Electronic Colloquium on Computational Complexity, Report No. 110 (2004)

An Application of Quantum Finite Automata to Interactive Proof Systems *

Harumichi Nishimura   Tomoyuki Yamakami

Computer Science Program, Trent University
Peterborough, Ontario, Canada K9J 7B8

Abstract: Quantum finite automata have been studied intensively since their introduction in late 1990s as a natural model of a quantum computer with finite-dimensional quantum memory space. This paper seeks their direct application to interactive proof systems in which a mighty quantum prover communicates with a quantum-automaton verifier through a common communication cell. Our quantum interactive proof systems are juxtaposed to Dwork-Stockmeyer's classical interactive proof systems whose verifiers are two-way probabilistic automata. We demonstrate strengths and weaknesses of our systems and further study how various restrictions on the behaviors of quantum-automaton verifiers affect the power of quantum interactive proof systems.

Keywords: quantum finite automaton, quantum interactive proof system, quantum measurement, quantum circuit

1 Development of Quantum Finite Automata

A quantum computer—quantum-mechanical computing device—has drawn wide attention as a future computing paradigm since the pioneering work of Feynman [20], Deutsch [16], and Benioff [9] in the 1980s. Over the decades, such a device has been mathematically modeled in numerous ways to deliver a coherent theory of quantum computation. Of all computational models, Moore and Crutchfield [34] as well as Kondacs and Watrous [32] proposed a (one-head) *quantum finite automaton* (*qfa*, in short) as a simple but natural model of a quantum computer that is equipped with finite-dimensional quantum memory space†. Parallel to classical automata theory, the theory of quantum finite automata has been well established to study the nature of quantum computation. Performing a series of unitary operations as its tape head scans input symbols, a qfa may eventually enter accepting or rejecting inner states to halt. Any entry of such a unitary operation is a complex number, called a *(transition) amplitude*. A quantum computation is seen as an evolution of a quantum superposition of the machine's configurations, where a configuration is a pair of an inner state and a head position of the machine. As quantum physics dictates, a quantum evolution is reversible in nature. A special operation called a *(quantum) measurement* is performed to "observe" whether the qfa enters an accepting inner state, a rejecting inner state, or a non-halting inner state. Of all the variations of qfa's discussed in the past literature, we shall focus our study only on the early models of Moore and Crutchfield and of Kondacs and Watrous for our application to interactive proof systems.

In 1997, Kondacs and Watrous [32] introduced two types of qfa's: a *1-way quantum finite automaton* (*1qfa*, in short) whose head always moves rightward and a *2-way quantum finite automaton* (*2qfa*, in short) whose head moves in all directions. Both qfa's perform a so-called projection measurement (or von Neumann measurement) after every move of them. Because of a finite memory constraint, no 1qfa recognizes even the regular language $Zero = \{x0 \mid x \in \{0,1\}^*\}$ with small error probability [32]. In the model of Moore and Crutchfield, on the contrary, a 1qfa performs a measurement only once after the tape head scans the right endmarker. Their model is often referred to as a *measure-once 1-way quantum finite automaton* (*mo-1qfa*, in short). The qfa model of Kondacs and Watrous is by contrast called a *measure-many 1-way quantum finite automaton*. As Brodsky and Pippenger [11] showed, mo-1qfa's are so restrictive that they are fundamentally equivalent in

*An extended abstract appeared in the Proceedings of the 9th International Conference on Implementation and Application of Automata, Lecture Notes in Computer Science, Springer-Verlag, Kingston, Canada, July 22–24, 2004. This work was in part supported by the Natural Sciences and Engineering Research Council of Canada.

†The tape head of a quantum finite automaton may exist in a superposition.

ISSN 1433-8092

power to "permutation" automata, which recognize exactly group languages. Unlike the 1qfa's, 2qfa's can simulate deterministic finite automata with probability 1. Moreover, Kondacs and Watrous [32] constructed a 2qfa that recognizes with small error probability the non-regular language $Upal = \{0^n1^n \mid n \geq 0\}$ (unique palindromes) in worst-case linear time by exploiting its quantum superposition. The power of a qfa may vary in general depending on the types of restrictions imposed on its behaviors: for instance, head move, measurement, quantum state, and so forth.

We are particularly interested in a qfa whose error probability is bounded above by a certain constant $\epsilon \in [0, 1/2)$ independent of input lengths. Such a qfa is conventionally called *bounded error*. We use the notation 1QFA (2QFA, resp.) to denote the class of all languages recognized by bounded-error 1qfa's (2qfa's, resp.) with arbitrary complex amplitudes. Similarly, let MO-1QFA be the class of all languages recognized by bounded-error mo-1qfa's. When the running time of a qfa is an issue, we use the notation 2QFA(*poly-time*) to denote the collection of all languages recognized by expected polynomial-time 2qfa's with bounded error, where an *expected polynomial-time 2qfa* is a 2qfa whose average running time on each input of length n is bounded above by a fixed polynomial in n. When all amplitudes are drawn from a designated amplitude set K, we emphatically write $2\mathrm{QFA}_K$ and $2\mathrm{QFA}_K$(*poly-time*). For comparison, we write REG for the class of all regular languages. Our current state of knowledge is summarized as follows: 1QFA $\subsetneq$ REG $\subsetneq$ 2QFA(*poly-time*) $\subseteq$ 2QFA. How powerful is 2QFA? It directly follows from [42] that any 2qfa with $\mathbb{A}$-amplitudes[‡] can be simulated by a probabilistic Turing machine (PTM, in short) using space $O(\log n)$ with unbounded error. Since any unbounded-error $s(n)$-space PTM can be simulated deterministically in time $2^{O(s(n))}$ [10], we conclude that $2\mathrm{QFA}_{\mathbb{A}} \subseteq \mathrm{P}$. For an overview of qfa's, see the textbook, e.g., [24].

In this paper, we seek a direct application of qfa's to an interactive proof system, which can be viewed as a two-player game between the players called a prover and a verifier. In our basic model, a qfa plays a role of a verifier and a prover can apply any operation that quantum physics allows. Such a system is generally called a weak-verifier quantum interactive proof system. We further place various restrictions on our basic model and study how such restrictions affect its computational power. In the following section, we take a quick tour of the notion of interactive proof systems as an introduction to our formalism of quantum interactive proof systems with qfa verifiers.

2 Basics of Interactive Proof Systems

In mid 1980s, Goldwasser, Micali, and Rackoff [21] and independently Babai [8] introduced the notion of a so-called (single-prover) *interactive proof system* (*IP system*, in short), which can be viewed as a two-player game in which a player P, called a *prover*, who has unlimited computational power tries to convince or fool the other player V, called a *verifier*, who runs a randomized algorithm. These two players can access a given input and share a common communication bulletin board on which they can communicate with each other by posting their messages in turn. The goal of the verifier is to decide whether the input is in a given language L with designated accuracy. We say that *L has an IP system (P, V)* (or *an IP system (P, V) recognizes L*) if there exists an error bound $\epsilon \in [0, 1/2)$ such that the following two conditions hold: (1) if the input x belongs to L, then the "honest" prover P convinces the verifier V to accept x with probability $\geq 1 - \epsilon$ and (2) if the input x is not in L, then the verifier V rejects x with probability $\geq 1 - \epsilon$ although it plays against any "dishonest" prover. Because of their close connection to cryptography, program checking, and list decoding, the IP systems have become one of the major research topics in computational complexity theory.

When a verifier is a polynomial-time PTM, Shamir [38] proved that the corresponding IP systems exactly characterize the complexity class PSPACE based on the work of Lund, Fortnow, Karloff, and Nisan [33] and on the result of Papadimitriou [37]. This demonstrates the power of interactions between mighty provers and polynomial-time PTM verifiers.

The major difference between the models of Goldwasser et al. [21] and of Babai [8] is the amount of the verifier's private information that is revealed to a prover. Goldwasser et al. considered the IP systems whose verifiers can hide his probabilistic moves from provers to prevent any malicious attack of the provers. Babai considered by contrast the IP systems in which verifiers' moves are completely revealed to provers. Although he named his IP system an *Arthur-Merlin game*, it is also known as an IP system with "public coins." Despite the difference of the models, Goldwasser and Sipser [22] later proved that the classes of all languages recognized by both IP systems with polynomial-time PTM verifiers coincide.

[‡]The set $\mathbb{A}$ consists of all algebraic complex numbers.

In early 1990s, Dwork and Stockmeyer [17] focused their research on IP systems with weak verifiers, particularly, bounded-error *2-way probabilistic finite automaton* (*2pfa*, in short) verifiers that may "privately" flip fair coins. Their research inspires us to apply quantum finite automata to interactive proof systems. For later use, let IP($2pfa$) be the class of all languages recognized by IP systems with 2pfa verifiers and let IP($2pfa$, *poly-time*) be the subclass of IP($2pfa$) where the verifiers run in expected polynomial time. When the verifiers flip only "public coins," we write AM($2pfa$) and AM($2pfa$, *poly-time*) instead. Dwork and Stockmeyer showed without any unproven assumption that the IP systems with 2pfa verifiers are more powerful than 2pfa's alone (which are viewed as IP systems without any prover). Moreover, they showed that the nonregular language $Pal = \{x \in \{0,1\}^* \mid x = x^R\}$ (palindromes), where x^R is x in the reverse order, separates IP($2pfa$, *poly-time*) from AM($2pfa$) and the language $Center = \{x1y \mid x, y \in \{0,1\}^*, |x| = |y|\}$ separates AM($2pfa$) from AM($2pfa$, *poly-time*). The IP systems of Dwork and Stockmeyer can be seen as a special case of a much broader concept of space-bounded IP systems. For their overview, the reader may refer to [13].

Recently, a quantum analogue of an IP system was introduced by Watrous [43] under the term (single-prover) *quantum interactive proof system* (*QIP system*, in short). The QIP systems with uniform polynomial-size quantum-circuit verifiers exhibit significant computational power of recognizing every language in PSPACE by exchanging only three messages between a prover and a verifier [28, 43]. The study of QIP systems, including their variants (such as multi-prover model [12, 30] and zero-knowledge model [29, 41]), has become a major topic in quantum complexity theory. In particular, quantum analogues of Babai's Merlin-Arthur games, called *quantum Merlin-Arthur games*, have drawn significant attention (e.g., [1, 2, 31, 40, 45]).

Motivated by the work of Dwork and Stockmeyer [17], this paper introduces a QIP system whose verifier is especially a qfa. In the subsequent sections, we give the formal definition of our basic QIP systems and explore their properties and relationships to the classical IP systems of Dwork and Stockmeyer.

3 Application of QFAs to QIP Systems

Following the success of IP systems with 2pfa verifiers, we wish to apply qfa's to QIP systems. A purpose of our study is to examine the power of "interaction" when a weak verifier, represented by a qfa, meets with a mighty prover. The main goal of our study is (i) to investigate the roles of the interactions between a prover and a weak verifier, (ii) to understand the influence of various restrictions and extensions of QIP systems, and (iii) to study the QIP systems under a broader but general framework. In addition, when the power of verifiers is limited, we may possibly prove without any unproven assumption the separations and collapses of certain complexity classes defined by QIP systems with such weak verifiers.

Throughout this paper, let $\mathbb{Q}$ and $\mathbb{C}$ respectively denote the sets of all rational numbers and of all complex numbers. Let $\mathbb{N}$ be the set of all natural numbers (i.e., nonnegative integers) and set $\mathbb{N}^+ = \mathbb{N} - \{0\}$. For any two integers m and n with $m < n$, the notation $[m, n]_{\mathbb{Z}}$ denotes the set $\{m, m+1, m+2, \ldots, n\}$ and $\mathbb{Z}_n$ in particular denotes the set $[0, n-1]_{\mathbb{Z}}$. All *logarithms* are to base 2 and all *polynomials* have integer coefficients. By $\tilde{\mathbb{C}}$, we denote the set of all polynomial-time approximable complex numbers, where a complex number is called *polynomial-time approximable* if its real part and imaginary part are both deterministically approximated to within 2^{-n} in polynomial time. Our input alphabet Σ is an arbitrary finite set, not necessarily limited to $\{0,1\}$. Following the convention, we write $\Sigma^n = \{x \in \Sigma^* \mid |x| = n\}$ and $\Sigma^{\leq n} = \{x \in \Sigma^* \mid |x| \leq n\}$, where $|x|$ denotes the length of x. Opposed to the notation Σ^*, Σ^∞ stands for the collection of all infinite sequences, each of which consists of symbols from Σ. For any symbol a in Σ, a^∞ denotes an element of Σ^∞, which is the infinite sequence made only of a. We assume the reader's familiarity with classical automata theory and the basic concepts of quantum computation (see, e.g., [24, 25, 35]).

3.1 Basic Definition

We first give a "basic" definition of a QIP system whose verifier is a qfa. Our basic definition is a natural concoction of the IP model of Dwork and Stockmeyer [17] and the qfa model of Kondacs and Watrous [32]. In the subsequent section, we discuss a major difference between our QIP systems and the circuit-based QIP systems of Watrous [43]. Our definition seemingly demands much stricter conditions than that of Dwork and Stockmeyer; however, our basic model serves a mold to build various QIP systems with qfa verifiers. In later sections, we shall restrict the behaviors of a verifier as well as a prover to obtain several variants of our basic QIP systems since these restricted models have never been addressed in the literature.

Hereafter, the notation (P, V) is used to denote the QIP system with the prover P and the verifier V. In such a QIP system (P, V), the 2qfa verifier V is particularly specified by a finite set Q of verifier's inner states, a finite input alphabet Σ, a finite communication alphabet Γ, and a verifier's transition function δ. The set Q is the union of three mutually disjoint subsets Q_{non}, Q_{acc}, and Q_{rej}, where any states in Q_{non}, Q_{acc}, and Q_{rej} are respectively called a *non-halting inner state*, an *accepting inner state*, and a *rejecting inner state*. Accepting inner states and rejecting inner states are simply called *halting inner states*. In particular, Q_{non} has the so-called *initial inner state* q_0. The input tape is indexed by natural numbers (the first cell is indexed 0). The two designated symbols ¢ and \$ not in Σ, called respectively the *left endmarker*[§] and the *right endmarker*, mark the left end and the right end of the input. For convenience, set $\check{\Sigma} = \Sigma \cup \{¢, \$\}$. Assume also that Γ contains the blank symbol #. At the beginning of the computation, an input string x over Σ of length n is written orderly from the first cell to the nth cell of the input tape. The tape head initially scans the left endmarker. The communication cell holds only a symbol in Γ and initially the blank symbol # is written in the cell. Similar to the original definition of [32], our input tape is *circular*; that is, whenever the verifier's head scanning ¢ (\$, resp.) on the input tape moves to the left (right, resp.), the head reaches to the right end (resp. left end) of the input tape.

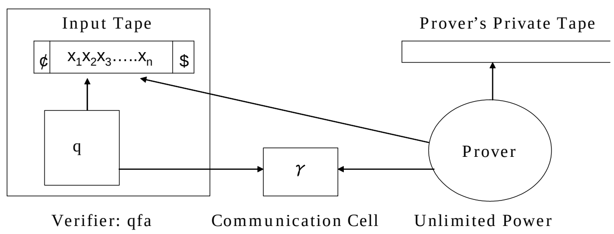

Figure 1: A schematic of a QIP system with a qfa verifier

A *(global) configuration of* (P, V) is a description of the QIP system (P, V) at a certain moment, comprising visible configurations of the two players. Each player can see only his portion of a global configuration. A *visible configuration* of the verifier V on an input of length n is represented by a triplet $(q, k, \gamma) \in Q \times \mathbb{Z}_{n+2} \times \Gamma$, which indicates that the verifier is in state q, the content of the communication cell is γ, and the verifier's head position is k on the input tape. Let $\mathcal{V}_n$ and $\mathcal{M}$ be respectively the Hilbert spaces spanned by the computational bases $\{|q, k\rangle \mid (q, k) \in Q \times \mathbb{Z}_{n+2}\}$ and $\{|\gamma\rangle \mid \gamma \in \Gamma\}$. The Hilbert space $\mathcal{V}_n \otimes \mathcal{M}$ is called the *verifier's visible configuration space* on inputs of length n. The *verifier's transition function* δ is a map from $Q \times \check{\Sigma} \times \Gamma \times Q \times \Gamma \times \{0, \pm 1\}$ to $\mathbb{C}$ and is interpreted as follows. For any $q, q' \in Q$, $\sigma \in \check{\Sigma}$, $\gamma, \gamma' \in \Gamma$, and $d \in \{0, \pm 1\}$, the complex number $\delta(q, \sigma, \gamma, q', \gamma', d)$ specifies the transition amplitude with which the verifier V scanning symbol σ on the input tape and symbol γ on the communication cell in state q changes q to q', replaces γ with γ', and moves the machine's head on the input tape in direction d.

For any input x of length n, δ induces the linear operator U_δ^x on $\mathcal{V}_n \otimes \mathcal{M}$ defined by $U_\delta^x |q, k, \gamma\rangle = \sum_{q', \gamma', d} \delta(q, x_{(k)}, \gamma, q', \gamma', d) |q', k', \gamma'\rangle$, where $x_{(k)}$ is the kth symbol in x and $k' = k + d \pmod{n+2}$. The verifier is called *well-formed* if U_δ^x is unitary on $\mathcal{V}_n \otimes \mathcal{M}$ for every string $x \in \Sigma^*$. Since we are interested only in well-formed verifiers, we henceforth assume that all verifiers are well-formed. For every input x of length n, the 2qfa verifier V starts with the initial superposition $|q_0, 0, \#\rangle$. A single step of the verifier on input x consists of the following process. First, V applies his operation U_δ^x to an existing superposition $|\phi\rangle$ and then $U_\delta^x|\phi\rangle$ becomes the new superposition $|\phi'\rangle$. Let $W_{acc} = \mathrm{span}\{|q, k, \gamma\rangle \mid (q, k, \gamma) \in Q_{acc} \times \mathbb{Z}_{n+2} \times \Gamma\}$, $W_{rej} = \mathrm{span}\{|q, k, \gamma\rangle \mid (q, k, \gamma) \in Q_{rej} \times \mathbb{Z}_{n+2} \times \Gamma\}$, and $W_{non} = \mathrm{span}\{|q, k, \gamma\rangle \mid (q, k, \gamma) \in Q_{non} \times \mathbb{Z}_{n+2} \times \Gamma\}$. Moreover, let k_{acc}, k_{rej}, and k_{non} be respectively the positive numbers representing "accept," "reject," and "non halt." The new superposition $|\phi'\rangle$ is then measured by the observable $k_{acc}E_{acc} + k_{rej}E_{rej} + k_{non}E_{non}$, where E_{acc}, E_{rej}, and E_{non} are respectively the projection operators on W_{acc}, W_{rej}, and W_{non}. Provided that $|\phi'\rangle$ is expressed as $|\psi_1\rangle + |\psi_2\rangle + |\psi_3\rangle$ for certain three vectors $|\psi_1\rangle \in W_{acc}$, $|\psi_2\rangle \in W_{rej}$, and $|\psi_3\rangle \in W_{non}$, we say that, at this step, V *accepts* x *with probability* $\||\psi_1\rangle\|^2$ and *rejects* x *with probability* $\||\psi_2\rangle\|^2$. Only the non-halting superposition $|\psi_3\rangle$ continues to the next step and V is said to *continue (to the next step) with probability* $\||\psi_3\rangle\|^2$. The probability that x is accepted (rejected, resp.) within the first t steps is thus the sum, over all $i \in [1, t]_{\mathbb{Z}}$, of the probabilities with which V accepts (rejects, resp.) x at the ith step. In particular, when

[§]For certain variants of qfa's, the left endmarker is redundant. See, e.g., [4].

the verifier is a 1qfa, the verifier's transition function δ must satisfy the following two additional conditions: (i) for every $q, q' \in Q$, $\sigma \in \check{\Sigma}$, and $\gamma, \gamma' \in \Gamma$, $\delta(q, \sigma, \gamma, q', \gamma', d) = 0$ if $d \neq 1$ (i.e., the head always moves to the right) and (ii) the verifier must enter halting states until the verifier's head moves off the right endmarker \$ (the head may halt at ¢ since the input tape is circular). This second condition makes all computation paths terminate. Therefore, on input x, a 1qfa verifier halts in at most $|x| + 2$ steps.

In contrast to the verifier, the prover P has an infinite private tape and accesses an input x and a communication cell. Let Δ be a finite set of the prover's private tape alphabet, which includes the blank symbol #. The prover is assumed to alter only a "finite" initial segment of his private tape at every step. Let $\mathcal{P}$ be the Hilbert space spanned by $\{|y\rangle \mid y \in \Delta^{\infty}_{fin}\}$, where Δ^{∞}_{fin} is the set of all finite series of tape symbols containing only a finite number of non-blank symbols; namely, $\Delta^* \times \{\#\}^{\infty}$. The *prover's visible configuration space* is the Hilbert space $\mathcal{M} \otimes \mathcal{P}$. Formally, the prover P on input x is specified by a series $\{U^x_{P,i}\}_{i\in\mathbb{N}^+}$ of unitary operators, each of which acts on the prover's visible configuration space, such that $U^x_{P,i}$ is of the form $S^x_{P,i} \otimes I$, where $\dim(S^x_{P,i})$ is finite and I is the identity operator. Such a series of operators is particularly called the *prover's strategy* on the input x. To refer to the strategy on x, we often use the notation P_x. For any function k from $\mathbb{N}^2$ to $\mathbb{N}$, we call the prover $k(n,i)$*-space bounded* if the prover uses at most the first $k(n,i)$ cells of his private tape; that is, at the ith step, $S^x_{P,i}$ is applied only to the first $k(n,i)$ cells of the prover's private tape in addition to the communication cell. We often consider the case where the value $k(n,i)$ is independent of i. If the prover has a string y in his private tape and scans symbol γ in the communication cell, then he applies $U^x_{P,i}$ to the quantum state $|\gamma\rangle|y\rangle$ at the ith step. If $U^x_{P,i}|\gamma\rangle|y\rangle = \sum_{\gamma',y'} \alpha^i_{\gamma',y'}|\gamma'\rangle|y'\rangle$, then the prover changes y into y' and replaces γ by γ' with amplitude $\alpha^i_{\gamma',y'}$.

Formally, a global configuration consists of the four items: V's inner state, V's head position, the content of a communication cell, and the content of P's private tape. We express a superposition of such configurations of (P,V) on input x as a vector in the Hilbert space $\mathcal{V}_{|x|} \otimes \mathcal{M} \otimes \mathcal{P}$, which is called the *(global) configuration space* of (P,V) on input x. The computation of (P,V) on input x constitutes a series of superpositions of configurations resulting by an alternate application of unitary operations of the verifier and the prover as well as the verifier's measurement. The computation on input x starts with the global initial configuration $|q_0, 0\rangle|\#\rangle|\#^{\infty}\rangle$, in which the verifier is in his initial configuration and the prover's private tape consists only of blank symbols. The two players apply their unitary operations U^x_{δ} and $P_x = \{U^x_{P,i}\}_{i\in\mathbb{N}^+}$ in turn starting with the verifier's move. Through the communication cell, the two players exchange communication symbols, which cause the two players entangled. A measurement is made after every move of the verifier to determine whether V is in a halting inner state. Each computation path therefore ends when V enters a certain halting inner state along this computation path. For convenience, we use the same notation (P,V) to mean a QIP system and also a protocol taken by the prover P and the verifier V. Furthermore, we define the overall probability that (P,V) *accepts* (*rejects*, resp.) the input x as the limit, as $t \to \infty$, of the probability that V accepts (rejects, resp.) x in at most t steps. We use the notation $p_{acc}(x,P,V)$ ($p_{rej}(x,P,V)$, resp.) to denote the overall acceptance (rejection, resp.) probability of x by (P,V). We say that V *always halts with probability* 1 if, for every input x and every prover P^*, (P^*,V) reaches halting inner states with probability 1. In general, V may not always halt with probability 1. When we discuss the entire *running time* of the QIP system, we count the number of all steps taken by the verifier as well as the prover.

Let a, b be any two real numbers in the unit interval $[0,1]$ and let L be any language. We say that L *has an* (a,b)*-QIP system* (P,V) (or *a* (a,b)*-QIP system* (P,V) *recognizes* L) if (P,V) is a QIP system and the following two conditions hold for (P,V):

1. (completeness) for any $x \in L$, (P,V) accepts x with probability at least a, and
2. (soundness) for any $x \notin L$ and any prover P^*, (P^*,V) rejects[¶] x with probability at least b.

Note that a (a,a)-QIP system has the error probability at most $1-a$. This paper discusses only the QIP systems whose error probabilities are bounded above by certain constants lying in the interval $[0,1/2)$.

Adapting the notational convention of Condon [13], we write $\mathrm{QIP}_{a,b}(\langle\mathcal{R}\rangle)$, where $\langle\mathcal{R}\rangle$ is a set of restrictions, to denote the collection of all languages recognized by certain (a,b)-QIP systems with the restrictions specified by $\langle\mathcal{R}\rangle$. Let $\mathrm{QIP}(\langle\mathcal{R}\rangle)$ be $\bigcup_{\epsilon>0} \mathrm{QIP}_{1/2+\epsilon,1/2+\epsilon}(\langle\mathcal{R}\rangle)$. If in addition the verifier's amplitudes are restricted to an amplitude set K (but there is no restriction for the prover), then we rather write $\mathrm{QIP}_K(\langle\mathcal{R}\rangle)$. Notice that $\mathrm{QIP}(\langle\mathcal{R}\rangle) = \mathrm{QIP}_{\mathbb{C}}(\langle\mathcal{R}\rangle)$. Mostly, we focus our attention on the following three basic restrictions $\langle\mathcal{R}\rangle$: $\langle 1qfa\rangle$

[¶]Generally, the QIP system may increase its power if we instead require (P^*,V) to *accept* x with probability $\leq 1-b$ for any prover P^*. Such a modification defines a *weak* QIP system. See, e.g., [17] for the classical case.

("measure-many" 1qfa verifiers), $\langle 2qfa \rangle$ ("measure-many" 2qfa verifiers), and $\langle$*poly-time*$\rangle$ (expected polynomial running time). For instance, QIP($2qfa$, *poly-time*) denotes the language class defined by QIP systems with expected polynomial-time 2qfa verifiers. Other types of restrictions will be discussed in later sections.

3.2 Comparison with Circuit Based QIP Systems

We briefly discuss the major difference between our automaton-based QIP systems and circuit-based QIP systems in which a prover and a verifier are both viewed as two finite series of quantum circuits intertwined each other in turn, sharing only message qubits. Here, assumed is the reader's familiarity with Watrous's circuit-based QIP model [43].

In the circuit-based model of Watrous, the measurement of the output qubit is performed only once at the end of the computation since any measurement during the computation can be postponed to the end (see, e.g., [35]). This is possible because the verifier uses his own private qubits and his running time is bounded. However, since our 2qfa verifier has no private tape and may not halt within a finite number of steps, the simulation of such a verifier on a quantum circuit requires a measurement of a certain number of qubits (as a halting flag) after each move of the verifier.

A verifier in the circuit-based model is allowed to carry out a large number of basic unitary operations in its single interaction round whereas a qfa verifier in our basic model is constantly under attack of a malicious prover after every move of the verifier. This comes from the belief that no malicious prover truthfully keeps the communication cell unchanged while awaiting for the verifier's next query. Therefore, such a malicious prover may exercise more influence on the verifier in our QIP model than in the circuit-based model. Later in Section 9, nevertheless, we shall introduce a variant of our basic QIP systems, in which we allow a verifier to make a series of transitions without communicating with a prover. This makes it possible for us to discuss the number of communications between a prover and a verifier necessary for the recognition of a given language.

4 One-Way QFA Verifiers against Mighty Provers

Following the definition of a qfa-verifier QIP systems, we shall demonstrate how well a qfa verifier plays against a powerful prover. We begin with our investigation on the power of QIP systems whose verifiers are particularly limited to 1qfa's.

Earlier, Kondacs and Watrous [32] demonstrated a weakness of 1qfa's; namely, no 1qfa recognizes the regular language $Zero$ and therefore, 1QFA cannot contain REG. In the following theorem, we show that the interaction between a prover and a 1qfa verifier complements such deficiency of 1qfa's and truly enhances the power of recognizing languages: QIP($1qfa$) equals REG. This gives a complete characterization of the QIP systems with 1qfa verifiers.

Theorem 4.1 1QFA $\subsetneqq$ QIP($1qfa$) = REG.

Note that the first inequality of Theorem 4.1 follows from the last equality since 1QFA $\neq$ REG. To prove this equality, we first claim in Proposition 4.2 that, for any *1-way deterministic finite automaton* (*1dfa*, in short) M, we can build a QIP system (P, V) in which the 1qfa verifier V simulates M in a reversible fashion. Since any move of a 1dfa is generally not reversible, we need to use an honest prover as an "eraser" which removes any irreversible information of M into the prover's private tape to maintain a history of the verifier's past inner states. This simulation establishes the desired inclusion.

Proposition 4.2 REG $\subseteq$ $\mathrm{QIP}_{1,1}(1qfa)$.

Proof. Let L be any regular language and let $M = (Q, \Sigma, \delta_M)$ be any 1dfa that recognizes L, where Q is the set of all inner states, Σ is the input alphabet, and δ_M is the transition function. We may assume for convenience that M's input tape has the left endmarker ¢ and the right endmarker \$ because this assumption does not change the recognition power of the 1dfa. For any pair (q, σ) of an inner state $q \in Q$ and an input symbol $\sigma \in \Sigma$, consider the set $S_{q,\sigma}$ of all inner states that lead to q while scanning σ; namely, $S_{q,\sigma} = \{p \in Q \mid \delta_M(p, \sigma) = q\}$.

Our goal is to define a QIP system that recognizes L with probability 1. Consider the following QIP protocol that simulates M by forcing a prover to act as an eraser. In what follows, let $\Gamma = \{\#\} \cup \left(\bigcup_{q\in Q,\sigma\in\Sigma} S_{q,\sigma}\right)$ be our communication alphabet, provided that the symbol # is not in Q. The verifier V is defined to simulate truthfully each move of M. Let us assume that, at an arbitrary step $i \in [1, n+2]_{\mathbb{Z}}$, V is in inner state p scanning symbol σ. Now, consider the case where $\delta_M(p,\sigma) = q$; in other words, M enters state q just after it scans symbol σ in state p. The verifier V behaves as follows. In scanning the current communication symbol, whenever it is not #, V immediately rejects the input. Assuming that the communication symbol is #, V enters the state q by passing the communication symbol p to a prover. Note that, if the prover always returns #, V eventually ends its computation at the time when the head reaches the endmarker \$. If M enters an accepting inner state, then V simply accepts the input; otherwise, V rejects the input. We design our honest prover P to return # at every communication step.

Let x be any input to our QIP system (P,V). First, consider the case where x belongs to L. Since the honest prover P erases the information on V's inner state at every step, V can simulate each move of M in a reversible fashion. Hence, V accepts x with probability 1. On the contrary, when $x \notin L$, a dishonest prover P^* cannot return any symbol except for # (or any superposition of such symbols) to optimize his adversarial strategy because, otherwise, V can increase his rejection probability by immediately entering a rejecting inner state in a deterministic manner. If P^* always returns #, however, V correctly simulates M and eventually enters a rejecting inner state with probability 1. Therefore, (P,V) recognizes L with certainty. □

To show that $\mathrm{QIP}(1qfa) \subseteq \mathrm{REG}$—the opposite direction of Proposition 4.2, we use two results: Lemmas 4.3 and 4.4. To state these lemmas, we need the notion of resource-bounded QIP systems. Let s and t be any functions mapping $\mathbb{N}$ to $\mathbb{N}$. A *$(t(n),s(n))$-bounded QIP system* is obtained from a QIP system by forcing the QIP protocol to "terminate" after $t(|x|)$ steps on each input x with $s(|x|)$-space bounded provers. After the $t(|x|)$th measurement, we actually stop the entire computation of the QIP system and make any non-halting inner state collapse to the special output symbol "*I don't know*". We say that a language L *has a $(t(n),s(n))$-bounded QIP system* (or *a $(t(n),s(n))$-bounded QIP system recognizes L*) if the system satisfies the completeness and soundness conditions given in Section 3 for L with error probability at most ϵ, where ϵ is a certain constant drawn from the interval $[0,1/2)$. The following lemma connects basic QIP systems to bounded QIP systems.

Lemma 4.3 *Let L be any language in* $\mathrm{QIP}(1qfa)$. *There exists a constant $c \in \mathbb{N}^+$ such that L has an $(n+2,c)$-bounded QIP system with a 1qfa verifier.*

Lemma 4.3 is a direct consequence of Lemma 5.5, which we shall prove in the subsequent section. Another ingredient, Lemma 4.4, relates to the notion of *1-tiling complexity* [14]. For any language L over alphabet Σ, we define the infinite binary matrix M_L whose rows and columns are indexed by the strings over Σ in the following fashion: any (x,y)-entry of M_L is 1 if $xy \in L$ and 0 otherwise. Furthermore, for each $n \in \mathbb{N}$, $M_L(n)$ denotes the submatrix of M_L whose rows and columns are indexed by the strings of length $\leq n$. A *1-tile* of $M_L(n)$ is a nonempty submatrix M of $M_L(n)$ such that (i) all the entries of M are specified by a certain index set $R \times C$, where $R, C \subseteq \Sigma^{\leq n}$, and (ii) all the entries of M have the same value 1. For convenience, we often identify $R \times C$ with M itself. A *1-tiling* of $M_L(n)$ is a set S of 1-tiles of $M_L(n)$ such that every 1-valued entry of $M_L(n)$ is covered by at least one element of S. The *1-tiling complexity* of L is the function $T^1_L(n)$ whose value is the minimal size of a 1-tiling of $M_L(n)$.

Lemma 4.4 *Let L be any language, let $c \in \mathbb{N}^+$, and let $\epsilon \in [0,1/2)$. If an $(n+2,c)$-bounded QIP system (P,V) with a 1qfa verifier recognizes L with error probability at most ϵ, then the 1-tiling complexity of L is at most $4^d\lceil 2\sqrt{2}(1+2d^2)/(1-2\epsilon)\rceil^{2d+1}$, where d equals $|Q||\Gamma||\Delta|^c$ for the set Q of the verifier's inner states, the prover's tape alphabet Δ, and the communication alphabet Γ.*

Proof. Let L be any language recognized by an $(n+2,c)$-bounded QIP system (P,V) with a 1qfa verifier with error probability at most $\epsilon < 1/2$. Let Q, Δ, and Γ be respectively the set of V's inner states, V's tape alphabet, and the communication alphabet. Recall that, for every input $x \in \Sigma^*$ and every step $i \in [1,|x|+1]_{\mathbb{Z}}$, $U^x_{P,i}$ denotes P's ith operation on x, which is described as a $|\Delta|^c$-dimensional unitary matrix since P is c-space bounded. Since P's strategy may differ on a different input, we use the notation P_x to indicate that P always takes the strategy $\{U^x_{P,i}\}_{i\in\mathbb{N}^+}$ on any given input. Write d for $|Q||\Gamma||\Delta|^c$ and μ for $(1/2-\epsilon)/(1+2d^2)$.

Consider the binary matrix M_L induced from L. Our goal is to present a 1-tiling of $M_L(n)$, for each $n \in \mathbb{N}$, of size at most $(2\lceil\sqrt{2}/\mu\rceil)^{2d}\lceil 1/\mu\rceil \leq 4^d\lceil\sqrt{2}/\mu\rceil^{2d+1} = 4^d\lceil\frac{2\sqrt{2}(1+2d^2)}{1-2\epsilon}\rceil^{2d+1}$. Note that, for any 1-valued (x,y)-entry of $M_L(n)$, since $xy \in L$, the QIP protocol (P_{xy},V) accepts xy with probability at least $1-\epsilon$. Notationally, for each vector $\mathbf{p}$ and any index $\mathbf{i}$, $[\mathbf{p}]_{\mathbf{i}}$ represents the $\mathbf{i}$-entry of $\mathbf{p}$.

For any fixed input x, a quadruple (j_1,j_2,j_3,j_4) in the set $Q \times [1,|x|+2]_{\mathbb{Z}} \times \Delta^c \times \Gamma$ represents a *global configuration* of the $(n+2,c)$-bounded QIP system (P,V), in which V is in inner state j_1 with its head scanning the j_2th cell, the communication cell contains j_3, and the prover's private tape consists of j_4. If the head position j_2 is ignored, we call the remaining triplet (j_1, j_3, j_4) a *semi-configuration.* Let $I = Q \times \Delta^c \times \Gamma$ (the set of all semi-configurations) and, for each $n \in \mathbb{N}$, let I_n be $Q \times [1, n+2]_{\mathbb{Z}} \times \Delta^c \times \Gamma$ (the set of all global configurations on any input of length n).

In the following definition of a 1-tiling, we arbitrarily fix an integer $n \in \mathbb{N}^+$ and two strings x and y of length $\leq n$ satisfying that $xy \in L$. Since V is fixed, we drop the letter V out of $p_{acc}(x, P, V)$. To compute $p_{acc}(xy, P_{xy})$, we introduce two types of vectors. The *configuration amplitude vector* $\mathbf{p}_{x,y}$ is the unique $(d+1)$-dimensional vector $\mathbf{p}_{x,y}$ whose first d entries are indexed by the semi-configurations. For simplicity, all the semi-configurations are assumed to be enumerated. For any semi-configuration $\mathbf{i} = (i_1, i_2, i_3) \in I$, the $\mathbf{i}$-entry $[\mathbf{p}_{x,y}]_{\mathbf{i}}$ is set to be 0 if i_1 is a halting inner state; otherwise, $[\mathbf{p}_{x,y}]_{\mathbf{i}}$ is the amplitude of the configuration $(i_1, |x|+1, i_2, i_3)$ in the superposition obtained after the $|x|+1$st application of V's unitary operation. In addition, the final $d+1$st entry of $\mathbf{p}_{x,y}$ indicates the probability that xy is accepted within the first $|x|+1$ steps of V.

We further define additional d-dimensional vectors to describe the transition amplitudes of the protocol (P, V). For each index $\mathbf{j} = (j_1, j_2, j_3, j_4) \in I_{|y|}$, let $\mathbf{r}^{\mathbf{j}}_{x,y}$ be the d-dimensional vector whose entries are indexed by the semi-configurations $\mathbf{i} = (i_1, i_2, i_3)$. If j_1 is an accepting inner state, then the (i_1, i_2, i_3)-entry of $\mathbf{r}^{\mathbf{j}}_{x,y}$ indicates the transition amplitude from the configuration $(i_1, |x|+1, i_2, i_3)$ to the configuration $(j_1, |x|+j_2+1, j_3, j_4)$; otherwise, $[\mathbf{r}^{\mathbf{j}}_{x,y}]_{\mathbf{i}}$ is 0. It immediately follows that $\sum_{\mathbf{j} \in I_{|y|}} |[\mathbf{r}^{\mathbf{j}}_{x,y}]_{\mathbf{i}}|^2 \leq 1$ for any fixed semi-configuration $\mathbf{i} \in I$.

Using the aforementioned vectors, we can calculate the acceptance probability $p_{acc}(xy, P_{xy})$ of input xy by the protocol (P_{xy}, V) as follows: $p_{acc}(xy, P_{xy}) = \sum_{\mathbf{j} \in I_{|y|}} |\mathbf{p}'_{x,y} \cdot \mathbf{r}^{\mathbf{j}}_{x,y}|^2 + [\mathbf{p}_{x,y}]_{d+1}$, where $\mathbf{p}'_{x,y}$ is the d-dimensional vector obtained from $\mathbf{p}_{x,y}$ by deleting its last entry and the notation $\cdot$ denotes the inner product.

First, letting $\mathbb{C}_1 = \{r \in \mathbb{C} \mid \exists a, b[r = a + ib \;\&\; |a|, |b| \leq 1]\}$, we partition the $(d+1)$-dimensional complex space $\mathbb{C}_1^d \times [0,1]$ into $(2\lceil \sqrt{2}/\mu \rceil)^{2d} \lceil 1/\mu \rceil$ hyper-cuboids of diameter μ in each $\mathbb{C}_1$ and $[0,1]$; i.e., a $\frac{\mu}{\sqrt{2}} \times \frac{\mu}{\sqrt{2}}$ square in each of the first d coordinates and a real line segment of length μ in the $d+1$st coordinate. Note that some hyper-cuboids near the boundary may have diameter less than μ in certain coordinates. Note that each hyper-cuboid has volume at most $\mu^{2d+1}/2^d$. Second, we associate each hyper-cuboid C with the rectangle R_C defined as $R_C = \{x \mid \exists y' (\mathbf{p}_{x,y'} \in C \;\wedge\; xy' \in L)\} \times \{y \mid \exists x' (\mathbf{p}_{x',y} \in C \;\wedge\; x'y \in L)\}$. To complete the proof, it suffices to prove that R_C is a 1-tile of $M_L(n)$ for every hyper-cuboid C whose rectangle is non-empty since, if so, every 1-valued entry of $M_L(n)$ is covered by a certain 1-tile R_C and therefore, the collection T of all such rectangles forms a 1-tiling of $M_L(n)$. Hence, the 1-tiling complexity of L is bounded by $T^1_L(n) \leq |T| = (2\lceil \sqrt{2}/\mu \rceil)^{2d} \lceil 1/\mu \rceil$.

Let C be any hyper-cuboid whose rectangle is non-empty and let (x, y) be any pair of strings of length $\leq n$ in R_C. Toward a contradiction, we assume that R_C is not a 1-tile; namely, $xy \notin L$. This implies that, for any prover P^*, (P^*, V) accepts xy with probability $\leq \epsilon$. Since $(x, y) \in R_C$, there exists a pair (x', y') of strings of length $\leq n$ such that $\mathbf{p}_{x,y'}$ and $\mathbf{p}_{x',y}$ are both in C. It follows that $p_{acc}(x'y, P_{x'y}) \geq 1 - \epsilon$ since $x'y \in L$. Now, consider the special prover P' that simulates $P_{xy'}$ while reading x and then simulates $P_{x'y}$ while reading y. By the definition of P', it follows that $p_{acc}(xy, P') = \sum_{\mathbf{j} \in I_{|y|}} |\mathbf{p}'_{x,y'} \cdot \mathbf{r}^{\mathbf{j}}_{x',y}|^2 + [\mathbf{p}_{x,y'}]_{d+1}$.

We wish to claim that $p_{acc}(xy, P') > \epsilon$. The difference between $p_{acc}(xy, P')$ and $p_{acc}(x'y, P_{x'y})$ is upper-bounded by:

$$\begin{aligned}
&|p_{acc}(xy, P_{xy}) - p_{acc}(x'y, P_{x'y})| \\
&\leq \quad |[\mathbf{p}_{x',y}]_{d+1} - [\mathbf{p}_{x,y'}]_{d+1}| + \sum_{\mathbf{j}} \left| \left| \sum_{\mathbf{i}} [\mathbf{p}'_{x,y'}]_{\mathbf{i}} [\mathbf{r}^{\mathbf{j}}_{x'y}]_{\mathbf{i}} \right|^2 - \left| \sum_{\mathbf{i}} [\mathbf{p}'_{x',y}]_{\mathbf{i}} [\mathbf{r}^{\mathbf{j}}_{x',y}]_{\mathbf{i}} \right|^2 \right| \\
&\leq \quad |[\mathbf{p}_{x',y}]_{d+1} - [\mathbf{p}_{x,y'}]_{d+1}| + \sum_{\mathbf{j}} \sum_{\mathbf{i},\mathbf{i}'} \left| ([\mathbf{p}'_{x,y'}]_{\mathbf{i}} [\mathbf{p}'_{x,y'}]^*_{\mathbf{i}'} - [\mathbf{p}'_{x',y}]_{\mathbf{i}} [\mathbf{p}'_{x',y}]^*_{\mathbf{i}'}) [\mathbf{r}^{\mathbf{j}}_{x',y}]_{\mathbf{i}} [\mathbf{r}^{\mathbf{j}}_{x',y}]^*_{\mathbf{i}'} \right|.
\end{aligned}$$

The first term $|[\mathbf{p}_{x',y}]_{d+1} - [\mathbf{p}_{x,y'}]_{d+1}|$ is at most μ since $\mathbf{p}_{x',y}$ and $\mathbf{p}_{x,y'}$ are in the same hyper-cuboid. The last term is also bounded above by $2\mu \sum_{\mathbf{j}} \sum_{\mathbf{i},\mathbf{i}'} \left| [\mathbf{r}^{\mathbf{j}}_{x',y}]_{\mathbf{i}} [\mathbf{r}^{\mathbf{j}}_{x',y}]^*_{\mathbf{i}'} \right|$. This comes from the following bound:

$$|[\mathbf{p}'_{x,y'}]_{\mathbf{i}} [\mathbf{p}'_{x,y'}]^*_{\mathbf{i}'} - [\mathbf{p}'_{x',y}]_{\mathbf{i}} [\mathbf{p}'_{x',y}]^*_{\mathbf{i}'}| \leq |[\mathbf{p}'_{x,y'}]_{\mathbf{i}} ([\mathbf{p}'_{x,y'}]^*_{\mathbf{i}'} - [\mathbf{p}'_{x',y}]^*_{\mathbf{i}'})| + |[\mathbf{p}'_{x',y}]^*_{\mathbf{i}'} ([\mathbf{p}'_{x,y'}]_{\mathbf{i}} - [\mathbf{p}'_{x',y}]_{\mathbf{i}})| \leq 2\mu.$$

This term $2\mu \sum_{\mathbf{j}} \sum_{\mathbf{i},\mathbf{i}'} \left| [\mathbf{r}^{\mathbf{j}}_{x',y}]_{\mathbf{i}} [\mathbf{r}^{\mathbf{j}}_{x',y}]^*_{\mathbf{i}'} \right|$ is further bounded by $2\mu \sum_{\mathbf{i},\mathbf{i}'} \sqrt{\sum_{\mathbf{j}} |[\mathbf{r}^{\mathbf{j}}_{x',y}]_{\mathbf{i}}|^2} \sqrt{\sum_{\mathbf{j}} |[\mathbf{r}^{\mathbf{j}}_{x',y}]^*_{\mathbf{i}'}|^2}$ using the Cauchy-Schwarz inequality and is thus at most $2\mu d^2$. Overall, the term $|p_{acc}(xy, P') - p_{acc}(x'y, P_{x'y})|$ is upper-bounded by $\mu(1 + 2d^2)$, which also equals $1/2 - \epsilon$ by the choice of μ. Therefore, the desired inequality $p_{acc}(xy, P') \geq 1/2 > \epsilon$ follows immediately from $p_{acc}(x'y, P_{x'y}) \geq 1 - \epsilon$. This implies that (P', V) accepts xy

with probability $> \epsilon$. This contradicts our assumption that $p_{acc}(xy, P^*) \le \epsilon$ for any prover P^*. Therefore, R_C is a 1-tile of $M_L(n)$. □

At length, we obtain the containment QIP($1qfa$) $\subseteq$ REG by combining Lemmas 4.3 and 4.4, which indicates that every language in QIP($1qfa$) has 1-tiling complexity $O(1)$. Recall from [14] that a language is regular if and only if its 1-tiling complexity is bounded above by a certain constant. Therefore, it immediately follows that QIP($1qfa$) $\subseteq$ REG, as requested. This completes the proof of Theorem 4.1.

5 Two-Way QFA Verifiers against Mighty Provers

We have seen in the previous section that, using interactions with provers, 1qfa verifiers can exercise a remarkable power of recognizing the regular languages. This section turns our interest to the 2qfa-verifier QIP systems; namely, QIP($2qfa$) and QIP($2qfa$, *poly-time*). First, observe that a verifier can completely eliminate any intrusion of a prover by simply ignoring the communication cell (i.e., applying the identity operation). This observation yields the following simple containments: 2QFA $\subseteq$ QIP($2qfa$) and 2QFA(*poly-time*) $\subseteq$ QIP($2qfa$, *poly-time*).

Now, we demonstrate the power of QIP($2qfa$, *poly-time*).

Theorem 5.1 REG $\subsetneqq$ QIP($2qfa$, *poly-time*) $\nsubseteq$ AM($2pfa$).

The first proper containment follows immediately from the facts that REG $\subsetneqq$ 2QFA(*poly-time*) [32] and 2QFA(*poly-time*) $\subseteq$ QIP($2qfa$, *poly-time*). To prove the second separation, we first introduce a variation of Pal, briefly called $Pal_\#$, which is defined as $Pal_\# = \{x\#x^R \mid x \in \{0,1\}^*\}$ over the alphabet $\{0,1,\#\}$, where $\#$ is a separator not in $\{0,1\}$. Similar to Pal [17, Theorem 3.4], we can show that this language $Pal_\#$ does not belong to AM($2pfa$). In the following lemma, we further claim that $Pal_\#$ is indeed in QIP($2qfa$, *poly-time*). Theorem 5.1 naturally follows from this lemma.

Lemma 5.2 *For any constant $\epsilon \in (0, 1/2]$, $Pal_\# \in$ QIP$_{1,1-\epsilon}$($2qfa$, poly-time).*

Proof. We slightly modify the classical IP protocol given in [17] for Pal. Let $\Sigma = \{0,1,\#\}$ be our input alphabet, let $\Gamma = \{0,1,\#\}$ be our communication alphabet. Let ϵ be any error bound in $(0,1/2]$ and set $d = \lceil \log_2(1/\epsilon) \rceil$. Note that $d \ge 1$ since $\epsilon \le 1/2$. Our QIP system (P, V) for $Pal_\#$ is given as follows. We begin with the description of the 2qfa verifier V who runs in worst-case linear time. Recall that the verifier's head is initially scanning the endmarker ¢ with the blank symbol # in the communication cell. Let x be any input string. The verifier runs the following quantum algorithm by stages, creating the total of 2^d independent computation paths. The initial stage is assumed to be $s = \lambda$, the empty string.

> Repeat the following procedure (*) until $|s| = d$. During this procedure, V always unalters the communication cell (such a verifier is said to make a *one-way communication*). Assume that V is in stage $s \in \{0,1\}^{\le d-1}$.
>
> (*) In the first phase, the head moves rightward. If there is no # in x, then V rejects x when V scans the right endmarker. In scanning #, V generates a superposition of two independent branches by entering two inner states $q_{1,s}$ and $q_{2,s}$ with the equal amplitude $1/\sqrt{2}$. In the branch starting with $q_{1,s}$, the head moves leftward; in the other branch with $q_{2,s}$, it moves rightward. During this phase, whenever a prover returns any non-blank symbol, V rejects x immediately. In the second phase, visiting each cell, V receives a communication symbol, say a, from a prover. The head checks whether it is currently scanning a in the input tape unless the head arrives at an endmarker. If V discovers a discrepancy, then it enters a rejecting inner state. When the head reaches an endmarker, V rejects x if a prover sends a non-blank symbol. The head at the left endmarker ¢ stays still for another step and enters $q_{0,s0}$ whereas the head at the right endmarker \$ enters $q_{0,s1}$ by moving right to ¢ (since the input tape is circular‖). Go to the next stage.
>
> Along each computation path s, if x is not yet rejected after executing (*) d times, then V enters an accepting inner state.

Table 1 describes the formal transitions of V. Note that the running time of V is $O(n)$ even in the worst case. Consider the case where $x = y\#y^R$ for a certain string y. In each round, the honest prover P must pass the

‖The circularity of the input tape is used to simplify the description of the transitions and is not necessary for the lemma.

string y^R bit by bit to the verifier after V splits into two branches. With this honest prover P, V never enters any rejecting inner state. Hence, after d rounds, V finally accepts x with probability 1.

$V_{¢}\|q_{0,s}\rangle\|\#\rangle = \|q'_{0,s}\rangle\|\#\rangle$	$V_{¢}\|q_{1,s}\rangle\|\#\rangle = \|q_{0,s0}\rangle\|\#\rangle$
$V_{¢}\|q_{i,s}\rangle\|a\rangle = \|r_{i,s}\rangle\|a\rangle$	$V_{\$}\|q_{i,s}\rangle\|a\rangle = \|r_{i,s}\rangle\|a\rangle$
$V_{\$}\|q'_{0,s}\rangle\|\#\rangle = \|r_{0,s}\rangle\|\#\rangle$	$V_{\$}\|q_{2,s}\rangle\|\#\rangle = \|q_{0,s1}\rangle\|\#\rangle$
$V_{\#}\|q'_{0,s}\rangle\|\#\rangle = \frac{1}{\sqrt{2}}(\|q_{1,s}\rangle\|\#\rangle + \|q_{2,s}\rangle\|\#\rangle)$	$V_{\#}\|q_{i,s}\rangle\|b\rangle = \|r_{i,s}\rangle\|b\rangle$
$V_a\|q'_{0,s}\rangle\|\#\rangle = \|q'_{0,s}\rangle\|\#\rangle$	
$V_a\|q_{i,s}\rangle\|a\rangle = \|q_{i,s}\rangle\|a\rangle$	$V_a\|q_{i,s}\rangle\|a'\rangle = \|r_{i,s}\rangle\|a'\rangle$
$D(q_{1,s}) = -1$	$D(q_{2,s}) = 1$
$D(q'_{0,s}) = 1$	$D(q_{0,s0}) = 0$
$D(q_{0,s1}) = 1$	

Table 1: Transitions of V for $Pal_{\#}$ with stage $s \in \{0,1\}^{\leq d-1}$, $i \in \{1,2\}$, $a \in \{0,1\}$, $a' \in \Gamma$ with $a \neq a'$, and $b \in \Gamma$. The rejecting inner states are $r_{0,s}$ and $r_{i,s}$. The unitary operator V_σ for each $\sigma \in \check{\Sigma}$ acts on the Hilbert space spanned by $Q \times \Gamma$. The transition function δ of V is then induced by setting $\delta(q,\sigma,\gamma,q',\gamma',d) = \langle q',\gamma'|U_\sigma|q,\gamma\rangle$ and $d = D(q)$ for any $q,q' \in Q$ and any $\gamma,\gamma' \in \Gamma$.

Next, assume that x is not in $Pal_{\#}$. It suffices to consider only the case where x is of the form $y\#z^R$ since, if there is no #, V rejects x with probability 1. In each round, since V makes only one-way communication with a dishonest prover, the prover's visible configuration is exactly the same along two branches. In other words, the prover answers in exactly the same way along these two branches. In the second phase, a dishonest prover P^* may return a superposition of 0 and 1. Since V's two branches never interfere with each other in each round, V can eliminate at least one of them by entering a rejecting inner state. This gives the rejection probability of at least 1/2 since the squared magnitude of the superposition obtained along each branch is exactly 1/2. Since we repeat (*) d times, the total rejection probability sums up to at least $\sum_{i=1}^{d} 2^{-i} = 1 - 1/2^d$, which is lower-bounded by $1-\epsilon$ by the choice of d. Thus, V rejects x with probability $\geq 1-\epsilon$. Therefore, (P,V) is a $(1,1-\epsilon)$-QIP system that recognizes $Pal_{\#}$. □

Supplementing Theorem 5.1, we now present an upper bound of QIP($2qfa$, *poly-time*): with an appropriate choice of amplitudes, QIP($2qfa$, *poly-time*) is located in the complexity class NP, where NP is the class consisting of all languages recognized by nondeterministic Turing machines in polynomial time. This can be compared with a result of Dwork and Stockmeyer [17], who proved that AM($2pfa$) $\subsetneq$ IP($2pfa$, *poly-time*) $\subseteq$ PSPACE.

Theorem 5.3 QIP$_{\tilde{\mathbb{C}}}$($2qfa$, *poly-time*) $\subseteq$ NP.

To show the desired upper-bound of QIP$_{\tilde{\mathbb{C}}}$($2qfa$, *poly-time*), we need the following lemma, which is similar to Lemma 4.3.

Lemma 5.4 *Every language in* QIP($2qfa$, *poly-time*) *has a* $(t(n), c\log n + c)$*-bounded QIP system for a certain polynomial* t *and a certain constant* $c > 0$.

Lemma 5.4 (as well as Lemma 4.3) directly comes from the following lemma whose proof is based on the result of Kobayashi and Matsumoto [30]. Lemma 5.5 states that, without changing the acceptance probability, the prover's visible configuration space can be reduced in size to the verifier's visible configuration space.

Lemma 5.5 *Let* (P,V) *be any QIP system with a 2qfa (1qfa, resp.) verifier and let* Q,Γ *be respectively the sets of all inner states and of all communication symbols. There exists another prover* P' *that satisfies the following two conditions: for every input* x *and every* $i \in \mathbb{N}^+$*, (i) the prover's* i*th operation* $U^x_{P',i}$ *is a* $|Q||\Gamma|(|x|+2)$*-dimensional (*$|Q||\Gamma|$*-dimensional, resp.) unitary operator, and (ii)* (P',V) *accepts* x *with the same probability as* (P,V) *does.*

Proof. Take an arbitrary QIP system (P,V) with the set Q of all inner states of V, the communication alphabet Γ, and the transition function δ of V. In this proof, we consider only the case where V is a 1qfa. The remaining case where V is a 2qfa can be similarly proven if we further include the information on V's head position.

For convenience, we view our QIP system (P, V) as a quantum circuit of three registers. The first register represents the inner state of V together with the head position of the input tape, the second register represents the communication cell, and the third register represents a prover's private tape. Let x be any input of length n. Recall the Hilbert spaces $\mathcal{V}_n$, $\mathcal{M}$, and $\mathcal{P}$ associated with (P, V) on input x. The Hilbert space $\mathcal{V}_n$ is the tensor product of the $|Q|$-dimensional space $\mathcal{V}$ and the $(n+2)$-dimensional space $\mathcal{V}'_n$. Henceforth, we can omit the description of qubits on $\mathcal{V}'_n$ since V is a 1qfa. The initial superposition of (P, V) is $|\chi_0\rangle = |q_0\rangle|\#\rangle|\lambda\rangle$. Note that, at each step $i \in [1, n+1]_{\mathbb{Z}}$, without changing V's acceptance probability, we can swap the application order of V's ith measurement E_{non} and P's ith operation $U^x_{P,i}$. For each index $i \in [1, n+1]_{\mathbb{Z}}$, the three superpositions $|\phi_i\rangle$, $|\psi_i\rangle$, and $|\chi_i\rangle$ are inductively defined as follows: $|\chi_i\rangle = E_{non}|\psi_i\rangle$, $|\psi_i\rangle = U^x_{P,i}|\phi_i\rangle$, and $|\phi_i\rangle = U^x_\delta|\chi_{i-1}\rangle$. In addition, let $|\phi_{n+2}\rangle = U^x_\delta|\chi_{n+1}\rangle$, which is the superposition obtained just before the final measurement.

For brevity, write $\mathcal{P}'$ for the $|Q||\Gamma|$-dimensional Hilbert space that corresponds to the private tape of a $|Q||\Gamma|$-space bounded prover. Our goal is to define the prover P' that works on $\mathcal{M} \otimes \mathcal{P}'$. Hereafter, we define the strategy $\{U^x_{P',i}\}_{i\in\mathbb{N}^+}$ of P' on input x. For convenience, set $|\chi'_0\rangle = |\chi_0\rangle$. It follows from [35, page 110] that, for every index $i \in [1, n+1]_{\mathbb{Z}}$, there exists a vector $|\psi'_i\rangle$ in $\mathcal{V} \otimes \mathcal{M} \otimes \mathcal{P}'$ satisfying that $\mathrm{tr}_{\mathcal{P}'}|\psi'_i\rangle\langle\psi'_i| = \mathrm{tr}_{\mathcal{P}}|\psi_i\rangle\langle\psi_i|$ since the dimension of $\mathcal{P}'$ is the same as that of $\mathcal{V}\otimes\mathcal{M}$. We further define the vectors $|\chi'_i\rangle$ and $|\phi'_i\rangle$ as follows: let $|\chi'_i\rangle = E_{non}|\psi'_i\rangle$ for $i \in [1, n+1]_{\mathbb{Z}}$ and $|\phi'_i\rangle = U^x_\delta|\chi'_{i-1}\rangle$ for any $i \in [1, n+2]_{\mathbb{Z}}$. Note that $\mathrm{tr}_{\mathcal{P}}|\phi_1\rangle\langle\phi_1| = \mathrm{tr}_{\mathcal{P}'}|\phi'_1\rangle\langle\phi'_1|$. Now, fix $j \in [2, n+2]_{\mathbb{Z}}$ arbitrarily. Since U^x_δ and E_{non} act on neither $\mathcal{P}$ nor $\mathcal{P}'$, we obtain:

$$\mathrm{tr}_{\mathcal{P}}|\phi_j\rangle\langle\phi_j| = U^x_\delta E_{non}(\mathrm{tr}_{\mathcal{P}}|\psi_{j-1}\rangle\langle\psi_{j-1}|)E_{non}(U^x_\delta)^\dagger = U^x_\delta E_{non}(\mathrm{tr}_{\mathcal{P}'}|\psi'_{j-1}\rangle\langle\psi'_{j-1}|)E_{non}(U^x_\delta)^\dagger = \mathrm{tr}_{\mathcal{P}'}|\phi'_j\rangle\langle\phi'_j|,$$

which further implies: for any $i \in [1, n+1]_{\mathbb{Z}}$,

$$\mathrm{tr}_{\mathcal{M}\otimes\mathcal{P}'}|\psi'_i\rangle\langle\psi'_i| = \mathrm{tr}_{\mathcal{M}\otimes\mathcal{P}}|\psi_i\rangle\langle\psi_i| = \mathrm{tr}_{\mathcal{M}\otimes\mathcal{P}}|\phi_i\rangle\langle\phi_i| = \mathrm{tr}_{\mathcal{M}\otimes\mathcal{P}'}|\phi'_i\rangle\langle\phi'_i|,$$

where the second equality comes from the fact that $U^x_{P,i}$ is applied only to the space $\mathcal{M}\otimes\mathcal{P}$. Since $\mathrm{tr}_{\mathcal{M}\otimes\mathcal{P}'}|\psi'_i\rangle\langle\psi'_i| = \mathrm{tr}_{\mathcal{M}\otimes\mathcal{P}'}|\phi'_i\rangle\langle\phi'_i|$, there exists a unitary operator U_i acting on $\mathcal{M} \otimes \mathcal{P}'$ satisfying that $(I \otimes U_i)|\phi'_i\rangle = |\psi'_i\rangle$ [26, 39]. The desired operation $U^x_{P',i}$ of P' is set to be this $I \otimes U_i$.

Next, we compare the acceptance probabilities of the two QIP systems (P, V) and (P', V). We have $\mathrm{tr}_{\mathcal{P}}|\psi_i\rangle\langle\psi_i| = \mathrm{tr}_{\mathcal{P}'}|\psi'_i\rangle\langle\psi'_i|$ for every $i \in [1, n+1]_{\mathbb{Z}}$ as well as $\mathrm{tr}_{\mathcal{P}}|\phi_{n+2}\rangle\langle\phi_{n+2}| = \mathrm{tr}_{\mathcal{P}'}|\phi'_{n+2}\rangle\langle\phi'_{n+2}|$. Thus, for every $i \in [1, n+2]_{\mathbb{Z}}$, the acceptance probability of x produced by the ith measurement of (P, V) equals the acceptance probability of x caused by the ith measurement of (P', V). This completes the proof. □

Lemma 5.5 lets us focus our attention only on $(n^{O(1)}, O(\log(n)))$-bounded QIP systems. To simulate such a system, we need to approximate the prover's unitary operations using only a fixed universal set of quantum gates. Lemma 5.6 relates to an upper bound of the number of quantum gates necessary to approximate a given unitary operator. The lemma, explicitly stated in [36], can be obtained from the Solovay-Kitaev theorem [27, 35] following the standard decomposition of unitary matrices. We fix an appropriate universal set of quantum gates consisting of the Controlled-NOT gate and a finite number of single-qubit gates, with $\tilde{\mathbb{C}}$-amplitudes, that generate a dense subset of SU(2) with their inverse. Write $\log^k n$ for $(\log n)^k$ for any constant $k \in \mathbb{N}^+$.

Lemma 5.6 *For any sufficiently large $k \in \mathbb{N}^+$, any k-qubit unitary operator U_k, and any real number $\epsilon > 0$, there exists a quantum circuit C of size at most $2^{3k}\log^3(1/\epsilon)$ acting on k qubits such that $\|U_C - U_k\| < \epsilon$, where U_C is the unitary operator corresponding to C, where $\|A\| = \sup_{|\phi\rangle\neq 0}\|A|\phi\rangle\|/\||\phi\rangle\|$.*

A quantum circuit C built in Lemma 5.6 can be further encoded into a binary string, provided that the encoding length is at least the size of the quantum circuit. This enables us to prove the simulation result of any bounded QIP system with $\tilde{\mathbb{C}}$-amplitudes. We say that a function f from $\mathbb{N}$ to $\mathbb{N}$ is *polynomial-time computable* if there exists a deterministic Turing machine that, on any input 1^n, outputs $1^{f(n)}$.

Proposition 5.7 *Let s and t be any polynomial-time computable functions from $\mathbb{N}$ to $\mathbb{N}$. Any language that has a $(t(n), s(n))$-bounded QIP system with a 2qfa verifier using $\tilde{\mathbb{C}}$-amplitudes belongs to the complexity class* $\mathrm{NTIME}(n^{O(1)}t(n)2^{O(s(n))}\log^{O(1)} t(n))$.

The proof of Proposition 5.7 is outlined as follows. Given a bounded QIP system, we first guess a binary string that encodes a quantum circuit representing the prover's strategy. We then simulate the verifier's move followed by the prover's operation. This simulation can be done deterministically by listing all the verifier's configurations and simulating their amplitudes at each step. After each step of the verifier, we calculate the probability of reaching any halting configuration instead of performing measurement. Now, we give the formal proof of Proposition 5.7.

Proof of Proposition 5.7. Let (P, V) be any $(t(n), s(n))$-bounded QIP system with a 2qfa verifier and let A be the language recognized by (P, V) with error probability at most $1/2 - \epsilon$ for a certain fixed constant $\epsilon \in (0, 1/2]$. Let x be any input string of length n. By translating the prover's tape alphabet Δ to $\{0,1\}^{\lceil \log |\Delta| \rceil}$ and the communication alphabet Γ to $\{0,1\}^{\lceil \log |\Gamma| \rceil}$, we can assume without loss of generality that our prover uses at most $\lceil \log |\Delta| \rceil s(n)$ qubits on his private tape and writes $\lceil \log |\Gamma| \rceil$-qubit strings in the communication cell. Now, let $s'(n) = \lceil \log |\Delta| \rceil s(n) + \lceil \log |\Gamma| \rceil$ for any $n \in \mathbb{N}$.

A prover comprises a series of $t(n)$ unitary matrices on $s'(n)$ qubits, say $U_1, U_2, \ldots, U_{t(n)}$. For each U of such matrices, Lemma 5.6 gives a quantum circuit C_U of size at most $2^{3s'(n)} \log^3(dt(n))$ such that the unitary operator associated with C_U approximates U to within $1/dt(n)$, where d is a constant satisfying $d > 2/\epsilon$. This makes it possible to replace the prover P by the series of $t(n)$ quantum circuits $(C_{U_1}, C_{U_2}, \ldots, C_{U_{t(n)}})$, which is hereafter abbreviated C. Note that the cumulative approximation error is bounded above by $\sum_{i=1}^{t(n)} \frac{1}{dt(n)} = 1/d$, which is smaller than $\epsilon/2$. Using this C as a prover, V proceeds his computation and accepts (rejects, resp.) x with probability $\geq (1/2 + \epsilon) - \epsilon/2 = 1/2 + \epsilon/2$ if $x \in A$ ($x \notin A$, resp.). Choose an effective encoding $\langle C \rangle$ of C satisfying that $|\langle C \rangle| \leq ct(n) \cdot 2^{3s'(n)} \log^3(dt(n))$ for a certain constant $c > 0$. Note that any configuration of (C, V) requires $s'(n) + O(\log n)$ qubits.

Using the encoding $\langle C \rangle$, we give a classical simulation of the computation of (C, V) on input x. Note that the verifier V can be represented by the product of $t(n) + 1$ unitary matrices of dimension polynomial in n and the "prover" C consists of $t(n)$ unitary matrices of dimension $2^{s'(n)}$. Note that all the gates in C and verifier's transition function use only polynomial-time approximable amplitudes. Within time polynomial in n and $\log t(n)$, we can approximate such amplitudes to within $\frac{1}{t(n)2^{r(n)}}$ for any fixed polynomial r. By choosing a sufficiently large polynomial r, we can deterministically simulate with high accuracy the computation of (C, V) in polynomial time. Such a simulation gives an approximation of the acceptance probability $p_{acc}(x, C, V)$. Now, we accept the input x if the approximated acceptance probability exceeds $1/2$, and reject x otherwise. For a certain polynomial p independent of n, we therefore obtain a $t(n)2^{O(s(n))}p(n, \log t(n))$-time deterministic algorithm that approximately simulates V with a fixed prover C.

At last, we consider the following nondeterministic algorithm $\mathcal{A}$:

> On input x ($n = |x|$), nondeterministically guess $\langle C \rangle$, where C is a series of $t(n)$ quantum circuits of size $\leq 2^{3s'(n)} \log^3(dt(n))$. If the aforementioned deterministic simulation of (C, V) leads to acceptance, then accept x, or else reject x.

It is easy to verify that $\mathcal{A}$ recognizes L in time $p'(n, \log t(n)) \cdot 2^{O(s(n))} t(n)$ for an appropriate polynomial p'. Therefore, L belongs to $\mathrm{NTIME}(n^{O(1)} t(n) 2^{O(s(n))} \log^{O(1)} t(n))$. □

We return to the proof of the second part of Theorem 5.3. Take any language L in $\mathrm{QIP}_{\tilde{\mathbb{C}}}(2qfa, \textit{poly-time})$. Lemma 5.4 guarantees the existence of a bounded-error $(t(n), s(n))$-bounded QIP system recognizing L using $\tilde{\mathbb{C}}$-amplitudes, where $t(n)$ is a polynomial and $s(n)$ is a logarithmic function. From Proposition 5.7, it follows that L belongs to the complexity class $\mathrm{NTIME}(n^{O(1)} t(n) 2^{O(s(n))} \log^{O(1)} t(n))$, which clearly coincides with NP. This ends the proof of Theorem 5.3.

In the end of this section, we present a closure property of QIP systems with 2qfa verifiers.

Proposition 5.8 *$\mathrm{QIP}(2qfa)$ and $\mathrm{QIP}(2qfa, \textit{poly-time})$ are closed under union.*

Proposition 5.8 is shown in the following fashion. For any two 2qfa-verifier QIP systems (P_1, V_1) and (P_2, V_2) that respectively correspond to L_1 and L_2, the verifier for $L_1 \cup L_2$ first asks a prover to choose the minimal index $i \in \{1, 2\}$ for which (P_i, V_i) accepts x (if i exists). The verifier then simulates the protocol (P_i, V_i) to check whether (P_i, V_i) truly accepts x. The formal proof below shows the validity of this protocol.

Proof of Proposition 5.8. We prove only the closure property of $\mathrm{QIP}(2qfa)$ under union because a similar proof shows the closure property of $\mathrm{QIP}(2qfa, \textit{poly-time})$. Take any two languages $L_1, L_2 \in \mathrm{QIP}(2qfa)$ and, for each $i \in \{1, 2\}$, choose a QIP system (P_i, V_i) that recognizes L_i with error probability $\leq \epsilon$, where ϵ is any fixed constant in $[0, 1/2)$. Without loss of generality, we may assume that the set of all inner states of V_1 and that of V_2 are mutually disjoint. Consider the following protocol of a new verifier V to determine whether any given input x belongs to $L_1 \cup L_2$. At the first move, V sends the communication symbol # to a prover without moving its tape head and waits for the prover's reply $i \in \{1,2\}$. Whenever the reply i is neither 1 nor 2, V immediately rejects x to prevent the prover from tampering. On the contrary, if i is truly in $\{1,2\}$, then V simulates V_i. On any input x in $L_1 \cup L_2$, our honest prover P first returns the minimal index $i \in \{1,2\}$ such that $x \in L_i$ and then behaves like P_i.

Henceforth, we prove that (P,V) recognizes $L_1 \cup L_2$. Let x be an arbitrary input. First, assume that $x \in L_1 \cup L_2$. Obviously, if $x \in L_1$, then the protocol (P,V) simulates (P_1,V_1) and otherwise, (P,V) simulates (P_2,V_2). Hence, V accepts x with probability at least $1-\epsilon$. Next, assume that $x \notin L_1 \cup L_2$. To maximize the acceptance probability of V on the input x, a dishonest prover should return either 1 or 2 (or their superposition). However, V simulates V_i when he receives i, and the computation paths of V that simulate V_1 and V_2 do not interfere with each other. Thus, for any prover P^*, (P^*,V) rejects x with probability at least $1-\epsilon$. This completes the proof. □

6 How Often is Measurement Performed?

Measurement is one of the most fundamental operations in quantum computation. Although a measurement is necessary to "know" the content of a target quantum state, the measurement collapses the quantum state and thus causes a quantum computation irreversible. Since a qfa uses only a finite amount of memory space, the number of times when measurements are conducted affects the computational power in general. Recall measure-once 1qfa's or mo-1qfa's from Section 1. We define an *mo-1qfa verifier* as a 1qfa verifier who does not perform any measurement until he applies the final unitary operation while visiting the right endmarker \$. This indicates that a measurement takes place only once after the verifier makes exactly $|x|+2$ moves on input x. We use the restriction $\langle mo\text{-}1qfa\rangle$ to indicate that a verifier is an mo-1qfa. This section makes a comparison between mo-1qfa verifiers and 1qfa verifiers in our QIP systems. As mentioned in Section 1, mo-1qfa's and 1qfa's are quite different in power because of the different numbers of measurement operations performed during a computation.

In what follows, we show that (i) the QIP systems with mo-1qfa verifiers are more powerful than mo-1qfa's alone and (ii) mo-1qfa verifiers are more prone to be fooled by dishonest provers than 1qfa verifiers.

Theorem 6.1 MO-1QFA $\subsetneqq$ QIP$(mo\text{-}1qfa)$ $\subsetneqq$ QIP$(1qfa)$.

Theorem 6.1 is a direct consequence of Proposition 6.2, which refers to a closure property of QIP$(mo\text{-}1qfa)$. Conventionally, a complexity class $\mathcal{C}$ is said to be *closed under complementation* if, for any language A over alphabet Σ in $\mathcal{C}$, its complement $\Sigma^* - A$ is also in $\mathcal{C}$.

Proposition 6.2 QIP$(mo\text{-}1qfa)$ *is not closed under complementation.*

Theorem 6.1 follows from Proposition 6.2 because QIP$(1qfa)$ (= REG) and MO-1QFA are known to be closed under complementation [34].

To prove Proposition 6.2, it suffices to show that (i) the unary language $L_a = \{a\}^* - \{\lambda\}$ is in QIP$_{1,1}(mo\text{-}1qfa)$ and (ii) the language $\{\lambda\}$ is not in QIP$(mo\text{-}1qfa)$. We first show that $L_a \in$ QIP$_{1,1}(mo\text{-}1qfa)$. We set out alphabets Σ and Γ as $\Sigma = \{a\}$ and $\Gamma = \{a, \#\}$. The transition of our verifier V is given in Table 2. At the first step, V stays in the initial inner state q_0 with passing the symbol # to a prover. If the input is λ, then, in reaching the endmarker \$ in state q_0, V enters the rejecting inner state q_{rej}. Clearly, V rejects the input with certainty no matter how the prover behaves. In the opposite case where the input is nonempty, if V scans a for the first time in the initial inner state q_0, V sends the symbol a to a prover and then enters the inner state q_1. When the honest prover modifies it back to #, V keeps the current inner state q_1 and the current communication symbol until V reads \$. Finally, V enters the accepting inner state q_{acc}. With the honest prover, V correctly accepts the input with certainty. Hence, (P,V) recognizes L_a with certainty.

We next prove the remaining claim that $\{\lambda\} \notin$ QIP$(mo\text{-}1qfa)$. More generally, we claim that no finite language belongs to QIP$(mo\text{-}1qfa)$. This claim is a consequence of the following lemma, which gives a more general limit to the power of the QIP systems with mo-1qfa verifiers.

Lemma 6.3 *Let L be a language over a nonempty alphabet Σ and let M be its minimal deterministic automaton. Assume that there exist an input symbol $a \in \Sigma$, an accepting inner state q_1, and a rejecting inner state q_2 satisfying: (1) if M reads a in the state q_1, then M enters the state q_2 and (2) if M reads a in the state q_2, then*

$V_{¢}\|q_0\rangle\|\#\rangle = \|q_0\rangle\|\#\rangle$	
$V_a\|q_0\rangle\|\#\rangle = \|q_1\rangle\|a\rangle$	$V_a\|q_1\rangle\|\#\rangle = \|q_1\rangle\|\#\rangle$
$V_{\$}\|q_0\rangle\|b\rangle = \|q_{rej}\rangle\|b\rangle$	$V_{\$}\|q_1\rangle\|\#\rangle = \|q_{acc}\rangle\|\#\rangle$

Table 2: Transitions of V for L_a with $b \in \{a, \#\}$. The unitary operator V_σ for each $\sigma \in \check{\Sigma}$ acts on the Hilbert space span$\{|q,\gamma\rangle \mid (q,\gamma) \in Q \times \Gamma\}$. The transition function δ of V is then induced by letting $\delta(q,\sigma,\gamma,q',\gamma',1) = \langle q',\gamma'|V_\sigma|q,\gamma\rangle$ for every $q, q' \in Q$ and $\gamma, \gamma' \in \Gamma$.

M stays in the state q_2. Figure 2 illustrates these transitions. The language L is then outside of QIP(mo-$1qfa$).

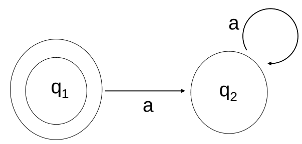

Figure 2: Transitions included in the minimal automaton for L

To prove Lemma 6.3, we use the following well-known result in [11].

Lemma 6.4 [11] *Let U be any unitary matrix and let ϵ be any positive real number. There exists a number $n \in \mathbb{N}^+$ such that $\|(I - U^n)x\|^2 < \epsilon$ for any vector x with $\|x\|^2 \leq 1$.*

We give the proof of Lemma 6.3.

Proof of Lemma 6.3. From the characteristics of the minimal automaton, there exists an input string $y \in \Sigma^*$ such that M enters q_1 after reading y and enters q_2 after reading ya^n for any positive integer n. Hereafter, we fix such a string y. Assume toward a contradiction that L belongs to QIP(mo-$1qfa$). Take a real number $\eta > 0$, an honest prover P, and an mo-1qfa verifier V satisfying the following: (P, V) accepts y with probability at least $1/2 + \eta$ while, for any prover P^* and any number $n \in \mathbb{N}^+$, (P^*, V) rejects ya^n with probability $\geq 1/2 + \eta$. Consider the following prover P' that works on input ya^n: P' first simulates P on input y while V is reading y and, whenever V passes a symbol s to P', P' returns the same s to V.

To lead to a contradiction, we utilize Lemma 6.4. Let $|\phi_y\rangle$ be the superposition of configurations obtained after V finishes reading y. Let V_b be the unitary operator corresponding to the transition of V while scanning symbol $b \in \check{\Sigma}$. By setting $\epsilon = \eta^2$, Lemma 6.4 guarantees the existence of a positive integer n such that $\||\phi_y\rangle - V_a^n|\phi_y\rangle\|^2 < \eta^2$, which equals $\||\phi_y\rangle - V_a^n|\phi_y\rangle\| < \eta$. For readability, we write $p_{acc}(y, P)$ for $p_{acc}(y, P, V)$. Since $p_{acc}(y, P)$ and $p_{acc}(ya^n, P')$ are obtained respectively by measuring the final superpositions $V_{\$}|\phi_y\rangle$ and $V_{\$}V_a^n|\phi_y\rangle$, we conclude:

$$|p_{acc}(y, P) - p_{acc}(ya^n, P')| \leq \|V_{\$}|\phi_y\rangle - V_{\$}V_a^n|\phi_y\rangle\| = \||\phi_y\rangle - V_a^n|\phi_y\rangle\| < \eta,$$

where the first inequality is a folklore (see, e.g., [44, Lemma 8]). Since $p_{acc}(y, P) \geq 1/2 + \eta$, it follows that $p_{acc}(ya^n, P') \geq (1/2 + \eta) - \eta = 1/2$, which contradicts our assumption that, for any prover P^*, $p_{acc}(ya^n, P^*) \leq 1/2 - \eta < 1/2$. Therefore, $L \notin$ QIP(mo-$1qfa$). □

Earlier, Brodsky and Pippenger [11] gave a group-theoretic characterization of MO-1QFA. Such a characterization is not yet known for QIP(mo-$1qfa$).

7 Is a Quantum Prover Stronger than a Classical Prover?

Our prover can perform any operation that quantum physics allows. We want to restrict the power of a prover. If the prover is limited to wield only "classical" power, we may call such a prover "classical." More precisely, a prover is called *classical* if the prover's move is dictated by a unitary operator whose entries are either 0s or

1s. By contrast, we sometimes refer to any standard prover as a *quantum prover*. Remember that any classical prover is a quantum prover. Although any classical-prover QIP system seems to be directly simulated by a similar QIP system using a quantum prover, it is not yet known that this is truly the case in general because, intuitively, more powerful the prover becomes, more easily may the weak verifier be convinced as well as fooled. Hereafter, the restriction $\langle$*c-prover*$\rangle$ indicates that a prover behaves classically. In our qfa-verifier QIP systems, a classical prover may play an essentially different role from a quantum prover's.

We consider the 1qfa-verifier case first. Similar to the quantum prover case, we can show that 1QFA $\subseteq$ QIP($1qfa$, *c-prover*). By expanding this containment, we can show the following stronger containment, which makes a bridge between quantum provers and classical provers.

Proposition 7.1 *QIP($1qfa$) $\subseteq$ QIP($1qfa$, c-prover).*

Proof. It is easy to show in a way similar to Proposition 4.2 that QIP($1qfa$, *c-prover*) contains all regular languages. Since QIP($1qfa$) = REG by Theorem 4.1, QIP($1qfa$, *c-prover*) therefore includes QIP($1qfa$). $\Box$

Whether QIP($1qfa$, *c-prover*) coincides with QIP($1qfa$) is unclear due to the soundness condition of a QIP system.

Next, we examine the 2qfa-verifier case. Unlike the 1qfa verifier case, any containment between QIP($2qfa$) and QIP($2qfa$, *c-prover*) is unknown. Nonetheless, we can easily show that QIP($2qfa$, *poly-time*, *c-prover*) contains 2QFA(*poly-time*). The proper inclusion REG $\subsetneqq$ QIP($2qfa$, *poly-time*, *c-prover*) is a direct consequence of the result in [32] that REG $\subsetneqq$ 2QFA(*poly-time*). The following theorem greatly strengthens this separation.

Theorem 7.2 *1. AM($2pfa$) $\subsetneqq$ QIP($2qfa$, c-prover).*
2. AM($2pfa$, poly-time) $\subsetneqq$ QIP($2qfa$, poly-time, c-prover) $\nsubseteq$ AM($2pfa$).

Proof. In the proof of Lemma 5.2, we have shown that $Pal_{\#}$ is in QIP($2qfa$, *poly-time*). Notice that the same proof works for classical provers. This places $Pal_{\#}$ in QIP($2qfa$, *poly-time*, *c-prover*). Hence, similar to Theorem 5.1, the separation between AM($2pfa$) and QIP($2qfa$, *poly-time*, *c-prover*) naturally follows. This separation further leads to the inequality between AM($2pfa$) and QIP($2qfa$, *c-prover*) (also between AM($2pfa$, *poly-time*) and QIP($2qfa$, *poly-time*, *c-prover*)). Therefore, in this proof, it suffices to show that AM($2pfa$) $\subseteq$ QIP($2qfa$, *c-prover*). Since our proof works for any time-bounded case, we also obtain the remaining claim that AM($2pfa$, *poly-time*) $\subseteq$ QIP($2qfa$, *poly-time*, *c-prover*).

The important starting point is the fact that the complexity class AM($2pfa$) can be characterized by bounded-error finite automata with probabilistic and nondeterministic moves. Such an automaton is called a *2npfa* in [14]. Let L be any language in AM($2pfa$) over alphabet Σ. Take a finite automaton $M = (Q, \Sigma, \delta_M)$ with nondeterministic states and probabilistic states that recognizes L with error probability at most ϵ, where $0 \leq \epsilon < 1/2$. To simplify our proof, we make two inessential assumptions for M's head move. Assume that (i) M's head always moves either to the right or to the left and (ii) whenever M tosses a fair coin, the head moves only to the right. Based on this M, we shall construct a QIP system (P, V) for L.

Let x be any input of length n. The verifier V carries out the following procedure, in which V simulates M step by step with $Q' = \{p, \hat{p} \mid p \in Q\}$ as the set of inner states and $\Gamma = (Q' \times \{\pm 1\}) \cup \{\#, \$\}$ as the communication alphabet, where $\hat{p}$ is a new inner state associated with p and \$ is a new non-blank symbol. Consider any step at which M tosses a fair coin in probabilistic state p by the transition $\delta_M(p, \sigma) = \{(p_0, 1), (p_1, 1)\}$ for certain distinct states $p_0, p_1 \in Q$. The verifier V checks whether the communication cell is blank. If not, V rejects x at this simulation step; otherwise, V makes the corresponding transition $V_\sigma|p\rangle|\#\rangle = \frac{1}{\sqrt{2}}(|p_0\rangle|(p,1)\rangle + |p_1\rangle|(p,1)\rangle)$. Here, V_σ is the unitary operator defined by $\delta(p, \sigma, \gamma, q, \gamma', D(q)) = \langle q, \gamma'|V_\sigma|p, \gamma\rangle$ with the transition function δ of V and D is the function from Q' to $\{0, \pm 1\}$. The verifier expects a prover to erase the symbol p in the communication cell by overwriting it with the blank symbol $\#$. This erasure guarantees V's move to be unitary.

Next, consider any step at which M makes a nondeterministic choice in state p by the transition $\delta_M(p, \sigma) = \{(p_0, d_0), (p_1, d_1), \ldots, (p_m, d_m)\}$, where $m \in \mathbb{N}$. Notice that a deterministic move is treated as a special case of a nondeterministic move. In this case, V takes two steps to simulate M's move. The verifier V enters a rejecting inner state immediately unless the communication cell contains the blank symbol. Now, assume that the communication cell is blank. Without moving its head, V first sends the designated symbol \$ to a prover, requesting a pair (p', d') in $Q \times \{\pm 1\}$ to return. This is done by the transition $V_\sigma|p\rangle|\#\rangle = |\hat{p}\rangle|\$\rangle$. The verifier forces a prover to return a valid nondeterministic choice (i.e., $(p', d') \in \delta_M(p, \sigma)$) by entering a rejecting inner state if the prover writes any other symbol. Once V receives a valid pair (p', d'), V makes the transition $V_\sigma|\hat{p}\rangle|(p', d')\rangle = |p'\rangle|(\hat{p}, d')\rangle$ and expects a prover to erase the communication symbol $(\hat{p}, d')$.

The honest prover P must blank the communication cell at the end of each simulation step of V and return a "correct" nondeterministic choice on request of the verifier V. If $x \in L$, there are a series of nondeterministic choices along which M accepts x with probability at least $1-\epsilon$. With the help of the honest prover P, V can successfully simulate M with the same error probability. Consider the case where $x \notin L$, on the contrary. In this case, no matter how nondeterministic choices are made, M rejects x with probability at least $1-\epsilon$. Take a dishonest classical prover P^* that maximizes the acceptance probability of V on x. This prover P^* must clear out the communication cell whenever V asks him to do so since, otherwise, V immediately rejects x. Since P^* is classical, all the computation paths of V have nonnegative amplitudes which cause only constructive interference. This indicates that P^* cannot annihilate any existing computation path of V. On request for a nondeterministic choice, P^* must return any one of valid nondeterministic choices. With a series of nondeterministic choices of P^*, if V rejects x with probability less than $1-\epsilon$, then our simulation implies that M rejects x with probability less than $1-\epsilon$. This is a contradiction against our assumption. Hence, V rejects x with probability at least $1-\epsilon$. Therefore, (P,V) is a $(1-\epsilon,1-\epsilon)$-QIP system for L. □

In the above proof, we cannot replace a classical prover by a quantum prover. The major reason is that a quantum prover may (i) return a superposition of two nondeterministic choices instead of choosing one of the two choices and (ii) use negative amplitudes to make the verifier's quantum simulation destructive.

In the end of this section, we present a QIP protocol with a classical prover for the non-regular language *Center*, which is known to be in AM($2pfa$) but not in AM($2pfa$, *poly-time*) [17]. In our QIP system, a prover signals the location of the center bit of an input and then a verifier tests the correctness of the location by employing the quantum Fourier transformation (QFT, in short) in a fashion similar to [32].

Lemma 7.3 *For any* $\epsilon \in (0,1)$, *Center* $\in \mathrm{QIP}_{1,1-\epsilon}(2qfa$, *poly-time*, *c-prover*).

Proof. Let ϵ be any error bound in the real interval $(0,1)$ and set $N = \lceil 1/\epsilon \rceil$. We give a QIP protocol witnessing the membership of *Center* to $\mathrm{QIP}_{1,1-\epsilon}(2qfa$, *poly-time*, *c-prover*). Let $\Sigma = \{0,1\}$ be our input alphabet and let $\Gamma = \{\#,1\}$ be our communication alphabet. Our QIP protocol comprises four phases. Let x be an arbitrary input. In the first phase, the verifier checks whether $|x|$ is odd by moving the head toward the right endmarker \$ together with switching two inner states q_0 and q_1. To make deterministic moves, the verifier forces a prover to return only the blank symbol #. When $|x|$ is odd, the verifier enters the state q_3 after stepping back to ¢. Hereafter, we consider only the case where input x has an odd length.

In the second phase, V moves its head rightward by passing the communication symbol # to a prover until V receives 1 from the prover. Receiving 1 from the prover, V rejects x unless scanning 1 in the input tape. Otherwise, the third phase starts. During the third and fourth phases, whenever the prover changes the communication symbol 1 to #, V immediately rejects the input. Assume that the head is now scanning 1. In the third phase, the computation splits into N parallel branches (the first split) generating the N distinct inner states $r_{1,0}, r_{2,0}, \ldots, r_{N,0}$ with equal amplitudes $1/\sqrt{N}$. The head then moves deterministically toward the right endmarker \$ in the following manner: along the jth path ($1 \leq j \leq N$) associated with the inner state $r_{j,0}$, the head idles for $2(N-j)$ steps in each tape cell before moving to the next one. When the head reaches \$, it steps back two cells and starts the fourth phase. During the fourth phase, the head along the jth path keeps moving leftward by idling in each cell for j steps until the head reaches ¢. At the left endmarker, the computation splits again into N parallel branches by the QFT (the second split), yielding either the accepting inner state t_N or one of the rejecting inner states $\{t_j \mid 1 \leq j < N\}$.

The formal description of the transitions of V is given in Table 3. From this table, it is not difficult to check that the verifier is well-formed (i.e., U_δ^x is unitary for every $x \in \Sigma^*$). The honest prover P should return 1 exactly at the time when the verifier scans the center bit of an input and at the time when the verifier sends # during the third and fourth phases. At any other step, P should perform the identity operation.

The following is the proof of the completeness and soundness of the QIP system (P,V) for *Center*. First, consider a positive instance x, which is of the form $y1z$ for certain strings y and z of the same length, say n. Since the honest prover P signals when the verifier reads the center bit of x, the first split occurs exactly after n steps of V from the start of the second phase. Along the jth path ($1 \leq j \leq N$) chosen at the first split, V idles for $2n(N-j)$ steps while reading y and also idles for $(|x|-1)j$ steps while reading the whole input except for its rightmost symbol. Overall, the idling time elapses for the duration of $2n(N-j) + 2nj = 2nN$, which is independent of j. Hence, all the N^2 paths created at the two splits have the same length. The QFT then converges them to the verifier's visible accepting configuration $|t_N\rangle|\#\rangle$. Therefore, V accepts x with probability 1.

$V_{¢}\vert q_0\rangle\vert\#\rangle = \vert q_0\rangle\vert\#\rangle$	$V_{\$}\vert q_0\rangle\vert\#\rangle = \vert q_{rej,0}\rangle\vert\#\rangle$
$V_{¢}\vert q_2\rangle\vert 1\rangle = \vert q_{rej,0}\rangle\vert\#\rangle$	$V_{\$}\vert q_1\rangle\vert\#\rangle = \vert q_2\rangle\vert\#\rangle$
$V_{¢}\vert s_{j,0}\rangle\vert 1\rangle = \frac{1}{\sqrt{N}}\sum_{l=1}^{N}\exp(2\pi\imath jl/N)\vert t_l\rangle\vert\#\rangle$ $(1 \leq j \leq N)$	$V_{\$}\vert q_0\rangle\vert 1\rangle = \vert q_{rej,1}\rangle\vert\#\rangle$
$V_{¢}\vert q_2\rangle\vert\#\rangle = \vert q_3\rangle\vert\#\rangle$	$V_{\$}\vert q_1\rangle\vert 1\rangle = \vert q_{rej,1}\rangle\vert 1\rangle$
	$V_{\$}\vert r_{j,0}\rangle\vert 1\rangle = \vert s'_{j,0}\rangle\vert 1\rangle$ $(1 \leq j \leq N)$
$V_b\vert q_0\rangle\vert\#\rangle = \vert q_1\rangle\vert\#\rangle$	$V_b\vert q_1\rangle\vert\#\rangle = \vert q_0\rangle\vert\#\rangle$
$V_b\vert q_0\rangle\vert 1\rangle = \vert q_{rej,0}\rangle\vert\#\rangle$	$V_b\vert q_1\rangle\vert 1\rangle = \vert q_{rej,0}\rangle\vert 1\rangle$
$V_b\vert q_2\rangle\vert\#\rangle = \vert q_2\rangle\vert\#\rangle$	$V_b\vert q_2\rangle\vert 1\rangle = \vert q_{rej,1}\rangle\vert 1\rangle$
$V_b\vert q_3\rangle\vert\#\rangle = \vert q_3\rangle\vert\#\rangle$	
$V_1\vert q_3\rangle\vert 1\rangle = \frac{1}{\sqrt{N}}\sum_{j=1}^{N}\vert r_{j,0}\rangle\vert\#\rangle$	$V_0\vert q_3\rangle\vert 1\rangle = \vert q_{rej,-1}\rangle\vert\#\rangle$
$V_b\vert r_{j,0}\rangle\vert 1\rangle = \vert r'_{j,N-j}\rangle\vert 1\rangle$ $(1 \leq j \leq N-1)$	$V_b\vert r_{j,0}\rangle\vert\#\rangle = \vert q_{rej,j}\rangle\vert 1\rangle$ $(1 \leq j \leq N-1)$
$V_b\vert r_{j,k}\rangle\vert 1\rangle = \vert r'_{j,k}\rangle\vert 1\rangle$ $(1 \leq k \leq N-j,\ 1 \leq j \leq N-1)$	
$V_b\vert r'_{j,k}\rangle\vert 1\rangle = \vert r_{j,k-1}\rangle\vert 1\rangle$ $(2 \leq k \leq N-j,\ 1 \leq j \leq N-1)$	
$V_b\vert r'_{j,1}\rangle\vert 1\rangle = \vert r_{j,0}\rangle\vert 1\rangle$, $(1 \leq j \leq N)$	$V_b\vert r_{N,0}\rangle\vert 1\rangle = \vert r_{N,0}\rangle\vert 1\rangle$
$V_b\vert s'_{j,0}\rangle\vert 1\rangle = \vert s_{j,0}\rangle\vert\#\rangle$ $(1 \leq j \leq N)$	$V_b\vert s_{j,0}\rangle\vert 1\rangle = \vert s_{j,j}\rangle\vert 1\rangle$ $(1 \leq j \leq N)$
$V_b\vert s_{j,k}\rangle\vert 1\rangle = \vert s_{j,k-1}\rangle\vert 1\rangle$ $(2 \leq k \leq j,\ 1 \leq j \leq N)$	
$V_b\vert s_{j,1}\rangle\vert 1\rangle = \vert s_{j,0}\rangle\vert 1\rangle$ $(1 \leq j \leq N)$	$V_b\vert s_{j,0}\rangle\vert\#\rangle = \vert q_{rej,N+j}\rangle\vert\#\rangle$ $(1 \leq j \leq N)$
$D(q_0) = D(q_1) = D(q_3) = 1$, $D(q_2) = -1$	$D(r_{j,0}) = 1$ $(1 \leq j \leq N)$
$D(r_{j,k}) = D(r'_{j,k}) = 0$ $(1 \leq j \leq N-1,\ k \neq 0)$	$D(s_{j,0}) = D(s'_{j,0}) = -1$ $(1 \leq j \leq N)$
$D(s_{j,k}) = 0$ $(1 \leq j \leq N,\ k \neq 0)$	$D(t_j) = 0$ $(1 \leq j \leq N)$

Table 3: Transitions of V for *Center* with $b \in \{0,1\}$. In this table, t_N is the only accepting inner state while $q_{rej,j}$ $(-1 \leq j \leq 2N-1)$ and t_l $(1 \leq l < N)$ are rejecting inner states. The table, however, excludes obvious transitions to rejecting inner states when a prover changes the communication symbol 1 to # during the third and fourth phases. The transition function δ is induced from V as $\delta(q,\sigma,\gamma,q',\gamma',d) = \langle q',\gamma'|V_\sigma|q,\gamma\rangle$ if $D(q') = d$ and 0 otherwise.

On the contrary, suppose that $x = y0z$, where $|y| = |z| = n$. Consider the second, third and fourth phases. To minimize the rejection probability, a dishonest prover P^* should send the symbol 1 at the moment when V scans 1 in the input tape in the second phase and then maintain 1 after the first split because, otherwise, V immediately rejects x and no classical prover passes both 1 and # in a form of superposition. Now assume that the eth symbol of x is 1 and P^* sends 1 during the eth interaction, where $1 \leq e \leq 2n+1$. Note that $e \neq n+1$ because the center bit of x is 0. For any $j \in [1,N]_{\mathbb{Z}}$, let p_j be the computation path following the jth branch generated at the first split. Along this path p_j toward the left endmarker ¢, the idling time totals $2(|x|-e)(N-j)+2nj = 2(n+1-e)(N-j)+2nN$. For any distinct values j and j', the two paths p_j and $p_{j'}$ have different lengths. For each of such paths, the QFT further generates N parallel paths; however, only one of them reach $|t_N\rangle|\#\rangle$. Hence, the probability of V reaching such an acceptance configuration is no more than $1/N^2$. Since there are N paths $\{p_j\}_{1\leq j\leq N}$, the overall acceptance probability is at most $N \times (1/N^2) = 1/N$. It is easy to see that V rejects x with probability $\geq 1 - 1/N \geq 1 - \epsilon$. □

8 What If a Verifier Reveals His Private Information?

The strength of a prover's strategy hinges on the amount of the information that a verifier reveals. For instance, when a verifier makes only one-way communication (as in the proof of Lemma 5.2), no prover gains more than the information on the number of the verifier's moves. The prover therefore knows little of the verifier's configurations. In Babai's "public" IP systems by contrast, a verifier completely reveals his configurations. The notion of "public coins" forces the verifier to pass *only* his choice of next moves, which allows the prover to reconstruct the verifier's computation. In this section, we consider a straightforward analogy of public IP systems in the quantum setting and call our QIP system *public* for convenience. Formally, we introduce a *public QIP system* as follows.

Definition 8.1 A qfa-verifier QIP system (P,V) is called *public* if the verifier V writes his choice of non-halting inner state and head direction in the communication cell at every step; that is, the verifier's transition function

δ satisfies that, for any x, q, k, and γ, $U^x_\delta|q,k,\gamma\rangle = \sum_{q',\xi,d}\delta(q,x_{(k)},\gamma,q',\xi,d)|q',k+d \pmod{n+2},\xi\rangle$, where $\xi=(q',d)$ whenever q' is a non-halting inner state.

In particular, when the verifier V is a 1qfa, we can omit the head-direction information d from the communication symbol $\xi=(q',d)$ in the above definition since V always moves its head to the right. To emphasize the public QIP system, we use the restriction $\langle public\rangle$.

Let us begin our study on the complexity class QIP$(1qfa, public)$.

Proposition 8.2 $\mathrm{QIP}_{1,1}(1qfa, public) \nsubseteq$ 1QFA.

Recall the language *Zero*. Proposition 8.2 is obtained by proving that *Zero* belongs to QIP$(1qfa, public)$ since *Zero* resides outside of 1QFA [32]. The following proof exploits the prover's ability to inform the location of the rightmost bit 0 of an instance in *Zero*.

Proof of Proposition 8.2. We want to show that *Zero* has an error-free public QIP system (P,V) with a 1qfa verifier. Since no 1qfa recognizes the language *Zero* [32], we therefore obtain the proposition. To describe the desired protocol (P,V), let $\Sigma=\{0,1\}$ be its input alphabet and let $Q_{non}=\{q_0,q_1\}$, $Q_{acc}=\{q_{acc,0},q_{acc,1},q_{acc,-1}\}$ and $Q_{rej}=\{q_{rej,0},q_{rej,1},q_{rej,-1}\}$ be respectively the sets of all non-halting inner states, accepting inner states and rejecting inner states of V.

As mentioned before, we abbreviate communication symbol $(q,1)$ for $q\in Q$ as q since V's head direction is always $+1$. Our communication alphabet Γ is thus $\{\#,q_0,q_1\}$. The protocol of V is described in the following. Let $x=yb$ be any input string, where $b\in\{0,1\}$. The verifier V stays in the initial state q_0 by sending the communication symbol q_0 to a prover until the prover returns #. Whenever V receives #, he immediately rejects x if its current scanning symbol is different from 0. On the contrary, if V is scanning 0, then he waits for the next tape symbol. If the next symbol is \$, then he accepts x; otherwise, he rejects x. See Table 4 for the formal description of V's transitions. Our honest prover P does not alter the communication cell until V reaches the end of ¢y and he must return # exactly when V reads the rightmost symbol of ¢y.

$V_{¢}\|q_0\rangle\|\#\rangle = \|q_0\rangle\|q_0\rangle$	$V_1\|q_0\rangle\|q_0\rangle = \|q_0\rangle\|q_0\rangle$	$V_0\|q_0\rangle\|q_0\rangle = \|q_0\rangle\|q_0\rangle$
$V_{\$}\|q_0\rangle\|q_i\rangle = \|q_{rej,i}\rangle\|\#\rangle$	$V_1\|q_0\rangle\|\#\rangle = \|q_{rej,-1}\rangle\|\#\rangle$	$V_0\|q_0\rangle\|\#\rangle = \|q_1\rangle\|q_1\rangle$
$V_{\$}\|q_1\rangle\|q_i\rangle = \|q_{acc,i}\rangle\|\#\rangle$	$V_1\|q_1\rangle\|q_i\rangle = \|q_{rej,i}\rangle\|q_0\rangle$	$V_0\|q_1\rangle\|q_i\rangle = \|q_{rej,i}\rangle\|q_0\rangle$
	$V_1\|q_0\rangle\|q_1\rangle = \|q_{rej,1}\rangle\|\#\rangle$	$V_0\|q_0\rangle\|q_1\rangle = \|q_{rej,1}\rangle\|\#\rangle$

Table 4: Transitions of V for *Zero* with $i\in\{0,\pm1\}$. The symbol q_{-1} denotes #.

It still remains to prove that (P,V) recognizes *Zero* with certainty. Consider the case where our input x is of the form $y0$ for a certain string y. Since $x\in Zero$, the honest prover P returns # exactly when V reads the rightmost symbol of ¢y. This information helps V locate the end of y. Now, V confirms that the current scanning symbol is 0 and then enters an accepting inner state with probability 1 after it encounters the right endmarker. On the contrary, assume that $x=y1$. Clearly, the best adversary P^* needs to return either q_0 or # (or their superposition). If P^* keeps returning q_0, then V eventually rejects x and increases the rejection probability. Since V's computation is deterministic, this only weakens the strategy of P^*. To make the best of the adversary's strategy, P^* must return the communication symbol # before V reaches \$. Nonetheless, although P^* returns it, V is designed to lead to a rejecting inner state. Therefore, the QIP system (P,V) recognizes *Zero* with certainty. □

A *1-way reversible finite automaton* (*1rfa*, in short) is a 1qfa whose transition amplitudes are either 0 or 1. Let 1RFA denote the collection of all languages recognized by certain 1rfa's. As Ambainis and Freivalds [5] showed, 1RFA is characterized as the collection of all languages that can be recognized by 1qfa's with success probability $\geq 7/9+\epsilon$ for certain numbers $\epsilon>0$.

Proposition 8.3 1RFA $\subsetneq \mathrm{QIP}_{1,1}(1qfa, public)$.

Proof. We first show that 1RFA $\subseteq \mathrm{QIP}_{1,1}(1qfa, public)$. Take an arbitrary set L recognized by a 1rfa $M=(Q,\Sigma,q_0,Q_{acc},Q_{rej},\delta_M)$. Without loss of generality, we can assume that, in the transition of M, the initial state q_0 appears only when M starts its computation.

$V_{¢}\|q_0\rangle\|\#\rangle = \|q\rangle\|q\rangle$ if $\delta_M(q_0, ¢) = q$
$V_b\|p\rangle\|p\rangle = \|q\rangle\|q\rangle$ if $\delta_M(p, b) = q \in Q_{non}$
$V_b\|p\rangle\|p\rangle = \|q\rangle\|\#\rangle$ if $\delta_M(p, b) = q \in Q_{acc} \cup Q_{rej}$ and $b \neq ¢$

Table 5: Transitions of V for L with $b \in \Sigma$ and $p, q \in Q$

The protocol of V is given as follows. Assume that V is in inner state p scanning symbol b. Whenever M changes its inner state from p to q while scanning b, V does so by sending the communication symbol p to a prover if q is a non-halting inner state. As soon as V finds that the communication symbol has been altered by the prover, V immediately rejects the input. Table 5 gives the list of V's unitary operators induced from M's transition function δ_M. The honest prover P is the one who does not alter any communication symbol. On any input x, the QIP system (P, V) clearly accepts x with certainty if $x \in L$. Consider the opposite case where $x \notin L$. It is easy to see that the best strategy for a dishonest classical prover P^* is to keep any communication symbol unchanged because any alteration of a communication symbol causes V to reject x immediately. Even with such a prover P^*, V rejects x with certainty. Therefore, (P, V) recognizes L with certainty. Since L is arbitrary, we obtain the desired inclusion $\mathrm{1RFA} \subseteq \mathrm{QIP}_{1,1}(1qfa, \mathit{public})$. Finally, the separation between 1RFA and $\mathrm{QIP}_{1,1}(1qfa, \mathit{public})$ comes from Proposition 8.2. This completes the proof. $\square$

We further examine public QIP systems with 2qfa verifiers. Similar to Theorem 7.2(2), we can give the following separation.

Theorem 8.4 *1.* $\mathrm{QIP}(2qfa, \mathit{public}, \mathit{poly\text{-}time}) \nsubseteq \mathrm{AM}(2pfa, \mathit{poly\text{-}time})$.

2. $\mathrm{QIP}(2qfa, \mathit{public}, \mathit{poly\text{-}time}, \mathit{c\text{-}prover}) \nsubseteq \mathrm{AM}(2pfa, \mathit{poly\text{-}time})$.

A language that separates the public QIP systems from $\mathrm{AM}(2pfa, \mathit{poly\text{-}time})$ is $Upal$. Since $Upal$ resides outside of $\mathrm{AM}(2pfa, \mathit{poly\text{-}time})$ [17] and $Upal$ belongs to $\mathrm{2QFA}(\mathit{poly\text{-}time})$ [32], the separation $\mathrm{2QFA}(\mathit{poly\text{-}time}) \nsubseteq \mathrm{AM}(2qfa, \mathit{poly\text{-}time})$ follows immediately. This separation, however, does not directly imply Theorem 8.4 because it is not clear whether $\mathrm{2QFA}(\mathit{poly\text{-}time})$ is included in $\mathrm{QIP}(2qfa, \mathit{public}, \mathit{poly\text{-}time})$ or in $\mathrm{QIP}(2qfa, \mathit{public}, \mathit{poly\text{-}time}, \mathit{c\text{-}prover})$. Therefore, we still need to prove in Lemma 8.5 that $Upal$ is indeed in both $\mathrm{QIP}(2qfa, \mathit{public}, \mathit{poly\text{-}time})$ and $\mathrm{QIP}(2qfa, \mathit{public}, \mathit{poly\text{-}time}, \mathit{c\text{-}prover})$. Our public QIP system for $Upal$, nevertheless, is essentially a slight modification of the 2qfa given in [32] for $Upal$.

Lemma 8.5 *For any constant* $\epsilon \in (0, 1]$, $Upal \in \mathrm{QIP}_{1,1-\epsilon}(2qfa, \mathit{public}, \mathit{poly\text{-}time}) \cap \mathrm{QIP}_{1,1-\epsilon}(2qfa, \mathit{public}, \mathit{poly\text{-}time}, \mathit{c\text{-}prover})$.

Proof. We show that $Upal$ belongs to $\mathrm{QIP}_{1,1-\epsilon}(2qfa, \mathit{public}, \mathit{poly\text{-}time})$ since the proof that $Upal$ belongs to $\mathrm{QIP}_{1,1-\epsilon}(2qfa, \mathit{public}, \mathit{poly\text{-}time}, \mathit{c\text{-}prover})$ is similar. Let $N = \lceil 1/\epsilon \rceil$. We define our public QIP system (P, V) as follows. The verifier V acts as follows. In the first phase, it determines whether an input x is of the form 0^m1^n. The rest of the verifier's algorithm is similar in essence to the one given in the proof of Lemma 7.3. In the second phase, V generates N branches with amplitude $1/\sqrt{N}$ by entering N different inner states, say $r_1, r_2, \ldots, r_N$. In the third phase, along the jth branch starting with r_j ($j \in [1, N]_{\mathbb{Z}}$), the head idles for $N - j$ steps at each tape cell containing 0 and idles for j steps at each cell containing 1 until the head finishes reading 1s. In the fourth phase, V applies the QFT to collapse all the paths to a single accepting inner state if $m = n$. Otherwise, all the paths do not interfere with each other since the head reaches the right endmarker at different times along different branches. During the first and second phases, V publicly reveals the information (q', d') on his next move and then checks whether the prover rewrites it with a different symbol. To constrain the prover's strategy, V immediately enters a rejecting inner state if the prover alters the content of the communication cell. The honest prover P always applies the identity operation at every step.

We show the completeness and soundness for our QIP system (P, V). This is done in a fashion similar to the proof of Lemma 7.3. With the honest prover for any input $x \in Upal$, (P, V) obviously accepts x with probability 1. Assume that $x = 0^m1^n$ with $m \neq n$. Consider a dishonest prover P^* who maximizes the acceptance probability of V on x. Against V's rejection criteria, the prover P^* cannot change the content of the communication cell at any step. Since the head arrives at the endmarker \$ at different moments, no two branches apply the QFT simultaneously. This makes it impossible for P^* to force two or more branches to interfere. Along each branch, the probability that V enters an accepting inner state is at most $1/N^2$. Therefore,

(P^*, V) rejects x with probability bounded below by $1 - N \cdot (1/N^2)$, which is at least $1 - \epsilon$. □

As noted in the proof of Theorem 7.2, the classical public IP systems with 2pfa verifiers can be characterized by alternating automata that make nondeterministic moves and probabilistic moves. A natural question is whether our public QIP systems have a similar characterization in terms of a certain variation of qfa's. Moreover, Condon et al. [14] proved that any language in AM($2pfa$, *poly-time*) has polylogarithmic 1-tiling complexity. What is the 1-tiling complexity of languages in QIP($2qfa$, *public*, *poly-time*)?

9 How Many Interactions are Necessary?

In the previous sections, we have shown that quantum interactions between a prover and a qfa verifier notably enhance the qfa's ability to recognize certain types of languages. Since our basic model of QIP systems forces a verifier to communicate with a prover at every move, it is natural to ask whether such interactions are truly necessary. Throughout this section, we carefully examine the number of interactions between a prover and a verifier in a QIP system. To study such a number, we need to modify our basic systems so that a prover should alter a communication symbol in the communication cell exactly when the verifier asks the prover to do so. For such a modification, we first look into the IP systems of Dwork and Stockmeyer [17]. In their system, a verifier is allowed to do computation silently at any chosen time with no communication with a prover. The verifier interacts with the prover only when the help of the prover is needed. We interpret the verifier's silent mode as follows: if the verifier V does not wish to communicate with the prover, he writes a special communication symbol in the communication cell to signal the prover that he needs no help from the prover. Simply, we use the blank symbol # to condition that the prover is prohibited to tailor the content of the communication cell.

We formally introduce a new QIP system, in which no malicious prover P is permitted to cheat a verifier by tampering with the symbol # willfully. To describe a "valid" prover P independent of the choice of a verifier, we require the prover's strategy $P_x = \{U^x_{P,i}\}_{i\in\mathbb{N}^+}$ on input x, acting on the prover's visible configuration space $\mathcal{M} \otimes \mathcal{P}$, to satisfy the following condition. For each $i \in \mathbb{N}$, let $S_0 = \{\#^\infty\}$ and let S_i be the collection of all $y \in \Delta^\infty_{fin}$ such that, for a certain element $z \in S_{i-1}$ and certain communication symbols $\sigma, \tau \in \Gamma^*$, the superposition $U^x_{P,i}|\sigma\rangle|z\rangle$ contains the configuration $|\tau\rangle|y\rangle$ of non-zero amplitude. Note that these S_i's are all finite. For every $i \in \mathbb{N}^+$ and every $y \in S_{i-1}$, we require the existence of a pure quantum state $|\psi_{x,y,i}\rangle$ in the Hilbert space spanned by $\{|z\rangle \mid z \in \Delta^\infty_{fin}\}$ for which $U^x_{P,i}|\#\rangle|y\rangle = |\#\rangle|\psi_{x,y,i}\rangle$. A prover who meets this condition is briefly referred to as *committed*. A trivial example of such a committed prover is the prover P_I, who always applies the identity operation. A committed prover lets the verifier safely make a number of moves without any "direct" interaction with him. Observe that this new model with committed provers is in essence close to the circuit-based QIP model discussed in Section 3.2. We name our new model an *interaction-bounded QIP system* and use the new notation $\mathrm{QIP}^{\#}(1qfa)$ for the class of all languages recognized with bounded error by such interaction-bounded QIP systems with 1qfa verifiers. Since $\mathrm{QIP}^{\#}(1qfa)$ naturally contains $\mathrm{QIP}(1qfa)$, our interaction-bounded QIP systems can also recognize the regular languages. This simple fact will be used later.

Lemma 9.1 $\mathrm{REG} \subseteq \mathrm{QIP}^{\#}(1qfa)$.

Next, we need to clarify the meaning of the number of interactions. Consider any non-halting global configuration in which V on input x communicates with a prover (i.e., writes a non-blank symbol in the communication cell). For convenience, we call such a global configuration a *query configuration* and, at a query configuration, V is said to *query* a word to a prover. The *number of interactions* in a given computation means the maximum number, over all computation paths γ, of all the query configurations of non-zero amplitudes along its computation path γ. Let L be any language and let (P, V) be any interaction-bounded QIP system recognizing L. We say that the QIP protocol (P, V) *makes i interactions* on input x if i equals the number of interactions during the computation of (P, V) on x. Furthermore, we call (P, V) *k-interaction bounded* if, for every x, if $x \in L$ then (P, V) makes at most k interactions on input x^{**} and otherwise, for every committed prover P^*, (P^*, V) makes at most k interactions on input x. At last, let $\mathrm{QIP}^{\#}_k(1qfa)$ denote the class of all languages recognized with bounded error by k-interaction bounded QIP systems with 1qfa verifiers. Obviously, $1\mathrm{QFA} \subseteq \mathrm{QIP}^{\#}_k(1qfa) \subseteq \mathrm{QIP}^{\#}_{k+1}(1qfa) \subseteq \mathrm{QIP}^{\#}(1qfa)$ for any number $k \in \mathbb{N}$. In particular,

**Instead, we may possibly consider a stronger condition like: for every x and every committed prover P^*, (P^*, V) makes at most k interactions.

$\mathrm{QIP}_0^{\#}(1qfa) = \mathrm{1QFA}$.

As the main theorem of this section, we show in Theorem 9.2 that (i) 1-iteration helps a verifier but (ii) 1-iteration does not achieve the full power of $\mathrm{QIP}^{\#}(1qfa)$.

Theorem 9.2 $\mathrm{QIP}_0^{\#}(1qfa) \subsetneq \mathrm{QIP}_1^{\#}(1qfa) \subsetneq \mathrm{QIP}^{\#}(1qfa)$.

Theorem 9.2 is a direct consequence of Lemma 9.3 and Proposition 9.4. For the first inequality of Theorem 9.2, we use the language *Odd* defined as the set of all binary strings of the form 0^m1z, where $m \in \mathbb{N}$, $z \in \{0,1\}^*$, and z contains an odd number of 0s. Since $Odd \notin \mathrm{1QFA}$ [6], it is enough for us to show in Lemma 9.3 that *Odd* belongs to $\mathrm{QIP}_1^{\#}(1qfa)$. For the second inequality, we shall demonstrate in Proposition 9.4 that $\mathrm{QIP}_1^{\#}(1qfa)$ does not include the regular language *Zero*. Since $\mathrm{REG} \subseteq \mathrm{QIP}^{\#}(1qfa)$ by Lemma 9.1, *Zero* belongs to $\mathrm{QIP}^{\#}(1qfa)$ and we therefore obtain the desired separation.

The rest of this section is devoted to prove Lemma 9.3 and Proposition 9.4. As the first step, we prove Lemma 9.3.

Lemma 9.3 $Odd \in \mathrm{QIP}_1^{\#}(1qfa)$.

Proof. We give a 1-interaction bounded QIP system (P,V) that recognizes *Odd*. Now, let $\Sigma = \{0,1\}$ and $\Gamma = \{\#, a\}$ be respectively the input alphabet and the communication alphabet for (P,V). Let $Q = \{q_0, q_1, q_2, q_{acc}, q_{rej,0}, q_{rej,1}\}$ be the set of V's inner states with $Q_{acc} = \{q_{acc}\}$ and $Q_{rej} = \{q_{rej,0}, q_{rej,1}\}$. The protocol of the verifier V is given as follows. With no query to a committed prover, V continues to read the input symbols until the head scans 1 in the input tape. When V reads 1, V queries the symbol a to a committed prover. If the prover returns a, then V immediately rejects the input. Otherwise, the verifier checks whether the substring of the input after 1 includes an odd number of 0s. This check can be done by the verifier alone. Table 6 gives the formal description of V's transitions. The honest prover P, whenever receiving the symbol a from the verifier, returns the symbol # and writes a in the first blank cell of his private tape. Technically speaking, to make P unitary, we need to map visible configuration $|\#\rangle|y\rangle$ for certain y's not appeared yet in P's private tape to superposition $|a\rangle|\phi_{x,y}\rangle$ with an appropriate vector $|\phi_{x,y}\rangle$. By a right implementation, we can make P a committed prover.

$V_¢|q_0\rangle|\#\rangle = |q_0\rangle|\#\rangle$	$V_0|q_0\rangle|\#\rangle = |q_0\rangle|\#\rangle$	$V_1|q_0\rangle|\#\rangle = |q_1\rangle|a\rangle$
$V_\$|q_0\rangle|\#\rangle = |q_{rej,0}\rangle|\#\rangle$	$V_0|q_1\rangle|\#\rangle = |q_2\rangle|\#\rangle$	$V_1|q_1\rangle|\#\rangle = |q_1\rangle|\#\rangle$
$V_\$|q_1\rangle|\#\rangle = |q_{rej,1}\rangle|\#\rangle$	$V_0|q_2\rangle|\#\rangle = |q_1\rangle|\#\rangle$	$V_1|q_2\rangle|\#\rangle = |q_2\rangle|\#\rangle$
$V_\$|q_2\rangle|\#\rangle = |q_{acc}\rangle|\#\rangle$	$V_0|q_1\rangle|a\rangle = |q_{rej,0}\rangle|\#\rangle$	$V_1|q_1\rangle|a\rangle = |q_{rej,0}\rangle|\#\rangle$

Table 6: Transitions of V for *Odd*

We show that (P,V) recognizes *Odd* with probability 1. Let x be any input. First, consider the case where x is in *Zero*. Assume that x is of the form 0^m1y, where y contains an odd number of 0s. The honest prover P erases a that is sent from the verifier when V reads 1. Since V can check whether y includes an odd number of 0s, V accepts x with certainty. Next, assume that $x \notin Odd$. In the special case where $x \in \{0\}^*$, V can reject x with certainty with no query to a committed prover. Now, consider the remaining case where x contains a 1. Assume that x is of the form 0^m1y, where y contains an even number of 0s. The verifier V sends a to a committed prover when he reads 1. Note that V's protocol is deterministic. To maximize the acceptance probability of V, a dishonest prover needs to return # to V since, otherwise, V immediately rejects x in a deterministic fashion. Since V can check whether y includes an odd number of 0s without making any query to the prover, for any committed prover P^*, (P^*,V) rejects x with certainty. Since the number of interactions in the protocol is at most 1, *Odd* therefore belongs to $\mathrm{QIP}_1^{\#}(1qfa)$, as requested. □

As the second step, we prove Proposition 9.4. The language *Zero* is known to be outside of 1QFA [32]; in other words, $Zero \notin \mathrm{QIP}_0^{\#}(1qfa)$. Proposition 9.4 expands this result and shows that *Zero* is not even in $\mathrm{QIP}_1^{\#}(1qfa)$.

Proposition 9.4 $Zero \notin \mathrm{QIP}_1^{\#}(1qfa)$.

Now, we begin with the proof of Proposition 9.4. Towards a contradiction, we first assume that a 1-iteration bounded QIP system (P,V) with 1qfa verifier recognizes $Zero$ with error probability $\leq 1/2-\eta$ for a certain constant $\eta>0$. Let Q and Γ be respectively the set of V's inner states and the communication alphabet. Write Σ for our alphabet $\{0,1\}$ for simplicity. Without loss of generality, we assume henceforth that V does not query at the time when it enters a halting inner state; in particular, the time when the head is scanning the endmarker \$.

First, we introduce the notions of "1-iteration condition" and "query weight." We fix an input x and let P' be any committed prover. For readability, we use the notation $Comp_V(P',x)$ to denote the computation of (P',V) on the input x. Moreover, $PComp_V(P',x)$ denotes the partial computation obtained by executing the QIP protocol (P',V) on any input whose prefix is x while the head is reading ¢x (i.e., between the first step at ¢ and the step at which the head reads the rightmost symbol of x and moves off x). When we consider a computation path, we understand that a computation path *terminates* either at a halting configuration or at a non-halting configuration ξ of zero amplitude.

For convenience, a committed prover P' is said to satisfy *the 1-iteration condition* at x with V if, for any query configuration ξ of non-zero amplitude in $Comp_V(P',x)$, no other query configuration exists between ξ and the initial configuration in the computation. Let $C^{(1)}_{x,V}$ be the collection of all committed provers P' who satisfy the 1-iteration condition at x with V. It is important to note that, whenever a prover in $C_{x,V}$ answers to V with non-blank communication symbols with non-zero amplitude, V must change these symbols back to blank immediately since, otherwise, V is considered to make a second query. Choose any prover P' in $C^{(1)}_{x,V}$ and consider the computation $Comp_V(P',x)$. By introducing an extra projection, we modify $Comp_V(P',x)$ as follows. Whenever V conducts a measurement, we then apply a projection, mapping onto the Hilbert space span$\{|\#\rangle\}$, to the communication cell. This projection makes all non-blank symbols collapse. If the communication cell is blank, then V continues to the next step. Observe that this modified computation is independent of the choice of a committed prover. For this modified computation of V on x, we use the notation $MComp_V(x)$. Figure 3 illustrates the difference between a modified computation and two computations with different provers. The *query weight* $wt^{(x)}_V(y)$ *of* V *at* y *conditional to* x is the sum of all the squared magnitudes of the amplitudes of query configurations, in $MComp_V(xy)$, where V makes queries while reading y. For brevity, let $wt_V(y)=wt^{(\lambda)}_V(y)$, where λ is the empty string. By its definition, a query weight ranges between 0 and 1 and satisfies that $wt_V(x)+wt^{(x)}_V(y)=wt_V(xy)$ for any $x,y\in\Sigma^*$.

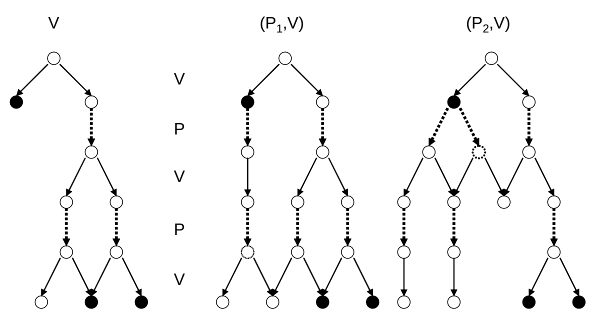

Figure 3: Example of a modified computation. The leftmost graph depicts the modified computation of V on input x. The latter two graphs are computations of V on x using different provers P_1 and P_2. The black circles indicate query configurations whereas the white circles indicate non-query configurations. The dotted circle is the place where prover P_2 forces V to generate a new computation path that destructively interferes with an existing path in the modified computation of V.

Recall that (P,V) is 1-iteration bounded and recognizes $Zero$. The following lemma holds for the query weight of V. In the lemma, *one round* in a computation comprises the following series of executions: a prover first applies his strategy including the return of the blank symbol (if not the first round) and V then makes his move followed by a measurement. Note that the first round does not include a prover's move. In a modified computation, *one round* is similar but further includes an extra projection (described above) after V's own measurement.

Lemma 9.5 *Let P' be any committed prover and let x,y be any strings.*

1. *Any committed prover satisfies the 1-iteration condition at x with V.*

2. *Any query configuration ξ of non-zero amplitude at round i in $Comp_V(P',x)$ must appear at the same round i in $MComp_V(x)$ with the same amplitude for any $i \in [1,|x|+1]_{\mathbb{Z}}$.*
3. *The query weight $wt_V^{(x)}(y)$ is greater than or equal to the sum of all the squared magnitudes of amplitudes of query configurations in $PComp_V(P',xy)$ while V's head is reading y.*

Proof. 1) Take any committed prover P'. In the case where $x \notin Zero$, since the QIP protocol (P',V) makes at most 1 iteration on x, P' clearly satisfies the 1-iteration condition at x with V. In contrast, assuming that $x \in Zero$, consider the partial computation $PComp_V(P',x)$. Note that this partial computation is also a partial computation of (P',V) on $x1$. Since $x1 \notin Zero$, P' must satisfy the 1-iteration condition at $x1$ with V. Therefore, P' satisfies the 1-iteration condition also at x.

2) We prove the claim by induction on $i \in [1,|x|+1]_{\mathbb{Z}}$. The basis case $i=1$ is trivially true since there is no prover's strategy. Consider the ith round. Let ξ be any query configuration of non-zero amplitude occurring at round i in $Comp_V(P',x)$. Note from the first claim that every committed prover satisfies the 1-iteration condition at x with V.

We first show that ξ appears with non-zero amplitude at round i in $MComp_V(x)$. Assume otherwise that ξ appears at round i in $Comp_V(P',x)$ but not in $MComp_V(x)$. This implies that, at a certain early round, as a response to a query configuration η in $Comp_V(P',x)$ of non-zero amplitude, P' forces V to generate ξ with non-zero amplitude later at the round i since, otherwise, V generates ξ with no query and ξ is therefore in $MComp_V(x)$, a contradiction. Since η and ξ are in the same computation path, this clearly violates the 1-iteration condition of P'. As a consequence, ξ must appear with non-zero amplitude at round i in $MComp_V(x)$.

Next, we show that ξ's amplitude in $Comp_V(P',x)$ is the same as in $MComp_V(x)$. Towards a contradiction, we assume that the amplitudes of ξ in $Comp_V(P',x)$ and in $MComp_V(x)$ are different. Now, consider all computation paths that reach ξ. Note that, if such a path contains no query configuration (other than ξ), this path must appear in $MComp_V(x)$. There are two cases to discuss: either a new computation path leading to ξ is added or an existing computation path to ξ is annihilated.

(Case 1) Consider any computation path γ leading to ξ in $Comp_V(P',x)$ whose amplitude contributes to the difference of ξ's amplitudes in $Comp_V(P',x)$ and in $MComp_V(x)$. Such a path γ should not be present in $MComp_V(x)$. The 1-iteration condition of P' implies that, since ξ's amplitude is not 0, the path γ cannot contain any query configuration of non-zero amplitude before reaching ξ. Hence, the path γ must be in $MComp_V(x)$, a contradiction.

(Case 2) The remaining case is that, at an early round, P' forces V to generate a number of computation paths that destructively interfere with an existing computation path δ leading to ξ in $MComp_V(x)$. This interference annihilates the path δ, which causes the change of ξ's amplitude in $Comp_V(P',x)$. Figure 3 illustrates this case. We modify the strategy of P' by changing its amplitudes (but not the tape/communication symbols) so that δ narrowly survives. Note that such a modification is possible because V moves exactly in the same way as before and therefore the modification does not incur any change of the computation $Comp_V(P',x)$ except for the amplitude distribution. As a result, the path δ connects two query configurations of non-zero amplitudes. This contradicts the first claim; namely, the 1-iteration condition of any committed prover.

In either case, we reach a contradiction. Therefore, the claim holds.

3) This follows directly from the second claim. □

We continue the proof of Proposition 9.4. Now, consider the value ν defined as the supremum, over all strings w in $Zero$, of the query weight of V at w. Observe that $0 \leq \nu \leq 1$ by Lemma 9.5. We examine the two cases $\nu = 0$ and $\nu > 0$ separately. For readability, we omit the letter V whenever it is clear from the context.

(Case 1: $\nu = 0$) Obviously, $wt(w) = 0$ for all $w \in Zero$. Toward a contradiction, it suffices to give a bounded-error 1qfa that recognizes $Zero$ since $Zero \notin$ 1QFA. Let P_I be the committed prover who applies only the identity operator at every step. The desired 1qfa M behaves as follows. On input x, M simulates V on x with the "imaginary" prover P_I by maintaining the content of the communication cell as an integrated part of M's inner states. This is possible by defining M's inner state (q,σ) to reflect both V's inner state q and a symbol σ in the communication cell. Now, we claim that M recognizes $Zero$ with bounded error. If input x is in $Zero$, then, since any communication with a prover has the zero amplitude, M correctly accepts x with probability $\geq 1/2+\eta$. Similarly, we can verify that, if x is not in $Zero$, M rejects x with probability $\geq 1/2+\eta$ because (P_I,V) must reject x with the same probability. Therefore, M recognizes $Zero$ with error probability $\leq 1/2-\eta$, as requested.

(Case 2: $\nu > 0$) Recall that the notation P_w refers to the strategy of P on input w. Note that, for every real number $\gamma \in (0,\nu]$, there exists a string w in $Zero$ such that $wt(w) \geq \nu - \gamma$. For each $y \in \Sigma^*$, set $\gamma_y =$

$\min\{\eta^2/16(|y|+1)^2, \nu\}$ and choose the lexicographically minimal string $w_y \in Zero$ such that $wt(w_y) \geq \nu - \gamma_y$.

For each $y \in \Sigma^*$, define the new prover P'_y that behaves on input $w_y y 01^m$ for every $m \in \mathbb{N}^+$ in the following fashion: P'_y takes the strategy $P_{w_y y 0}$ while V's head is reading $¢w_y$ and then P'_y behaves as P_I (i.e., applies the identity operator) while V is reading the remaining portion $y01^m\$$. For readability, we abbreviate $w_y y$ as $\tilde{y}$. We then claim the following.

Claim. *For any string $y \in \Sigma^*$, $p_{acc}(\tilde{y}0, P'_y) \geq 1/2 + \eta/2$.*

Proof of Claim. Let y be an arbitrary input string. Note that the protocol (P'_y, V) works in the same way as $(P_{\tilde{y}0}, V)$ while V is reading $¢w_y$. Consider $wt^{(w_y)}(y0)$. Note that $wt(w_y) + wt^{(w_y)}(y0) \leq \nu$. It thus follows that $wt^{(w_y)}(y0) \leq \gamma_y$ using the inequality that $wt(w_y) \geq \nu - \gamma_y$. Lemma 9.5(3) implies that, for any committed prover P^*, $wt^{(w_y)}(y0)$ bounds the sum of all the squared magnitudes of query configurations, while the head is reading $y0$, in the computation of (P^*, V) on the input $w_y y 0$. Therefore, a simple calculation (as in, e.g., [44, Lemma 9]) shows that

$$|p_{acc}(\tilde{y}0, P'_y) - p_{acc}(\tilde{y}0, P_{\tilde{y}0})| \leq 2\left(wt^{(w_y)}(y0)\right)^{1/2} |y0| \leq 2\sqrt{\gamma_y}(|y|+1) \leq \eta/2.$$

Since $p_{acc}(\tilde{y}0, P_{\tilde{y}0}) \geq 1/2 + \eta$, it follows that $p_{acc}(\tilde{y}0, P'_y) \geq (1/2+\eta) - \eta/2 \geq 1/2 + \eta/2$. □

Recall the set Q of inner states and the communication alphabet Γ. Set $d = |Q||\Gamma|$ for brevity. Using Lemma 4.3, for each $y \in \Sigma^*$, there exists a $(|\tilde{y}0|+2, d)$-bounded QIP system $(P^{(1)}_y, V)$ that simulates (P'_y, V) on input $\tilde{y}0$. The initial superposition is $|q_0, \#, \#^d\rangle$, where we omit the qubits representing the head position of V because V is a 1qfa verifier. Let $\mathcal{V} = \mathrm{span}\{|q\rangle \mid q \in Q\}$, let $\mathcal{M} = \mathrm{span}\{|\sigma\rangle \mid \sigma \in \Gamma\}$, and let $\mathcal{P}$ be the d-dimensional Hilbert space representing the prover's private tape. Let $|\psi_y\rangle$ be the superposition in the global configuration space $\mathcal{V} \otimes \mathcal{M} \otimes \mathcal{P}$ obtained just after V's head moves off the right end of $¢\tilde{y}0$ and $P^{(1)}_y$ replies to V. For each number $n \in \mathbb{N}^+$, consider a $(|\tilde{y}0|+2+n, d)$-bounded QIP system $(P^{(2)}_{y,n}, V)$ where $P^{(2)}_{y,n}$ simulates P'_y while reading $¢\tilde{y}0$ and applies the identity operator while reading $1^n\$$. Noting that the prover P'_y does nothing after V have read $¢w_y$, we can verify that $(P^{(2)}_{y,n}, V)$ simulates (P'_y, V) on the input $\tilde{y}01^n$. Letting $\mu = \inf_{y\in\Sigma^*}\{\||\psi_y\rangle\|\}$, we consider the two subcases $\mu \leq \eta/4$ and $\mu > \eta/4$.

(Subcase a: $\mu \leq \eta/4$) There exists a string y such that $\mu \leq \||\psi_y\rangle\| < \mu + \eta/4 \leq \eta/2$. This means that, after reading $¢\tilde{y}0$, the halting probability of V increases by no more than $(\eta/2)^2$. Consider the input $\tilde{y}01$. Since $p_{acc}(\tilde{y}0, P^{(1)}_y) \geq 1/2+\eta/2$, it follows that $p_{acc}(\tilde{y}01, P^{(2)}_{y,1}) \geq (1/2+\eta/2) - \||\psi_y\rangle\|^2 \geq 1/2$. However, this contradicts our assumption that, for any committed prover P^*, (P^*, V) accepts $\tilde{y}01$ with probability $\leq 1/2 - \eta < 1/2$.

(Subcase b: $\mu > \eta/4$) Let ϵ be any sufficiently small positive real number and choose a string y such that $\||\psi_y\rangle\| \in [\mu, \mu+\epsilon)$. The superposition of global configurations obtained after the operation of the protocol $(P^{(2)}_y, V)$ on $\tilde{y}01^j$ just before V scans \$ becomes $(P_I E_{non} V_1)^j |\psi_y\rangle$, where V_b ($b \in \check{\Sigma}$) is the unitary operation of V when V is scanning the symbol b and P_I is the identity operation of a prover. For convenience, write W for $P_I E_{non} V_1$. For any integer $j \geq 1$, $\mu \leq \|W^j|\psi_y\rangle\| < \mu + \epsilon$. By a similar analysis in [32] (see also [24, Lemma 4.1.12]), there exist a constant $c > 0$ independent of ϵ and a number $m \in \mathbb{N}^+$ such that $\||\psi_y\rangle - W^m|\psi_y\rangle\| < c\cdot\epsilon^{1/4}$. From this inequality follows

$$|p_{acc}(\tilde{y}0, P^{(1)}_y) - p_{acc}(\tilde{y}01^m, P^{(2)}_{y,n})| \leq \|V_\$|\psi_y\rangle - V_\$ W^m|\psi_y\rangle\| = \||\psi_y\rangle - W^m|\psi_y\rangle\| < c\epsilon^{1/4},$$

where the first inequality is obtained as in the proof of Lemma 6.3. We thus obtain the upper bound that $|p_{acc}(\tilde{y}0, P'_y) - p_{acc}(\tilde{y}01^m, P'_y)| \leq c\epsilon^{1/4}$. Let $\epsilon = \left(\frac{\eta}{2c}\right)^4$. Since $p_{acc}(\tilde{y}0, P'_y) \geq 1/2 + \eta/2$, it follows that $p_{acc}(\tilde{y}01^m, P'_y) \geq (1/2+\eta/2) - c\epsilon^{1/4} = 1/2$. This contradicts our assumption that (P^*, V) accepts $\tilde{y}01^m$ with probability $\leq 1/2 - \eta/2 < 1/2$ for any committed prover P^*. Therefore, $Zero \notin \mathrm{QIP}^{\#}_1(1qfa)$, as requested. This completes the proof of Proposition 9.4.

Since a 1qfa verifier cannot remember the number of queries, we may not directly generalize the proof of Theorem 9.2 to claim that $\mathrm{QIP}^{\#}_k(1qfa) \neq \mathrm{QIP}^{\#}_{k+1}(1qfa)$ for any constant k in $\mathbb{N}^+$. Nevertheless, we still conjecture that this claim holds.

10 Future Directions

There have been a surge of interests in QIP systems [12, 28, 30, 43, 45] partly because a QIP system embodies an essence of quantum computation and communication. Our research on weak-verifier QIP systems was inspired by the work of Dwork and Stockmeyer [17], who extensively studied $\mathrm{IP}(2pfa)$ and $\mathrm{AM}(2pfa)$. Having started with our basic qfa-verifier QIP systems, we have discussed several variants of restricted QIP systems and have demonstrated strengths and weaknesses of these QIP systems. Nonetheless, the theory of weak-verifier QIP systems is still vastly uncultivated. The development of new proof techniques is needed to settle down, for instance, all the pending questions left in this paper. We strongly hope that further research will unearth the crucial characteristics of the QIP systems.

This final section discusses six important directions that lead to fruitful future research on weak-verifier QIP systems.

- **Modifying Verifier's Ability.** Our verifier is a quantum finite automaton against a mighty prover who can apply any unitary operation. This paper has dealt with only three major qfa's: mo-1qfa's, 1qfa's, and 2qfa's. It is important to study the nature of quantum interactions between provers and different types of verifiers. There have been several variants of 2qfa's proposed in the literature. For instance, Amano and Iwama [3] studied so-called a 1.5qfa, which is a 2qfa whose head never moves to the left. Recently, Ambainis and Watrous [7] considered a 2qfa whose head move is particularly classical. Instead of restricting the ability of qfa's, we can supplement an additional device to gain more computational power of qfa's. As an example, Golovkins [23] lately studied a qfa that is equipped with a pushdown stack. Using these qfa models as verifiers, we need to conduct a comprehensive study on the corresponding QIP systems.
- **Curtailing Prover's Strategy.** Another direction is to limit the prover's power. Instead of strengthening a verifier, for instance, we can restrict the size of the prover's strategy. Having already seen in Lemma 5.5, without diminishing the recognition power, we can limit the size of prover's private tape space to the size of the verifier's visible configuration space. If we further constrain the prover's strategy, how powerful is the corresponding QIP system? In the 1990s, Condon and Ladner [15] studied IP systems with restricted provers who take only polynomial-size strategy. They showed that, with polynomial-size strategy, the IP systems with polynomial-time PTM verifiers exactly characterize Babai's class MA. Analogously, for instance, we can consider the QIP systems in which 2qfa verifiers play against $O(\log\log n)$-space bounded provers. Such QIP systems still recognize certain non-regular languages.
- **Communicating through a Classical Channel.** We may understand our QIP protocol as a 2-party communication protocol exchanging messages through a quantum channel. Recall that a classical prover performs only a unitary operation of entries either 0 or 1. Seen as a communication protocol, we instead restrict a communication channel between two players, a prover and a verifier, to be classical. Such a communication may be realized by performing a measurement on the communication cell just before each player makes an access to the cell. The communication cell then becomes a probabilistic mixture of classical states. It is, nonetheless, unclear whether this QIP system is as powerful as our classical-prover QIP system.
- **Using Prior Entanglement.** Quantum entanglement is of significant importance in quantum computation and communication. The EPR pair[††], for instance, is used to teleport a quantum state using a quantum correlation between two qubits. Consider the case where a verifier shares limited prior entanglement with a prover in such a way that, before the start of a QIP protocol, a certain number of the verifier's inner states and a finite segment of the prover's private tape are entangled in a predetermined manner. This simple model of limited prior entanglement, nevertheless, does not enrich the computational resource of the QIP systems because, similar to the proof of $\mathrm{QIP}(1qfa) = \mathrm{REG}$, we can prove that the aforementioned limited prior entanglement makes the corresponding QIP systems recognize only regular languages. Therefore, other types of models are needed to explore the usefulness of prior entanglement.
- **Playing against Multiple Provers.** A natural extension of our basic QIP systems is obtained by providing each QIP system with multiple provers against a single verifier. In the polynomial-time setting, Kobayashi and Matsumoto [30] studied the QIP systems in which a uniform polynomial-size quantum-circuit verifier plays against multiple provers. These provers may further share prior entanglement among them (but not with a verifier). Multiple-prover QIP systems of Kobayashi and Matsumoto are shown

[††]The EPR pair is the 2-qubit quantum state $|\Phi^+\rangle = (|00\rangle + |11\rangle)/\sqrt{2}$, which was proposed by Einstein, Podolsky, and Rosen in 1935.

to characterize the complexity class NEXP [30]. In a classical case, Feige and Shamir [19] constructed a 2-prover IP system with a 2pfa verifier (using the model of Dwork and Stockmeyer) for each recursive language. Naturally, we expect the multiple-prover QIP systems with qfa verifiers to demonstrate a similar increase in power over the single-prover QIP systems.

- **Making Knowledge-Based Interactions.** Lately, a great attention has been paid to a quantum zero-knowledge proof systems (QZKP systems, in short) [29, 41]. As a followup to their 2pfa-verifier IP systems, Dwork and Stockmeyer also studied zero-knowledge proof systems played between provers and 2pfa verifiers [18]. It is desirable to develop a theory of QZKP systems with qfa verifiers in connection to quantum cryptography.

Acknowledgment. The first author is grateful to Hirotada Kobayashi for a detailed presentation of his result.

References

[1] D. Aharonov and T. Naveh. Quantum NP – a survey. Available at arXiv quant-ph/0210077.

[2] D. Aharonov and O. Regev. A lattice problem in quantum NP. In *Proc. 44th IEEE Symposium on Foundations of Computer Science*, pp.210–219, 2003.

[3] M. Amano and K. Iwama. Undecidability on quantum finite automata. In *Proc. 31st ACM Symposium on Theory of Computing*, pp.368–375, 1999.

[4] A. Ambainis, M. Beaudry, M. Golovkins, A. Ķikusts, M. Mercer, and D. Thérien. Algebraic results on quantum automata. In *Proc. 21st International Symposium on Theoretical Aspect of Computer Science*, Lecture Notes in Comput. Sci., Vol.2996, pp.93–104, 2004.

[5] A. Ambainis and R. Freivalds. 1-way quantum finite automata: strengths, weakness and generalizations. In *Proc. 39th IEEE Symposium on Foundations of Computer Science*, pp.332–341, 1998.

[6] A. Ambainis, A. Ķikusts, and M. Valdats. On the class of languages recognizable by 1-way quantum finite automata. In *Proc. 18th International Symposium on Theoretical Aspect of Computer Science*, Lecture Notes in Comput. Sci., Vol.2010, pp.75–86, 2001.

[7] A. Ambainis and J. Watrous. Two-way finite automata with quantum and classical states. *Theor. Comput. Sci.* **287** (2002) 299–311.

[8] L. Babai. Trading group theory for randomness. In *Proc. 17th ACM Symposium on Theory of Computing*, pp.421–429, 1985.

[9] P. Benioff. The computer as a physical system: a microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines. *J. Stat. Phys.* **22** (1980) 563–591.

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

[11] A. Brodsky and N. Pippenger. Characterizations of 1-way quantum finite automata. *SIAM J. Comput.* **31** (2002) 1456–1478.

[12] R. Cleve, P. Høyer, B. Toner, and J. Watrous. Consequences and limits of nonlocal strategies. In *Proc. 19th IEEE Conference on Computational Complexity*, pp.236–249, 2004.

[13] A. Condon. The complexity of space bounded interactive proof systems. In *Complexity Theory: Current Research* (eds. Ambos-Spies, et al.), Cambridge University Press, pp.147–189, 1993.

[14] A. Condon, L. Hellerstein, S. Pottle, and A. Wigderson. On the power of finite automata with both nondeterministic and probabilistic states. *SIAM J. Comput.* **27** (1998) 739–762.

[15] A. Condon and R. Ladner. Interactive proof systems with polynomially bounded strategies. *J. Comput. Syst. Sci.* **50** (1995) 506–518.

[16] D. Deutsch. Quantum theory, the Church-Turing principle and the universal quantum computer. *Proc. Roy. Soc. London Ser. A,* **400** (1985) 96–117.

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

[18] C. Dwork and L. Stockmeyer. Finite state verifiers II: zero knowledge. *J. ACM* **39** (1992) 829–858.

[19] U. Feige and A. Shamir. Multi-oracle interactive protocols with constant space verifiers. *J. Comput. System Sci.* **44** (1992) 259–271.

[20] R. Feynman. Simulating physics with computers. *Int. J. Theor. Phys.* **21** (1982) 467–488.

[21] S. Goldwasser, S. Micali and C. Rackoff. The knowledge complexity of interactive proof systems. *SIAM J. Comput.* **18** (1989) 186–208.

[22] S. Goldwasser and M. Sipser. Private coins versus public coins in interactive proof systems. In *Proc. 18th ACM Symposium on Theory of Computing*, pp.59–68, 1986.

[23] M. Golovkins. Quantum pushdown automata. In *Proc. 27th Conference on Current Trends in Theory and Practice of Informatics*. Lecture Notes in Comput. Sci. Vol.1963, pp.336–346, 2000.

[24] J. Gruska. *Quantum Computing*. McGraw Hill, 1999.

[25] J. Hopcroft, R. Motwani, and J. Ullman. *Introduction to Automata Theory, Languages, and Computation, Second Edition*. Addison-Wesley, 2001.

[26] L. Hughston, R. Jozsa, and W. Wootters. A complete classification of quantum ensembles having a given density matrix. *Physics Letters A* **183** (1993) 14–18.

[27] A. Kitaev. Quantum computations: algorithms and error correction. *Russian Math. Surveys* **52** (1997) 1191–1249.

[28] A. Kitaev and J. Watrous. Parallelization, amplification, and exponential time simulation of quantum interactive proof systems. In *Proc. 32nd ACM Symposium on Theory of Computing*, pp.608–617, 2000.

[29] H. Kobayashi. Non-interactive quantum perfect and statistical zero-knowledge. In *Proc. 14th International Symposium on Algorithms and Computation*, Lecture Notes in Comput. Sci. Vol.2906, pp.178–188, 2003.

[30] H. Kobayashi and K. Matsumoto. Quantum multi-prover interactive proof systems with limited prior entanglement. *J. Comput. Syst. Sci.* **66** (2003) 429–450.

[31] H. Kobayashi, K. Matsumoto, and T. Yamakami. Quantum Merlin-Arthur proof systems: Are multiple Merlins more helpful to Arthur? In *Proc. 14th International Symposium on Algorithms and Computation*, Lecture Notes in Comput. Sci. Vol.2906, pp.189–198, 2003.

[32] A. Kondacs and J. Watrous. On the power of quantum finite state automata. In *Proc. 38th IEEE Symposium on Foundations of Computer Science*, pp.66–75, 1997.

[33] C. Lund, L. Fortnow, H. Karloff, and N. Nisan. Algebraic methods for interactive proof systems. *J. ACM* **39** (1992) 859–868.

[34] C. Moore and J. Crutchfield. Quantum automata and quantum grammars. *Theor. Comput. Sci.* **237** (2000) 275–306

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

[36] H. Nishimura and T. Yamakami. Polynomial time quantum computation with advice. *Inform. Process. Lett.* **90** (2004) 195–204.

[37] C. Papadimitriou. Games against nature. *J. Comput. Syst. Sci.* **31** (1985) 288–301.

[38] A. Shamir. IP=PSPACE. *J. ACM* **39** (1992) 869–877.

[39] A. Uhlmann. Parallel transport and "quantum holonomy" along density operators. *Reports on Mathematical Physics* **24** (1986) 229–240.

[40] J. Watrous. Succinct quantum proofs for properties of finite groups. In *Proc. IEEE 41st Symposium on Foundations of Computer Science*, pp.537–546, 2000.

[41] J. Watrous. Limits on the power of quantum statistical zero-knowledge. In *Proc. 43rd IEEE Symposium on Foundations of Computer Science*, pp.495–468, 2002.

[42] J. Watrous. On quantum and classical space-bounded processes with algebraic transition amplitudes. In *Proc. 40th IEEE Symposium on Foundations of Computer Science*, pp.341–351, 1999.

[43] J. Watrous. PSPACE has constant-round quantum interactive proof systems. *Theor. Comput. Sci.* **292** (2003) 575–588.

[44] T. Yamakami. Analysis of quantum functions. *Int. J. Found. Comput. Sci.* **14** (2003) 815-852. A preliminary version appeared in the *Proc. 19th Conference on Foundations of Software Technology and Theoretical Computer Science*. Lecture Notes in Comput. Sci., Vol.1738, pp.407–419, 1999.

[45] A. Yao. Interactive proofs for quantum computation. In *Proc. 14th International Symposium on Algorithms and Computation*, Lecture Notes in Comput. Sci. Vol.2906, p.1, 2003.

ECCC ISSN 1433-8092
http://www.eccc.uni-trier.de/eccc
ftp://ftp.eccc.uni-trier.de/pub/eccc
ftpmail@ftp.eccc.uni-trier.de, subject 'help eccc'