbers of rounds and provers required by proof systems for problems in MIP, and it plays an important role in known proofs of the inclusion NEXP $\subseteq$ MIP. The failure of this technique in the presence of

The input is a collection of constrains $\varphi = (C_1, \ldots, C_m)$ on variables $x_1, \ldots, x_n$

Verifier's step 1: Select a constraint C_j uniformly at random, and send C_j to Alice. Also select a variable x_i on which C_j acts, uniformly at random, and send x_i to Bob.

Provers' actions: Alice replies with an assignment to all variables in C_j. Bob replies with an assignment to x_i.

Verifier's step 2: Output 1 if and only if the provers' assignments are consistent on x_i and satisfy the constraint C_j.

Figure 6.3: Clause-versus-variable interactive game.

entanglement between the provers is a source of considerable difficulty in working with the class QMIP*.

Oracularization leverages the presence of multiple provers by using one of the provers to check that the others provide answers in a non-adaptive manner. As an example demonstrating this technique, consider the one-round two-prover *clause-versus-variable* interactive game described in Figure 6.3. In this game, the provers' goal is to convince the verifier of the satisfiability of a set of constraints φ, where each constraint acts on a subset of n variables $x_1, \ldots, x_n$ taking values in some finite alphabet Γ. Similar ideas play an important role in the proof that MIP = NEXP.

Consider first the value of this game when the two provers, Alice and Bob, are restricted to classical deterministic strategies. In this case, a strategy for Bob is a function mapping each variable x_i to an element of the alphabet Γ, so that the strategy coincides with a complete assignment to the variables. For each constraint C that this assignment fails to satisfy, there is a probability at least $1/\ell$ that the provers will fail, for ℓ being the number of variables in the constraint C; either Alice's assignments fail to satisfy C, or they must differ from the assignment represented by Bob's strategy on at least one variable. Consequently, for V_φ being the verifier defined in the clause-versus-variable game for φ, one has that

$$\omega(\varphi) \leq \omega(V_\varphi) \leq 1 - \frac{1-\omega(\varphi)}{\ell}, \tag{6.21}$$

where $\omega(\varphi)$ is the maximum fraction of constraints that are simultaneously satisfiable in φ, and where it has been assumed that ℓ variables appear in each constraint.

Unfortunately this technique fails in the presence of shared entanglement between the provers, as is demonstrated by the following example.

Example 6.2 (The Magic Square game). Consider a 3 × 3 matrix of Boolean variables

$$\begin{pmatrix} X_1 & X_2 & X_3 \\ X_4 & X_5 & X_6 \\ X_7 & X_8 & X_9 \end{pmatrix}, \tag{6.22}$$

and define a one-round two-prover interactive game as follows.

1. The verifier first chooses either a row or a column in the 3 × 3 matrix of Boolean random variables, uniformly at random from the 6 possible choices, and sends these variables to the first prover Alice. The verifier also selects one of the three variables in the chosen row or column, uniformly at random from the 3 possibilities, and sends this variable to the second prover Bob.
2. The provers must respond with Boolean assignments to the variables they were sent.
3. The verifier outputs 1 (i.e., accepts) if and only if the following conditions hold:

   (a) Both Alice and Bob give the same assignment to the one variable they were sent in common.

   (b) If the verifier initially selected a *row* in the 3 × 3 matrix of Boolean random variables, then Alice's assignments to these variables must have *even parity.*

(b) If the verifier initially selected a *column* in the 3×3 matrix of Boolean random variables, then Alice's assignments to these variables must have *odd parity.*

If the provers employ a classical strategy in this game, their probability of causing the verifier to accept is at most 17/18. This follows from the fact that any deterministic strategy for Bob must determine an assignment to the Boolean variables $X_1, \ldots, X_9$, and no assignment to these variables can satisfy all six of the parity constraints (because the parity of all 9 Boolean variables cannot be both even and odd). It is straightforward to see that there exists a deterministic strategy for the provers that succeeds with probability exactly 17/18, which establishes that this upper-bound is achievable. (For instance, Alice may respond with assignments (0,0,0), (0,0,0), and (1,1,0) for rows 1, 2, and 3, respectively, and assignment (0,0,1) for all three columns; and Bob may answer in a manner consistent with the assigment $(X_1, \ldots, X_9) = (0,0,0,0,0,0,1,1,0)$. This strategy only loses in the case that the verifier asks Alice for an assignment of column 3 and Bob for an assignment to X_9.)

In contrast, entangled provers have a perfect strategy for this game—they can win with certainty. One strategy for the provers that achieves this goal is based on the construction of nine ± 1-*observables* $H_1, \ldots, H_9 \in \mathrm{Herm}(\mathbb{C}^4)$, meaning that they are Hermitian operators whose eigenvalues are all either 1 or -1, having the following properties: if H_i is placed in the i-th position of the 3×3 magic square, then

(i) the operators appearing in the same row or in the same column must commute, and

(ii) the product of the operators appearing in each row is $\mathbb{1}$, and the product of the operators appearing in each column is $-\mathbb{1}$.

Such operators can be constructed from the Pauli operators

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{6.23}$$

as follows:

$$\begin{pmatrix} H_1 & H_2 & H_3 \\ H_4 & H_5 & H_6 \\ H_7 & H_8 & H_9 \end{pmatrix} = \begin{pmatrix} \sigma_x \otimes \sigma_y & \sigma_y \otimes \sigma_x & \sigma_z \otimes \sigma_z \\ \sigma_y \otimes \sigma_z & \sigma_z \otimes \sigma_y & \sigma_x \otimes \sigma_x \\ \sigma_z \otimes \sigma_x & \sigma_x \otimes \sigma_z & \sigma_y \otimes \sigma_y \end{pmatrix}. \tag{6.24}$$

From each of these observables a projective measurement $\{\Pi_0^k, \Pi_1^k\}$, for $k = 1, \ldots, 9$, can be defined with $\Pi_a^k = (\mathbb{1} \otimes \mathbb{1} + (-1)^a H_k)/2$, for $a \in \{0,1\}$, being the projector on the eigenspace of H_k with associated eigenvalue $(-1)^a$. Suppose the provers share the entangled state

$$|\psi\rangle = \frac{1}{2}|00\rangle|00\rangle + \frac{1}{2}|01\rangle|01\rangle + \frac{1}{2}|10\rangle|10\rangle + \frac{1}{2}|11\rangle|11\rangle, \tag{6.25}$$

in which they both hold two qubits. This state has the property that

$$\langle\psi|A \otimes B|\psi\rangle = \frac{1}{4}\operatorname{Tr}(AB^{\intercal}) \tag{6.26}$$

for any $A, B \in \mathrm{L}(\mathbb{C}^4)$. When Alice (or Bob) receives the labels of some variables, she will perform the measurements described above, in sequence, to the pair of qubits she holds, using the measurement $\{\Pi_0^k, \Pi_1^k\}$ to determine the assignment she responds with for the variable X_k. Property (i) of the H_i ensures that measurements within any single row or column commute, so it does not matter which order Alice would choose to perform these measurements. Using property (ii) and (6.26), it may be verified that the required parity conditions will always hold for the outcomes of these measurements, and that Alice and Bob will always produce the same assignment to the variable they both received.

Based on Example 6.2 a simple 3-SAT formula φ with 9 variables and 24 clauses can be devised such that φ is not satisfiable but the clause-vs-variable interactive game defined from this formula can be won with certainly by entangled provers. In contrast, the unentangled value is strictly less than 1.

Consequently, one has that the oracularization technique does not extend directly to the case of entangled provers. In the following section it is shown that this failure is not limited to specific examples such as the Magic Square game, but can affect the verification possibilities of broad classes of verifiers in multi-prover interactive games.

6.2.3 XOR games

XOR games are a class of two-prover one-round interactive games in which the verifier is restricted to have the following form. The verifier's message (also called its *question*) to each prover in the first turn is classical. Each prover's message (its *answer*) to the verifier in the second turn is classical and consists of a single bit. Finally the verifier decides on its output bit based solely on the parity of the provers' answers. It is possible to consider XOR games with any number of provers, but in this section we focus on the case of two-prover XOR games.

The class of promise problems that can be decided by verifiers having this restricted form is denoted $\oplus\mathrm{MIP}_{a,b}(2,2)$ in case the provers are unentangled, and $\oplus\mathrm{MIP}^*_{a,b}(2,2)$ with entangled provers. An important result in the field of hardness of approximation states that the unentangled-prover class is powerful enough to capture all problems in NEXP, meaning that the inclusion

$$\mathrm{NEXP} \subseteq \oplus\mathrm{MIP}_{a,b}(2,2) \tag{6.27}$$

holds for a specific choice of constants $0 < b < a < 1$. In contrast, allowing entanglement between the provers reduces the verifier's decision power (under the assumption that PSPACE is properly contained in NEXP):

$$\oplus\mathrm{MIP}^*_{a,b}(2,2) \subseteq \mathrm{PSPACE}, \tag{6.28}$$

which holds for any $0 \leq b < a \leq 1$ separated by at least an inverse polynomial. Thus, the introduction of entanglement has the effect of collapsing the verifier's ability to decide problems, from NEXP to PSPACE.

The inclusion (6.28) can be shown by giving a direct simulation of any $\oplus\mathrm{MIP}^*(2,2)$ verifier by a $\mathrm{QIP}(2)$ verifier, concluding via the inclusion $\mathrm{QIP} \subseteq \mathrm{PSPACE}$ described in Section 4.4. In the remainder of this section we will describe the weaker inclusion $\oplus\mathrm{MIP}^*_{a,b}(2,2) \subseteq \mathrm{EXP}$, which has the advantage that it can be proven by expressing the entangled value ω^* of an XOR game directly as the optimum of a semidefinite program.

The verifier V in an XOR game can be specified explicitly as a pair (π, V), consisting of a distribution π on pairs of questions $(x, y) \in X \times Y$ and a predicate $V(c \mid x, y)$ that dictates the parities $c = a \oplus b$ for the provers' answers a and b that cause the verifier to accept. A strategy for the provers consists of a choice of Hilbert spaces $\mathcal{V}$ and $\mathcal{W}$, a pure state $|\psi\rangle \in \mathcal{V} \otimes \mathcal{W}$, and two families of binary-valued measurements

$$\{P_0^x, P_1^x\} \quad \text{and} \quad \{Q_0^y, Q_1^y\} \tag{6.29}$$

on the spaces $\mathcal{V}$ and $\mathcal{W}$, respectively. Upon receiving questions (x, y), the probability that the provers return answers (a, b) is

$$\langle\psi|P_a^x \otimes Q_b^y|\psi\rangle. \tag{6.30}$$

Thus, $\omega^*(V)$ is equal to the supremum value of the expression

$$\sum_{x,y} \pi(x,y) \sum_c V(c \,|\, x,y) \sum_{\substack{a,b \\ a\oplus b=c}} \langle\psi|P_a^x \otimes Q_b^y|\psi\rangle, \tag{6.31}$$

over all strategies for the provers, as described above.

Up to an additive scaling of ω^*, one may assume that for each pair (x, y), there is a unique $c \in \{0, 1\}$ such that $V(c \,|\, x, y) = 1$. For each question pair (x, y), let $r(x, y) = (-1)^c \pi(x, y)$ for this unique choice of c. It holds that

$$\omega^*(V) = \frac{1}{2} + \frac{1}{2}\beta^*(V), \tag{6.32}$$

where the *bias* $\beta^*(V)$ is defined as

$$\beta^*(V) = \sup \sum_{x,y} r(x,y)\langle\psi|A_x \otimes B_y|\psi\rangle, \tag{6.33}$$

where $A_x = P_0^x - P_1^x$ and $B_y = Q_0^y - Q_1^y$, and the supremum is over all strategies as before. (The operators A_x and B_y are observables that represent the binary-valued measurements $\{P_0^x, P_1^x\}$ and $\{Q_0^y, Q_1^y\}$.)

Example 6.3 (CHSH game)**.** A simple XOR game is the CHSH game, named after its inventors Clauser, Horne, Shimony and Holt [44]. In this game, the verifier's questions consist of a single bit each, the distribution π is uniform on $\{0, 1\} \times \{0, 1\}$, and the predicate representing the verifier's final decision is defined as

$$V(c \,|\, x, y) = \begin{cases} 1 & \text{if } c = x \wedge y \\ 0 & \text{if } c \neq x \wedge y. \end{cases} \tag{6.34}$$

It therefore holds that $r(x, y) = (-1)^{x \wedge y}/4$. The bias $\beta^*(\text{CHSH})$ is given by the expression

$$\sup_{|\psi\rangle, A_0, A_1, B_0, B_1} \frac{1}{4} \langle\psi|\big(A_0 \otimes B_0 + A_1 \otimes B_0 + A_0 \otimes B_1 - A_1 \otimes B_1\big)|\psi\rangle, \quad (6.35)$$

where the supremum is over all bipartite states $|\psi\rangle$ and observables A_0, A_1, B_0, B_1. By considering the choices

$$\begin{aligned} |\psi\rangle &= \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle), \\ A_0 = \sigma_x, &\quad B_0 = (\sigma_x + \sigma_z)/\sqrt{2}, \\ A_1 = \sigma_z, &\quad B_1 = (\sigma_x - \sigma_z)/\sqrt{2}, \end{aligned} \quad (6.36)$$

one finds that $\beta^*(\text{CHSH}) \geq \sqrt{2}/2$. That this holds with equality will be show below. In contrast, unentangled provers are easily seen to achieve a bias at most $\beta(\text{CHSH}) = 1/2$.

There exists a natural semidefinite programming relaxation for the bias of a given XOR game as follows. First, let R be a matrix indexed by the disjoint union $X \sqcup Y$ of the question sets, defined as

$$R(x, y) = R(y, x) = \frac{r(x, y)}{2} \quad \text{and} \quad R(x, x') = R(y, y') = 0 \quad (6.37)$$

for all $x, x' \in X$ and $y, y' \in Y$. Next, for a given strategy, defined by a shared entangled state $|\psi\rangle$ and collections of ± 1-observables $\{A_x\}$ and $\{B_y\}$, define unit vectors

$$u_x = (A_x \otimes \mathbb{1})|\psi\rangle \quad \text{and} \quad v_y = (\mathbb{1} \otimes B_y)|\psi\rangle, \quad (6.38)$$

and observe that

$$\langle\psi|A_x \otimes B_y|\psi\rangle = \langle u_x, v_y\rangle \quad (6.39)$$

for every pair $(x, y) \in X \times Y$. One finds that the bias obtained by this particular strategy is given by

$$\sum_{x,y} r(x, y)\langle\psi|A_x \otimes B_y|\psi\rangle = \langle R, Z\rangle, \quad (6.40)$$

for Z the Gram matrix of the collection $\{u_x : x \in X\} \cup \{v_y : y \in Y\}$, i.e.,

$$\begin{aligned} Z(x, y) = \langle u_x, v_y\rangle, &\quad Z(x, x') = \langle u_x, u_{x'}\rangle, \\ Z(y, x) = \langle v_y, u_x\rangle, &\quad Z(y, y') = \langle v_y, v_{y'}\rangle. \end{aligned} \quad (6.41)$$

(In the present case, one has that each of the values $Z(x, y)$ is real and satisfies $Z(x, y) = Z(y, x)$ because the value $\langle \psi | A_x \otimes B_y | \psi \rangle$ is a real number.)

It is therefore the case that

$$\beta^*(V) \leq \mathrm{SDP}(V) = \sup_Z \langle R, Z \rangle, \tag{6.42}$$

where the supremum is over all positive semidefinite matrices Z indexed by $X \sqcup Y$ and satisfying $Z(x, x) = Z(y, y) = 1$ for each $x \in X$ and $y \in Y$ (which reflects the fact that the vectors $\{u_x\} \cup \{v_y\}$ are unit vectors).

The relaxation (6.42) is very useful to prove upper bounds on the bias of two-prover XOR games. For the case of the CHSH game (as described in Example 6.3), the matrix R is a 4×4 matrix with both its 2×2 diagonal blocks equal to 0, and each off-diagonal block equal to

$$\frac{1}{8} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \tag{6.43}$$

The dual problem to (6.42) is

$$\mathrm{SDP}^*(\mathrm{CHSH}) = \inf_H \mathrm{Tr}(H), \tag{6.44}$$

where the infimum is over all Hermitian matrices H such that $H \geq R$. Using the fact that R squares to $(1/32)\mathbb{1}$, one finds that $H = (\sqrt{2}/8)\mathbb{1}$ provides a dual certificate with objective value $\sqrt{2}/2$. Because this value is achieved by the strategy described earlier, it follows by weak duality that $\beta^*(\mathrm{CHSH}) = \mathrm{SDP}(\mathrm{CHSH}) = \sqrt{2}/2$.

This is not a coincidence: the equality $\beta^*(V) = \mathrm{SDP}(V)$ always holds. There is an explicit mapping, due to Tsirelson [158], that shows how any feasible solution to the semidefinite program (corresponding to the operator Z above) can be transformed into a strategy for the provers (a state $|\psi\rangle$ and binary-valued measurements $\{P_0^x, P_1^x\}$ and $\{Q_0^y, Q_1^y\}$) achieving a bias equal to the objective value given by Z in (6.42).

The characterization of the bias of two-prover XOR games as the optimum of a semidefinite program has multiple consequences. First, it allows one to replace the supremum in (6.33) by an efficiently computable quantity. The inclusion of $\mathrm{MIP}^* \subseteq \mathrm{EXP}$ follows, as an explicit representation of the matrix R specifying an XOR game can be computed in exponential time from a description of a quantum circuit for the verifier, and the optimum of the resulting exponential-size semidefinite program can be approximated to within exponential precision in time polynomial in its size.

A second noteworthy consequence is a bound on the entanglement of optimal strategies in XOR games. The optimum of (6.42) is always achieved by a matrix of dimension $N = |X| + |Y|$, whose Gram factorization involves vectors of the same dimension. Tsirelson's transformation can be used to map these vectors onto a state and two collections of measurements in which each prover holds $\lfloor N/2 \rfloor$ qubits. Thus, for every XOR game there exists an optimal strategy that uses a number of qubits linear in the number of questions in the game. This is not true of more general interactive games, as demonstrated for instance by the game from Example 6.1, for which the entangled value is only achieved in the limit as the dimension of the provers' shared entangled state goes to infinity.

6.3 Using entanglement in multi-prover games

This section is devoted to the presentation of structural results, such as parallelization and perfect completeness, that apply to the class QMIP*. Some of these results parallel similar properties known to hold for classical multi-prover interactive proof systems. Proofs of the latter type of results, however, usually rely on the technique of oracularization, which was shown to fail in the presence of entangled provers in the preceding section. Thus, a direct extension of the classical results to the entangled-prover setting is not generally possible, and different proofs must be devised.

The reductions established in this section will make crucial use of entanglement between the provers—it will typically be the case that, even if honest unentangled provers could win with high probability in a certain interactive game, provers in the modified game will nevertheless still need to make use of prior shared entanglement in order to win with high probability. In some cases, entanglement will be used to achieve reductions unlikely to hold in the classical setting, such as a reduction to public-coin systems. The following properties will be shown:

1. *Perfect completeness.* Multi-prover quantum interactive proof systems can be made perfectly complete.
2. *Parallelization.* Multi-prover quantum interactive proof systems can be parallelized to three turns of interaction. Moreover, any three-turn multi-prover quantum interactive proof system can be transformed into one that is *public-coin*: the verifier's unique message is a single random bit broadcast to all provers. In addition, public-coin proof systems can be further parallelized to only two turns of interaction by introducing an additional prover.
3. *Classical verifiers.* Any multi-prover quantum interactive proof system can be transformed into one in which the verifier is classical at the cost of considering two additional provers and a polynomial increase in the number of rounds of interaction.

Putting these properties together, any k-prover quantum interactive proof system can be transformed into a one-round proof system with $k+1$ provers, perfect completeness, and soundness bounded away from 1 by an inverse polynomial.

Theorem 6.4. For every polynomially bounded functions k and m it holds that

$$\mathrm{QMIP}^*(k,m) \subseteq \mathrm{QMIP}^*_{1,1-1/p}(k+1,2), \tag{6.45}$$

for some choice of a polynomially bounded function $p = O(m^2)$.

It is not known whether the soundness parameter of multi-prover interactive proof systems with quantum verifiers can be amplified in parallel with the same set of provers, a problem that will be discussed in Section 6.5.1. Thus, amplifying the inverse-polynomial gap in completeness and soundness from (6.45) requires a polynomial increase in either the number of provers or the number of rounds of interaction (q.v. Proposition 6.1).

Allowing for a polynomial number of rounds of interaction, the verifier can further be made classical.

Theorem 6.5. For all polynomially bounded functions k, m and q, it holds that

$$\mathrm{QMIP}^*(k, m) \subseteq \mathrm{MIP}^*_{1,2^{-q}}(k + 2, p), \qquad (6.46)$$

for some choice of a polynomially bounded function $p = O(q \cdot m^2)$.

When comparing these results with those known to hold for unentangled provers, there is a significant gap: to determine whether or not the number of provers be reduced. There is currently no compelling evidence in favor of $\mathrm{QMIP}^*(k, \mathrm{poly})$ being a larger class than $\mathrm{QMIP}^*(2, \mathrm{poly})$, but also there is no known transformation allowing a reduction of the number of provers.

6.3.1 Perfect completeness

The standard transformation to achieve perfect completeness for the class MIP proceeds as follows. Given a verifier V, a modified verifier V' is defined that executes the same procedure as V many times in parallel with a carefully chosen set of distinct private random strings. The strings are chosen so as to guarantee that, provided the provers had successful strategies for at least half of the possible choices of a random string for V', there will always be at least one string in the set for which the provers can convince the verifier V to output 1 with certainty, when V is executed with this choice of randomness.

In the case of a quantum verifier, this sort of transformation is meaningless—there is no discrete set of "random bits" for the verifier that parametrizes its verification procedure and can be easily manipulated. For this reason a different transformation is required. The reduction to be described will be similar in spirit to the one introduced in Section 4.2.1 for the single-prover case, with an important twist. Recall that, in that transformation, during the last round of interaction, the prover is required to apply a certain unitary transformation on its private register to disentangle it from the message register (q.v. Eq. 4.18). If the corresponding register in the multi-prover setting is shared between multiple provers, it may not be possible for them to implement such a unitary transformation locally. A more complicated transformation, which requires that the number of turns in the game increases from m to $3m$, will allow the provers to achieve the desired effect: they execute the entire game backward (m extra turns), and then forward again (m extra turns). It is interesting to note that even if the original verifier was classical, and the provers could achieve their maximum success probability without using any prior entanglement, the new verifier will make use of quantum messages and in general the provers may need to use prior entanglement in order to achieve the optimal success probability of 1 in the modified game.

Suppose a verifier V in a quantum multi-prover interactive game is given, along with a target threshold $\alpha \geq 1/2$ for its maximum acceptance probability $\omega^*(V)$. We will describe a transformation mapping V to a new verifier V' such that the following properties hold:

1. If V is an m-turn verifier, then V' is a $3m$-turn verifier.
2. If it is the case that $\omega^*(V) \geq \alpha$, then $\omega^*(V') = 1$.
3. It always holds that $\omega^*(V') \leq 1/2 + 2\sqrt{\omega^*(V)} + 5\omega^*(V)/2$.

By this transformation one may conclude that the following theorem holds.

Theorem 6.6. Let $a \geq 1/2$ and $b \leq 1/25$. For every choice of k and m it holds that

$$\mathrm{QMIP}^*_{a,b}(k, m) \subseteq \mathrm{QMIP}^*_{1,c}(k, 3m), \tag{6.47}$$

for $c = 1/2 + 2\sqrt{b} + 5b/2$.

To explain the idea behind the reduction, it will be convenient to replace the assumption $\omega^*(V) \geq \alpha$ in item 2 by the more specific requirement that there exists a fixed strategy for the provers with the property that the optimal success probability of this strategy, when maximized over all possible initial states of the provers' private registers, is exactly $1/2$. This is easily achieved by allowing the provers to force a rejection in order to artificially lower their success probability, along the same lines as was discussed in the single-prover setting. It will also be convenient to assume that the first turn of the game is executed by the verifier, sending a message to each of the provers.

Assuming V is given in purified form, the construction of the $3m$-turn verifier V' can be described as follows.

1. V' simulates V for the first m turns, up to but not including the final measurement of the output qubit of V.
2. V' chooses a bit $b \in \{0, 1\}$ uniformly at random. If $b = 0$ it executes the *rewinding test* described in step 3. If $b = 1$ it performs the *invertibility test* described in step 4.
3. *Rewinding test:*

   (i) V' measures the output qubit of V. If the result is 1 it stops the game and outputs 1. If it is 0, the original interactive game is executed backward in time for m turns, interacting with the provers as needed. At the last step V' applies V_1^{-1} to the registers $(\mathsf{Z}_1, \mathsf{X}_1^1, \ldots, \mathsf{X}_1^k)$, obtaining $(\mathsf{Z}_0, \mathsf{Y}_0^1, \ldots, \mathsf{Y}_0^k)$.

   (ii) V' performs a controlled-phase flip Z, multiplying the phase by -1 if all the qubits in Z_0 are in state $|0\rangle$.

   (iii) V' executes the original interactive game forward in time for m turns. It measures the output qubit of V and returns the outcome.

4. *Invertibility test:*

   (i) V' executes the original interactive game backward in time for m turns.

   (ii) After applying V_1^{-1} it applies the measurement $\{\Pi_{init}, \mathbb{1} - \Pi_{init}\}$, which measures all qubits of register Z_0 in the computational basis. If the outcome associated with Π_{init}, corresponding to all qubits being in the $|0\rangle$ state, is obtained V' returns the outcome 1; otherwise it returns 0.

Consider first the case where there exist k provers $P^1, \ldots, P^k$ such that, with the optimal choice of initial state of their private registers $\mathsf{W}_0^1, \ldots, \mathsf{W}_0^k$, the provers cause V to accept with probability exactly $1/2$. Define new provers $R^1, \ldots, R^k$ who perform precisely the same actions as the original provers when asked by the verifier, including performing the reverse action when asked to do so. It is clear that such provers will always cause the verifier to output 1 with certainty in the invertibility test. That they also cause the verifier to output 1 with certainty in the rewinding test follows from a similar analysis as was performed in the single-prover case in Section 4.2.1.

Now suppose the provers' maximum probability to convince V to accept is less than $1/25$. Let $R^1, \dots, R^k$ be arbitrary provers in the interactive game specified by V', and let $|\psi\rangle$ be the initial state of all parties' private registers, including the provers' shared entanglement, at the beginning of the game. We may introduce three unitary operators that capture the actions performed jointly by the verifier and provers in the *forward*, *backward*, and *forward* phases of the game. Unitary U_1 implements all parties' actions in the *forward* phase, including the verifier's last unitary operation $V_{m/2+1}$, but without measuring the output qubit. Unitary U_2 implements all parties' actions in the *backward* phase, starting with the verifier's application of $V^*_{m/2+1}$ and ending with V^*_1. Finally, unitary U_3 implements all parties' actions in the second *forward* phase. For the verifier, these are the same transformations that were used in the definition of U_1, but in general the provers' actions may be different.

With respect to the operators U_1, U_2, and U_3 just defined, the provers' success probability in the game may be characterized as follows. Let $p_1/2$, where $p_1 = \|\Pi_1 U_1|\psi\rangle\|^2$, be the probability that the verifier stops and accepts in step 3(i). It holds that $p_1 \leq \omega^*(V)$. The probability that the verifier stops and accepts in step 3(iii) is $p_2/2$, where

$$p_2 = \|\Pi_1 U_3 Z U_2(\mathbb{1} - \Pi_1)U_1|\psi\rangle\|^2. \tag{6.48}$$

Finally let $p_3 = \|\Pi_{init} U_2 U_1|\psi\rangle\|^2$, so that the probability that the verifier stops and accepts in step 4 is $p_3/2$.

The value $\omega^*(V') = (p_1 + p_2 + p_3)/2$ is bounded by expressing a tradeoff between p_2 and p_3. Either U_2 is such that the combined unitary U_2U_1 brings the state of all registers into one that is consistent with a possible initial state of the game specified by V. In this case the invertibility test will accept, but p_2 will be bounded by $\omega^*(V)$. Alternatively, the provers' actions in the *backwards* phase of the game are such that the combined action U_2U_1 results in a state in which the verifier's private register is not in the $|0\rangle$ state, in which case the invertibility test will reject and p_3 will be small. This tradeoff can be expressed by applying the triangle inequality as follows.

$$\begin{aligned}\sqrt{p_2} &= \|\Pi_1 U_3 Z U_2 (\mathbb{1} - \Pi_1) U_1 |\psi\rangle\| \\ &\leq \|\Pi_1 U_3 Z U_2 \Pi_1 U_1 |\psi\rangle\| + \|\Pi_1 U_3 Z U_2 U_1 |\psi\rangle\| \\ &\leq \sqrt{\omega^*(V)} + \|\Pi_1 U_3 Z (\mathbb{1} - \Pi_{\mathit{init}}) U_2 U_1 |\psi\rangle\| \\ &\qquad + \|\Pi_1 U_3 Z \Pi_{\mathit{init}} U_2 U_1 |\psi\rangle\| \\ &\leq \sqrt{\omega^*(V)} + \sqrt{1 - p_3} + \sqrt{\omega^*(V)},\end{aligned} \tag{6.49}$$

where the last term is bounded by using the fact that $\Pi_{\mathit{init}} U_2 U_1 |\psi\rangle$ is a valid initial state for the game specified by V, and can therefore not lead to a higher acceptance probability than $\omega^*(V)$.

Putting everything together, one has

$$\begin{aligned}\omega^*(V') &\leq \frac{1}{2}\Big(\omega^*(V) + \Big(2\sqrt{\omega^*(V)} + \sqrt{1-p_3}\Big)^2 + p_3\Big) \\ &\leq \frac{1}{2}\Big(\omega^*(V) + \Big(1 + 4\sqrt{\omega^*(V)} + 4\omega^*(V) - p_3\Big) + p_3\Big) \\ &= \frac{1}{2} + 2\sqrt{\omega^*(V)} + \frac{5}{2}\omega^*(V),\end{aligned} \tag{6.50}$$

as desired.

6.3.2 Parallelization and public-coin systems

Classical multi-prover interactive proof systems can be parallelized to a single round of interaction by using the oracularization technique. Starting from a verifier V in an m-turn interactive game, the two-turn verifier V' selects a random string r representing the private random bits of V and asks a first prover to provide a complete transcript, including all messages that would have been exchanged between the verifier and all provers, for the execution of the m-turn game using the random string r. The other provers are used to check that the transcript is one that could indeed have arisen in the original game, and in particular that messages from the provers in a certain turn, as described in the transcript, do not depend on messages sent by the verifier in subsequent turns. To check this condition, the provers are only given access to those random bits that determine messages from V sent in the first t turns, where $1 \leq t \leq m$ is randomly chosen; they are asked for a transcript of the game until that round. The verifier V' checks the transcripts received from the provers for consistency. The same transformation allows for a reduction of the number of provers to two.

For quantum interactive games the notion of a transcript is ill-defined, a difficulty that was already encountered in the single-prover settings discussed in Chapters 4 and 5. As was also discussed previously, the technique of oracularization will in general not apply even to classical verifiers in the presence of prior shared entanglement between the provers.

Fortunately, it turns out that the same transformation used to parallelize single-prover quantum interactive proof systems does extend to the multi-prover setting. Recall that this transformation requires the provers to start the interaction in the state they would be in halfway through the original interaction, proceeding either *forward* or *backward* depending on a coin-flip made by the verifier. The same idea can be applied to multiple provers, who will have no more latitude to cheat than in the single-prover setting. As for the transformation achieving perfect completeness, honest provers may be required to use entanglement in order to succeed in the modified proof system, irrespective of whether it is required by honest provers in the original proof system. This is because the joint state of their message registers halfway through the original game may contain entanglement generated by the verifier's messages. The consequence of this transformation for interactive proof systems is stated in the following theorem.

Theorem 6.7. For all polynomially bounded functions k and $m \geq 4$, and for every function $\varepsilon : \mathbb{N} \to [0,1]$, it holds that

$$\mathrm{QMIP}^*_{1,1-\varepsilon}(k,m) \subseteq \mathrm{QMIP}^*_{1,1-\delta}(k,3) \tag{6.51}$$

for $\delta = \varepsilon/m^2$.

Even if the verifier in the original m-turn game is classical, the same transformation will require a quantum verifier to execute the three-turn game, and it is not known if a similar transformation can be performed while keeping the verifier classical. If the soundness property of the original proof system is only known to hold against unentangled provers (that is, the proof system is a QMIP proof system), the reduction will not apply, as the provers must be able to use shared entanglement in order to succeed even in the honest case. To handle this case, the quantum verifier would first have to be simulated by a classical verifier through the circuitous route described in Section 6.2.1 (involving encoding the problem decided by the QMIP verifier as an MIP problem, and going through the constructions proving QMIP $\subseteq$ NEXP $\subseteq$ MIP). The resulting classical verifier can be parallelized to a single round using oracularization as described earlier. It is an open question whether QMIP systems can be directly parallelized to a single round of interaction with a classical verifier and without requiring the addition of a prover.

Quantum multi-prover interactive proof systems can be further parallelized to a single round (two turns) of interaction by introducing an additional prover. The transformation proceeds in two steps, each of which is of interest in its own right. The first step establishes that any three-turn verifier can be transformed into one whose single message to each prover consists of a uniformly random bit broadcast simultaneously to all provers. This public-coin form is unique to quantum multi-prover games, and it is unlikely to be achievable for classical games: because the verifier's message to all provers is publicly known, all provers receive the same information and can coordinate their actions perfectly. Thus MIP$^{\text{pub}}$, the public-coin variant of MIP, collapses to the single-prover class IP, and MIP = MIP$^{\text{pub}}$ would imply PSPACE = NEXP.

What makes this result possible in the case of quantum provers is that, even upon receiving the same message, the provers are still restricted to applying a local transformation on their respective registers. The verifier thus has the guarantee that the joint states of the message registers that could be sent by the provers in the third turn are related by the action of a quantum channel in tensor product form. This guarantee turns out to be sufficient to establish soundness of the public-coin proof system.

Given a three-turn verifier $V = (V_1, V_2)$, a three-turn public-coin verifier V' can be constructed as follows.

1. V' receives message register Z_1 from the first prover, and nothing

from the other provers.

2. V' chooses $c \in \{0,1\}$ uniformly at random and broadcasts it to all provers.
3. V' receives register Y_1^i from the i-th prover, for $i = 1, \dots, k$.
   (i) If $c = 0$, V' applies V_2 to the qubits in $(\mathsf{Z}_1, \mathsf{Y}_1^1, \dots, \mathsf{Y}_1^k)$, and outputs the outcome of the measurement performed by V_2.
   (ii) If $c = 1$, V' applies V_1^* to the qubits in $(\mathsf{Z}_1, \mathsf{Y}_1^1, \dots, \mathsf{Y}_1^k)$ and produces the output 1 if and only if all the qubits in Z_0 are in state $|0\rangle$.

The analysis of the completeness and soundness properties of V' follows along the same lines as the analysis of the min-max formulation of the value of a QIP(3) interactive game given in Section 4.4.1. The consequence for interactive proofs is stated in the following theorem.

Theorem 6.8. For every polynomially bounded function k and every function $b : \mathbb{N} \to [0,1]$, it holds that

$$\mathrm{QMIP}^*_{1,b}(k,3) \subseteq \mathrm{QMIP}^{*,\mathrm{pub}}_{1,(1+\sqrt{b})/2}(k,3), \tag{6.52}$$

where $\mathrm{QMIP}^{*,\mathrm{pub}}_{a,b}(k,m)$ is the class of promise problems having quantum k-prover m-turn interactive proof systems in which all the verifier's messages to the provers are public coins.

In the second step of the parallelization procedure it is shown how any three-turn public-coin verifier V interacting with k provers can be transformed into a two-turn verifier V' (no longer public-coin) interacting with $k+1$ provers. Because V is public-coin, we may assume that its first action V_1 consists of generating uniformly random bits and broadcasting them to the provers. At the end of the game the verifier applies a unitary V_2 to the joint state formed by the provers' message registers $\mathsf{Y}_1^1, \dots, \mathsf{Y}_1^2$ received in the first turn, $\mathsf{Y}_2^1, \dots, \mathsf{Y}_2^k$ received in the third turn, and its own private register. Define the new verifier V' as follows:

1. V' broadcasts public coins to the first k provers exactly as V would. No message is sent to the $(k+1)$-st prover.

2. V' applies the unitary V_2 to the messages received, treating the first k provers' answer registers as if they contained the provers' messages in the second turn of the original game, and the $(k+1)$-st prover's message as if it contained the joint state of all provers' messages in the first turn of the original game. V' then measures the output qubit and produces the outcome.

First we claim that for any strategy for the provers $P^1, \dots, P^k$ in the interactive game specified by V there exists a strategy for the provers $R^1, \dots, R^{k+1}$ with the same probability of being accepted by V'. To achieve this $R^1, \dots, R^k$ can simulate the actions of $P^1, \dots, P^k$ in the first turn of their interaction with V, handing over their joint message registers to R^{k+1} before the interaction with V' starts (which is allowed as part of the provers' prior shared entanglement). When the game specified by V' is initiated, R^{k+1} sends all its registers to V' and $R^1, \dots, R^k$ continue as if they were $P^1, \dots, P^k$ interacting with V. The new provers' probability of being accepted by V' is identical to the original provers' probability of being accepted by V.

Conversely, fix a strategy for provers $R^1, \dots, R^{k+1}$ in an interaction with V' and define a strategy for $P^1, \dots, P^k$ that has the same probability of being accepted by V as follows. $P^1, \dots, P^k$ initialize their private registers exactly as $R^1, \dots, R^k$ would, except that for each $i \in \{1, \dots, k\}$ prover P^i is also given the register $\mathsf{W}_1^{k+1,i}$ sent by R^{k+1} to V' that would have been interpreted as message register Y_1^i by V' in the game. In the first turn of their interaction with V each prover sends $\mathsf{W}_1^{k+1,i}$. In the second turn they behave exactly as $R^1, \dots, R^k$ would have in their interaction with V'. Once again, the probability of $P^1, \dots, P^k$ being accepted by V is identical to the probability of $R^1, \dots, R^{k+1}$ being accepted by V'.

Through the transformation just described, one concludes the following theorem.

Theorem 6.9. For every polynomially bounded function k and all functions $a, b : \mathbb{N} \to [0,1]$ such that $a > b$, it holds that

$$\mathrm{QMIP}^{*,\mathrm{pub}}_{a,b}(k,3) \subseteq \mathrm{QMIP}^{*,\mathrm{pub}}_{a,b}(k+1,2). \tag{6.53}$$

Combining Theorem 6.6, Theorem 6.7, Theorem 6.8 and Theorem 6.9 proves Theorem 6.4.

6.3.3 Classical verifiers

In Section 6.2.1 it is argued, albeit rather indirectly, that quantum verifiers interacting with multiple unentangled provers are no more powerful than their classical counterparts. In the presence of entangled provers it may seem that the possibility for the verifier to exchange quantum messages is essential, and indeed this is the case for some of the reductions discussed in the preceding section. Nevertheless, it is still the case that any quantum multi-prover interactive proof system with entangled provers can be transformed into one in which the verifier is classical, provided the number of provers is allowed to increase by two and the number of rounds to a polynomial. This fact was stated as Theorem 6.5 earlier in this section.

The reduction from quantum to classical verifiers that underlies the theorem just mentioned is highly non-trivial. Its completeness requires honest provers to share polynomially many qubits of entanglement, and its soundness rests on the property of *entanglement rigidity*. Informally speaking, this property states that certain correlations generated by the provers, as witnessed by a high success probability in certain interactive games (such as the CHSH game, Example 6.3), are *rigid* in the sense that they can only be obtained by performing measurements on a specific entangled state, up to local isometries that could be performed by the provers. (In the case of the CHSH game, this state is an EPR pair.) Rigidity can be leveraged by the verifier to exert a tight control over the provers' actions: by verifying that they are able to successfully play the CHSH game, it is possible to assert that, up to local isometries acting on their private registers, the provers share an EPR pair on which they apply specific measurements.

Using additional ideas, it is possible to devise a proof system whereby a classical verifier V' is able to "orchestrate" $k+2$ provers $R^1, \dots, R^{k+2}$, using only classical messages, so as to reproduce any polynomial-time interaction between a quantum verifier V and k provers. In this orchestration, one of the additional provers, say R^{k+1}, plays the role of V, and the other, R^{k+2}, is used to control the actions of R^{k+1} via a form of distributed process tomography. The original proof system may call for quantum messages to be exchanged between V and the provers. In the new proof system, such messages are simulated via teleportation between R^{k+1} and the first k provers, where V' uses classical messages to relegate the required correction bits between the provers. The EPR pairs used for teleportation are tested by executing a sufficiently large numbers of CHSH games in sequence and verifying that the provers achieve a success rate close to the optimal $\omega^*(\text{CHSH})$. This large number of CHSH games leads to a polynomial blow-up in the number of rounds of interaction of V', even if V is single-round.

6.4 Containment of NEXP in QMIP*

As was previously discussed, the introduction of entanglement between provers can sometimes give them a significant advantage in a multi-prover interactive game. As a result, it is not immediately clear that the complexity class QMIP* is larger than the single-prover class QIP, as the soundness property of the multi-prover interactive proof system constructions that establish NEXP $\subseteq$ MIP could be compromised by entanglement between the provers. Indeed, as was mentioned in Section 6.2.3, a collapse of this sort does occur for the restricted case of XOR proof systems: $\oplus$MIP* $\subseteq$ PSPACE, while $\oplus$MIP $=$ NEXP. The following theorem shows that this does not happen in the more general setting (assuming PSPACE $\neq$ NEXP), and more precisely that quantum interactive proof systems with entangled provers are powerful enough to decide all problems in nondeterministic exponential time.

Theorem 6.10. Every language in NEXP has a three-prover one-round interactive proof system in which completeness $a = 1$ can be achieved by unentangled provers, and soundness $b = 1/2$ holds against entangled provers. In particualr, it holds that

$$\text{NEXP} \subseteq \text{QMIP}^*. \tag{6.54}$$

Additional properties of verifiers that establish the containment (6.54) are also known. For instance, the verifier can be taken to be classical, to send messages to two out of three provers chosen at random, and to receive a number of bits from each prover that scales as $O(\log(1/b))$, where b is the desired soundness parameter. If one is willing to relax the condition of perfect completeness, the inclusion

$$\text{NEXP} \subseteq \oplus\text{MIP}^*_{1-\varepsilon,1/2+\delta}(3,2) \qquad (6.55)$$

(three-prover one-round XOR games) is also known to hold for any choice of constants $\delta, \varepsilon > 0$.

In this section we sketch some of the ingredients that go into the proof of Theorem 6.10. The starting point is the proof system introduced by Babai, Fortnow, and Lund [21] in their proof of NEXP $\subseteq$ MIP. This proof system has two main components, both of which need to be made "entanglement-resistant," meaning that their soundness guarantee can be extended to hold against entangled provers.

The first component is the technique of oracularization. This is used in combination with the technique of arithmetization (introduced for the proof of IP = PSPACE) to devise a basic two-prover proof system for a certain NEXP-complete problem. As discussed earlier, oracularization fails in general with entangled provers. In Section 6.4.1 two workarounds are described that establish a weaker form of oracularization with entangled provers, first by using three, and then two, provers.

The second component is an interactive game called the *multilinearity test*. This test is used as a means to enforce that the provers' answers are determined according to a multilinear function of the message received from the verifier, which is interpreted as a point $x \in \mathbb{F}^m$ for some large finite field $\mathbb{F}$. The statement and analysis of a multilinearity test with entangled provers requires care, and the main ideas are discussed in the simpler context of the linearity test in Section 6.4.2. In Section 6.4.3 the two components are combined into a brief sketch of the proof of Theorem 6.10.

6.4.1 Games with three provers and monogamy of entanglement

Consider the following (apparently trivial) modification of the oracularization technique. Given a verifier V specifying a two-prover one-round interactive game, define a three-prover one-round verifier V' as follows.

At the start of the game, V' selects a permutation of the three provers uniformly at random and assigns them labels *Alice*, *Bob*, and *Charlie*. V' then plays the two-prover game specified by V with the provers that were designated as Alice and Bob, ignoring the prover designated as Charlie. Each prover is assigned its name when it is sent its first message.

If the provers employ classical deterministic (or even randomized) strategies, the presence of Charlie makes no difference whatsoever; and so it holds that $\omega(V) = \omega(V')$. This equality no longer holds with entangled provers. The reason is related to a property of entanglement called *entanglement monogamy*. Informally speaking, monogamy states that there exist strong correlations that can be realized between two parties sharing entanglement that cannot be extended to three or more parties. For instance, three qubits cannot be in a state in which each pair of qubits forms an EPR pair. Thus, while a random string shared between two parties can just as easily be shared among three, a bipartite entangled state cannot in general be extended to a tripartite state reproducing the bipartite correlations among any subset of two out of three of the parties.

Example 6.4 (Three-prover CHSH game CHSH_3)**.** The transformation described above can be applied to the CHSH game, which was presented in Example 6.3. In the new, three-prover variant of this game, the verifier selects two provers at random to play the roles of Alice and Bob, and sends them questions as in the CHSH game. The third prover is ignored. It is evident that the classical value of this game coincides with that of the two-prover variant: $\omega(\mathrm{CHSH}_3) = \omega(\mathrm{CHSH}) = 3/4$. What is perhaps more surprising is that the entangled value is no larger: $\omega^*(\mathrm{CHSH}_3) = \omega(\mathrm{CHSH}_3) = 3/4$.[2] This fact is representative of the monogamy of quantum correlations.

Figure 6.4 describes a three-prover variant of the clause-versus-variable game introduced in Section 6.2.3, demonstrating further the monogamy phenomenon. The following analogue of (6.21) can be estab-

[2] Indeed, it holds that the so-called *no-signaling* value of this game is 3/4, which can be proved through the use of linear programming. As the no-signaling value upper-bounds the entangled value, it follows that $\omega^*(\mathrm{CHSH}_3) \leq 3/4$.

The input is a collection of constrains $\varphi = (C_1, \ldots, C_m)$ on variables $x_1, \ldots, x_n$, where each constraint C_j is an arbitrary constraint involving at most ℓ of the variables, each ranging over a finite alphabet Γ.

1. Select a random permutation of the three provers, and name the first *Alice*, the second *Bob*, and the third *Charlie*.
2. Run the 2-prover clause-versus-variable verifier V_φ with Alice and Bob, ignoring Charlie.

Figure 6.4: 3-prover clause-versus-variable verifier T_φ.

lished for the verifier T_φ described in Figure 6.4: there exists a constant $c > 1$ such that, for all φ,

$$\omega(\varphi) \leq \omega^*(T_\varphi) \leq 1 - \left(\frac{1-\omega(\varphi)}{n}\right)^c. \tag{6.56}$$

The remainder of this section is devoted to a proof of the implication $[\omega^*(T_\varphi) = 1] \Rightarrow [\varphi \text{ is satisfiable}]$. (The converse implication is immediate.) Taking the contrapositive, this statement already implies that if $\omega(\varphi) < 1$ it must also be that $\omega^*(\varphi) < 1$. The quantitative bound provided by the second inequality in (6.56) can be derived using the same proof outline, but requires substantially more technical work to keep track of the losses incurred in all inequalities.

Applying the bound (6.56) to an exponential-sized family of constraints determining membership in an NEXP-complete language yields the following complexity-theoretic consequence:

$$\text{NEXP} \subseteq \text{MIP}^*_{1,1-2^{-\text{poly}}}(3,2). \tag{6.57}$$

Although the inclusion is non-trivial, the exponentially small gap between the completeness and soundness parameters is too small to be amplified by any efficient method. This small gap is a consequence of the dependence on the number of variables n in the right-hand side of (6.56), which is exponential in the input size for an NEXP-complete language. For the case of the unentangled value $\omega(T_\varphi)$, as seen from (6.21), there is no such dependence. For the entangled value it is not known if some dependence on n is necessary.

The analysis of the entangled value of T_φ rests on a stand-alone *consistency test.* Let X and Γ be finite sets and let π be a distribution on X. The test only requires two provers, but its analysis extends to the case where it is played in the presence of additional (passive) provers.

Consistency test $\textsc{cons}(X, \Gamma, \pi)$

Given finite sets X and Γ, and a distribution π on X, perform the following steps:

1. Choose $x \in X$ according to π, and send x to two provers.
2. Receive answers $a, b \in \Gamma$ respectively. Accept if and only if $a = b$.

A strategy for the provers in $\textsc{cons}(X, \Gamma, \pi)$ can be described succinctly by specifying an initial shared entangled state $|\psi\rangle \in \mathcal{V} \otimes \mathcal{W}$ and measurements $\{P_a^x : a \in \Gamma\}$ and $\{Q_b^x : b \in \Gamma\}$, for every $x \in X$, corresponding respectively to the first and second provers' measurements upon receiving message x from the verifier. The properties of the test are summarized in the following lemma.

Lemma 6.11. Suppose a strategy for the provers, specified by measurements $\{P_a^x : a \in \Gamma\}$ and $\{Q_b^x : b \in \Gamma\}$ and a shared entangled state $|\psi\rangle \in \mathcal{V} \otimes \mathcal{W}$, succeeds with probability 1 in the game $\textsc{cons}(X, \Gamma, \pi)$. For every $x \in X$ such that $\pi(x) > 0$, and for all $a \in \Gamma$, it holds that

$$\begin{aligned} \Big((P_a^x)^2 \otimes \mathbb{1}\Big)|\psi\rangle &= (P_a^x \otimes \mathbb{1})|\psi\rangle \\ &= (\mathbb{1} \otimes Q_a^x)|\psi\rangle = \Big(\mathbb{1} \otimes (Q_a^x)^2\Big)|\psi\rangle. \end{aligned} \tag{6.58}$$

Proof. For each $x \in X$ satisfying $\pi(x) > 0$, define vectors

$$\begin{aligned} v_x &= \sum_{a \in \Gamma} |a\rangle \otimes (P_a^x \otimes \mathbb{1})|\psi\rangle, \\ w_x &= \sum_{a \in \Gamma} |a\rangle \otimes (\mathbb{1} \otimes Q_a^x)|\psi\rangle, \end{aligned} \tag{6.59}$$

and observe that

$$\langle w_x, v_x \rangle = \sum_{a \in \Gamma} \langle \psi | P_a^x \otimes Q_a^x | \psi \rangle, \tag{6.60}$$

which is a nonnegative real number in the interval $[0,1]$ for every choice of $x \in X$. One has that

$$\| v_x \|^2 = \sum_{a \in \Gamma} \langle \psi | (P_a^x)^2 \otimes \mathbb{1} | \psi \rangle \leq \sum_{a \in \Gamma} \langle \psi | P_a^x \otimes \mathbb{1} | \psi \rangle = 1, \tag{6.61}$$

where the inequality holds by virtue of the fact that $0 \leq P_a^x \leq \mathbb{1}$ for each $x \in X$ and $a \in \Gamma$, and the second equality follows from the assumption that $\{P_a^x\}$ is a measurement. Along similar lines, one finds that $\| w_x \| \leq 1$.

Now, under the assumption that the strategy succeeds with probability 1, it must hold that

$$1 = \sum_{x \in X} \pi(x) \sum_{a \in \Gamma} \langle \psi | P_a^x \otimes Q_a^x | \psi \rangle = \sum_{x \in X} \pi(x) \langle w_x, v_x \rangle, \tag{6.62}$$

and therefore $\langle w_x, v_x \rangle = 1$ for every $x \in X$ satisfying $\pi(x) > 0$. By the equality condition of the Cauchy–Schwarz inequality, one finds that $w_x = v_x$ for every $x \in X$ satisfying $\pi(x) > 0$, and moreover these vectors must all be unit vectors. Consequently

$$(P_a^x \otimes \mathbb{1}) | \psi \rangle = (\mathbb{1} \otimes Q_a^x) | \psi \rangle \tag{6.63}$$

for each $a \in \Gamma$, again for each $x \in X$ satisfying $\pi(x) > 0$.

Finally, observing the equivalence of the statements

(i) $\langle \psi | R \otimes \mathbb{1} | \psi \rangle = \langle \psi | S \otimes \mathbb{1} | \psi \rangle$ and

(ii) $(R \otimes \mathbb{1}) | \psi \rangle = (S \otimes \mathbb{1}) | \psi \rangle$

for all choices of positive semidefinite operators $0 \leq R \leq S$, along with the fact that the inequality in (6.61) must be an equality, one may conclude that

$$\Bigl((P_a^x)^2 \otimes \mathbb{1} \Bigr) | \psi \rangle = (P_a^x \otimes \mathbb{1}) | \psi \rangle \tag{6.64}$$

for every $a \in \Gamma$ and $x \in X$ satisfying $\pi(x) > 0$. The equality

$$(\mathbb{1} \otimes Q_a^x) | \psi \rangle = \Bigl(\mathbb{1} \otimes (Q_a^x)^2 \Bigr) | \psi \rangle \tag{6.65}$$

is proved through a similar methodology. □

One useful consequence of this lemma is that if the reduced density operator of $|\psi\rangle$ on the first prover's subspace has full support, then for any x such that $\pi(x) > 0$ the measurement $\{P_a^x\}$ is a projective measurement. A similar conclusion holds whenever the reduced density operator of $|\psi\rangle$ on the second prover's subspace has full support.

Suppose now that $\omega^*(T_\varphi) = 1$, so there is a strategy for the provers in the clause-versus-variable game that succeeds with probability 1. The strategy is specified by a set of *Alice-measurements* and *Bob-measurements*,

$$\{P^j_{b_1,\ldots,b_\ell} : b_1, \cdots b_\ell \in \Gamma\} \quad \text{and} \quad \{Q^i_a : a \in \Gamma\}, \tag{6.66}$$

respectively. The Alice-measurements result in a variable setting $b_1, \ldots, b_\ell$ for the ℓ variables appearing in each constraint C_j, while the Bob-measurements result in a variable setting a for each variable x_i. Because the game treats all three provers symmetrically, it is possible to argue that there is no loss of generality in taking $|\psi\rangle$ to be invariant under all permutations of its three registers, and one may also assume that it has full support on each provers' register. Similarly, one may assume that each prover always performs the same measurement when given the same name (i.e., Alice or Bob) by the verifier. Finally, one may assume that all of the measurements are projective measurements.

The goal is to show that φ is satisfiable. Define a distribution on assignments to the n variables as follows:

$$p(a_1, \ldots, a_n) = \left\| \left(\mathbb{1} \otimes Q^n_{a_n} \cdots Q^1_{a_1} \otimes \mathbb{1} \right) |\psi\rangle \right\|^2. \tag{6.67}$$

This is the distribution one would obtain by sequentially applying the Bob-measurements, in the order $i = 1, \ldots, n$, to the second prover's register of $|\psi\rangle$. This, of course, is not what the second prover does—the distribution p is only being defined in this way for the sake of the analysis. It must also be stressed that because the Bob-measurements do not necessarily commute, it is not immediate that this distribution is consistent with any prover's single Bob-measurement for a selected variable (except for the measurement associated with the variable x_1, which is the first measurement applied in (6.67)).

It will be shown that, if the strategy succeeds with probability 1, any assignment in the support of p must satisfy all of the constraints.

As there must be at least one assignment in the support of p, this will imply that φ is satisfiable. Toward this goal, for each $j \in \{1, \ldots, m\}$, define $q_j : \Gamma^\ell \rightarrow [0, 1]$ to be the marginal probability distribution on the possible assignments to the ℓ variables appearing in the constraint C_j that is obtained from the distribution p. We will prove that

$$q_j(b_1, \ldots, b_\ell) = \left\| (P^j_{b_1,\ldots,b_\ell} \otimes \mathbb{1} \otimes \mathbb{1}) |\psi\rangle \right\|^2. \tag{6.68}$$

Note that this will suffice to complete the proof. In greater detail, because the provers are assumed to cause the verifier to accept with certainty, Alice's answers always satisfy the clause she was asked about, and therefore (6.68) implies that every assignment in the support of p satisfies C_j. As this is so for all j, it must hold that every assignment in the support of p satisfies all of the constraints, as required.

It therefore remains to prove (6.68). For each $j \in \{1, \ldots, m\}$ and $i \in \{1, \ldots, \ell\}$, define measurements

$$\{R^{j,i}_a : a \in \Gamma\} \quad \text{and} \quad \{S^{j,i}_a : a \in \Gamma\} \tag{6.69}$$

as follows:

$$R^{j,i}_a = \sum_{\substack{b_1,\ldots,b_\ell \in \Gamma \\ b_i = a}} P^j_{b_1,\ldots,b_\ell} \quad \text{and} \quad S^{j,i}_a = Q^{k_i}_a, \tag{6.70}$$

where $k_1 < \cdots < k_\ell$ are the indices of the variables appearing in the constraint C_j. In words, the first measurement is equivalent to performing the measurement $\{P^j_{b_1,\ldots,b_\ell} : b_1, \ldots, b_\ell \in \Gamma\}$ and outputting just the assignment $a = b_i$ rather than the entire assignment $(b_1, \ldots, b_\ell)$, while the second measurement is equivalent to the Bob-measurement for the i-th variable appearing in C_j.

Observe that the provers' success in the consistency check performed by the clause-versus-variable verifier implies that the measurements $\{R^{j,i}_a\}$ and $\{S^{j,i}_a\}$ necessarily constitute a perfect strategy for the game $\textsc{cons}(X, \Gamma, \pi)$, where $X = \{1, \ldots, m\} \times \{1, \ldots, \ell\}$ and π is the distribution obtained by selecting a constraint and a variable appearing in that constraint, both uniformly at random. It therefore holds, for all choices of $j \in \{1, \ldots, m\}$, $i \in \{1, \ldots, \ell\}$, and $a \in \Gamma$, that

$$\bigl(R^{j,i}_a \otimes \mathbb{1} \otimes \mathbb{1}\bigr)|\psi\rangle = \bigl(\mathbb{1} \otimes S^{j,i}_a \otimes \mathbb{1}\bigr)|\psi\rangle, \tag{6.71}$$

which is remarkable because it implies that the probability for the first prover to assign the value a to a given variable is constant over all choices of the constraints in which that variable appears. (Recall that $S_a^{j,i}$ only depends on the label of the i-th variable appearing in C_j, but not on the clause C_j itself.) By using the permutation invariance of both $|\psi\rangle$ and the measurements performed by the three provers, the 5 similar equations to (6.71) obtained by permuting the 3 systems also hold. It therefore follows that

$$(Q_a^k \otimes \mathbb{1} \otimes \mathbb{1})|\psi\rangle = (\mathbb{1} \otimes Q_a^k \otimes \mathbb{1})|\psi\rangle = (\mathbb{1} \otimes \mathbb{1} \otimes Q_a^k)|\psi\rangle \tag{6.72}$$

for every $k \in \{1, \dots, n\}$ and $a \in \Gamma$. For instance, if k is the index of the i-th variable appearing in the constraint C_j, then one has

$$\begin{aligned}(Q_a^k \otimes \mathbb{1} \otimes \mathbb{1})|\psi\rangle &= (S_a^{j,i} \otimes \mathbb{1} \otimes \mathbb{1})|\psi\rangle = (\mathbb{1} \otimes \mathbb{1} \otimes R_a^{j,i})|\psi\rangle \\ &= (\mathbb{1} \otimes S_a^{j,i} \otimes \mathbb{1})|\psi\rangle = (\mathbb{1} \otimes Q_a^k \otimes \mathbb{1})|\psi\rangle.\end{aligned} \tag{6.73}$$

Note that (6.72) does not follow directly from the permutation invariance of $|\psi\rangle$ and the provers' measurements, and it makes essential use of all three systems.

Now, suppose that C_j is a constraint acting on the subset of variables indexed by $k_1 < \cdots < k_\ell$, and let $i_1 < \cdots < i_{n-\ell}$ be the indices of the remaining $n - \ell$ variables. Let us also take $(a_1, \dots, a_n)$ to be an assignment to the variables $(x_1, \dots, x_n)$ that is obtained from the assignment $(b_1, \dots, b_\ell)$ to the variables indexed by $(k_1, \dots, k_\ell)$ and an assignment $(c_1, \dots, c_{n-\ell})$ to the variables indexed by $(i_1, \dots, i_{n-\ell})$. By repeatedly applying (6.72) one finds that

$$\begin{aligned}&(Q_{a_n}^n \cdots Q_{a_1}^1 \otimes \mathbb{1} \otimes \mathbb{1})|\psi\rangle \\ &\quad = (\mathbb{1} \otimes Q_{b_\ell}^{k_\ell} \cdots Q_{b_1}^{k_1} \otimes Q_{c_{n-\ell}}^{i_{n-\ell}} \cdots Q_{c_1}^{i_1})|\psi\rangle,\end{aligned} \tag{6.74}$$

and therefore, by summing over all choices of $c_1, \dots, c_{n-\ell}$, one finds that

$$q_j(b_1, \dots, b_\ell) = \left\| (\mathbb{1} \otimes Q_{b_\ell}^{k_\ell} \cdots Q_{b_1}^{k_1} \otimes \mathbb{1})|\psi\rangle \right\|^2. \tag{6.75}$$

Finally, by applying (6.71) repeatedly, it follows that

$$\begin{aligned}(\mathbb{1} \otimes Q_{b_\ell}^{k_\ell} \cdots Q_{b_1}^{k_1} \otimes \mathbb{1})|\psi\rangle &= (R_{b_1}^{j,1} \otimes Q_{b_\ell}^{k_\ell} \cdots Q_{b_2}^{k_2} \otimes \mathbb{1})|\psi\rangle \\ = \cdots = (R_{b_1}^{j,1} \cdots R_{b_\ell}^{j,\ell} \otimes \mathbb{1} \otimes \mathbb{1})|\psi\rangle &= (P_{b_1,\dots,b_\ell}^j \otimes \mathbb{1} \otimes \mathbb{1})|\psi\rangle,\end{aligned} \tag{6.76}$$

where the last equality has made use of the fact that $\{P^j_{b_1,\dots,b_\ell}\}$ is a projective measurement. The equation (6.68) has therefore been proved, as required.

This concludes the analysis of the three-prover clause-versus-variable verifier T_φ. One of the provers, playing the role of Charlie, plays a completely passive role. It is not asked any questions, but its presence is necessary for the proof to go through, as Example 6.2 of the Magic Square game demonstrates. At a more technical level, all three registers of $|\psi\rangle$ played an essential role in the analysis above.

It is possible to define a different variant of the clause-versus-variable verifier that involves an interaction with only two provers and remains sound against entangled provers. In the three-prover game the intuition behind the role of the third prover is that it "confuses" the other provers into not knowing with which prover they should coordinate their answers. The idea for the two-prover variant is to induce the same type of confusion by sending an additional question, chosen uniformly at random, to the second prover in the clause-versus-variable game. The prover thus receives two variables: one taken from the first prover's constraint, and the other chosen uniformly at random. The fact that the prover is not told which variable it will be tested on makes its task harder.

2-prover confuse-SAT verifier F_φ

Given $\varphi = (C_1, \dots, C_m)$, where each C_i is a constraint acting on ℓ out of n variables $x_1, \dots, x_n$, the verifier proceeds as follows:

1. Select an index $j \in [m]$ uniformly at random. Let $\{x_{i_1}, \dots, x_{i_\ell}\}$ be the variables on which constraint C_j acts. Select $t \in \{1, \dots, \ell\}$ and $i_{t'} \in \{1, \dots, n\}$ uniformly at random. Send C_j to the first prover, and the unordered pair $\{i_t, i_{t'}\}$ to the second prover.
2. The first prover replies with an assignment $(a_1, \dots, a_\ell) \in \Gamma^k$. The second prover replies with an assignment $(b, b') \in \Gamma^2$.
3. Accept if and only if the first prover's answers satisfy clause C_j and the provers' answers are consistent on the variable they were both asked: $a_t = b$.

The same result as shown for the three-prover clause-versus-variable verifier T_φ holds: $\omega^*(F_\varphi) = 1$ if and only if φ is satisfiable, and a similar quantitative bound as the one in (6.56) holds as well. The proof follows the same outline. The main additional ingredient is the analysis of the following "confuse" test, that complements the consistency test $\textsc{cons}(X, \Gamma, \pi)$ introduced earlier.

Confuse test $\textsc{conf}(X, \Gamma, \nu)$

Given finite sets X, Γ and a distribution ν on $X \times X$, the verifier proceeds as follows:

1. Select $(x, y) \in X \times X$ according to ν, and send the unordered pair $\{x, y\}$ to the first prover and either x (with probability 1/2) or y (with probability 1/2) to the second.
2. Receive answers $(a, a') \in \Gamma^2$ and $b \in \Gamma$ respectively. Accept if and only if the provers' answers are consistent.

This game has the following soundness property: any strategy given by $\{P^{x,y}_{a,a'}\}$, $\{Q^x_b\}$, and $|\psi\rangle$ that has success probability 1 must be such that, for any $x, y \in X$ such that $\nu(x, y) > 0$ and all $a, b \in \Gamma$,

$$(P^{x,y}_{a,b} \otimes \mathbb{1})|\psi\rangle = (\mathbb{1} \otimes Q^x_a Q^y_b)|\psi\rangle = (\mathbb{1} \otimes Q^y_b Q^x_a)|\psi\rangle. \tag{6.77}$$

Furthermore, if $|\psi\rangle$ has full support on the second prover's space, then the measurement operators Q^x_a and Q^y_b must commute provided that $\nu(x, y) > 0$. These properties together with those that follow from the test $\textsc{cons}(X, \Gamma, \pi)$ suffice to prove soundness of the confuse-SAT verifier.

6.4.2 Linearity testing with entangled provers

Originally introduced in the context of efficient program checking, the *linearity test* of Blum, Luby and Rubinfeld [33] quickly found one of its most important applications in the study of classical multi-prover interactive proof systems. Based on this test alone, it is already possible to prove a weak form of the PCP theorem, establishing that all languages in NP have proofs of exponential length that can be verified with a constant gap between completeness and soundness by querying only a constant number of bits from the proof. This section is devoted to the formulation and analysis of the test when executed with entangled provers.

The linearity test can be formulated as a one-round interactive game played between a classical verifier and three provers. The verifier's messages to the provers are elements of $\mathbb{F}_2^n$, for some integer n. The provers' messages are elements of $\mathbb{F}_2$. The test is designed to certify that the provers' answers are consistent with a linear function $f : \mathbb{F}_2^n \to \mathbb{F}_2$, i.e., one that can be written as $f(x) = u \cdot x$ for some $u \in \mathbb{F}_2^n$.

Linearity test

The verifier performs either of the following with probability 1/2 each:

1. *(Consistency)* Select $x \in \mathbb{F}_2^n$ uniformly at random and send x to each prover. Accept if and only if all provers provide the same answer $a \in \mathbb{F}_2$.
2. *(Linearity)* Select $x, y \in \mathbb{F}_2^n$ uniformly at random, and set $z = x+y$. Send x to the first prover, y to the second, and z to the third. Accept if and only the provers' answers $a, b, c \in \mathbb{F}_2$ satisfy $a + b = c$.

The linearity test has perfect completeness: if the provers answer according to the same linear function, they are accepted by the verifier with probability 1. Its soundness property against classical deterministic provers can be stated as follows.

Theorem 6.12 (BLR linearity test)**.** Suppose that three classical deterministic provers succeed in the linearity test with probability $1 - \varepsilon$, and let $f_1, f_2, f_3 : \mathbb{F}_2^n \to \mathbb{F}_2$ be the functions describing their respective strategies. There exists a vector $u \in \mathbb{F}_2^n$ such that, for each $i \in \{1, 2, 3\}$, $f_i(x) = u \cdot x$ for all but a fraction at most 8ε of $x \in \mathbb{F}_2^n$.

Theorem 6.12 makes the assumption that the three provers are deterministic. In general the provers may use private or shared randomness. The result is easily extended by "fixing the randomness";

each private or shared random string that may be used by the provers corresponds to a deterministic strategy to which the theorem can be applied.

In the case of entangled provers it is not possible to "fix the quantumness" present in the provers' strategy, which in general is specified by families of binary-valued measurements $\{P_a^x\}$, $\{Q_b^y\}$, $\{R_c^z\}$ on the three parts of a tripartite state $|\psi\rangle \in \mathcal{X} \otimes \mathcal{Y} \otimes \mathcal{Z}$. What does it mean for such a strategy to be linear? Without "fixing the randomness," it is impossible to claim that the strategy is close to a single linear function f. A more reasonable statement would be that any strategy for the provers with a high probability of success in the test is *almost indistinguishable* from a "linear" strategy of the following form:

1. Each prover measures its share of the entangled state using the same measurement $\{S_u : u \in \mathbb{F}_2^n\}$, respectively obtaining outcomes $u_1, u_2, u_3 \in \mathbb{F}_2^n$ such that $u_1 = u_2 = u_3$ with high probability.
2. Upon receiving the verifier's message x, y, or z, the first, second, or third prover answers with $u_1 \cdot x$, $u_2 \cdot x$, or $u_3 \cdot x$, respectively.

The strength of this statement resides in the existence of the measurement $\{S_u\}$, its independence from the prover's question, and the claim that it faithfully reproduces the original strategy.

Whether the statement is meaningful or not rests on the precise quantification of the claim that the original and oblivious strategies are "almost indistinguishable." In the case of classical provers (i.e., Theorem 6.12), this is the statement that the oblivious strategy differs from the original one in a fraction at most 8ε of questions x. The case of entangled provers is more subtle, as it requires the introduction of a measure of distance between strategies that is

(i) strong enough that "nearby" strategies have a similar success probability, not only in the test itself but also in any proof system that would invoke the test as a sub-game; and

(ii) weak enough that it is possible to place bounds on the distance solely from the assumption that the provers have a high success in the linearity test.[3]

[3] This implies for instance that the operator norm between the provers' mea-

Such a distance measure can be defined as follows. Consider two strategies for the first prover, specified by measurements $\{P_a^x\}$ and $\{\tilde{P}_a^x\}$. Fixing a message x from the verifier, the distance between the post-measurement states resulting from these two measurements can be expressed as

$$\begin{aligned}\sum_a \left\| \left(\sqrt{P_a^x} \otimes \mathbb{1}\right)|\psi\rangle - \left(\sqrt{\tilde{P}_a^x} \otimes \mathbb{1}\right)|\psi\rangle \right\|^2 \\ = \sum_a \mathrm{Tr}\left(\left(\sqrt{P_a^x} - \sqrt{\tilde{P}_a^x}\right)^2 \rho\right),\end{aligned} \tag{6.78}$$

where ρ is the reduced density operator of $|\psi\rangle$ on the first prover's register. Provided the quantity in (6.78) is small, the provers' shared state is almost the same after the first prover has measured its subsystem using either of the two measurements and obtained an answer a. Thus, the joint distributions on outcomes obtained when the first prover measures using either measurement, and the other provers perform any measurement whatsoever, are close in statistical distance.

Theorem 6.13 (Entangled-prover linearity test)**.** Suppose three entangled provers succeed in the linearity test with probability at least $1-\varepsilon$ using a strategy specified by measurements $\{P_a^x\}$, $\{Q_b^y\}$, $\{R_c^z\}$ on an entangled state $|\psi\rangle$. There exists a measurement $\{S_u : u \in \mathbb{F}_2^n\}$, which is independent of x, such that, for

$$\tilde{P}_a^x = \sum_{\substack{u\in\mathbb{F}_2^n \\ u\cdot x = a}} S_u, \tag{6.79}$$

it holds that

$$\frac{1}{2^n} \sum_{x\in\mathbb{F}_2^n} \sum_{a\in\mathbb{F}_2} \mathrm{Tr}\left(\left(\sqrt{P_a^x} - \sqrt{\tilde{P}_a^x}\right)^2 \rho\right) = O(\sqrt{\varepsilon}), \tag{6.80}$$

where ρ is the reduced density operator of $|\psi\rangle$ on the first prover's register. A similar statement holds for $\{Q_b^y\}$ and $\{R_c^z\}$.

surements (and even more so the diamond norm between the associated quantum channels) would not be appropriate, as success in the test does not put constraints directly on the provers' measurements themselves, but only on their probability of obtaining certain outcomes when applied on the specific entangled state $|\psi\rangle$.

The measurement $\{S_u\}$ whose existence is claimed in the previous theorem has a simple definition. Given $\{P_a^x\}$, let its matrix-valued Fourier coefficient be defined as

$$\widehat{S}_u = 2^{-n} \sum_x (-1)^{x \cdot u} (P_0^x - P_1^x), \tag{6.81}$$

and for every u let

$$S_u = (\widehat{S}_u)^2. \tag{6.82}$$

It follows from Parseval's formula that $\{S_u\}$ is a well-defined measurement. For intuition on this definition it is useful to consider the special case of classical provers who may use shared randomness, for which $\{S_u\}$ corresponds to the following definition of an oblivious strategy. Let r be a value for the shared randomness. In the original strategy, r points to a function f_r according to which the prover would determine its answer to the verifier's question. In the new strategy, the oblivious prover uses r to sample a linear function $g : x \mapsto u \cdot x$, where u is chosen according to the distribution suggested by the Fourier spectrum of f_r.[4] Upon receiving question x the prover answers with $g(x) = u \cdot x$.

The fact that this is a good strategy follows from the classical proof of Theorem 6.12, which establishes that the Fourier coefficients of $g = (-1)^f$ are sharply concentrated. In the case of a randomized strategy, most functions f_r will have a large Fourier coefficient, and the strategy described above will with high probability provide answers that are consistent and chosen according to a linear function. (Which linear function this is depends on the random string r and may change with each interaction with the verifier.)

With the definition of S_u in hand, the proof of Theorem 6.13 is not too difficult—it uses similar arguments to those employed in the analysis of the three-prover clause-versus-variable game given in the previous section, and we omit the details.

6.4.3 NEXP ⊆ QMIP*

The proof of Theorem 6.10 follows the same broad outline as Babai, Fortnow, and Lund's proof that NEXP ⊆ MIP, with significant modi-

[4] Letting $g_r = (-1)^{f_r}$, this is the distribution induced by the $|\widehat{g_r}(u)|^2$. Parseval's identity shows that this is indeed a distribution.

fications to the analysis needed to ensure that soundness is preserved against entangled strategies. (Completeness is straightforward, and does not require honest provers to use any entanglement.) In the following discussion we give a high-level overview of the soundness analysis that emphasizes the most important modifications required and ties them to the results from the preceding sections; the reader is advised not to interpret the discussion too literally and is referred to [21] for a technical exposition.

The starting point is an encoding of an NEXP-complete problem, such as SUCCINCT-3-COLORABILITY, as an instance of the following problem: given a multilinear function f in $\ell + 3$ variables, do there exist functions $A_1, A_2, A_3 : \{0,1\}^\ell \to \{0,1\}$ such that $f(z, A_1(z), A_2(z), A_3(z)) = 0$ for every $z \in \{0,1\}^\ell$? Here the function f encodes the structure of the problem (such as the graph to be colored), the variable z is used to index all the required constraints (such as edges of the graph), and A_1, A_2 and A_3 provides an assignment (the coloring). The input size n is the size of an arithmetic circuit specifying f, and in particular the total number of variables z can be exponential in n.

Using the technique of arithmetization [124] it is possible to devise a three-prover proof system for this problem with these attributes:

1. The verifier selects random elements $z, b_1, b_2 \in \mathbb{F}^\ell$, where $\mathbb{F}$ is a field of size exponential in n known to all parties.
2. Based on z, the verifier has a multiple-round interaction with the first prover.
3. The verifier asks two other provers for values $\tilde{A}_i(b_1)$ and $\tilde{A}_i(b_2)$, for all $i \in \{1,2,3\}$, respectively, where $\tilde{A}_i$ is the unique multilinear extension of A_i to $\mathbb{F}^\ell$.[5]
4. The verifier decides to accept or reject based on the results of its interaction with all three provers.

The most important step in the analysis of this proof system consists of devising a test that can be used to guarantee that the functions $\mathbb{F}^\ell \to \mathbb{F}$

[5] A multilinear function is one that is linear in each of its variables; $\tilde{A}_i$ extends A_i in the sense that $\tilde{A}_i(z) = A_i(z)$ for every $z \in \{0,1\}^\ell \subseteq \mathbb{F}^\ell$.

used by the second and third prover in step 3 agree with a multilinear function on a large fraction of inputs. With the promise that this is the case, the remainder of the proof system can be proven sound in a very similar way whether the provers use entanglement or not, as it mostly relies on analyzing the single-prover interaction performed in step 2.

Thus, the analysis will be completed once it is shown how to test that a prover answers queries $z \in \mathbb{F}^\ell$ with a function $\tilde{A} : \mathbb{F}^\ell \to \mathbb{F}$ that is linear in each variable. The test for this is a natural extension of the linearity test described in the previous section, and can be formulated as a one-round three-prover interactive game as follows. The verifier selects a coordinate $i \in \{1, \ldots, \ell\}$, an $x \in \mathbb{F}^\ell$ uniformly at random, and $y_i, z_i \in \mathbb{F} \backslash \{x_i\}$. Let y and z be equal to x with the i-th coordinate replaced by y_i and z_i, respectively. The verifier asks the three provers for the value of $\tilde{A}$ at x, y and z respectively. Upon receiving answers a, b, c from each of them the verifier checks the identities

$$\frac{c-a}{z-x} = \frac{b-a}{y-x} = \frac{c-b}{z-y}. \tag{6.83}$$

The analysis proceeds by induction on ℓ. The case $\ell = 1$ is provided by the linearity test described in the previous section. Unfortunately, as one proceeds through the induction, the error (as measured by the fraction of points on which $\tilde{A}$ differs from a multilinear function) increases rapidly: even in the most optimistic case, it will be multiplied by a factor 2 in each step, eventually yielding an unmanageable exponential blow-up.

To handle this difficulty, Babai, Fortnow, and Lund introduce a "self-improvement lemma," which establishes the following: any function $B : \mathbb{F}^\ell \to \mathbb{F}$ that is such that

(i) B is very close to linear along "lines" (meaning that the functions $B_i : x_i \mapsto B(x_1, \ldots, x_n)$, for $i = 1, \ldots, \ell$, are close to linear for most choices of $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n$), and

(ii) B is globally "somewhat close" to a multilinear function,

must in fact be very close to multilinear. That is, if the error in (i) is some small $\varepsilon > 0$, and the error in (ii) is a possibly much larger $\eta < 1/4$, then in fact η is automatically much smaller than expected, of order

$\ell\varepsilon$. The proof of this claim is based on expansion properties of the hypercube, from which the factor of ℓ loss in the error originates.

The case of entangled strategies is substantially more difficult. In particular, the conclusion of the lemma can only be reached by a suitable modification of the provers' strategy: an additional step of active correction is required whereby the distance to a multilinear function (as measured by a statement of a form similar to the bound given in Theorem 6.13) is reduced by leveraging the entangled-prover analogue of assumption (i) above. The "improved" measurement can be defined as the optimum of a semidefinite program and shown to satisfy similar (though quantitatively weaker) error bounds as those promised by the self-improvement lemma in the classical case, thus enabling the induction to carry through.

Once the soundness of the multilinearity test against entangled provers has been established, the remainder of the analysis of Babai, Fortnow, and Lund's proof system for NEXP $\subseteq$ MIP goes through with minor modifications, leading to a proof of the inclusion NEXP $\subseteq$ MIP*. The main difference between the two end results is that, while the first is known to hold with two provers, the proof of the second containment seems to require three provers: although it is possible to formulate the multilinearity test as a two-prover game, only the three-prover variant is known to be sound against entangled provers. It is an open question to determine whether NEXP $\subseteq$ QMIP$^*(2, \mathrm{poly})$.

6.5 Further topics

This section is devoted to two topics on which relatively little is known—both are currently active areas of research. The first topic is error reduction through parallel repetition, and the second is the problem of placing computational upper bounds on the classes MIP* and QMIP*.

6.5.1 Parallel repetition

Given a multi-prover interactive proof system with completeness and soundness parameters $a > b$, Proposition 6.1 states that the difference $a - b$ can be amplified efficiently, either by repeating the protocol sequentially (which increases the number of turns), or with different sets of provers. Both procedures work for all types of proof systems considered in this chapter, with or without entanglement between the provers.

It is natural to ask if the same effect can be achieved without any increase in the number of provers or turns. The most standard approach to this question is to consider repeating the protocol in parallel with the same set of provers. This type of repetition was already considered in the single-prover setting (q.v. Section 4.3), and it is defined analogously with multiple provers. Given a one-round two-prover[6] verifier $V = (V_1, V_2)$ and a number of repetitions ℓ, another one-round two-prover verifier $V^{(\ell)}$ is defined as follows: $V^{(\ell)}$ executes ℓ independent copies of the first transformation V_1 of V, generating ℓ independent pairs of registers $(\mathsf{X}_1^{1,i}, \mathsf{X}_1^{2,i})_{i=1,\dots,\ell}$. $V^{(\ell)}$ then sends $(\mathsf{X}_1^{1,1}, \cdots, \mathsf{X}_1^{1,\ell})$ to the first prover and $(\mathsf{X}_1^{2,1}, \cdots, \mathsf{X}_1^{2,\ell})$ to the second. The provers are expected to return ℓ message registers each. $V^{(\ell)}$ measures each pair of answer registers independently according to V_2, and accepts if and only if all measurements produce the outcome 1.

Even though the repeated verifier $V^{(\ell)}$ has a simple product form, due to the fact that the provers receive all of their questions simultaneously they may in general apply an arbitrary quantum channel, introducing correlations between their answers that make this type of repetition harder to analyze. The following example demonstrates that, in contrast to the single-prover case, parallel repetition cannot be expected to perfectly amplify either the unentangled or entangled values of a multi-prover game.

Example 6.5 (Fortnow–Feige–Lovász game). Consider the following one-round two-prover verifier V. The verifier selects a pair of messages (s, t) uniformly from $\{(0, 0), (0, 1), (1, 0)\}$ and sends s to the first prover and t to the second. Each prover replies with a bit $a, b \in \{0, 1\}$ respectively. The verifier outputs 1 if and only if $s \wedge a \neq t \wedge b$.

It may be verified that $\omega(V) = \omega(V^{(2)}) = 2/3$, as well as $\omega^*(V) =$

[6] The transformation can be described more generally, but almost all known results apply to the one-round two-prover setting only.

2/3. (Indeed, even the no-signaling value of V is equal to 2/3.) Combined with the previous equalities, this implies $\omega^*(V^{(2)}) = 2/3$ as well, so that neither the classical or entangled value of the game described by V multiply under parallel repetition.

The only scenario for which there is a satisfactory understanding of the effect of parallel repetition is that of classical two-prover one-round games with unentangled provers, to which the following result applies.

Theorem 6.14. There exists a constant C such that the following holds. Let V be a one-round two-prover classical verifier, r an upper bound on the number of bits of each prover's message register, and ℓ an integer. Then the ℓ-th parallel repetition $V^{(\ell)}$ of V satisfies

$$\omega(V)^{(\ell)} \leq \omega(V^{(\ell)}) \leq \Big(1 - \big(1 - \omega(V)\big)^3\Big)^{C\ell/r}. \tag{6.84}$$

The theorem, proved in [140, 91], only applies to games with two provers. The situation for more than two provers is poorly understood and only very weak results are known.

For the case of entangled provers only partial results, that apply to specific classes of games with a classical verifier, are known; we briefly describe some of these results below. There is no known parallel repetition theorem that applies to fully quantum verifiers.[7]

Two classes of games for which good results are known are *free games* and *projection games*. A free game is one in which the verifier's messages to the provers are chosen according to a product distribution. An example of a free game is the CHSH game (Example 6.3). Projection games are characterized by the form of the verifier's acceptance criterion: for any pair of messages from the verifier, and any possible message from the first prover, there should always be at most one message from the second prover that will result in acceptance. An example of a projection game is the Magic Square game (Example 6.2). The CHSH game is also a projection game, but the Magic Square game is not a free game, and the game described in Example 6.5 is neither. For both classes of games it is known that the entangled value decreases

[7] See the chapter notes for results that apply to restricted quantum verifiers that either send quantum messages but receive classical answers, or vice versa.

exponentially fast with the number of repetitions, in a manner analogous to (6.84) except the exponent 3 is replaced by 3/2 for free games and some other universal constant for projection games; for the latter there is no dependence on the answer length r.

Free games and projection games encompass many interesting games, but not all. For a general game it is possible to consider a modified form of repetition. The simplest way to describe this transformation is as a transformation of the game itself.

Given a one-round two-prover verifier V, consider a verifier $V_\perp$ that performs the following:

1. Generate a pair of messages (x^1, x^2) as V would.
2. Independently and with probability 1/4 each, replace the message x^1 with a "dummy" message $\perp$, and the message x^2 with $\perp$.
3. Send the new messages to the provers.
4. If either message was replaced by a $\perp$, accept any answer from the provers. If neither message was modified, accept the provers' answers if and only if the original verifier V would have accepted them.

It is clear that this transformation can only increase the value of the game; in fact it is not hard to verify that

$$\omega(V_\perp) = \frac{1}{4} + \frac{3}{4}\,\omega(V) \quad \text{and} \quad \omega^*(V_\perp) = \frac{1}{4} + \frac{3}{4}\,\omega^*(V), \tag{6.85}$$

a simple affine scaling. In spite of its almost naïve simplicity, it turns out that this transformation allows to prove strong parallel repetition results: it is known that if $\omega^*(V) = 1 - \varepsilon$ then $\omega^*(V_\perp^{(\ell)}) \leq (1 - \varepsilon^c)^{\Omega(\ell)}$, for some constant $c > 1$. (The same holds for the classical value.) Intuitively, the role of the "dummy" question $\perp$ is to limit the provers' ability to exploit correlations present in their ℓ pairs of questions in order to succeed with substantially higher probability than a strategy which treats all repetitions independently; at a high level the transformation makes the game closer to a free game.

6.5.2 Upper bounds on QMIP*

As discussed in Section 6.2.1, the inclusion QMIP $\subseteq$ NEXP follows from the observation that optimal *unentangled* strategies for the provers can always be implemented using private registers that are of dimension at most exponential in the number of qubits exchanged between each prover and the verifier. Example 6.1 demonstrates that this fact no longer holds for entangled strategies, which may in general benefit from arbitrarily large amounts of prior shared entanglement.

The example suggests the introduction of a hierarchy of values

$$\omega(V) = \omega^{(1)}(V) \leq \cdots \leq \omega^{(d)}(V) \leq \cdots \leq \omega^*(V), \tag{6.86}$$

where $\omega^{(d)}(V)$ is the maximum success probability of provers whose initial shared entangled state has local dimension at most d. Example 6.1 provides a V for which the chain of inequalities (6.86) does not eventually collapse into a series of equalities for large enough d. For small values of d explicit examples show that the first few inequalities can be strict; for instance the CHSH game (Example 6.3) is such that $\omega(\text{CHSH}) = \omega^{(1)}(\text{CHSH}) < \omega^{(2)}(\text{CHSH}) = \omega^*(\text{CHSH})$. We refer to the chapter notes for pointers to further examples.

Such examples raise the question of whether the entangled value is even computable. It is possible to devise a procedure for approaching ω^* from below, with the implication that QMIP* $\subseteq$ RE, the class of problems that are recursively enumerable. Because ω^* is given by the supremum over all d of $\omega^{(d)}$, for every verifier V and $\varepsilon > 0$ there is a dimension d such that $\omega^{(d)}(V) \geq \omega^*(V) - \varepsilon$. Applying a union bound over the suitably discretized space of possible verifiers of a given size we may deduce the existence of an integer $d = d(\varepsilon, s)$ depending only on ε and $s \in \mathbb{N}$ such that, for all verifiers of size at most s, there is an entangled strategy using private registers of dimension at most d whose success probability is at least $\omega^*(V) - \varepsilon$. No estimates are known on the dependence of $d(\varepsilon, s)$ on either parameter.

To prove that every problem in QMIP* is decidable, it would suffice to devise a counterpart to the above procedure that approaches $\omega^*(V)$ from above. This problem, however, is not settled, even for the simplest case of classical one-round two-prover games. For the remainder of this section we focus on that setting and introduce a procedure that may, under plausible but difficult mathematical conjectures, provide the required sequence of approximations.

The procedure is based on a hierarchy of semidefinite programs whose optimum is always at least the entangled value. For the case of XOR games the first level of the hierarchy coincides with the semidefinite program introduced in Section 6.2.3. Higher levels introduce variables not only for the inner products between vectors $(P_a^x \otimes \mathbb{1})|\psi\rangle$ and $(\mathbb{1} \otimes Q_b^y)|\psi\rangle$ associated with each provers' possible measurement operators, but also for composite terms involving vectors such as $(P_a^x \otimes Q_b^y)|\psi\rangle$ (at the second level), $(P_a^x P_{a'}^{x'} \otimes Q_b^y)|\psi\rangle$ (at the third level), and so on. All natural constraints that should hold for projective strategies, such as

$$\langle\psi|\big(P_a^x \otimes \mathbb{1}\big)\big(P_a^x P_{a'}^{x'} \otimes Q_b^y\big)|\psi\rangle = \langle\psi|\big(\mathbb{1} \otimes Q_b^y\big)\big(P_a^x P_{a'}^{x'} \otimes \mathbb{1}\big)|\psi\rangle, \quad (6.87)$$

are enforced as constraints in the semidefinite program. This hierarchy, introduced in [129], can be shown to converge in the limit of infinitely many levels to a value called the *field-theoretic* value ω^{FV} of the game. The field-theoretic value has an alternative definition as the supremum over all *commuting strategies* for the provers of the probability that the verifier outputs 1. Commuting strategies are a relaxation of entangled strategies in which the provers share a single quantum system, not necessarily finite-dimensional and initialized in to an arbitrary pure state, on which they perform their measurements in sequence. The only restriction, which is required for the model to be well-defined, is that the measurement operators associated with distinct provers must commute pairwise.

If it is assumed that the provers' joint register is finite-dimensional, commuting-prover strategies are equivalent to entangled strategies. This follows from a well-known (but nontrivial) result from the theory of C^*-algebras showing that for finite-dimensional algebras, pairwise commutation implies the existence of a decomposition of the algebra as a direct sum of tensor products. If the associated Hilbert space is allowed to be infinite dimensional, as seems necessary for ω^{FV} to coincide with the limiting value of the hierarchy introduced above, then the equality $\omega^*(V) \overset{?}{=} \omega^{\mathrm{FV}}(V)$ forms the content of *Tsirelson's problem.*

Tsirelson's problem is equivalent to a range of long-standing conjectures in the theory of C^*-algebras, including Connes' embedding conjecture and Kirchberg's QWEP conjecture. An affirmative resolution of any of these equivalent conjectures would imply that all problems in $\mathrm{MIP}^*(2,2)$ are decidable. The converse is not necessarily true; the decidability of all problems in MIP^* could hold even if the conjectures fail. In particular, deciding problems in MIP^* only requires one to compute a constant-factor approximation, rather than an arbitrarily close one, of the entangled value. No such procedure is known, making the question of (un)decidability of languages in MIP^* and QMIP^* one of the most intriguing open problems in the area of quantum multi-prover interactive proofs.

6.6 Chapter notes

The class MIP was first considered in [29], where its introduction was motivated by the study of zero-knowledge proof systems. Many results about MIP were discovered soon after, in the late 1980s and early 1990s (see [64, 21, 59], for instance). Modern accounts tend to emphasize a point of view on the class that results from its connection to probabilisticaly checkable proofs and the PCP theorem [17, 18], with its many applications to the hardness of approximation problems [57]. We refer the interested reader to [133] for a brief history of the developments that led to the PCP theorem.

The origins of quantum multi-prover interactive games can be traced back to the study of Bell inequalities in the foundations of quantum mechanics. This study was pioneered by Bell [26], with the simplest non-trivial inequality, the so-called CHSH inequality, being explicitly written as such by Clauser, Horne, Shimony, and Holt [44]. This inequality is equivalent to the bound $\omega(\mathrm{CHSH}) \leq 3/4$ discussed in Example 6.3, but formulated in a different language. The perspective of Bell inequalities as interactive games came through the work of Mermin [127], Peres [139], Cabello [41], Cleve, and others, and the connection to the soundness of interactive proof systems was observed in [46]. The Magic Square game (Example 6.2) is attributed to Mermin [127] and Peres [139]; our formulation can be found in [14] and [46].

Multi-prover games with quantum messages and the associated complexity classes QMIP and QMIP* were introduced in [119], where it was shown that QMIP = NEXP. This equality was also extended to the case of provers who may use prior shared entanglement on a number of qubits bounded by a polynomial in the input length.

The class MIP* was first defined in [46], whose focus is on the special case of XOR proof systems, corresponding to the class $\oplus\text{MIP}^*(2,2)$. The inclusion NEXP $\subseteq \oplus\text{MIP}_{a,b}(2,2)$ for constants $0 < b < a < 1$ is due to Håstad [89]. The inclusion $\oplus\text{MIP}^*(2,2) \subseteq \text{EXP}$ is implicit in [46]. In [170] it was shown that the inclusion can be improved to $\oplus\text{MIP}^*_{a,b}(2) \subseteq \text{QIP}_{a,b}(2)$ for any a, b separated by an inverse polynomial gap, which therefore implies $\oplus\text{MIP}^*_{a,b}(2) \subseteq \text{PSPACE}$.

In [47] it was shown that XOR games obey a perfect parallel repetition theorem. Slofstra [153] investigated the question of entanglement in XOR games, and described a game for which the dimension of entanglement required for an optimal strategy matches the upper bound obtained from Tsirelson's construction.

An extension of XOR games that allows for quantum messages from the verifier to the prover ("quantum XOR games") was considered in [141]. (See also [50] for a closely related model.) It was proved in [141] that Example 6.1 of the coherent state exchange game can be cast as a quantum XOR game with quantum messages for which the optimal success probability can only be reached in the limit as the dimension of the provers' shared entangled state goes to infinity.

The results on perfect completeness, parallelization and public-coin systems presented in Section 6.3 appear in [109]. An alternative proof of the perfect completeness property appears in [118]. Ito [92] showed that games with classical verifiers could be parallelized to 4 turns while keeping the verifier classical, provided one is granted the promise that there exists a classical strategy for the provers that achieves the completeness parameter and soundness holds against entangled-prover strategies. (The corresponding proof systems are called *entanglement-resistant* MIP systems.)

An interesting variant on the multi-prover quantum interactive proof system model, in which the verifier is quantum and the provers are permitted to communicate classically (but not to share prior entanglement) was considered in [30]. The resulting complexity class was proved to contain NEXP.

Theorem 6.5 on simulating a quantum verifier by a classical one was proved in [142]. (See also [143] for a high-level exposition of the results.) This result is based on the rigidity of sequential repetitions of the CHSH game. Rigidity of a single repetition of the CHSH game is proved in [126].

Theorem 6.10 stating the inclusion NEXP $\subseteq$ QMIP* was proved in [95], where an analysis of Babai, Fortnow, and Lund's multilinearity test with entangled provers was given, generalizing the linearity test presented in Section 6.4.2. Letting QMA$_{\mathrm{EXP}}$ denote the exponential-length proof variant of QMA (defined in [75]), one has that QMA$_{\mathrm{EXP}} \subseteq$ QMIP$^*_{a,b}(5,2)$ for some choice of a and b such that $a - b > 2^{-p(n)}$ for a polynomially bounded function p [60], suggesting that a stronger inclusion may be achievable. This result was improved to the inclusion QMA$_{\mathrm{EXP}} \subseteq$ MIP$^*_{a,b}(4,1)$ by [102], again for some choice of a and b separated by an inverse exponential. The exponential gap between completeness and soundness in both results is too small to be amplified via standard techniques, and whether or not the inclusion of QMA$_{\mathrm{EXP}}$ in QMIP* holds is an interesting open problem.

The phenomenon of entanglement monogamy is pervasive in quantum information theory. This terminology is generally attributed to Bennett; one of the first times it appears in print is in [155]. The three-prover CHSH game described in Example 6.4 was introduced and analyzed in [156]. A more general phenomenon is known to hold; for any two-prover one-round game G with classical messages of length at most t, the associated $(1+2^t)$-prover game G' (with the original game being played with two randomly chosen provers) has value $\omega^*(G') = \omega(G)$. The 3-prover clause-versus-variable verifier T_φ appeared in [108], and the two-prover variant is due to [93].

The procedure of parallel repetition for classical multi-prover interactive proof systems was first suggested in [64]. We refer the interested reader to e.g. [53, 38] for recent developments. The fact that the no-signaling value of the Fortnow–Feige–Lovász game is $2/3$ was proved in [91], and the game itself was considered by Fortnow [62] and Feige and Lovasz [59], who were the first to observe that it provides a counter-example to the perfect exponentiation of the classical value of a multi-prover game under parallel repetition. Parallel repetition for two-prover one-round free multi-prover games with classical messages and entangled provers was proved in [42, 98]. These results were extended to any number of provers and games with quantum messages from the provers in [172]. The case of projection games was considered in [54]. The transformation described at the end of Section 6.5.1 is introduced in [24]. The transformation is inspired by earlier work of Feige and Kilian [58] who considered a slightly more complicated transformation in the setting of two-prover classical games. The transformation of Feige and Kilian, and its analysis, was extended to the case of games with entangled provers in [111].

Brunner et al. [40] described a verifier V such that $\omega^{(2)}(V) < \omega^{(3)}(V)$. Moreover there is a single one-round two-prover verifier V that demonstrates the same inequalities, but V chooses its messages to the provers from a continuous set of possibilities (the provers provide binary answers). Vertesi and Pal [163] construct a two-message two-prover classical verifier $V = V_d$ such that $\omega^{(d)}(V) > \omega^{(\lceil \log d \rceil - 1)}(V)$.

Tsirelson's problem was shown to be equivalent [103, 65] to a range of conjectures in the theory of C^*-algebras, including Connes' embedding conjecture [48] and Kirchberg's QWEP conjecture [112]. (See the papers by Ozawa [136, 137] for surveys on the equivalence between the latter two conjectures). Tsirelson's formulation of his problem is available as [157]; see also [149] for a discussion.

Acknowledgements

We thank Mark Wilde and an anonymous reviewer for helpful comments, corrections, and suggestions. Thomas Vidick acknowledges support from the IQIM, an NSF Physics Frontiers Center (NFS Grant PHY-1125565) with support of the Gordon and Betty Moore Foundation (GBMF-12500028). John Watrous acknowledges support from Canada's NSERC and the Canadian Institute for Advanced Research.

References

[1] S. Aaronson, S. Beigi, A. Drucker, B. Fefferman, and P. Shor. The power of unentanglement. *Theory of Computing*, 5(1):1–42, 2009.

[2] S. Aaronson and A. Drucker. A full characterization of quantum advice. *SIAM Journal on Computing*, 43(3):1131–1183, 2014.

[3] S. Aaronson and G. Kuperberg. Quantum versus classical proofs and advice. *Theory of Computing*, 3(7):129–157, 2007.

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

[5] D. Aharonov, I. Arad, Z. Landau, and U. Vazirani. The detectability lemma and quantum gap amplification. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing*, pages 417–426, 2009.

[6] D. Aharonov, I. Arad, and T. Vidick. Guest column: the quantum PCP conjecture. *ACM SIGACT News*, 44(2):47–79, 2013.

[7] D. Aharonov, M. Ben-Or, F. Brandão, and O. Sattath. The pursuit for uniqueness: extending Valiant–Vazirani theorem to the probabilistic and quantum settings. Available as arXiv e-Print 0810.4840, 2008.

[8] D. Aharonov, D. Gottesman, S. Irani, and J. Kempe. The power of quantum systems on a line. *Communications in Mathematical Physics*, 287(1):41–65, 2009.

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

[10] D. Aharonov and O. Regev. Lattice problems in NP $\cap$ coNP. *Journal of the ACM*, 52(5):749–765, 2005.

[11] W. Aiello and J. Håstad. Statistical zero-knowledge languages can be recognized in two rounds. *Journal of Computer and System Sciences*, 42(3):327–345, 1991.

[12] A. Ambainis. On physical problems that are slightly more difficult than QMA. In *Proceedings of the 29th Conference on Computational Complexity*, pages 32–43, 2014.

[13] A. Ambainis, A. Rosmanis, and D. Unruh. Quantum attacks on classical proof systems: The hardness of quantum rewinding. In *55th Annual IEEE Symposium on Foundations of Computer Science*, pages 474–483, 2014.

[14] P. Aravind. A simple demonstration of Bell's theorem involving two observers and no probabilities or inequalities. Available as arXiv.org e-Print quant-ph/0206070, 2002.

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

[16] S. Arora and S. Kale. A combinatorial, primal-dual approach to semidefinite programs. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing*, pages 227–236, 2007.

[17] 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.

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

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

[20] L. Babai. Local expansion of vertex-transitive graphs and random generation in finite groups. In *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing*, pages 164–174, 1991.

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

[22] 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.

[23] 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.

[24] M. Bavarian, T. Vidick, and H. Yuen. Anchoring games for parallel repetition. Available as arXiv.org e-Print 1509.07466, 2015.

[25] S. Beigi, P. Shor, and J. Watrous. Quantum interactive proofs with short messages. *Theory of Computing*, 7:101–117, 2011.

[26] J. Bell. On the Einstein–Podolsky–Rosen paradox. *Physics*, 1:195–200, 1964.

[27] A. Ben-Aroya, O. Schwartz, and A. Ta-Shma. Quantum expanders: Motivation and construction. *Theory of Computing*, 6(3):47–79, 2010.

[28] M. Ben-Or, E. Feig, D. Kozen, and P. Tiwari. A fast parallel algorithm for determining all roots of a polynomial with real roots. In *Proceedings of the 18th Annual ACM Symposium on Theory of Computing*, pages 340–349, 1986.

[29] M. Ben-Or, S. Goldwasser, J. Kilian, and A. Wigderson. Multi-prover interactive proofs: how to remove intractability assumptions. In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing*, pages 113–131, 1988.

[30] M. Ben-Or, A. Hassidim, and H. Pilpel. Quantum multi prover interactive proofs with communicating provers. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 467–476, 2008.

[31] D. Bini and V. Pan. Computing matrix eigenvalues and polynomial zeros where the output is real. *SIAM Journal on Computing*, 27(4):1099–1115, 1998.

[32] H. Blier and A. Tapp. All languages in NP have very short quantum proofs. In *Proceedings of the 2009 3rd International Conference on Quantum, Nano and Micro Technologies*, pages 34–37, 2009.

[33] M. Blum, M. Luby, and R. Rubinfeld. Self-testing/correcting with applications to numerical problems. *Journal of Computer and System Sciences*, 47(3):549–595, 1993.

[34] A. Bookatz. QMA-complete problems. *Quantum Information & Computation*, 14(5&6):361–383, 2014.

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

[36] 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.

[37] F. Brandão, M. Christandl, and J. Yard. A quasipolynomial-time algorithm for the quantum separability problem. In *Proceedings of the 43rd Annual ACM Symposium on Theory of Computing*, pages 343–352, 2011.

[38] Mark Braverman and Ankit Garg. Small value parallel repetition for general games. In *Proceedings of the 47th Annual ACM on Symposium on Theory of Computing*, pages 335–340. ACM, 2015.

[39] S. Bravyi. Efficient algorithm for a quantum analogue of 2-SAT. *Contemporary Mathematics*, 536:33–48, 2011.

[40] N. Brunner, S. Pironio, A. Acin, N. Gisin, A. Méthot, and V. Scarani. Testing the dimension of Hilbert spaces. *Physical Review Letters*, 100:210503, 2008.

[41] A. Cabello. Bell's theorem without inequalities and without probabilities for two observers. *Physical Review Letters*, 86:1911–1914, 2001.

[42] A. Chailloux and G. Scarpa. Parallel repetition of entangled games with exponential decay via the superposed information cost. In *Automata, Languages, and Programming*, volume 2014 of *Lecture Notes in Computer Science*, pages 296–307. Springer, 2014.

[43] A. Chiesa and M. Forbes. Improved soundness for QMA with multiple provers. *Chicago Journal of Theoretical Computer Science*, 2013:1–23, 2013.

[44] J. Clauser, M. Horne, A. Shimony, and R. Holt. Proposed experiment to test local hidden-variable theories. *Physical Review Letters*, 23:880–884, 1969.

[45] R. Cleve, A. Ekert, C. Macchiavello, and M. Mosca. Quantum algorithms revisited. *Proceedings of the Royal Society*, A454:339–354, 1998.

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

[47] R. Cleve, W. Slofstra, F. Unger, and S. Upadhyay. Perfect parallel repetition theorem for quantum XOR proof systems. *Computational Complexity*, 17(2):282–299, 2008.

[48] A. Connes. Classification of injective factors cases II_1, II_∞, III_λ, $\lambda \neq 1$. *Annals of Mathematics*, 104(1):73–115, 1976.

[49] S. Cook. The complexity of theorem proving procedures. In *Proceedings of the 3rd Annual ACM Symposium on Theory of Computing*, pages 151–158, 1971.

[50] T. Cooney, M. Junge, C. Palazuelos, and D. Pérez-García. Rank-one quantum games. *Computational Complexity*, 24(1):133–196, 2011.

[51] T. Cubitt and A. Montanaro. Complexity classification of local Hamiltonian problems. In *Proceedings of the 55th Annual IEEE Symposium on Foundations of Computer Science*, pages 120–129, 2014.

[52] I. Damgård and C. Lunemann. Quantum-secure coin-flipping and applications. In *Advances in Cryptology – ASIACRYPT 2009*, volume 5912 of *Lecture Notes in Computer Science*, pages 52–69. Springer, 2009.

[53] I. Dinur and D. Steurer. Analytical approach to parallel repetition. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pages 624–633, 2014.

[54] I. Dinur, D. Steurer, and T. Vidick. A parallel repetition theorem for entangled projection games. In *Proceedings of the 29th Conference on Computational Complexity*, pages 197–208, 2014.

[55] J. Edmonds. Minimum partition of a matroid into independent subsets. *Journal of Research of the National Bureau of Standards Section B: Mathematics and Mathematical Physics*, 69B(1–2):67–72, 1965.

[56] J. Edmonds. Paths, trees, and flowers. *Canadian Journal of Mathematics*, 17(3):449–467, 1965.

[57] U. Feige, S. Goldwasser, L. Lovász, S. Safra, and M. Szegedy. Interactive proofs and the hardness of approximating cliques. *Journal of the ACM*, 43(2):268–292, 1996.

[58] U. Feige and J. Kilian. Two-prover protocols—low error at affordable rates. *SIAM Journal on Computing*, 30(1):324–346, 2000.

[59] U. Feige and L. Lovász. Two-prover one-round proof systems: Their power and their problems. In *Proceedings of the 24th Annual ACM Symposium on Theory of Computing*, pages 733–744, 1992.

[60] J. Fitzsimons and T. Vidick. A multiprover interactive proof system for the local Hamiltonian problem. In *Proceedings of the 6th Conference on Innovations in Theoretical Computer Science*, pages 103–112, 2015.

[61] L. Fortnow. The complexity of perfect zero-knowledge. In S. Micali, editor, *Randomness and Computation*, volume 5 of *Advances in Computing Research*, pages 327–343. Greenwich: JAI Press, 1989.

[62] L. Fortnow. *Complexity-Theoretic Aspects of Interactive Proof Systems*. PhD thesis, Massachusetts Institute of Technology, 1989.

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

[64] L. Fortnow, J. Rompel, and M. Sipser. On the power of multi-prover interactive protocols. In *Proceedings of the 3rd Annual Structure in Complexity Theory Conference*, pages 156–161, 1988.

[65] T. Fritz. Tsirelson's problem and Kirchberg's conjecture. *Reviews in Mathematical Physics*, 24(05):1250012, 2012.

[66] S. Gharibian. Strong NP-hardness of the quantum separability problem. *Quantum Information & Computation*, 10(3):343–360, 2010.

[67] S. Gharibian and J. Kempe. Hardness of approximation for quantum problems. In *Automata, Languages, and Programming*, volume 7391 of *Lecture Notes in Computer Science*, pages 387–398. Springer, 2012.

[68] S. Gharibian and J. Sikora. Ground state connectivity of local Hamiltonians. Available as arXiv.org e-Print 1409.3182, 2014.

[69] O. Goldreich. Zero-knowledge twenty years after its invention. Electronic Colloquium on Computational Complexity Report 2002/186, 2002.

[70] O. Goldreich, A. Sahai, and S. Vadhan. Honest verifier statistical zero knowledge equals general statistical zero knowledge. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pages 23–26, 1998.

[71] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof systems. In *Proceedings of the 17th Annual ACM Symposium on Theory of Computing*, pages 291–304, 1985.

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

[73] 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. Greenwich: JAI Press, 1989.

[74] D. Gosset and D. Nagaj. Quantum 3-SAT is QMA1-complete. In *Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science*, pages 756–765, 2013.

[75] D. Gottesman and S. Irani. The quantum and classical complexity of translationally invariant tiling and Hamiltonian problems. In *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 95–104, 2009.

[76] A. Grilo, I. Kerenidis, and J. Sikora. Quantum NP - a survey. Available as arXiv.org e-Print quant-ph/0210077, 2002.

[77] A. Grilo, I. Kerenidis, and J. Sikora. QMA with subset state witnesses. Available as arXiv.org e-Print 1410.2882, 2014.

[78] L. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing*, pages 212–219, 1996.

[79] L. Gurvits. Classical deterministic complexity of Edmonds' problem and quantum entanglement. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing*, pages 1–19, 2003.

[80] G. Gutoski. Upper bounds for quantum interactive proofs with competing provers. In *Proceedings of the 20th Conference on Computational Complexity*, pages 334–343, 2005.

[81] G. Gutoski. *Quantum Strategies and Local Operations.* PhD thesis, University of Waterloo, 2009.

[82] G. Gutoski, P. Hayden, K. Milner, and M. Wilde. Quantum interactive proofs and the complexity of separability testing. *Theory of Computing*, 11(3):59–103, 2015.

[83] G. Gutoski and J. Watrous. Quantum interactive proofs with competing provers. In *Proceedings of the 22nd Symposium on Theoretical Aspects of Computer Science*, volume 3404 of *Lecture Notes in Computer Science*, pages 605–616. Springer, 2005.

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

[85] G. Gutoski and X. Wu. Parallel approximation of min-max problems. *Computational Complexity*, 22(2):385–428, 2013.

[86] S. Hallgren, A. Kolla, P. Sen, and S. Zhang. Making classical honest verifier zero knowledge protocols secure against quantum attacks. In *Proceedings of the 35th International Colloquium on Automata, Languages and Programming*, volume 5126 of *Lecture Notes in Computer Science*, pages 592–603. Springer, 2008.

[87] S. Hallgren, A. Smith, and F. Song. Classical cryptographic protocols in a quantum world. In *Advances in Cryptology – CRYPTO 2011*, volume 6841 of *Lecture Notes in Computer Science*, pages 411–428. Springer, 2011.

[88] A. Harrow and A. Montanaro. Testing product states, quantum Merlin–Arthur games and tensor optimization. *Journal of the ACM*, 60(1):3, 2013.

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

[90] P. Hayden, K. Milner, and M. Wilde. Two-message quantum interactive proofs and the quantum separability problem. *Quantum Information & Computation*, 14(5&6):384–416, 2014.

[91] T. Holenstein. Parallel repetition: Simplifications and the no-signaling case. *Theory of Computing*, 5:141–172, 2009.

[92] T. Ito. Parallelization of entanglement-resistant multi-prover interactive proofs. *Information Processing Letters*, 114(10):579–583, 2014.

[93] T. Ito, H. Kobayashi, and K. Matsumoto. Oracularization and two-prover one-round interactive proofs against nonlocal strategies. In *Proceedings of the 24th Conference on Computational Complexity*, pages 217–228, 2009.

[94] T. Ito, H. Kobayashi, and J. Watrous. Quantum interactive proofs with weak error bounds. In *Proceedings of the 3rd Conference on Innovations in Theoretical Computer Science*, pages 266–275, 2012.

[95] T. Ito and T. Vidick. A multi-prover interactive proof for NEXP sound against entangled provers. *Proceedings of the 53rd IEEE Symposium on Foundations of Computer Science*, pages 243–252, 2012.

[96] R. Jain, Z. Ji, S. Upadhyay, and J. Watrous. QIP = PSPACE. *Journal of the ACM*, 58(6):30, 2011.

[97] R. Jain, I. Kerenidis, G. Kuperberg, M. Santha, O. Sattath, and S. Zhang. On the power of a unique quantum witness. *Theory of Computing*, 8(17):375–400, 2012.

[98] R. Jain, A. Pereszlényi, and P. Yao. A parallel repetition theorem for entangled two-player one-round games under product distributions. In *Proceedings of the 29th Conference on Computational Complexity*, pages 209–216, 2014.

[99] R. Jain, S. Upadhyay, and J. Watrous. Two-message quantum interactive proofs are in PSPACE. In *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 534–543, 2009.

[100] R. Jain and J. Watrous. Parallel approximation of non-interactive zero-sum quantum games. In *Proceedings of the 24th Conference on Computational Complexity*, pages 243–253, 2009.

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

[102] Z. Ji. Classical verification of quantum proofs. Available as arXiv.org e-Print 1505.07432, 2015.

[103] M. Junge, M. Navascues, C. Palazuelos, D. Perez-Garcia, V. Scholz, and R. Werner. Connes' embedding problem and Tsirelson's problem. *Journal of Mathematical Physics*, 52(1):012102, 2011.

[104] S. Kale. *Efficient Algorithms Using the Multiplicative Weights Update Method*. PhD thesis, Princeton University, 2007.

[105] R. Karp. Reducibility among combinatorial problems. In R. Miller and J. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, New York, 1972.

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

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

[108] J. Kempe, H. Kobayashi, K. Matsumoto, B. Toner, and T. Vidick. Entangled games are hard to approximate. *SIAM Journal on Computing*, 40(3):848–877, 2011.

[109] J. Kempe, H. Kobayashi, K. Matsumoto, and T. Vidick. Using entanglement in quantum multi-prover interactive proofs. *Computational Complexity*, 18:273–307, 2009.

[110] J. Kempe and O. Regev. 3-local Hamitonian is QMA-complete. *Quantum Information & Computation*, 3(3):258–264, 2003.

[111] J. Kempe and T. Vidick. Parallel repetition of entangled games. In *Proceedings of the 43rd Annual ACM Symposium on Theory of Computing*, pages 353–362, 2011.

[112] E. Kirchberg. On non-semisplit extensions, tensor products and exactness of group C^*-algebras. *Inventiones Mathematicae*, 112(1):449–489, 1993.

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

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

[115] 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.

[116] H. Kobayashi. Non-interactive quantum perfect and statistical zero-knowledge. In *Proceedings of the 14th International Symposium on Algorithms and Computation*, volume 2906 of *Lecture Notes in Computer Science*, pages 178–188. Springer, 2003.

[117] H. Kobayashi. General properties of quantum zero-knowledge proofs. In *Proceedings of the 5th IACR Theory of Cryptography Conference*, volume 4948 of *Lecture Notes in Computer Science*, pages 107–124. Springer, 2008.

[118] H. Kobayashi, F. Le Gall, and H. Nishimura. Stronger methods of making quantum interactive proofs perfectly complete. In *Proceedings of the 4th Conference on Innovations in Theoretical Computer Science*, pages 329–352, 2013.

[119] H. Kobayashi and K. Matsumoto. Quantum multi-prover interactive proof systems with limited prior entanglement. *Journal of Computer and System Sciences*, 66(3):429–450, 2003.

[120] H. Kobayashi, K. Matsumoto, and T. Yamakami. Quantum Merlin–Arthur proof systems: Are multiple Merlins more helpful to Arthur? In *Algorithms and Computation*, volume 2906 of *Lecture Notes in Computer Science*, pages 189–198. Springer, 2003.

[121] D. Leung, B. Toner, and J. Watrous. Coherent state exchange in multi-prover quantum interactive proof systems. *Chicago Journal of Theoretical Computer Science*, 2013:11, 2013.

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

[123] 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.

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

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

[126] M. McKague, T. Yang, and V. Scarani. Robust self-testing of the singlet. *Journal of Physics A: Mathematical and Theoretical*, 45(45):455304, 2012.

[127] D. Mermin. Simple unified form for the major no-hidden-variables theorems. *Physical Review Letters*, 65(27):3373–3376, 1990.

[128] R. Mittal and M. Szegedy. Product rules in semidefinite programming. In *Fundamentals in Computation Theory*, volume 4639 of *Lecture Notes in Computer Science*, pages 435–445. Springer, 2007.

[129] M. Navascués, S. Pironio, and A. Acín. A convergent hierarchy of semidefinite programs characterizing the set of quantum correlations. *New Journal of Physics*, 10(7):073013, 2008.

[130] C. Neff. Specified precision polynomial root isolation is in NC. *Journal of Computer and System Sciences*, 48(3):429–463, 1994.

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

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

[133] R. O'Donnell. A history of the PCP theorem, 2005. Available at `http://courses.cs.washington.edu/courses/cse533/05au/pcp-history.pdf`.

[134] T. Okamoto. On relationships between statistical zero-knowledge proofs. *Journal of Computer and System Sciences*, 60(1):47–108, 2000.

[135] R. Oliveira and B. Terhal. The complexity of quantum spin systems on a two-dimensional square lattice. *Quantum Information & Computation*, 8(10):900–924, 2008.

[136] N. Ozawa. About the QWEP conjecture. *International Journal of Mathematics*, 15(05):501–530, 2004.

[137] N. Ozawa. About the Connes embedding conjecture. *Japanese Journal of Mathematics*, 8(1):147–183, 2013.

[138] C. Papadimitriou and M. Yannakakis. The complexity of facets (and some facets of complexity). *Journal of Computer and System Sciences*, 28(2):244–259, 1984.

[139] A. Peres. Incompatible results of quantum measurements. *Physics Letters A*, 151(3–4):107–108, 1990.

[140] R. Raz. A parallel repetition theorem. *SIAM Journal on Computing*, 27(3):763–803, 1998.

[141] O. Regev and T. Vidick. Quantum XOR games. In *Proceedings of the 28th Conference on Computational Complexity*, pages 144–155, 2013.

[142] B. Reichardt, F. Unger, and U. Vazirani. A classical leash for a quantum system: Command of quantum systems via rigidity of CHSH games. Available as arXiv.org e-Print 1209.0448, 2012.

[143] B. Reichardt, F. Unger, and U. Vazirani. Classical command of quantum systemes. *Nature*, 496(7446):456–460, 2013.

[144] B. Rosgen. Additivity and distinguishability of random unitary channels. *Journal of Mathematical Physics*, 49(10):102107, 2008.

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

[146] B. Rosgen. *Computational Distinguishability of Quantum Channels.* PhD thesis, University of Waterloo, 2009.

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

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

[149] V. Scholz and R. Werner. Tsirelson's problem. Available as arXiv.org e-Print 0812.4305.

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

[151] 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.

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

[153] W. Slofstra. Lower bounds on the entanglement needed to play XOR non-local games. *Journal of Mathematical Physics*, 52(10):102202, 2011.

[154] R. Spekkens and T. Rudolph. Degrees of concealment and bindingness in quantum bit-commitment protocols. *Physical Review A*, 65(1):123410, 2001.

[155] B. Terhal. Is entanglement monogamous? *IBM Journal of Research and Development*, 48:71–78, 2004.

[156] B. Toner. Monogamy of non-local quantum correlations. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 465, pages 59–69, 2009.

[157] B. Tsirelson. Bell inequalities and operator algebras. Available at `http://www.tau.ac.il/~tsirel/download/bellopalg.pdf`.

[158] B. Tsirel'son. Quantum analogues of the Bell inequalities: The case of two spatially separated domains. *Journal of Soviet Mathematics*, 36:557–570, 1987.

[159] D. Unruh. Quantum proofs of knowledge. In *Advances in Cryptology – Eurocrypt 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 135–152. Springer, 2012.

[160] L. Valiant and V. Vazirani. NP is as easy as detecting unique solutions. *Theoretical Computer Science*, 47:85–93, 1986.

[161] W. van Dam and P. Hayden. Universal entanglement transformations without communication. *Physical Review A*, 67(6):060302, 2003.

[162] J. van de Graaf. *Towards a Formal Definition of Security for Quantum Protocols*. PhD thesis, Université de Montréal, 1997.

[163] T. Vértesi and K. Pál. Bounding the dimension of bipartite quantum systems. *Physical Review A*, 79:042106, 2009.

[164] M. Warmuth and D. Kuzmin. Online variance minimization. In *Proceedings of the 19th Annual Conference on Learning Theory*, volume 4005 of *Lecture Notes in Computer Science*, pages 514–528. Springer, 2006.

[165] J. Watrous. PSPACE has constant-round quantum interactive proof systems. In *Proceedings of the 40th Annual IEEE Symposium on Foundations of Computer Science*, pages 112–119, 1999.

[166] 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.

[167] 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.

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

[169] J. Watrous. Zero-knowledge against quantum attacks. *SIAM Journal on Computing*, 39(1):25–58, 2009.

[170] 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, 2006.

[171] X. Wu. Equilibrium value method for the proof of QIP=PSPACE. Available as arXiv.org e-Print 1004.0264, 2010.

[172] X. Wu, K.-M. Chung, and H. Yuen. Parallel repetition for entangled k-player games via fast quantum search. In *Proceedings of the 30th Conference on Computational Complexity*, volume 33 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 512–536. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2015.