PSPACE has constant-round quantum interactive proof systems

John Watrous*
Department of Computer Science
University of Calgary
Calgary, Alberta, Canada T2N 1N4

January 17, 2001

Abstract

In this paper we introduce quantum interactive proof systems, which are interactive proof systems in which the prover and verifier may perform quantum computations and exchange quantum messages. It is proved that every language in PSPACE has a quantum interactive proof system that requires a total of only three messages to be sent between the prover and verifier and has exponentially small (one-sided) probability of error. It follows that quantum interactive proof systems are strictly more powerful than classical interactive proof systems in the constant-round case unless the polynomial time hierarchy collapses to the second level.

1 Introduction

Several recent papers have provided compelling evidence, and proof in some cases, that certain computational, cryptographic, and information-theoretic tasks can be performed more efficiently by models based on quantum physics than those based on classical physics. For example, Shor [30] has shown that integers can be factored in expected polynomial time by quantum computers, a quantum key distribution protocol of Bennett and Brassard [10] that does not rely on intractability assumptions has been proven to be secure [12, 25, 31], and Raz [27] has shown an exponential separation between quantum and classical two-party communication complexity models. In this paper we introduce the quantum analogue of another concept—interactive proof systems—and provide strong evidence that additional power is gained by interactive proof systems in the quantum setting.

Interactive proof systems were first introduced by Babai [4] and Goldwasser, Micali, and Rackoff [21]. An interactive proof system consists of an interaction between a computationally unbounded prover and a computationally bounded probabilistic verifier. The prover attempts to convince the verifier that a given input string satisfies some property, while the verifier tries to determine the validity of this "proof". A language L is said to have an interactive proof system if there exists a polynomial-time verifier V such that (i) there exists a prover P (called an honest prover) that can always convince V to accept when the given input is in L, and (ii) no prover P' can convince V to accept with non-negligible probability when the input is not in L. The class of languages having interactive proof systems is denoted IP.

*Email: jwatrous@cpsc.ucalgary.ca. This research was supported by Canada's NSERC and was done while the author was at the Département IRO, Université de Montréal, Montréal, Québec, Canada H3C 3J7.

Based on the work of Lund, Fortnow, Karloff, and Nisan [24], Shamir [28] proved that the quantified boolean formula (QBF) problem, and therefore every language in PSPACE, has an interactive proof system. Since any language having an interactive proof system is in PSPACE [19], this implies IP = PSPACE. All known protocols for PSPACE require a nonconstant number of rounds of communication between the prover and verifier, and cannot be parallelized to require only a constant number of rounds under the assumption that the polynomial time hierarchy is proper. This follows from the fact that the class of languages having constant-round interactive proof systems is equivalent to AM [4, 22], and hence is contained in Π_2^p.

The main result we prove in this paper is as follows.

Theorem 1 *Every language in PSPACE has a three-message quantum interactive proof system with exponentially small one-sided error.*

This result contrasts with the facts mentioned above regarding classical interactive proof systems, as it shows there are languages having constant-round quantum interactive proof systems that do not have constant-round classical interactive proof systems unless AM = PSPACE.

Subsequent to the publication of the preliminary version of the present paper, Kitaev and Watrous [23] have proved a stronger result than Theorem 1, which is that any polynomial-round *quantum* interactive proof system can be parallelized to just three messages. The reader interested in the more general result should therefore refer to that paper. However, the techniques used by Kitaev and Watrous and in the present paper differ considerably, and we believe that both techniques will potentially find other applications.

We now summarize informally our technique for proving Theorem 1, which is essentially to show that we may parallelize a classical interactive proof system for the QBF problem by allowing the prover and verifier to send and process quantum information.

Consider the following unsuccessful method for trying to reduce the number of rounds required by a nonconstant-round protocol for PSPACE to a single round: define the verifier so that it first generates all of its random numbers, sends them all to the prover in one message, receives all the responses from the prover, and checks the validity of the interaction. This will clearly not work, since the prover may cheat by "looking ahead" and basing its responses on random numbers that would have been sent in later rounds in the nonconstant-round case—the fact that the prover must commit to certain answers before seeing the verifier's subsequent messages is essential for the soundness of existing protocols. However, using interactive proofs based on quantum physics, this technique can be made to work, as the abovementioned behavior on the part of the prover can be detected by a quantum verifier. We now sketch the method for doing this—a formal description of the protocol appears in Section 3.

Suppose V_c is a classical verifier and P_c is a classical prover for QBF. It is assumed that V_c is nonadaptive in the sense that each of its messages is simply a sequence of coin-flips that is independent of the prover's messages. We will let V and P denote the quantum verifier and prover in the resulting three-message quantum protocol. The quantum prover P first generates a superposition over all possible conversations between V_c and P_c. The prover P sends this state to the verifier V, but for technical reasons the prover will keep a copy of the messages corresponding to V_c for himself. At this point V and P share an entangled quantum state between them. It will be shown that P cannot cheat by preparing a state that is biased towards certain random choices for V_c, since the verifier V will be able to later check that the state was close to uniform. The verifier V checks that the conversations sent by P are valid according to the classical protocol for QBF by performing a suitable measurement. Naturally V rejects in case it sees a conversation that is not valid.

The verifier V now needs to check that the quantum state sent by P really corresponded to some classical prover P_c. Such a prover P_c would necessarily fix its response to each message of V_c before seeing messages that would be sent in later rounds. The verifier V does this by randomly choosing some round in the classical protocol and challenging the prover P to demonstrate that there is no correlation between the verifier-messages after the chosen round and the prover-messages before and including the chosen round. Let us say that the prover- and verifier-messages before and including the chosen round have *low-index*, and the remaining messages in the conversation have *high-index*. Thus, the low-index prover-messages should not have any correlation with the high-index verifier-messages. The challenge is as follows: V sends back to P all of the high-index prover-messages but keeps the low-index prover-messages and all of the verifier-messages, allows P to reverse its computation of the high-index prover-messages, and also allows P to send back its copy of all of the verifier-messages. The verifier now checks that the superposition of high-index verifier-messages is uniform by performing an appropriately defined measurement. If there is significant correlation between the low-index prover responses and the high-index verifier-messages, the uniformity test will fail with high probability and V will reject.

By performing the above process in parallel a polynomial number of times, the probability a cheating prover escapes detection is made exponentially small, while the protocol still requires only three messages to be communicated. We prove that the prover cannot cheat by entangling parallel executions of the protocol.

It is interesting to note that whereas any constant-message interactive proof system can be parallelized to just two messages in the classical case, it is apparently not straightforward to apply similar techniques in the quantum case—we have not been able to reduce our three-message quantum protocol to require only two messages, and it is an interesting open question whether this is possible in general.

The remainder of the paper is organized as follows. In Section 2 we give a formal definition of quantum interactive proof systems based on the quantum circuit model. In Section 3 we prove Theorem 1 by presenting a three-message quantum interactive proof system for the quantified boolean formula problem and proving its correctness. We conclude with Section 4, which mentions some open problems.

2 Definitions

We now give a formal definition of quantum interactive proof systems. We restrict our attention to constant-round quantum interactive proof systems, although the definition is easily extended to a nonconstant number of rounds. See Kitaev and Watrous [23] for further details. The model for quantum computation that provides a basis for our definition is the quantum circuit model. We will not define quantum circuits or discuss them in detail, as this has been done elsewhere (see Yao [33], Berthiaume [11], and Nielsen and Chuang [26] for example).

An m-message verifier V is a polynomial-time computable mapping $V : \Sigma^* \times \{1, \ldots, k\} \to \Sigma^*$ for $k = \lfloor m/2 + 1 \rfloor$, where each $V(x, j)$ is an encoding of a quantum circuit composed of quantum gates from some appropriately chosen universal set of gates. Universal sets of gates/transformations have been investigated in several papers [1, 7, 8, 15, 16]; for the purposes of this paper, we will assume only that this set includes the Hadamard gate and any gate or collection of gates universal for reversible computation, such as the Fredkin gate or Toffoli gate. Each encoding $V(x, j)$ is identified with the quantum circuit it encodes. It is assumed that this encoding is such that the size of a circuit is polynomial in the length of its encoding, so that each circuit $V(x, j)$ is polynomial in size.

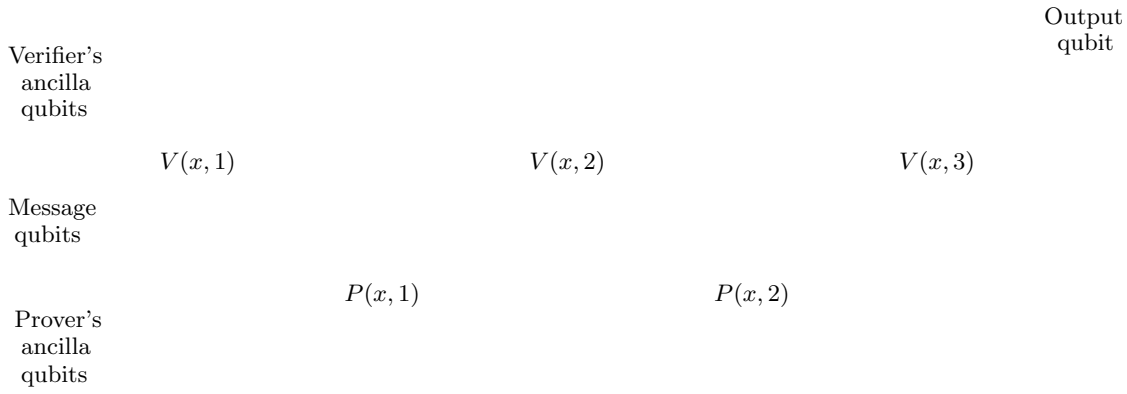

Figure 1: Quantum circuit for a 4-message quantum interactive proof system

The qubits upon which each $V(x,j)$ acts are divided into two groups: message qubits and ancilla qubits. The message qubits represent the communication channel between the prover and verifier, while the ancilla qubits represent qubits that are private to the verifier. One of the verifier's ancilla qubits is specified as the output qubit.

An m-message prover P is a mapping from $\Sigma^* \times \{1,\ldots,k'\}$ to the set of all quantum circuits, where $k' = \lfloor m/2+1/2 \rfloor$. No restrictions are placed on the size of each $P(x,j)$ or on the gates from which these circuits are composed. Similar to the case of the verifier, the qubits of the prover are divided into message qubits and ancilla qubits. Note that although the prover is all-powerful in a computational sense, since there is no bound on the complexity of the mapping P or on the size of each $P(x,j)$, we of course require that the prover obeys the laws of physics! This is enforced by requiring that the prover's actions correspond to valid quantum circuits.

Given a prover/verifier pair (P,V), consider a quantum circuit composed as shown in Figure 1 (the case $m=4$ is shown). The probability that a pair (P,V) accepts a given input x is defined to be the probability that an observation of the output qubit in the $\{|0\rangle,|1\rangle\}$ basis yields $|1\rangle$ when the circuits $V(x,1)$, $P(x,1)$, $V(x,2)$, ..., $P(x,k')$, $V(x,k)$ in case m is even, or $P(x,1)$, $V(x,1)$, ..., $P(x,k')$, $V(x,k)$ in case m is odd, are applied in sequence as illustrated, assuming all qubits are initially in the $|0\rangle$ state.

Now, we say that a language L has an m-message quantum interactive proof system with error probability ϵ if there exists an m-message verifier V such that

1. There exists an m-message prover P such that if $x \in L$ then (P,V) accepts x with probability 1.
2. For all m-message provers P', if $x \notin L$ then (P',V) accepts x with probability at most ϵ.

A few notes regarding the above definition are in order. First, we note that there are other ways in which we could have defined quantum interactive proof systems, such as a definition based on quantum Turing machines or a definition requiring that each circuit as above be given by $V(|x|,i)$ or $P(|x|,i)$, with x supplied as input to each circuit. We have chosen the above definition because of its simplicity. Given the apparent robustness of the class of polynomial-time computable quantum transformations, we suspect these definitions to be equivalent, although we have not investigated this question in detail. Second, we assume that each circuit corresponds to a unitary operator (e.g., no "measurement gates" are used). The action of any general quantum gate (i.e., a gate corresponding to a trace-preserving, completely positive linear map on mixed states of qubits) can

always be simulated by some unitary gate, possibly adding more ancilla qubits [2]. As this will not increase the size of a verifier's circuit by more than a polynomial factor, and will not affect the complexity of the mapping V significantly, our definition is equivalent to a definition allowing more general quantum gates.

3 Three-message quantum interactive proof systems for QBF

Recall that a quantified boolean formula is a formula of the form $Q_1 x_1 \cdots Q_n x_n \psi(x_1, \ldots, x_n)$, where each Q_i is an existential or universal quantifier ($\exists$ or $\forall$) and $\psi(x_1, \ldots, x_n)$ is a boolean formula without quantifiers in the variables $x_1, \ldots, x_n$. The quantified boolean formula (QBF) problem is to determine if a given quantified boolean formula is true.

To prove Theorem 1, it is sufficient to prove that there exists a three-message quantum interactive proof system with exponentially small error for the QBF problem. This is because the verifier may first compute a polynomial-time reduction from a given problem in PSPACE to the QBF problem, then execute the protocol for QBF, adjusting various parameters in the protocol to reduce error as necessary.

3.1 Classical QBF protocol

Our three-message quantum interactive proof system for the QBF problem is based on a variant of the Lund–Fortnow–Karloff–Nisan protocol due to Shen [29], to which the reader is referred for a detailed description. In this section we review some facts regarding this protocol that will later be helpful.

Let $\phi = Q_1 x_1 \cdots Q_n x_n \psi(x_1, \ldots, x_n)$ be a fixed input formula. Also let $\mathbb{F}$ be a finite field, whose size will be chosen later, and let $N = \binom{n+1}{2} + n$. The protocol is as follows. For $j = 1, \ldots, N-1$, the prover sends the verifier a polynomial p_j over $\mathbb{F}$ of degree at most d (which is determined by the protocol and may be taken to be polynomial in n), and the verifier chooses $r_j \in \mathbb{F}$ and sends r_j to the prover. The prover then sends a polynomial p_N to the verifier in the final round, and the verifier chooses $r_N \in \mathbb{F}$. The verifier then evaluates a particular polynomial-time predicate $E(\phi, r_1, \ldots, r_N, p_1, \ldots, p_N)$ and accepts if and only if the predicate evaluates to true.

A formal description of E may be derived from the paper of Shen. Since the details of the predicate are not necessary for our discussion, we will only state certain properties of E. First, for any sequence of random field elements $r_1, \ldots, r_N \in \mathbb{F}$ there exist polynomials $q_1, \ldots, q_N$, where each polynomial q_j depends only on $r_1, \ldots, r_{j-1}$, that correspond to the answers that should be given by an honest prover. These polynomials, which are well-defined regardless of the boolean value of ϕ, satisfy the following properties:

1. If ϕ is true, then $E(\phi, r_1, \ldots, r_N, q_1, \ldots, q_N)$ is true.
2. If ϕ is false, then for all $r_1, \ldots, r_N$ and $p_2, \ldots, p_N$, $E(\phi, r_1, \ldots, r_N, q_1, p_2, \ldots, p_N)$ is false.
3. If ϕ is false, then for all $k \leq N-1$, $r_1, \ldots, r_{k-1}$, and $p_1, \ldots, p_k$, the following holds in case $p_k \neq q_k$: there are at most d values of r_k for which there exist $r_{k+1}, \ldots, r_N$ and $p_{k+2}, \ldots, p_N$ such that $E(\phi, r_1, \ldots, r_N, p_1, \ldots, p_k, q_{k+1}, p_{k+2}, \ldots, p_N)$ is true.
4. If ϕ is false, then for any $r_1, \ldots, r_{N-1}$ and polynomials $p_1, \ldots, p_N$ such that $p_N \neq q_N$, there are at most d values of r_N for which $E(\phi, r_1, \ldots, r_N, p_1, \ldots, p_N)$ is true.

For given $r_1, \ldots, r_{k-1}$, we call the polynomial q_k the *correct* polynomial corresponding to $r_1, \ldots, r_{k-1}$.

Clearly, if ϕ evaluates to true, an honest prover can always convince the verifier to accept by sending the correct polynomials $q_1, \ldots, q_N$ corresponding to the verifiers random numbers $r_1, \ldots, r_{N-1}$.

Now suppose that ϕ evaluates to false. By item 2, a cheating prover cannot send the correct polynomial q_1 on the first round, for the prover rejects with certainty in this case. Hence the prover must send $p_1 \neq q_1$ if the verifier is to accept. Now suppose for $1 \leq k \leq N-1$ and $r_1, \ldots, r_{k-1}$ the prover has sent polynomials $p_1 \neq q_1, \ldots, p_k \neq q_k$ during rounds $1, \ldots, k$. Unless the verifier randomly chooses one of d particular values for r_k, the prover may not send q_{k+1} on the next round without causing the verifier to reject. Hence, if the prover sends an incorrect polynomial on round k, then with probability at least $1 - d/|\mathbb{F}|$ it must send an incorrect polynomial on round $k+1$. Finally, if the prover does not send the correct polynomial q_N during the last round, the verifier accepts with probability at most $d/|\mathbb{F}|$. Hence, the total probability that the verifier accepts may not exceed $(dN)/|\mathbb{F}|$.

Since the error probability of the protocol depends on the size of $\mathbb{F}$, $\mathbb{F}$ may be chosen sufficiently large at the start of the protocol. It will be convenient for us to take $\mathbb{F}$ to be the field with 2^k elements for k polynomial in n, which yields an exponentially small probability of error. For any chosen k, the verifier and prover may use a deterministic procedure to implement arithmetic in $\mathbb{F}$ in the following common way: an irreducible polynomial g of degree k over $GF(2)$ is computed in deterministic polynomial time [32], elements of $\mathbb{F}$ are identified with polynomials over $GF(2)$ of degree at most $k-1$, and arithmetic is taken to be the usual arithmetic on polynomials modulo g. This yields a natural correspondence between k bit strings and elements of $\mathbb{F}$.

3.2 Quantum verifier's protocol and proof of completeness

We now describe the three-message quantum verifier's protocol for QBF along with the protocol for the honest prover. It will be straightforward to demonstrate that this gives a complete proof system—the soundness of the proof system is proved in the next subsection.

We use the following conventions when describing the protocol. Collections of qubits upon which various transformations are performed are referred to as *registers*, and are labeled by capital letters in boldface. The registers required by the protocol are $\mathbf{R}_{i,j}$, $\mathbf{S}_{i,j}$, and $\mathbf{P}_{i,j}$ for $1 \leq i \leq m$ and $1 \leq j \leq N$, where N is as in the classical protocol described in Section 3.1 and m is some polynomial in the input size, chosen depending on the desired error bound as described later. Each register $\mathbf{R}_{i,j}$ and $\mathbf{S}_{i,j}$ consists of k qubits, where 2^k is to be the size of the field $\mathbb{F}$. We view the classical states of these registers as elements in $\mathbb{F}$, and identify these registers with the random field elements chosen by the verifier in the classical protocol. As for m we will later choose k to be some polynomial in order to obtain our desired error bound. Each $\mathbf{P}_{i,j}$ consists of $d+1$ collections of k qubits, for d as in the classical protocol, and we view the classical states of these registers as polynomials of degree at most d with coefficients in $\mathbb{F}$. These registers are identified with the polynomials sent by the prover to the verifier in the classical protocol.

The verifier may also use any polynomial number of additional ancilla qubits in order to perform the transformations described, and will also store various auxiliary variables, such as the random vector u described below, needed for the protocol. As there will be no need for the verifier to perform quantum operations on these values, we consider them as being stored classically. There is no difference in the behavior of the protocol if they are thought of as being stored in quantum registers, however.

It will be convenient to refer to certain collections of the quantum registers mentioned above; for a given vector $u \in \{1, \ldots, N\}^m$ we let $\mathbf{R}^u$ be the collection of registers $\mathbf{R}_{i,1}, \ldots, \mathbf{R}_{i,u_i-1}$ for

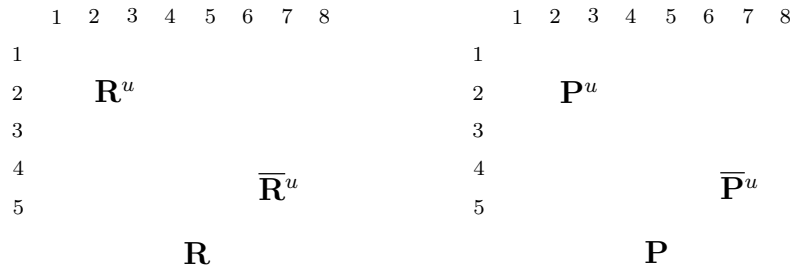

Figure 2: Example division of $\mathbf{R}$ and $\mathbf{P}$ for $\mathbf{N} = 8$, $\mathbf{m} = 5$, and $\mathbf{u} = (6,4,7,2,5)$.

$i = 1, \ldots, m$, and we let $\mathbf{P}^u$ be the collection of registers $\mathbf{P}_{i,1}, \ldots, \mathbf{P}_{i,u_i}$ for $i = 1, \ldots, m$. See Figure 2 for an example. We also let $\mathbf{R}_i$ and $\mathbf{P}_i$ denote the vectors $(\mathbf{R}_{i,1}, \ldots, \mathbf{R}_{i,N})$ and $(\mathbf{P}_{i,1}, \ldots, \mathbf{P}_{i,N})$, respectively.

Aside from reversible computations that may be described classically, the only quantum transformation used by the verifier is the Hadamard transform H that acts on a single qubit and is defined by $H : |0\rangle \mapsto (|0\rangle + |1\rangle)/\sqrt{2}$ and $H : |1\rangle \mapsto (|0\rangle - |1\rangle)/\sqrt{2}$ as usual.

We now describe the actions of the verifier and honest prover. When describing the actions of the honest prover and states of the entire system, it is assumed that the input QBF ϕ is true.

Step 1 (prover). Given a QBF ϕ and an $m \times N$ matrix R of elements in $\mathbb{F}$, let $Q(R)$ denote the corresponding matrix of correct polynomials as defined in Section 3.1. For each i, $Q(R)_{i,1}, \ldots, Q(R)_{i,N}$ is thus the sequence of polynomials the honest prover returns in the classical protocol given random field elements $R_{i,1}, \ldots, R_{i,N}$. The honest (quantum) prover prepares superposition

$$2^{-kmN/2} \sum_R |R\rangle|R\rangle|Q(R)\rangle$$

in registers $\mathbf{R}$, $\mathbf{S}$, and $\mathbf{P}$, sends $\mathbf{R}$ and $\mathbf{P}$ to the verifier, and keeps the register $\mathbf{S}$. The state of the system is now

$$2^{-kmN/2} \sum_R \left\{ |R\rangle|Q(R)\rangle \right\}_{\text{verifier}} \left\{ |R\rangle \right\}_{\text{prover}}.$$

Step 1 (verifier). The verifier rejects if $(\mathbf{R}_i, \mathbf{P}_i)$ contain an invalid proof that the input formula ϕ evaluates to true for any $i \in \{1, \ldots, m\}$ with respect to the classical protocol described in Section 3.1. This check is performed by reversibly computing the predicate E, so as not to alter superpositions of valid pairs (R, P). In the case that ϕ is true and the prover is honest the verifier never rejects on this step. The verifier now chooses $u \in \{1, \ldots, N\}^m$ uniformly at random and sends u and $\overline{\mathbf{P}}^u$ to the prover. The state of the system is now

$$2^{-kmN/2} \sum_R \left\{ |R\rangle|Q(R)^u\rangle \right\}_{\text{verifier}} \left\{ |R\rangle|\overline{Q(R)}^u\rangle \right\}_{\text{prover}}.$$

Step 2 (prover). Let $T_{i,j}$ be a unitary transformation such that $T_{i,j} : |R\rangle|0\rangle \mapsto |R\rangle|Q(R)_{i,j}\rangle$ for each i, j. Upon receiving u and $\overline{\mathbf{P}}^u$, the honest prover applies transformation $T_{i,j}^{-1}$ to $\mathbf{S}$ together with $\mathbf{P}_{i,j}$ for each pair i, j. This returns each register of $\overline{\mathbf{P}}^u$ to its initial zero value. The prover then sends $\mathbf{S}$ to the verifier. At this point the state of the system is

$$2^{-kmN/2} \sum_R \left\{ |R\rangle|R\rangle|Q(R)^u\rangle \right\}_{\text{verifier}} \left\{ |0\rangle \right\}_{\text{prover}}.$$

Here, the state $|0\rangle$ for the prover means that each register of $\overline{\mathbf{P}}^u$ contains 0, and so the prover has completely disentangled itself from the verifier's registers.

Step 2 (verifier). The verifier subtracts $\mathbf{R}_{i,j}$ from $\mathbf{S}_{i,j}$ for each i, j. In the case of the honest prover this has the effect of simply erasing $\mathbf{S}$. Next, the verifier applies the Hadamard transform to every qubit in each register of $\overline{\mathbf{R}}^u$. If $\overline{\mathbf{R}}^u$ now contains only 0 values, the verifier accepts, otherwise the verifier rejects. In short, the verifier is projecting the state of $\overline{\mathbf{R}}^u$ onto the state where $\overline{\mathbf{R}}^u$ is uniformly distributed over all possible values. It is easy to check that in the case of the honest prover the registers $\overline{\mathbf{R}}^u$ are not entangled with any other registers, as each register of $\mathbf{P}^u$ depends only on those of $\mathbf{R}^u$, and are in a uniform superposition over all possible values. Thus, the verifier accepts with certainty in this case.

This completes the description of the protocol and the proof of completeness.

3.3 Proof of soundness

Now we show that the verifier accepts with exponentially small probability in case ϕ is false, given any prover. Recall that the verifier acts as follows.

Step 1 (verifier). Receive quantum registers $\mathbf{R}$ and $\mathbf{P}$ from the prover. Reject if $(\mathbf{R}_i, \mathbf{P}_i)$ contain an invalid proof that the input formula ϕ evaluates to true for any $i \in \{1, \ldots, m\}$. Choose $u \in \{1, \ldots, N\}^m$ uniformly at random and send u and $\overline{\mathbf{P}}^u$ to the prover.

Step 2 (verifier). Receive $\mathbf{S}$ from the prover and subtract $\mathbf{R}_{i,j}$ from $\mathbf{S}_{i,j}$ for each i, j. Apply a Hadamard transform to every qubit in each register of $\overline{\mathbf{R}}^u$ and observe. If $\overline{\mathbf{R}}^u$ now contains only 0 values then accept, otherwise reject.

Let us examine the state of the prover and verifier as the protocol is executed. The prover first sends registers $\mathbf{R}$ and $\mathbf{P}$ to the verifier. The state of the system at this point may be expressed as

$$\sum_{R,P} \alpha(R,P)|R\rangle|P\rangle|\xi(R,P)\rangle, \tag{1}$$

where each $\alpha(R,P)$ is a complex number and $|\xi(R,P)\rangle$ is a normalized vector representing the state of the prover's ancilla registers, which may be entangled with $\mathbf{R}$ and $\mathbf{P}$ in any manner the prover chooses. Since the verifier rejects any pair R, P for which each (R_i, P_i) is not a valid proof that ϕ is true, we may assume the state in Eq. 1 is a superposition over such valid pairs for the purposes of bounding the probability that the verifier accepts.

The verifier chooses u randomly and sends u and $\overline{\mathbf{P}}^u$ to the prover. The prover applies some transformation to its registers (now including $\overline{\mathbf{P}}^u$), sends some register $\mathbf{S}$ to the verifier, and the verifier subtracts the contents of $\mathbf{R}$ from $\mathbf{S}$. The state of the system may now be described by

$$\sum_{R,P^u} \beta(R,u,P^u)|R\rangle|P^u\rangle|\eta(R,u,P^u)\rangle, \tag{2}$$

where each $\beta(R,u,P^u)$ is some complex number and $|\eta(R,u,P^u)\rangle$ is a normalized vector describing the state of the prover's registers as well as register $\mathbf{S}$.

The verifier now performs the Hadamard transforms on $\overline{\mathbf{R}}^u$ and observes, rejecting if any bit is not set to 0 and accepting otherwise. In order to calculate the resulting probability of acceptance, let us for now assume that u is fixed. Denote by $H^{\otimes k}$ the unitary transformation on k qubits obtained

by applying the Hadamard transform to each qubit individually. Let $l = \sum_{i=1}^{m}(N - u_i + 1)$, so that l denotes the number of registers (i.e., field elements) in $\overline{\mathbf{R}}^u$. Now, we have that the probability of acceptance is

$$\left\| \sum_{R,P^u} \beta(R,u,P^u)|R^u\rangle\langle 0 \,|\, H^{\otimes k} \,|\, R_{1,u_1}\rangle\langle 0 \,|\, H^{\otimes k} \,|\, R_{1,u_1+1}\rangle \cdots \langle 0 \,|\, H^{\otimes k} \,|\, R_{m,u_m}\rangle|P^u\rangle|\eta(R,u,P^u)\rangle \right\|^2 .$$

Noting that $\langle 0 \,|\, H^{\otimes k} \,|\, x\rangle = 2^{-k/2}$ for every $x \in \mathbb{F}$, this simplifies to

$$2^{-lk} \left\| \sum_{R,P^u} \beta(R,u,P^u)|R^u\rangle|P^u\rangle|\eta(R,u,P^u)\rangle \right\|^2 \;=\; 2^{-lk} \sum_{R^u,P^u} \left\| \sum_{\overline{R}^u} \beta(R,u,P^u)|\eta(R,u,P^u)\rangle \right\|^2 .$$

By the triangle inequality, this probability is at most

$$2^{-lk} \sum_{R^u,P^u} \left(\sum_{\overline{R}^u} |\beta(R,u,P^u)| \right)^2 . \tag{3}$$

It remains to prove an upper bound on the probability in Eq. 3 given that u is chosen uniformly at random. In order to do this, let us associate with each register $\mathbf{R}_{i,j}$ and each register $\mathbf{P}_{i,j}$ a random variable. The probability with which each random variable takes a particular value is precisely the probability that an observation of the associated register yields the given value, assuming that the observation takes place while the entire system is in the state described in Eq. 1. As we have done for registers, we may consider collections of random variables as being single random variables, abbreviated by $\mathbf{R}^u$, $\mathbf{P}^u$, etc.

Now, note that

$$|\beta(R,u,P^u)|^2 \;=\; \Pr\left[\mathbf{R} = R,\, \mathbf{P}^u = P^u\right]$$

for each R and P^u. This follows from the fact that the registers in $\mathbf{R}$ and $\mathbf{P}^u$ are never touched between the points in the protocol corresponding to the states in Eq. 1 and Eq. 2. Therefore we may rewrite the probability in Eq. 3 as

$$2^{-lk} \sum_{R^u,P^u} \left(\sum_{\overline{R}^u} \sqrt{\Pr\left[\mathbf{R} = R, \mathbf{P}^u = P^u\right]} \right)^2 . \tag{4}$$

To bound this expression, it will be helpful to consider the following definition.

Definition 1 *For any finite set S and mapping $f : S \to [0,1]$, define*

$$\theta_S(f) \;=\; \sum_{s \in S} \sqrt{f(s)}.$$

Now, for each pair R^u, P^u, define $X_{R^u,P^u} : \mathbb{F}^l \to [0,1]$ as follows:

$$X_{R^u,P^u}\left(\overline{R}^u\right) \;=\; \Pr\left[\overline{\mathbf{R}}^u = \overline{R}^u \,|\mathbf{R}^u = R^u, \mathbf{P}^u = P^u\right].$$

The probability in Eq. 4 may be written as

$$2^{-lk} \sum_{R^u,P^u} \Pr\left[\mathbf{R}^u = R^u, \mathbf{P}^u = P^u\right] \left(\theta_{\mathbb{F}^l}(X_{R^u,P^u})\right)^2.$$

We next define certain events based on the above random variables. Recall the definition of $Q(R)$ from above (i.e., $Q(R)$ is the $m \times N$ matrix of correct polynomials an honest prover answers for given R). For $1 \leq i \leq m$, $1 \leq j \leq N-1$, and R denoting the contents of $\mathbf{R}$, define $A_{i,j}$ to be the event that $\mathbf{P}_{i,j'}$ does not contain $Q(R)_{i,j'}$ for $j' \leq j$ but $\mathbf{P}_{i,j+1}$ does contain $Q(R)_{i,j+1}$. Informally, the event $A_{i,j}$ means that the prover has tried to "sneak in" a correct polynomial in register $\mathbf{P}_{i,j+1}$. Also, for $1 \leq i \leq m$, define $A_{i,N}$ to be the event that $\mathbf{P}_{i,j'}$ does not contain $Q(R)_{i,j'}$ for every j'. This event means that the prover never tries to put a correct polynomial in registers $\mathbf{P}_{i,1}, \ldots, \mathbf{P}_{i,N}$. Note that we must have $\Pr[A_{i,1} \cup \cdots \cup A_{i,N}] = 1$ for each i, as the verifier rejects in step 1 if $\mathbf{P}_{i,1}$ contains $Q(R)_{i,1}$. Finally, define the event B_u as $B_u = \bigcup_i A_{i,u_i}$. The event B_u, which we will later show is very likely to occur when u is chosen uniformly, means that the verifier has chosen some u that catches at least one of the locations where the prover is cheating, i.e., trying to "sneak in" correct polynomials. Now define mappings $Y_{R^u,P^u} : \mathbb{F}^l \to [0,1]$ and $Z_{R^u,P^u} : \mathbb{F}^l \to [0,1]$ as follows:

$$\begin{aligned} Y_{R^u,P^u}\left(\overline{R}^u\right) &= \Pr\left[\overline{\mathbf{R}}^u = \overline{R}^u \middle| \mathbf{R}^u = R^u, \mathbf{P}^u = P^u, B_u\right], \\ Z_{R^u,P^u}\left(\overline{R}^u\right) &= \Pr\left[\overline{\mathbf{R}}^u = \overline{R}^u \middle| \mathbf{R}^u = R^u, \mathbf{P}^u = P^u, \neg B_u\right], \end{aligned}$$

for events B_u and $\neg B_u$. We have

$$\theta_{\mathbb{F}^l}(X_{R^u,P^u}) = \theta_{\mathbb{F}^l}\left(\lambda_u Y_{R^u,P^u} + (1-\lambda_u) Z_{R^u,P^u}\right).$$

for $\lambda_u = \Pr\left[B_u \,|\mathbf{R}^u = R^u, \mathbf{P}^u = P^u\right]$.

Now consider the number of values of $\overline{R}^u$ for which $Y_{R^u,P^u}\left(\overline{R}^u\right) = 0$; we claim that this number is at least $\left(1 - dm2^{-k}\right) 2^{kl}$ for every R^u, P^u. This may be argued as follows. First, fix values for R^u, P^u, and i, and assume event A_{i,u_i} takes place. By the properties of the classical protocol discussed in Section 3.1, there are at most d values of R_{i,u_i} that do not cause the classical protocol to reject in this case. Thus, the number of values of $\overline{R}^u$ for which we have

$$\Pr\left[\overline{\mathbf{R}}^u = \overline{R}^u \middle| \mathbf{R}^u = R^u,\, \mathbf{P}^u = P^u,\, A_{i,u_i}\right] \neq 0$$

is at most $d\, 2^{k(l-1)}$. Since

$$\sum_{i=1}^{m} \Pr\left[\overline{\mathbf{R}}^u = \overline{R}^u \middle| \mathbf{R}^u = R^u,\, \mathbf{P}^u = P^u,\, A_{i,u_i}\right] \geq Y_{R^u,P^u}\left(\overline{R}^u\right) \geq 0,$$

the total number of values of $\overline{R}^u$ for which we have $Y_{R^u,P^u}\left(\overline{R}^u\right) \neq 0$ is at most $dm2^{k(l-1)}$.

Next, let us bound $\Pr[B_u]$ assuming u is chosen uniformly from $\{1, \ldots, N\}^m$. We have

$$\begin{aligned} \Pr[B_u] &= \sum_P \Pr[B_u|\mathbf{P} = P] \Pr[\mathbf{P} = P] \\ &= 1 - \left(1 - \frac{1}{N}\right)^m \\ &> 1 - e^{-m/N}. \end{aligned}$$

At this point, a simple lemma is required.

Lemma 1 *Let $f, g : S \to [0,1]$ satisfy $\sum_{s \in S} f(s) \leq 1$ and $\sum_{s \in S} g(s) \leq 1$, let $\lambda \in [0,1]$, and let $r = |\{s \in S | f(s) = 0\}|$. Then $\theta_S(\lambda f + (1-\lambda)g) \leq \sqrt{(1-\lambda)r} + \sqrt{|S| - r}$.*

Proof. First note that for any function $h : S \to [0,1]$ such that $\sum_{s \in S} h(s) \leq 1$, we have $\theta_S(h) \leq \sqrt{|S|}$ by the Cauchy-Schwarz inequality. Define $T = \{s \in S | f(s) = 0\}$. We have

$$\begin{aligned} \theta_S(\lambda f + (1-\lambda)g) &= \sqrt{1-\lambda}\,\theta_T(g) + \theta_{S\setminus T}(\lambda f(s) + (1-\lambda)g(s)) \\ &\leq \sqrt{(1-\lambda)|T|} + \sqrt{|S\setminus T|} \\ &= \sqrt{(1-\lambda)r} + \sqrt{|S| - r} \end{aligned}$$

as claimed. ■

We apply Lemma 1 to obtain

$$\begin{aligned} 2^{-lk} \left(\theta_{\mathbb{F}^l}\left(X_{R^u,P^u}\right)\right)^2 &\leq 2^{-lk}\left(\sqrt{(1-\lambda_u)(2^{lk} - dm2^{k(l-1)})} + \sqrt{dm2^{k(l-1)}}\right)^2 \\ &\leq 1 - \Pr\left[B_u \,|\mathbf{R}^u = R^u, \mathbf{P}^u = P^u\right]\left(1 - dm2^{-k}\right) + 2\sqrt{dm2^{-k}}. \end{aligned}$$

Hence, the probability that the verifier accepts is at most

$$2^{-lk} \sum_{R^u,P^u} \Pr\left[\mathbf{R}^u = R^u, \mathbf{P}^u = P^u\right]\left(\theta_{\mathbb{F}^l}\left(X_{R^u,P^u}\right)\right)^2 \leq 1 - \left(1 - e^{-m/N}\right)\left(1 - dm2^{-k}\right) + 2\sqrt{dm2^{-k}}.$$

By initially choosing m and k to be sufficiently fast growing polynomials in n (e.g., $m = (n+1)N$ and $k = 2n + dm + 6$), this probability may be made smaller than 2^{-n}, which completes the proof.

4 Conclusions and Open Problems

We have defined in this paper a natural quantum analogue of the notion of an interactive proof system, and proved that there exist three-message quantum interactive proof systems with exponentially small error for any PSPACE language.

Currently the best upper bound known on the power of quantum interactive proof systems is that any language having a quantum interactive proof system is in EXP [23]. We do not know if EXP has constant-round quantum interactive proofs, if quantum interactive proofs characterize PSPACE, or if the class of languages having quantum interactive proofs lies strictly between PSPACE and EXP. Another interesting class is the class of languages having two-message quantum interactive proof systems. How does this class relate to PSPACE, for instance?

Several variants on interactive proof systems have been studied, such as multi-prover interactive proofs [5, 9, 13, 18, 20], probabilistically checkable proofs [3, 20], and interactive proof systems having verifiers with very limited computing power [14, 17]. How do quantum analogues of these models compare with their classical counterparts?

Acknowledgments

I would like to thank Gilles Brassard, Anne Condon, Christiane Lemieux, and the anonymous referees for many helpful suggestions.

References

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

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

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

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

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

[6] 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.

[7] A. Barenco. A universal two-bit gate for quantum computation. *Proceedings of the Royal Society of London*, 449:679–683, 1995.

[8] A. Barenco, C. H. Bennett, R. Cleve, D. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. Smolin, and H. Weinfurter. Elementary gates for quantum computation. *Physical Review A*, 52:3457–3467, 1995.

[9] M. Ben-Or, S. Goldwasser, J. Kilian, and A. Wigderson. Multi-prover interactive proofs: how to remove intractability assumptions. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, pages 113–131, 1988.

[10] C. H. Bennett and G. Brassard. Quantum cryptography: Public key distribution and coin tossing. In *Proceedings of the IEEE International Conference on Computers, Systems, and Signal Processing*, pages 175–179, 1984.

[11] A. Berthiaume. Quantum computation. In L. Hemaspaandra and A. Selman, editors, *Complexity Theory Retrospective II*, pages 23–50. Springer, 1997.

[12] E. Biham, M. Boyer, P. O. Boykin, T. More, and V. Roychowdhury. A proof of the security of quantum key distribution. In *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing*, pages 715–724, 2000.

[13] J. Cai, A. Condon, and R. Lipton. PSPACE is provable by two provers in one round. *Journal of Computer and System Sciences*, 48(1):183–193, 1994.

[14] A. Condon and R. Ladner. Interactive proof systems with polynomially bounded strategies. *Journal of Computer and System Sciences*, 50(3):506–518, 1995.

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

[16] D. DiVincenzo. Two-bit gates are universal for quantum computation. *Physical Review A*, 50:1015–1022, 1995.

[17] C. Dwork and L. Stockmeyer. Finite state verifiers I: the power of interaction. *Journal of the ACM*, 39(4):800–828, 1992.

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

[19] P. Feldman. The optimum prover lies in PSPACE. Manuscript, M.I.T., 1986.

[20] L. Fortnow, J. Rompel, and M. Sipser. On the power of multi-prover interactive protocols. *Theoretical Computer Science*, 134:545–557, 1994.

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

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

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

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

[25] D. Mayers. Unconditional security in quantum cryptography. Los Alamos Preprint Archive, quant-ph/9802025, 1998.

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

[27] R. Raz. Exponential separation of quantum and classical communication complexity. In *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing*, pages 358–376, 1999.

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

[29] A. Shen. IP = PSPACE: simplified proof. *Journal of the ACM*, 39(4):878–880, 1992.

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

[31] P. Shor and J. Preskill. Simple proof of security of the BB84 quantum key distribution protocol. Los Alamos Preprint Archive, quant-ph/0003004, 2000.

[32] V. Shoup. New algorithms for finding irreducible polynomials over finite fields. *Mathematics of Computation*, 54:435–447, 1990.

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