PSPACE has constant-round quantum interactive proof systems

John Watrous*

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

A number of 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 [29] 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 secure under a wide variety of attacks [23, 24], and Raz [26] 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 [19]. An interactive proof system consists of an interaction between a computationally unbounded prover and a polynomial-time 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 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.

Based on the work of Lund, Fortnow, Karloff, and Nisan [22], Shamir [27] proved that every language in PSPACE has an interactive proof system. Since any language having an interactive proof system is in PSPACE [25], this implies $\mathrm{IP} = \mathrm{PSPACE}$. 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, 20], 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 $\mathrm{AM} = \mathrm{PSPACE}$.

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 constant: define the verifier so that it first generates all of its random numbers, sends them all to the prover in one round (or in a constant number of rounds), receives all the responses from the prover, and

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

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 aforementioned 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.

The prover first sends a superposition of sequences of random numbers and corresponding responses to the verifier, and the verifier checks that the responses are valid according to a classical protocol for PSPACE. (It will be shown that the prover cannot cheat by giving the verifier a superposition that is biased towards certain random sequences—the verifier will be able to later check that the superposition is close to uniform.) The verifier then chooses randomly one of the positions in the list of random numbers and responses, sends back to the prover its responses starting at this position in the list and challenges the prover to invert the computation it performed to obtain these responses. Let us say that the random numbers and responses up to the chosen position in the list have *low-index*, and the remaining random numbers and responses have *high-index*. The low-index responses, which were not sent back to the prover in the second round, should now depend only on the low-index random numbers (for otherwise the prover has cheated). The verifier may now check that the superposition of high-index random numbers is uniform by performing an appropriately defined measurement. However, if the prover has cheated by basing its low-index responses on high-index random numbers, the low-index responses and high-index random numbers will be entangled in a manner detectable by the verifier; with high probability, the high-index random numbers will fail the uniformity test.

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 2 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 3-message quantum protocol to require only 2 messages, and it is an interesting open question whether this is possible in general. On the other hand, and similar to the classical case, it seems quite unlikely that 1-message quantum interactive proof systems are as powerful as 3-message quantum interactive proof systems: any language in "quantum MA" is contained in PP [21].

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 3-message quantum interactive proof system for the quantified boolean formula problem and proving its correctness. We conclude with Section 4, which mention 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. 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 [31] and Berthiaume [11], 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 a number of papers [1, 7, 8, 14, 15]; for the purposes of this paper, we will assume only that this set includes the Hadamard gate and any universal gate 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. 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 here $k' = \lfloor m/2 + 1/2 \rfloor$). 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 (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 obey the laws of physics! This is en-

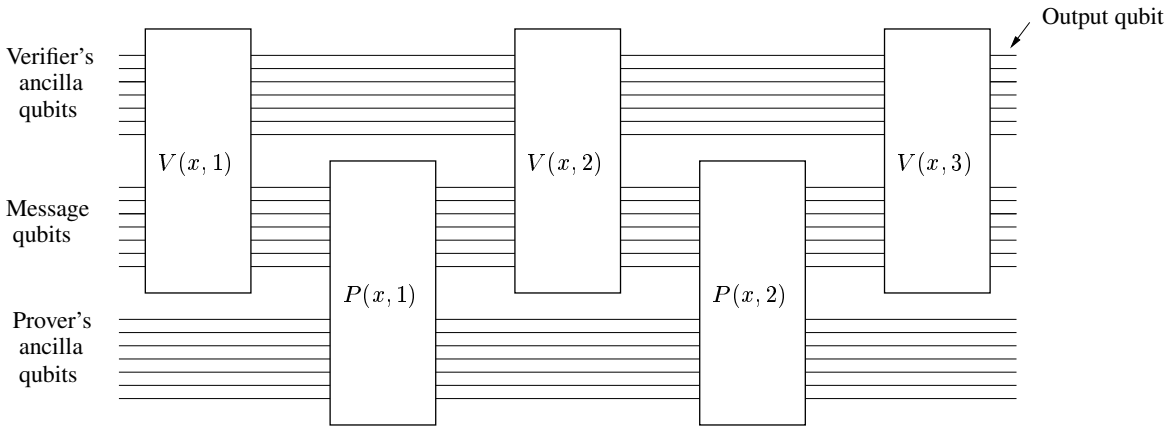

Figure 1. Quantum circuit for a 4-message quantum interactive proof system

forced by requiring that the prover's actions correspond to 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 a number of 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, for example. 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 the QBF problem

Recall that a quantified boolean formula is a formula of the form $Q_1 x_1 \cdots Q_n x_n B(x_1, \ldots, x_n)$, where each Q_i is an existential or universal quantifier ($\exists$ or $\forall$) and $B(x_1, \ldots, x_n)$ 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 3-message quantum interactive proof system with exponentially small error for the QBF problem. This is because a verifier (and any honest prover) 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 3-message quantum interactive proof system for the QBF problem is based on a variant of the Lund–Fortnow–Karloff–Nisan protocol due to Shen [28], 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 $Q = Q_1 x_1 \cdots Q_n x_n B(x_1, \ldots, x_n)$ be a fixed input formula. Also let $\mathbb{F}$ be a finite field, write $N = \binom{n+1}{2} + n$, and let d be the length of Q (with a slight modification of the protocol, $d = 3$ is sufficient). The protocol is as follows.

For $j = 1, \ldots, N-1$, the prover sends the verifier a polynomial f_j over $\mathbb{F}$ of degree at most d, and the verifier chooses $r_j \in \mathbb{F}$ and sends r_j to the prover. The prover then sends a polynomial f_N to the verifier in the final round, and the verifier chooses $r_N \in \mathbb{F}$ (there is no need for r_N to be sent to the prover). The verifier then evaluates a particular polynomial-time predicate $E(Q, r_1, \ldots, r_N, f_1, \ldots, f_N)$ 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 numbers $r_1, \ldots, r_N \in \mathbb{F}$ there exist polynomials $c_1, \ldots, c_N$, where each polynomial c_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 Q, satisfy the following properties:

1. If Q is true, then $E(Q, r_1, \ldots, r_N, c_1, \ldots, c_N)$ is true for all $r_1, \ldots, r_N$.
2. If Q is false, then for $r_1, \ldots, r_N$ and $f_2, \ldots, f_N$, $E(Q, r_1, \ldots, r_N, c_1, f_2, \ldots, f_N)$ is false.
3. If Q is false, then for all $k \leq N-1$, $r_1, \ldots, r_{k-1}$, and $f_1, \ldots, f_k$, the following holds in case $f_k \neq c_k$: there are at most d values of r_k for which there exist $r_{k+1}, \ldots, r_N$ and $f_{k+2}, \ldots, f_N$ such that $E(Q, r_1, \ldots, r_N, f_1, \ldots, f_k, c_{k+1}, f_{k+2}, \ldots, f_N)$ is true.
4. If Q is false, then for any $r_1, \ldots, r_{N-1}$ and polynomials $f_1, \ldots, f_N$ such that $f_N \neq c_N$, there are at most d values of r_N for which $E(Q, r_1, \ldots, r_N, f_1, \ldots, f_N)$ is true.

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

Clearly, if Q evaluates to true, an honest prover can always convince the verifier to accept by sending the correct polynomials $c_1, \ldots, c_N$ corresponding to the verifiers random numbers $r_1, \ldots, r_{N-1}$.

Now suppose that Q evaluates to false. By item 2, a cheating prover cannot send the correct polynomial c_1 on the first round, for the prover rejects with certainty in this case. Hence the prover must send $f_1 \neq c_1$ 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 $f_1 \neq c_1, \ldots, f_k \neq c_k$ during rounds $1, \ldots, k$. Unless the verifier randomly chooses one of d particular values for r_k, the prover may not send c_{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 c_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 (hence yielding exponentially small probability of error). For any chosen k, the verifier (and honest prover) may use a deterministic procedure to implement arithmetic in $\mathbb{F}$—specifically, compute an irreducible polynomial g of degree k over $GF(2)$ in deterministic polynomial time [30], identify elements of $\mathbb{F}$ with polynomials over $GF(2)$ of degree at most $k-1$, and take arithmetic 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 for QBF

We now describe the quantum verifier's protocol for our 3-message quantum interactive proof system for the QBF problem.

We use the following conventions when describing the quantum circuits corresponding to the verifier's actions. 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 $\mathbf{F}_{i,j}$ 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 below. 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 $\mathbb{F}$ as described above. Each $\mathbf{F}_{i,j}$ 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 $\mathbb{F}$. The verifier may also use any polynomial number of additional ancilla qubits in order to perform the transformations described. In addition, the verifier may 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 (although there is no difference in the behavior of the protocol if they are thought of as being stored in quantum registers).

The error probability of the protocol will depend on m and k as described below in Section 3.3—we may take m and k to be fixed polynomials in the input size to obtain exponentially small error.

It will be convenient to refer to certain collections of the quantum registers mentioned above; for a given vector

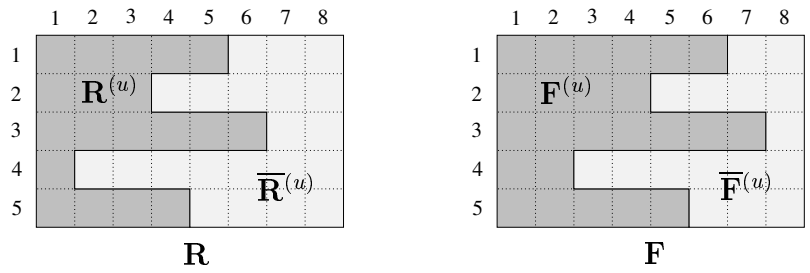

Figure 2. Example division of $\mathbf{R}$ and $\mathbf{F}$ for N = 8, m = 5, and u = (6,4,7,2,5).

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

The verifier's protocol is as follows.

Verifier's protocol for the QBF problem

1. Receive quantum registers $\mathbf{R}$ and $\mathbf{F}$ from the prover. Reject if $(\mathbf{R}_i, \mathbf{F}_i)$ contain an invalid proof that the input formula Q evaluates to true for any $i \in \{1, \ldots, m\}$.
2. Choose $u \in \{1, \ldots, N\}^m$ uniformly at random and send u and $\overline{\mathbf{F}}^{(u)}$ to the prover.
3. Receive $\mathbf{S}$ from the prover and subtract $\mathbf{R}_{i,j}$ from $\mathbf{S}_{i,j}$ for each i, j.
4. Apply transformation $H^{\otimes k}$ to each register of $\overline{\mathbf{R}}^{(u)}$. If $\overline{\mathbf{R}}^{(u)}$ now contains only 0 values, then accept, otherwise reject.

The check in step 1 refers to the classical protocol described in Section 3.1. Naturally this check is performed by reversibly computing the predicate E, so as not to alter superpositions of valid pairs (R, F). The transformation $H^{\otimes k}$ in step 4 is the Hadamard transform H applied to each qubit of the register in question, with H 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. The random choice of the vector u in step 2 can be simulated efficiently with negligible error using the Hadamard transform appropriately. (This negligible error will not change the fact that the protocol has one-sided error.)

3.3. Proof of correctness

We now prove that the above protocol is correct. First we show that there exists an honest prover P such that (P, V) accepts with certainty whenever the input formula Q evaluates to true.

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

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

in registers $\mathbf{R}$, $\mathbf{S}$, and $\mathbf{F}$ and sends $\mathbf{R}$ and $\mathbf{F}$ to the verifier (and keeps the register $\mathbf{S}$). Under the assumption Q is true, each pair (R_i, F_i) the verifier receives is a valid pair with respect to the classical protocol, so the verifier will not reject in step 1.

The behavior of the honest prover in the second round is as follows. For each i, j, let $T_{i,j}$ be a unitary transformation such that $T_{i,j} : |R\rangle|0\rangle \mapsto |R\rangle|C(R)_{i,j}\rangle$. Upon receiving u and $\overline{\mathbf{F}}^{(u)}$ in the second round, the prover applies transformation $T_{i,j}^{-1}$ to $\mathbf{S}$ together with $\mathbf{F}_{i,j}$ for each appropriate pair i, j. This returns each register of $\overline{\mathbf{F}}^{(u)}$ to its initial zero value. The prover then sends $\mathbf{S}$ to the verifier. It may be checked that after subtracting each $\mathbf{R}_{i,j}$ from $\mathbf{S}_{i,j}$, the registers $\overline{\mathbf{R}}^{(u)}$ will not be entangled with any other registers (as each register of $\mathbf{F}^{(u)}$ depends only on those of $\mathbf{R}^{(u)}$), and are in a uniform superposition over all possible values. Thus, each register of $\overline{\mathbf{R}}^{(u)}$ is put into state 0 during step 4, and hence the verifier accepts with certainty.

Now we show that the verifier accepts with exponentially small probability in case Q is false, given any prover. We begin by examining the total state of the prover and verifier as the protocol is executed. In step 1 the prover sends registers $\mathbf{R}$ and $\mathbf{F}$ to the verifier. The state of the system at this point may be expressed as

$$|\psi\rangle \;=\; \sum_{R,F} \alpha(R, F)|R\rangle|F\rangle|\xi(R, F)\rangle,$$

where each $\alpha(R, F)$ is a complex number and $|\xi(R, F)\rangle$ is a normalized vector representing the state of the prover's ancilla registers (which may be entangled with $\mathbf{R}$ and $\mathbf{F}$ in any manner the prover chooses). Since the verifier rejects

any pair R, F for which each (R_i, F_i) is not a valid proof that Q is true, we may assume $|\psi\rangle$ is a superposition over such valid pairs for the purposes of bounding the probability that the verifier accepts.

At this point, let us associate with each register $\mathbf{R}_{i,j}$ and each register $\mathbf{F}_{i,j}$ 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 state $|\psi\rangle$. As we have done above for registers, we may consider collections of random variables as being single random variables, abbreviated by $\mathbf{R}^{(u)}$, $\mathbf{F}^{(u)}$, etc. For example,

$$\Pr[\mathbf{R} = R,\, \mathbf{F}^{(u)} = F^{(u)}] \;=\; \sum_{\overline{F}^{(u)}} |\alpha(R, F)|^2.$$

We also define a number of events based on these random variables. Recall the definition of $C(R)$ from above (i.e., $C(R)$ is the $m \times N$ matrix of correct polynomials an honest prover answers for given R). For $1 \leq i \leq m$ and $1 \leq j \leq N-1$, define $A_{i,j}$ to be the event that $\mathbf{F}_{i,j'}$ does not contain $C(R)_{i,j'}$ for $j' \leq j$ and $\mathbf{F}_{i,j+1}$ does contain $C(R)_{i,j+1}$, for R denoting the contents of $\mathbf{R}$. For $1 \leq i \leq m$, define $A_{i,N}$ to be the event that $\mathbf{F}_{i,j'}$ does not contain $C(R)_{i,j'}$ for every j'. Note that we must have $\Pr[A_{i,1} \cup \cdots \cup A_{i,N}] = 1$ for each i, as the verifier surely rejects in step 1 if $\mathbf{F}_{i,1}$ contains $C(R)_{i,1}$. Finally, for each $v \in \{1, \ldots, N\}^m$ define events B_v and D_v as $B_v = \bigcup_i A_{i,v_i}$ and $D_v = \bigcap_i A_{i,v_i}$.

In step 2 the verifier chooses u randomly and sends u and $\overline{\mathbf{F}}^{(u)}$ to the prover. The prover applies some transformation to its registers (now including $\overline{\mathbf{F}}^{(u)}$), sends some register $\mathbf{S}$ to the verifier, and the verifier subtracts the contents of $\mathbf{R}$ from $\mathbf{S}$ in step 3. The state of the system may now be described by

$$\sum_{R, F^{(u)}} \beta(R, u, F^{(u)}) |R\rangle |F^{(u)}\rangle |\eta(R, u, F^{(u)})\rangle,$$

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

$$\left|\beta(R, u, F^{(u)})\right|^2 \;=\; \Pr\left[\mathbf{R} = R,\, \mathbf{F}^{(u)} = F^{(u)}\right] \qquad (1)$$

for each R and $F^{(u)}$, as the actions of the prover and verifier have not modified the contents of registers $\mathbf{R}$ and $\mathbf{F}^{(u)}$ during these steps.

The verifier now executes step 4. Assuming for now that u is fixed, this results in acceptance with probability

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

where $l = \sum_{i=1}^{m} (N - u_i + 1)$ denotes the number of registers to which $H^{\otimes k}$ was applied. This follows from the fact that $\langle 0 \,|\, H^{\otimes k} \,|\, R_{i,j} \rangle = 2^{-k/2}$ for each value $R_{i,j}$. By the triangle inequality, this probability is at most

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

We now derive an upper bound on (2) by considering the random variables defined above. To this end, consider the definition and lemma that follow.

Definition 1 *For any nonempty, finite set $T \subseteq S$ and mapping $f : S \to \mathbb{R}^+$, define*

$$\theta_T(f) \;=\; \frac{1}{|T|} \left(\sum_{s \in T} \sqrt{f(s)} \right)^2.$$

Lemma 1 *Let $f, g : S \to \mathbb{R}^+$ satisfy $\sum_{s \in S} f(s) \leq 1$ and $\sum_{s \in S} g(s) \leq 1$, let $r = |\{s \in S | f(s) = 0\}|/|S|$, and let $\lambda \in [0, 1]$. Then $\theta_S(\lambda f + (1-\lambda) g) \leq 1 - \lambda r + 2\sqrt{1-r}$.*

Proof. First note that for any function $h : S \to \mathbb{R}^+$ and subset $T \subseteq S$ such that $\sum_{s \in T} h(s) \leq 1$, we have

$$\sum_{s \in T} \sqrt{h(s)} \;\leq\; \sqrt{|T|} \sqrt{\sum_{s \in T} h(s)}$$

by the Cauchy-Schwarz inequality, and thus $\theta_T(h) \leq 1$. Now define $S' = \{s \in S | f(s) = 0\}$. We have

$$\begin{aligned}
&\sqrt{\theta_S(\lambda f + (1-\lambda) g)} \\
&\quad= \frac{1}{\sqrt{|S|}} \sum_{s \in S} \sqrt{\lambda f(s) + (1-\lambda) g(s)} \\
&\quad= \frac{\sqrt{(1-\lambda) r}}{\sqrt{|S'|}} \sum_{s \in S'} \sqrt{g(s)} \\
&\qquad + \frac{\sqrt{1-r}}{\sqrt{|S \setminus S'|}} \sum_{s \in S \setminus S'} \sqrt{\lambda f(s) + (1-\lambda) g(s)} \\
&\quad= \sqrt{(1-\lambda) r} \sqrt{\theta_{S'}(g)} \\
&\qquad + \sqrt{1-r} \sqrt{\theta_{S \setminus S'}(\lambda f + (1-\lambda) g)} \\
&\quad\leq \sqrt{(1-\lambda) r} + \sqrt{1-r}.
\end{aligned}$$

Thus

$$\begin{aligned}
&\theta_S(\lambda f + (1-\lambda) g) \\
&\qquad\leq 1 - \lambda r + 2\sqrt{(1-\lambda) r (1-r)} \\
&\qquad\leq 1 - \lambda r + 2\sqrt{1-r}
\end{aligned}$$

as claimed. ∎

By (1), we may rewrite (2) as

$$2^{-lk} \sum_{R^{(u)},F^{(u)}} \left(\sum_{\overline{R}^{(u)}} \sqrt{\Pr\left[\mathbf{R} = R, \mathbf{F}^{(u)} = F^{(u)}\right]} \right)^2. \tag{3}$$

For each pair $R^{(u)}, F^{(u)}$, define $X_{R^{(u)},F^{(u)}} : \mathbb{F}^l \rightarrow [0,1]$ as follows:

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

The probability in (3) may be written as

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

Next, define mappings $Y_{R^{(u)},F^{(u)}} : \mathbb{F}^l \rightarrow [0,1]$ and $Z_{R^{(u)},F^{(u)}} : \mathbb{F}^l \rightarrow [0,1]$ as follows:

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

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

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

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

Now consider the number of values of $\overline{R}^{(u)}$ for which $Y_{R^{(u)},F^{(u)}}\left(\overline{R}^{(u)}\right) = 0$; we claim that this number is at least $\left(1 - dm2^{-k}\right) 2^{kl}$ for every $R^{(u)}, F^{(u)}$. This may be argued as follows. First, fix values for $R^{(u)}$, $F^{(u)}$, 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

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

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

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

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

Now we may apply Lemma 1 to obtain

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

and hence

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

It remains to bound the above quantity, given that u is chosen uniformly from $\{1, \ldots, N\}^m$. Let U denote the random variable corresponding to the verifier's choice of u. We bound $\Pr[B_U]$ by conditioning on the events D_v that describe the exact places where the prover tries to "sneak in" the correct polynomials. Specifically, we have

$$\begin{aligned} \Pr[B_U] &= \sum_u \Pr[B_u] \Pr[U = u] \\ &= N^{-m} \sum_{u,v} \Pr[B_u | D_v] \Pr[D_v] \\ &= N^{-m} \sum_v \left(N^m - (N-1)^m\right) \Pr[D_v] \\ &= 1 - \left(1 - \frac{1}{N}\right)^m \\ &> 1 - e^{-m/N}. \end{aligned}$$

Thus, the overall probability that the verifier accepts is at most

$$1 - \left(1 - e^{-m/N}\right)\left(1 - dm2^{-k}\right) + 2\sqrt{dm2^{-k}}.$$

By initially choosing m and k to be sufficiently fast growing polynomials in the input size $|x|$ (e.g., $m = (|x|+1)N$ and $k = 2|x| + 6 + \lceil \log(dm) \rceil$), this probability may be made smaller than $2^{-|x|}$, 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 3-message quantum interactive proof systems with exponentially small error for any PSPACE language. We do not know if constant-round quantum interactive proofs characterize PSPACE, or if there are such proof systems for (presumably) larger classes (e.g., does EXP have constant-round quantum interactive proofs?).

Several variants on interactive proof systems have been studied, such as multi-prover interactive proofs [5, 9, 12, 17, 18], probabilistically checkable proofs [3, 18], and interactive proof systems having verifiers with very limited computing power [13, 16]. How do quantum analogues of these models compare with their classical counterparts?

Acknowledgments

I would like to thank Gilles Brassard, Anne Condon, and Christiane Lemieux for providing a number of very helpful suggestions regarding this paper.

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 Letters 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] 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.

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

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

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

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

[17] 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.

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

[19] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1):186–208, 1989. Preliminary version appeared in *Proceedings of the Eighteenth Annual ACM Symposium on Theory of Computing*, pages 291–304, 1985.

[20] 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.

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

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

[23] D. Mayers. Quantum key distribution and string oblivious transfer in noisy channels. In *Advances in Cryptology: Proceedings of Crypto'96*, volume 1109 of *Lecture Notes in Computer Science*, pages 343–357. Springer-Verlag, 1996.

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

[25] C. Papadimitriou. Games against nature. In *Proceedings of the 24th Annual Symposium on Foundations of Computer Science*, pages 446–450, 1983.

[26] 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.

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

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

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

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

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