Interactive Proofs with Quantum Finite Automata

Harumichi Nishimura† and Tomoyuki Yamakami‡

Abstract

Following an early work of Dwork and Stockmeyer on interactive proof systems whose verifiers are two-way probabilistic finite automata, the authors studied in 2004 the computational power of quantum interactive proof systems whose verifiers are particularly limited to quantum finite automata. As a follow-up to this early study, we further investigate the quantum nature of interactions between provers and verifiers by studying how various restrictions on quantum interactive proof systems affect the language recognition power of systems. In particular, we examine three intriguing restrictions for which (i) provers always behave in a classical fashion, (ii) verifiers always reveal the information on next moves to provers, and (iii) the number of interactions between provers and verifiers is bounded.

Keywords: finite automaton, interactive proof system, quantum computing, classical prover, quantum prover, interaction

1 Overview

Quantum mechanics has provided an unconventional means to fast computing and secure communication since early 1980s. The potential power of quantum interactions between two parties—a prover and a verifier—motivated the authors [15] in 2004 to investigate a quantum analogy of an early work of Dwork and Stockmeyer [6] on (classical) interactive proof (IP) systems whose verifiers are restricted to two-way probabilistic finite automata (or 2pfa’s). Such weak verifiers can model computations that utilize only a finite amount of memory space. These IP systems are special cases of a much wider scope of space-bounded IP systems of Condon [4]. It has been shown that those IP systems behave quite differently from time-bounded IP systems. Although verifier’s power is limited to 2pfa’s, as Dwork and Stockmeyer dexterously demonstrated, 2pfa-verifier IP systems turn out to be significantly powerful, because a number of interactions with amighty prover can enhance 2pfa-verifier’s ability to recognize much more complicated languages than the 2pfa’s alone recognize. To describe succinctly such restricted IP systems, Dwork and Stockmeyer introduced a special notation IP(⟨restriction⟩) for a class of languages that are recognized by bounded-error IP systems with weak verifiers under restrictions specified by ⟨restriction⟩. For instance, the notation IP(2pfa,poly-time) expresses the language class characterized by IP systems with 2pfa verifiers operated in expected polynomial time. The seminal work of Dwork and Stockmeyer further studied numerous subjects (private versus public coins, etc.) founded on a basis model of 2pfa-verifier IP systems. To express Babai’s Arthur-Merlin proof systems [3], in particular, they introduced another notation AM(⟨restriction⟩) under restrictions given in ⟨restrictions⟩. An immediate advantage of studying such weak verifier models is to prove certain types of separations and collapses among the associated complexity classes without any unproven assumption. For instance, Dwork and Stockmeyer successfully separated IP(2pfa,poly-time) from AM(2pfa), and AM(2pfa,poly-time) from AM(2pfa). Another significant advantage is that their restricted models make it easier to analyze the behaviors of two parties—a prover and a verifier—during a usually complicated communication process between them.

Unlike Watrous’ model [17] of circuit-based quantum interactive proof (QIP) systems, a basic QIP model of the authors [15] uses a measure-many two-way quantum finite automaton (2qfa, in short) of Kondacs and Watrous [11] as a weak verifier who can communicate with a mighty quantum prover through a quantum communication bulletin board (implemented as a communication cell that holds at each moment one symbol from a communication alphabet). They also considered two additional variants of verifiers: a measure-once...
one-way quantum finite automaton (mo-1qfa, in short) of Moore and Crutchfield [13] and a measure-many one-way quantum finite automaton (1qfa, in short) of Kondacs and Watrous [11]. An initial study of QIP systems with those weak verifiers reveals their noticeable strength. Let us recall from [10] a general notation QIP(⟨restrictions⟩) for bounded-error QIP systems under restrictions given by (restrictions), analogous to the aforementioned notation IP(⟨restriction⟩) of Dwork and Stockmeyer. For instance, QIP(2qfa) is obtained by restricting all verifiers to 2qfa’s. Likewise, a use of mo-1qfa verifiers and 1qfa verifiers introduces the language classes QIP(mo-1qfa) and QIP(1qfa), respectively.

The power of quantum interaction was exemplified in [16]. (i) With mo-1qfa verifiers, it holds that MO-1QFA \nsubseteq QIP(mo-1qfa) \nsubseteq REG, where REG is the class of regular languages and MO-1QFA is the class of all languages recognized by bounded-error mo-1qfa’s. (ii) With 1qfa verifiers, we obtain 1QFA \nsubseteq QIP(1qfa) = REG, where 1QFA is the class of all languages recognized by bounded-error 1qfa’s. (iii) With 2qfa verifiers, it holds that QIP(1qfa) \nsubseteq QIP(2qfa, poly-time) \nsubseteq AM(2qfa). They also showed that 2QFA_{\mathcal{C}} \subseteq P and QIP_{\mathcal{C}}(2qfa, poly-time) \subseteq NP, where \mathcal{C} and \mathcal{A} respectively indicate that all (transition) amplitudes of qfa verifiers are polynomial-time “approximable” complex numbers and of algebraic complex numbers. The last result clearly contrasts with the following classical containments: AM(2pfa) \nsubseteq IP(2pfa, poly-time) \nsubseteq PSPACE [6].

We intend to continue a study of qfa-verifier QIP systems from various aspects of computational complexity. In particular, this paper aims at presenting three intriguing subjects that were discussed in [15] but have been excluded from our early journal publication [16]. In this current publication, we shall examine strengths and weaknesses of qfa-verifier QIP systems by observing how various restrictions on the QIP systems affect their power of language recognition. Our investigation is focused on the following three selected subjects.

1. Classical provers versus quantum provers. In our model of QIP systems [15, 16], provers are basically quantum machines, which can apply any predetermined unitary operators. In contrast, provers in IP systems of Dwork and Stockmeyer [6] are essentially “probabilistic machines,” which can flip privately owned coins and decide what messages to send back to 2pfa verifiers. These probabilistic machines, however, are known to be reduced to deterministic machines, which are naturally associated with unitary operators whose entries are only 0s and 1s because the provers can use an unlimited amount of private memory storage. For convenience, we briefly call such provers classical provers; in contrast, we call standard provers quantum provers. Naturally, we raise a simple question of whether our quantum provers are truly different in recognition power from the aforementioned classical provers.

It appears that a classical prover helps a 2qfa verifier much more than a quantum prover does. For instance, the language Center = \{x1y | x, y \in \{0, 1\}^*, |x| = |y|\} is not yet known to belong to QIP(2qfa); however, interactions with a classical prover allow a 2qfa verifier to recognize this particular language. Such a strength of using classical provers stems from a simple fact that an analysis of classical-prover QIP protocols is much easier than that of quantum-prover ones. This paper further shows the following containments and separations concerning classical provers. (i) QIP(1qfa) \nsubseteq QIP(1qfa, c-prover). (ii) AM(2pfa) \nsubseteq QIP(2qfa, c-prover). (iii) AM(2pfa, poly-time) \nsubseteq QIP(2qfa, poly-time, c-prover) \nsubseteq AM(2pfa). A core argument for these results is a technical construction of appropriate QIP protocols that recognize target languages. All the above results will be presented in Section 3

2. Public information versus private information. As noted earlier, Dwork and Stockmeyer [6] examined two different types of IP systems—IP(⟨restriction⟩) and AM(⟨restriction⟩)—when verifiers are limited to 2pfa’s. The latter IP system AM(⟨restriction⟩) refers to Babai’s model of Arthur-Merlin proof systems [3] (also known as “public-coin” IP systems), in which verifiers flip fair coins to decide next moves and reveal their outcomes, essentially showing a state of next internal configurations to mighty provers, who can keep track of the verifier’s past configurations. In comparison, standard IP systems are referred to as “private-coin” IP systems. A question concerning “public coins” versus “private coins,” which was a key subject in [6], is essentially whether the prover obtains “complete information” or “partial information” on the configurations of the verifier. It was shown in [6] that 2PFA \nsubseteq AM(2pfa) \nsubseteq IP(2pfa), highlighting a clear difference between public information and private information.

Likewise, we shall introduce a similar “publicness” notion to qfa-verifier QIP systems by demanding verifiers to reveal their next moves to provers at every step. To express the public-coin analogue of QIP systems, we use the notation (public) as a restriction of the systems. With this notation, for instance,
QIP(1qfa, public) denotes a language class obtained from QIP(1qfa) by restricting verifiers to publicly announcing their next moves. It turns out that public QIP systems remain significantly powerful. To be more precise, we shall prove the following three class relations. (i) 1RFA $\nsubseteq$ QIP(1qfa, public) $\nsubseteq$ 1QFA, (ii) QIP(2qfa, public, poly-time) $\nsubseteq$ AM(2qfa, poly-time), and (iii) QIP(2qfa, public, c-prover) $\nsubseteq$ AM(2qfa, poly-time), where 1RFA is the language family induced by one-way (deterministic) reversible finite automata (1fa’s, in short). In Section 4 we shall discuss those results in details.

3. Number of interactions between a prover and a verifier. As suggested in [10] Section 6], the number of interactions between a prover and a verifier in a weak-verifier QIP system may serve as a complexity measure of classifying various languages. Unlike Dwork-Stockmeyer IP systems, the original QIP systems of the authors [13, 14] were introduced as to force two parties—a prover and a verifier—to communicate with each other at every step and, through Sections 3–4 we shall take this definition of QIP systems. To study the precise effect of interactions, nevertheless, we need to modify this original model slightly so that the verifier can interact with the prover only when he needs any help from the prover.

To express those new QIP systems and their corresponding language classes, we invent two new notations QIP$^k$((restrictions)) and QIP$^k_*$((restrictions)), where $k$ indicates the maximal number of iterations made during a computation. In Section 5 we shall prove that QIP$^k_*(1qfa) \nsubseteq$ QIP$^k_1(1qfa) \subseteq$ QIP$^k(1qfa)$. The first separation between QIP$^k_0(1qfa)$ and QIP$^k_1(1qfa)$ comes from the fact that the language $\text{Odd} = \{0^n1^n | n \in \mathbb{N}, \text{odd} \}$ belongs to QIP$^k_0(1qfa)$ but it is not in QIP$^k_1(1qfa)$ since QIP$^k_0(1qfa)$ coincides with 1QFA. In contrast, the second separation of QIP$^0_0(1qfa)$ from QIP$^0_1(1qfa)$ is exemplified by the language $\text{Zero} = \{x0 | x \in \{0, 1\}^*\}$; however, the proof of $\text{Zero} \not\in$ QIP$^0_1(1qfa)$ is much more involved than the proof that $\text{Zero} \not\in$ 1QFA that appears in [11].

2 QFA-Verifier QIP Systems

Throughout this paper, $\mathbb{C}$ denotes the set of all complex numbers and $i$ is $\sqrt{-1}$. Let $\mathbb{N}$ be the set of all natural numbers (i.e., nonnegative integers) and set $\mathbb{N}^+ = \mathbb{N} \setminus \{0\}$. Given two integers $m$ and $n$ with $m \leq n$, the integer interval $[m, n]$ is the set $\{m, m+1, m+2, \ldots, n\}$ and $\mathbb{Z}_n$ in particular denotes the set $[0, n-1]$. All logarithms are to base 2 and all polynomials have integer coefficients. An alphabet is a finite nonempty set of “symbols” and our input alphabet $\Sigma$ is not necessarily limited to $\{0, 1\}$ throughout this paper. Following the standard convention, $\Sigma^*$ denotes the set of all finite sequences of symbols from $\Sigma$, and we write $\Sigma^n = \{x \in \Sigma^* | |x| = n\}$, where $|x|$ denotes the length of $x$. Opposed to the notation $\Sigma^*$, $\Sigma^\infty$ stands for the set of all infinite sequences, each of which consists of symbols from $\Sigma$. For any symbol $a$ in $\Sigma$, $a^\infty$ denotes an element of $\Sigma^\infty$, 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 (refer to, e.g., [3, 4, 5, 6] for its foundation). As underlying computation device, we extensively use measure-many one-way quantum finite automata (or 1qfa’s, in short) and of measure-many two-way quantum finite automata (or 2qfa’s), where we assume the reader’s familiarity with the definitions of those quantum automata [11].

2.1 Basic Model of QIP Systems

Let us review the fundamental definition of QIP system of the authors [13, 14], in which verifiers are particularly limited to quantum finite automata. In Sections 4–5 we shall further restrict the behaviors of those verifiers as well as provers to obtain three major variations of our basic QIP systems. The reader may refer to [10] for a brief discussion on the main difference between QIP systems based on uniform quantum circuits and those based on quantum finite automata.

We use the notation $(P, V)$ to denote a QIP protocol taken by prover $P$ and verifier $V$ (whose schematic diagram is illustrated in Figure 1). For convenience, we use the same notation $(P, V)$ to mean a QIP system with the prover $P$ and the verifier $V$. The 2qfa verifier $V = (Q, \Sigma \cup \{\#, \$\}, \Gamma, \delta, \eta_0, Q_{\text{acc}}, Q_{\text{rej}})$ is a 2qfa specified by a finite set $Q$ of verifier’s inner states, an input alphabet $\Sigma$ and a verifier’s transition function $\delta$, equipped further with a shared communication cell using a communication alphabet $\Gamma$. The set $Q$ is the union of three mutually disjoint subsets $Q_{\text{non}}, Q_{\text{acc}},$ and $Q_{\text{rej}}$, where any states in $Q_{\text{non}}, Q_{\text{acc}},$ and $Q_{\text{rej}}$ are respectively called a non-halting inner state, an accepting inner state, and a rejecting inner state. In contrast to $Q_{\text{non}}$, inner states in $Q_{\text{acc}} \cup Q_{\text{rej}}$ are simply called halting inner states. In particular, $Q_{\text{non}}$ contains a so-called initial inner state $q_0$. An input tape is indexed by natural numbers (where the first cell is indexed 0). Two designated symbols $\xi$ and $\$ not appearing in $\Sigma$, which are called respectively the left
and symbol \(d\) the input tape in direction our convenience, set \(\hat{\Sigma} = \Sigma\). Assume also that \(\Gamma\) contains a blank symbol \# with which the system \((P, V)\) begins in the communication cell.

The verifier's transition function \(\delta\) is a map from \(Q \times \hat{\Sigma} \times \Gamma \times Q \times \Gamma \times \{0, \pm 1\}\) to \(\mathbb{C}\) and is interpreted as follows. For any \(q, q' \in Q, \sigma \in \hat{\Sigma}, \gamma, \gamma' \in \Gamma, \) and \(d \in \{0, \pm 1\}\), the complex number \(\delta(q, \sigma, \gamma, q', \gamma', d)\) specifies the transition amplitude with which the verifier \(V\) in state \(q\) scanning symbol \(\sigma\) on the input tape and symbol \(\gamma\) on the communication cell changes \(q\) to \(q'\), replaces \(\gamma\) with \(\gamma'\), and moves his tape head on the input tape in direction \(d\). When the tape head is located in a cell indexed \(t\), it must move to the cell indexed \(t + d\).

At the beginning of the computation, an input string \(x\) over \(\Sigma\) of length \(n\) is written orderly from the first cell to the \(n\)th cell of the input tape. The tape head initially scans \(\xi\) in the 0th cell. The communication cell holds only a symbol in \(\Gamma\) and initially \# is written in the cell. Similar to the original definition of 2qfa in [11], our input tape is circular; that is, whenever the verifier’s tape head scanning \(\xi\) (\$, resp.) on the input tape moves to the left (right, resp.), the tape head reaches to the right end (resp. left end) of the input tape.

\[\begin{array}{c}
\text{A qfa verifier} \\
\text{An input tape} \\
\text{An alphabet } \Sigma \\
\text{A tape head} \\
\text{An alphabet } \Gamma \\
\text{A finite control unit} \\
\end{array}\]

\[\begin{array}{c}
\text{A prover} \\
\text{An input tape} \\
\text{An alphabet } \Lambda \\
\text{An infinite private tape} \\
\end{array}\]

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

Next, we explain two concepts of (global) configuration and visible configuration. A (global) configuration of the QIP protocol \((P, V)\) is a snapshot of a “computation” of the protocol, comprising the following 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 \(\gamma\), and the verifier’s tape head position is \(k\) on the input tape. Let \(V_n\) and \(M\) be respectively the Hilbert spaces spanned by the computational bases \(\{|q, k, \gamma\} : (q, k) \in Q \times \mathbb{Z}_{n+2}\}\) and \(\{|\gamma\} : \gamma \in \Gamma\\). The Hilbert space \(V_n \otimes M\) is called the verifier’s visible configuration space on inputs of length \(n\).

When a direction of every tape head’s move is completely determined by a pair \((q, \gamma)\), we call the verifier unidirectional and use the following notation to simplify a description of the verifier’s strategy (or transition function) \(\delta\). A \((Q \times \Gamma)\)-transition \(V_\sigma\) of the verifier \(V\) on \(\sigma\) for every input symbol \(\sigma \in \hat{\Sigma}\) is a unitary operator acting on \(\text{span}\{|q|\gamma\} : q \in Q, \gamma \in \Gamma\) of the form \(V_\sigma |q, \gamma\rangle = \sum_{q', \gamma'} \alpha_{q, \gamma, q', \gamma'} |q', \gamma'\rangle\), where \(q', \gamma' \in Q, \Gamma\), and \(\alpha_{q, \gamma, q', \gamma'} \in \mathbb{C}\). This means that, if \(M\) scans \(\sigma\) in state \(q\) with \(\gamma\), it changes \(q\) to \(q'\), \(\gamma\) to \(\gamma'\) and the tape head moves in direction \(d\) determined uniquely by \(\langle q', \gamma'\rangle\). For this reason, we often express it as \(D(q', \gamma') = d\). When \(d\) is independent of \(\gamma'\), by contrast, we succinctly write \(D(q') = d\). The verifier’s strategy \(\delta\) can be expressed as \(\delta(q, \sigma, q', \gamma', d) = \langle q', \gamma' | V_\sigma | q, \gamma\rangle\) if \(D(q', \gamma') = d\), and 0 otherwise.

For any input \(x\) of length \(n\) in \(\Sigma^*\), \(\delta\) automatically induces the linear operator \(U_\delta^x\) acting on the Hilbert space \(V_N \otimes M\) 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 \(x(0) = k\), \(x(n+2) = \$, \(x(i)\) (\(i \in [1, n+2]\)) is the \(i\)th symbol in \(x\), and \(k' = k + d \pmod{n+2}\). The verifier is called well-formed if \(U_\delta^x\) is unitary on \(V_N \otimes M\) 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 quantum state \(|q_0, 0, \#\rangle\). A single step of the verifier on \(x\) consists of the following process. First, \(V\) applies...
his operation $U^x_n$ to an existing superposition $|\phi\rangle$ in $V_n \otimes M$ and then $U^x_n|\phi\rangle$ becomes the new superposition $|\phi'\rangle$. Second, we define $W_{acc} = span\{(q, k, \gamma) \mid (q, k, \gamma) \in Q_{acc} \times \Sigma_n \times \Gamma\}$, $W_{rej} = span\{(q, k, \gamma) \mid (q, k, \gamma) \in Q_{rej} \times \Sigma_n \times \Gamma\}$, and $W_{non} = span\{(q, k, \gamma) \mid (q, k, \gamma) \in Q_{non} \times \Sigma_n \times \Gamma\}$. Moreover, let $k_{acc}$, $k_{rej}$, and $k_{non}$ be respectively the positive numbers representing "accepting," "rejecting," and "non halting." 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 onto $W_{acc}$, $W_{rej}$, and $W_{non}$. Provided that $|\phi'\rangle$ is expressed as $|\psi_1\rangle + |\psi_2\rangle + |\psi_3\rangle$ for certain three vectors $|\psi_i\rangle \in W_{acc}$, $|\psi_2\rangle \in W_{rej}$, and $|\psi_3\rangle \in W_{non}$, we say that, at this step, $V$ accepts $x$ with probability $|||\psi_1|||^2$ and rejects $x$ with probability $|||\psi_2|||^2$. 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 $|||\psi_3|||^2$. The probability that $x$ is accepted (rejected, resp.) within the first $t$ steps is thus the sum, over all $i \in [1, t]$, of the probabilities with which $V$ accepts (rejects, resp.) $x$ at the $i$th step. In particular, when the verifier is a 1qfa, the verifier’s transition function $\delta$ must satisfy the following additional condition: for all $q, q' \in Q$, $\sigma \in \Sigma$, and $\gamma, \gamma' \in \Gamma$, it holds that $\delta(q, \sigma, q', \gamma', \delta) = 0$ if $d \neq +1$ (i.e., the tape head does not move to the right). Unlike 2qfa verifiers, a 1qfa verifier must stop running after applying $\delta$ at scanning $S$ and then performing a projection measurement. In other words, the 1qfa verifier completely stops by the time the verifier’s tape head moves off $S$ (conceptually, the tape head stops at $t$ since the input tape is circular). Therefore, on any input $x$, the 1qfa verifier halts in at most $|x| + 2$ steps.

In contrast to the verifier, the prover $P$ has a semi-infinite private tape and accesses input $x$ and the communication cell. Let $\Delta$ be a tape alphabet, which includes a special blank symbol $\#$, for the prover’s private tape. The prover is assumed to alter only a "finite" initial segment of his private tape at every step. Let $P$ be the Hilbert space spanned by $\{|y\rangle \mid y \in \Delta^{fin}\}$, where $\Delta^{fin}$ is the set of all finite sequences of tape symbols containing only a finite number of non-blank symbols. The prover’s visible configuration space is the Hilbert space $\mathcal{M} \otimes P$. Formally, the prover $P$ is specified by a series $\{U_{P,i}\}_{x \in \Sigma^*, i \in \mathbb{N}^+}$ of unitary operators, each of which acts on the prover’s visible configuration space, such that $U_{P,i}$ is of the form $S_{P,i} \otimes I$, where $dim(S_{P,i})$ is finite and $I$ is the identity operator. Such a series of operators is often called the prover’s strategy on the input $x$. To refer to the strategy on $x$, we often use the notation $P_i$; namely, $P_i = \{U_{P,i}\}_{x \in \Sigma^*}$. With this notation, the prover can be expressed as $\{P_i\}_{x \in \Sigma^*}$. If the prover has string $y \in \Delta^{fin}_y$ on his private tape and scans symbol $\gamma$ in the communication cell, then he applies $U_{P,i}$ to the quantum state $|\gamma\rangle|y\rangle$ at the $i$th step of the prover’s turn. If $U_{P,i}|\gamma\rangle|y\rangle = \sum_{\gamma', y'} \alpha_{\gamma', y'}|\gamma', y'\rangle$, then the prover changes $y$ into $y'$ and replaces $\gamma$ by $\gamma'$ with amplitude $\alpha_{\gamma', y'}$.

A (global) configuration consists of the four items: $V$’s inner state, $V$’s tape head position, the content of the 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 $V_{x|} \otimes \mathcal{M} \otimes P$, which is called the (global) configuration space of $(P, V)$ on the input $x$; in other words, a (global) configuration is of the form $|q, k, \gamma\rangle|y\rangle$, indicating that $V$ is in inner state $q$, its tape head is at cell $k$, $\gamma$ is in the communication cell, and $P$’s private tape contains $y$. A global configuration $\xi$ is called a halting configuration (a non-halting configuration, resp.) if $\xi$ contains a halting (non-halting, resp.) inner state of $V$. A computation of the QIP protocol $(P, V)$ on the input $x$ constitutes a series of superpositions of configurations resulting by alternate applications of unitary operations of the prover and the verifier including his projection measurement in the following manner. The computation of $(P, V)$ on $x$ starts with the global initial configuration $|q_0, 0\rangle\#\rangle\#^\infty$), where the verifier is in his initial configuration and the prover’s private tape consists only of the blank symbol $\#$. The two players $P$ and $V$ apply their unitary operators $U^x_i$ (as well as measurement) and $P_x$ in turn, starting with the verifier’s move. A projection measurement is made after every move of the verifier to determine whether $V$ is in a halting inner state. Through the communication cell, the two players exchange communication symbols, which cause the two players to be entangled. When the prover (verifier, resp.) writes symbol $\sigma \in \Gamma$ in the communication cell, we customarily say that the prover (verifier, resp.) sends $\sigma$ to the verifier (prover, resp.). More precisely, when $(q, k, \gamma|y\rangle$ is a current (global) configuration, $V$ changes it into $(U^x_n \otimes I_1)|q, k, \gamma\rangle|y\rangle = U^x_n|q, k, \gamma\rangle|y\rangle$, where $I_1$ is the identity operator acting on $P$. After $V$ applies the projection measurement $E_{non}$, the global configuration becomes $(E_{non} \otimes I_1)(U^x_n \otimes I_1)|q, k, \gamma\rangle|y\rangle$. Finally, $P$ changes $(q, k, \gamma|y\rangle$ into $(I_2 \otimes U^x_n)|q, k, \gamma\rangle|y\rangle = |q, k\rangle \otimes U^x_n|\gamma\rangle|y\rangle$, where $I_2$ is the identity operator acting on $V_{x|}$. A superposition $|\Phi\rangle$ of global configurations at the $i$th step is defined recursively as $|\Phi_0\rangle = |q_0, 0\rangle\#\rangle\#^\infty$, $|\Phi_{2i+1}\rangle = (E_{non} \otimes I_1)(U^x_n \otimes I_1)|\Phi_{2i}\rangle$, and $|\Phi_{2i+2}\rangle = (I_2 \otimes U^x_{n+1})|\Phi_{2i+1}\rangle$ for every $i \in \mathbb{N}$. For example, the superposition of (global) configurations after the $2i + 1$ step becomes

$$(E_{non} \otimes I_1)(U^x_n \otimes I_1)(I_2 \otimes U^x_{n+1})\cdots(U^x_n \otimes I_1)(I_2 \otimes U^x_{n+1})(E_{non} \otimes I_1)(U^x_n \otimes I_1)|q_0, 0\rangle\#\rangle\#^\infty.$$
Given any global configuration $\xi$, a local computation path ending with (or leading to) $\xi$ in computation ($\ket{\Phi_0}, \ket{\Phi_1}, \ldots, \ket{\Phi_n}$) of the QIP protocol $(P,V)$ on a given input is a series $(\xi_0, \xi_1, \ldots, \xi_n)$ of global configurations satisfying the following four conditions: $|\xi_0\rangle = |q_0, 0\rangle |\#\rangle^\infty$, $(E_{\text{non}} \otimes I_1)(U^2_a \otimes I_1)|\xi_2\rangle$ contains $|\xi_{2i+1}\rangle$ with non-zero amplitude for all $i \in [0, [(m - 1)/2]]_Z$, $(I_2 \otimes U^2_{\Phi_{i+1}})|\xi_{2i+1}\rangle$ contains $|\xi_{2i+2}\rangle$ with non-zero amplitude for all $i \in [0, [(m - 2)/2]]_Z$, and $\xi_m$ equals $\xi$. Moreover, a (global) computation path ending with $\xi$ is a local computation path $(\xi_0, \xi_1, \ldots, \xi_m)$ ending with $\xi$ for which $|\xi_i\rangle$ appears in $|\Phi_i\rangle$ with non-zero amplitude for every $i \in [0, m]_Z$. Each (global) computation path ends when $V$ enters a certain halting inner state along this computation path. 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.) $x$ within the first $t$ steps. We use the notation $p_{\text{acc}}(x,P,V)$ ($p_{\text{rej}}(x,P,V)$, 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. Notice that, when we discuss the entire running time of the QIP system, we count the number of all steps taken by the verifier (including measurements) as well as the prover.

Let $a$ and $b$ 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 an $(a,b)$-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 $x \in L$, the QIP protocol $(P,V)$ accepts $x$ with probability at least $a$, and

2. (soundness) for any $x \not\in L$ and any prover $P^*$, the QIP protocol $(P^*, V)$ rejects $x$ with probability at least $b$.

Note that an $(a,a)$-QIP system has the error probability at most $1 - a$. This paper discusses only the QIP systems whose error probabilities are bounded from above by certain constants lying in the real interval $[0, 1/2]$. For simplicity, we say that $L$ has a QIP system if there exists a constant (an error bound) $\epsilon \in [0, 1/2]$ such that $L$ has a $(1 - \epsilon, 1 - \epsilon)$-QIP system.

Given any pair $a, b \in [0, 1]$, the notation $\text{QIP}_{a,b}(\langle R \rangle)$, where $\langle R \rangle$ is a set of restrictions, denotes a class of all languages recognized by certain $(a,b)$-QIP systems with the restrictions specified by $\langle R \rangle$. In addition, we define $\text{QIP}(\langle R \rangle)$ as the union $\bigcup_{0 < \epsilon < 1/2} \text{QIP}_{1/2 + \epsilon, 1/2 - \epsilon}(\langle R \rangle)$. In this paper, we shall focus our attention on the following three basic restrictions $\langle R \rangle$: $(1qfa)$ (i.e., 1qfa verifiers), $(2qfa)$ (i.e., 2qfa verifiers), and $(\text{poly-time})$ (i.e., expected polynomial running time). As an example, $\text{QIP}(2qfa, \text{poly-time})$ 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 What if Provers Behave Classically?

To promote a better understanding of the roles of provers in our QIP systems described in Section 2.1, we shall examine a variant of those systems. Recall that, in Dwork-Stockmeyer IP systems [6], mighty provers operate only unitary operators whose entries are all limited to 0 and 1. Significance of such operators is that, recognition power of the corresponding IP systems.

To promote a better understanding of the roles of provers in our QIP systems described in Section 2.1, we shall examine a variant of those systems. Recall that, in Dwork-Stockmeyer IP systems [6], mighty provers operate only unitary operators whose entries are all limited to 0 and 1. Significance of such operators is that, recognition power of the corresponding IP systems.

As Lipton [12] demonstrated, this form of the soundness condition cannot be, in general, replaced by the following weaker form: “$(P,V)$ accepts $x$ with probability at most $1 - b$.” See [6] for a discussion.

In a strict sense, a more exact analogy to deterministic prover may require that even a communication cell behaves classically.
Let us examine the power of classical-prover QIP systems when verifiers are limited to 1qfa’s. It is not difficult to prove that 1QFA ⊆ QIP(1qfa,c-prover) by forcing provers to unalter the communication cell at any step. However, it is not clear whether QIP(1qfa,c-prover) coincides with QIP(1qfa). In what follows, we shall demonstrate that QIP(1qfa,c-prover) actually contains QIP(1qfa).

**Proposition 3.1** QIP(1qfa) ⊆ QIP(1qfa,c-prover).

**Proof.** It was shown in [16 Proposition 4.2] that REG ⊆ QIP(1,1)(1qfa). In a similar way, we can prove that QIP(1qfa,c-prover) also contains all regular languages. Since QIP(1qfa) = REG [16 Theorem 4.1], QIP(1qfa,c-prover) therefore contains QIP(1qfa).

Next, we shall examine the case of 2qfa verifiers. Unlike the 1qfa-verifier case, any containment between QIP(2qfa) and QIP(2qfa,c-prover) is currently unknown; nonetheless, we can verify that QIP(2qfa,poly-time,c-prover) contains 2QFA(poly-time). Therefore, the proper inclusion REG ⊆ QIP(2qfa,poly-time,c-prover) follows as a direct consequence of the result REG ⊆ 2QFA(poly-time) [11].

The following theorem further strengthens this separation to Arthur-Merlin proof systems.

**Theorem 3.2**
1. AM(2pfa) ⊆ QIP(2qfa,c-prover).
2. AM(2pfa,poly-time) ⊆ QIP(2qfa,poly-time,c-prover) ⊆ AM(2pfa).

**Proof.** In a quantum-prover model, it was shown in [16 Lemma 5.2] that the language $\text{Pal}_\#$ = \{x#x}^R \mid x \in \{0,1\}^\ast\} (where x^R is x in the reverse order) of “marked” even-length palindromes belongs to QIP(2qfa,poly-time). By a careful examination of the proof, we find that the same proof works for classical provers. This fact immediately places $\text{Pal}_\#$ into QIP(2qfa,poly-time,c-prover). Hence, the separation between AM(2pfa) and QIP(2qfa,poly-time,c-prover) naturally follows because $\text{Pal}_\#$ is located outside of AM(2pfa) [6]. This separation further leads to the difference between AM(2pfa) and QIP(2qfa,c-prover).

To complete the proof, we shall prove that AM(2pfa) ⊆ QIP(2qfa,c-prover). Since the proof that begins below works for any time-bounded model, we also obtain the remaining claim that AM(2pfa,poly-time) ⊆ QIP(2qfa,poly-time,c-prover).

Let L be any language in AM(2pfa) over alphabet $\Sigma$. We want to show that L is also in QIP(1qfa,c-prover). The important starting point is to notice that L can be recognized by special finite automata $M$, called 2npfa’s [5], that make probabilistic moves and nondeterministic moves in turn as follows. If $x \in L$, then there exists a series of nondeterministic choices by which $M$ halts in accepting states with probability at least $1 - \epsilon$; otherwise, for every series of nondeterministic choices, $M$ halts in rejecting states with probability at least $1 - \epsilon$, where $\epsilon$ is a constant in $[0,1/2)$.

Now, we take a 2npfa $M = (Q, \Sigma \cup \{\#,\}, \delta_M, Q_{acc}, Q_{ rej})$ with nondeterministic states and probabilistic states that recognizes $L$ with error probability at most $\epsilon$, where $0 \leq \epsilon < 1/2$. To simplify our proof, we force $M$ to satisfy the following two conditions: (i) $M$’s tape head does not stay still at any step and (ii) whenever $M$ tosses a fair coin, the tape head moves only to the right. It is not difficult to modify any 2npfa to meet those two conditions. Based on this machine $M$, we shall construct the desired QIP system $(P, V)$ with classical prover $P$ for $L$.

Let $x$ be any input string of length $n$. Let $Q' = Q \cup \{\# \mid p \in Q\}$ be a set of inner states and let $\Gamma = (Q' \times \{\pm 1\}) \cup \{\#, \kappa\}$ be a communication alphabet, where $\kappa$ is a new inner state associated with $p$ and $\kappa$ is a fresh non-blank symbol. The verifier $V$ carries out the procedure that follows $\delta_M : Q \times \hat{\Sigma} P(Q \times \{\pm 1\})$, by which $V$ simulates $M$ step by step.

Let us consider any step at which $M$ tosses a fair coin in probabilistic state $p$ by applying a transition $\delta_M(p, \sigma) = \{(p_0, 1), (p_1, 1)\}$ for certain distinct states $p_0, p_1 \in Q$, where “1” means that the tape head moves rightward by Condition (ii). The verifier $V$ checks whether # is in the communication cell. If this is not the case, $V$ rejects $x$ immediately; otherwise, $V$ makes the corresponding $(Q \times \Gamma)$-transition $V_x[p]\# = \frac{1}{\sqrt{2}}|p_0|(|p_0(p_1)| + |p_1(p_1)|)$ with $D(p_0, (p_1)) = D(p_1, (p_1)) = +1$. The verifier expects a prover to erase the symbol $(p, 1)$ in the communication cell by overwriting it with #. This erasure of symbols guarantees $V$’s move to be unitary.

Next, consider any step at which $M$ makes a nondeterministic choice in nondeterministic state $p$ by a transition $\delta_M(p, \sigma) = \{(p_0, d_0), (p_1, d_1), \ldots, (p_m, d_m)\}$, where $m \in \mathbb{N}$ and $d_i \in \{0, \pm 1\}$ for every $i \in [1, m]$. Here, deterministic moves are treated as a special case of nondeterministic moves. 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 #. Now, assume that # is in the communication cell. Without moving its tape
head, $V$ first sends the designated symbol $\kappa$ to a prover, requesting a pair $(p', d')$ in $Q \times \{\pm 1\}$ to return. This is done by the special $(Q \times \Gamma)$-transition $V_\kappa[p\eta]|\#\rangle = |p\rangle |\kappa\rangle$ with $D(p, \kappa) = 0$. The verifer forces a prover to return a valid form of nondeterministic choice (namely, $(p', d') \in \delta_M(p, \sigma)$) by entering a rejecting inner state if the prover writes any other symbol. Once $V$ receives a valid pair $(p_i, d_i)$, he makes the $(Q \times \Gamma)$-transition $V_\kappa[p\eta]|(p_i, d_i)\rangle = |p_i\rangle |\hat{p}, d_i\rangle$ with $D(p_i, (\hat{p}, d_i)) = d_i$ and expects a prover to erase the communication symbol $(\hat{p}, d_i)$.

The honest prover $P$ must blank the communication cell at the end of every simulation step of $V$ and, in request of $V$ with the symbol $\kappa$, $P$ returns “correct” nondeterministic choice to $V$ (if any). If $x \in L$, then there are a series of nondeterministic choices along which $M$ accepts $x$ with probability at least $1 - \epsilon$. Since the honest prover $P$ sends such a series step by step, $P$ guides $V$ to make correct nondeterministic choices. Moreover, $P$ allows $V$ to simulate correctly $M$’s probabilistic moves by erasing $V$’s communication symbols. Hence, $V$ successfully reaches $M$’s outcomes with the same error probability, and thus the protocol $(P, V)$ accepts $x$ with probability at least $1 - \epsilon$.

Next, consider the case where $x \not\in L$. Notice that 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 particular prover $P^*$ must clear out the communication cell whenever $V$ asks him to do so since, otherwise, $V$ immediately rejects $x$ and further lowers the acceptance probability, a contradiction against the choice of $P^*$. Since $P^*$ is classical, all the computation paths of $V$ have nonnegative amplitudes, which cause only non-destructive interference. This indicates that $P^*$ cannot mimic any existing computation path of $V$. On request for a nondeterministic choice, $P^*$ must return any one of valid nondeterministic choices; otherwise, $V$ rejects immediately. With a series of nondeterministic choices of $P^*$, if $V$ rejects $x$ with probability less than $1 - \epsilon$, then our simulation implies that $M$ also 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 classical-prover $(1 - \epsilon, 1 - \epsilon)$-QIP system for $L$. \hfill \Box

In the above proof, we cannot replace a classical prover by a quantum prover, mainly because a certain quantum prover may fool the aforementioned verifier by (i) returning a superposition of those nondeterministic choices instead of choosing one of the two choices and (ii) using negative amplitudes to make the verifier’s quantum simulation destructive.

In the end of this section, we shall present a QIP protocol with classical provers for the non-regular language $\text{Center} = \{x1y \mid x, y \in \{0, 1\}^*, |x| = |y|\}$, which is known to be in $\text{AM}(2\text{pfa})$ but not in $\text{AM}(2\text{pfa}, \text{poly-time})$ [8]. In the QIP protocol described in the next proof, an honest prover signals the location of the center bit of a given input and then a verifier tests the correctness of the location by employing the quantum Fourier transform (or QFT, in short) in a fashion similar to [11]. An interaction in a QIP protocol constitutes a verifier’s transition, a projection measurement, and a prover’s move.

**Lemma 3.3** For any $\epsilon \in (0, 1)$, $\text{Center} \in \text{QIP}_{1,1-\epsilon}(2\text{qfa}, \text{poly-time}, c\text{-prover}).$

**Proof.** Let $\epsilon$ be any error bound in the real interval $(0, 1)$ and set $N = \lceil 1/\epsilon \rceil$. In what follows, we shall define a desired QIP protocol that witnesses the membership of $\text{Center}$ to QIP$_{1,1-\epsilon}(2\text{qfa}, \text{poly-time}, c\text{-prover})$. Let $\Sigma = \{0, 1\}$ be our input alphabet and let $\Gamma = \{\#, 1\}$ be our communication alphabet. Our QIP protocol $(P, V)$ comprises four phases. The formal description of the behavior of $V$ is given in Table 1 using $(Q \times \Gamma)$-transitions $\{V_\gamma\}_{\gamma \Sigma}$. Let $x$ be an arbitrary input.

1) In the first phase, the verifer $V$ checks whether $|x|$ is odd by moving the tape head toward $\$" together with switching two inner states $q_0$ and $q_1$. To make deterministic moves during this phase, $V$ forces a prover to return only the blank symbol $\#" at any step by entering a rejecting state whenever the prover sends back non-blank symbols. When $|x|$ is odd, $V$ enters the inner state $q_1$ after moving its tape head back to $\$. Hereafter, we consider only the case where the input $x$ has an odd length.

2) In the second phase, $V$ moves its tape head rightward by sending $\#$ to a prover until $V$ receives 1 from the prover. Receiving 1 from the prover, $V$ rejects $x$ unless its tape head is currently scanning 1 on 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.

3) Assume that the tape head is now scanning 1. In the third phase, the computation splits into $N$ parallel branches by applying $V_1|q_0\rangle|1\rangle$. This step is called the first split and it generates the $N$ distinct inner states $r_{1,0}, r_{2,0}, \ldots, r_{N,0}$ with equal amplitudes $1/\sqrt{N}$. The tape head then moves deterministically toward $\$ in the following manner: along the $j$th computation path ($1 \leq j \leq N$) associated with the inner
state $r_{j,0}$, the tape head idles for $2(N - j)$ steps in each tape cell before moving to the next one by changing inner states as

$$r_{j,0} \rightarrow r_{j,N-j} \rightarrow r_{j,N,j-1} \rightarrow r_{j,N-j-1} \rightarrow r_{j,N-j-2} \rightarrow \cdots \rightarrow r_{j,1} \rightarrow r_{j,0},$$

where each number over arrows indicates the direction of the tape head. When the tape head reaches $\$$, it steps back one cell by applying $V_4^\dagger$ or $V_5^\dagger$ and then starts the fourth phase.

4) During the fourth phase, the tape head along the $j$th computation path keeps moving leftward by idling in each cell for $j$ steps, changing inner states as

$$s_{j,0} \rightarrow s_{j,0} \rightarrow s_{j,1} \rightarrow s_{j,1} \rightarrow s_{j,2} \rightarrow \cdots \rightarrow s_{j,1} \rightarrow s_{j,0}$$

until the tape head reaches $\$$. At $\$$, the computation splits again into $N$ parallel branches (called the second split) by applying the QFT $V_4^\dagger$ or $V_5^\dagger$, yielding either the accepting inner state $t_N$ or one of the rejecting inner states in $\{t_l | 1 \leq l < N\}$.

From Table 1 it is not difficult to check that $V$ is indeed well-formed (namely, $U^\dagger$ is unitary for every $x \in \Sigma^*$). The honest prover $P$ should return 1 exactly at the time when $V$ scans the center bit of an input string and at the time when $V$ sends $\#$ to $P$ during the third and fourth phases. At any other step, $P$ should apply the identity operator.

Now, we shall check the completeness and soundness of the obtained 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 just before $V$ reads the center bit 1 of $x$, the first split given by

$$|q_0\rangle = \frac{1}{\sqrt{N}} \sum_{j=1}^{N} |r_{j,0}\rangle \#$$

occurs at the middle of $x$ during the third phase (more precisely, exactly after $n$ steps of $V$ from the start of the second phase) after reading $y1$. Along the $j$th computation path ($1 \leq j \leq N$) associated with the inner state $r_{j,0}$ chosen at the first split, $V$ idles for $2n(N - j)$ steps while reading $z$ and also idles for $2nj$ steps while reading the whole input. Overall, the idling time elapses for the duration of $2n(N - j) + 2nj = 2nN$, which is independent of $j$. Hence, all the $N$ computation paths created at the aforementioned first split have the same length, and thus the superposition of global configurations prior to the second split becomes

$$\frac{1}{\sqrt{N}} \sum_{j=1}^{N} |s_{j,0}, 0\rangle \# |\Psi\rangle$$

for an appropriate quantum state $|\Psi\rangle$ in the Hilbert space...
P associated with the prover’s private tape. The QFT given by $V_4 |s_{j,0}\rangle |1\rangle = \frac{1}{\sqrt{N}} \sum_{t=1}^{N} \exp(2\pi i jl/N) |t|(|\#\rangle)$ then converges all the global configurations to the verifier’s visible accepting configuration $|t_N\rangle |\#\rangle$; that is, $\frac{1}{\sqrt{N}} \sum_{t=1}^{N} \left( \frac{1}{\sqrt{N}} \sum_{j=1}^{N} \exp(2\pi i jl/N) \right) |t,0\rangle |\#\rangle \psi = |t_N,0\rangle |\#\rangle \psi$. Therefore, $V$ accepts $x$ with probability 1.

On the contrary, suppose that $x$ is a negative instance of the form $x = y0z$ with $|y| = |z| = n$. Consider the second, third, and fourth phases. To minimize the rejection probability, a dishonest prover $P^*$ must send the symbol 1 just before $V$ scans 1 on the input tape during the second phase and then $P^*$ must maintain 1 because, otherwise, $V$ immediately rejects $x$ and, moreover, there is no way for classical provers to pass both 1 and # in a form of superposition to deceive the verifier. Let us assume that the uth symbol of $x$ is 1 and $P^*$ sends 1 during the ith interaction, where $1 \leq e \leq 2n + 1$. Obviously, $e \neq n + 1$ because the center bit of $x$ is 0. Consider the first split caused by applying $V_1 |q_0\rangle |1\rangle = \frac{1}{\sqrt{N}} \sum_{j=1}^{N} |r_{j,0}\rangle |\#\rangle$.

For each index $j \in [1, N]$, let $p_j$ be the computation path following the jth branch that starts with the inner state $r_{j,0}$ generated at the first split. Along this computation path $p_j$, the idling time totals $2(|x| - e)(N - j) + 2nj = 2(n + 1 - e)(N - j) + 2nN$. Since $1 \leq j \leq N$, two computation paths $p_j$ and $p_{j'}$ for any distinct values $j$ and $j'$ must have different lengths. Just before the second split, along the jth computation path, we obtain a quantum state $\frac{1}{\sqrt{N}} |s_{j,0}\rangle 0\rangle |\#\rangle \psi_j + |\Delta_j\rangle$, where $|\Delta_j\rangle$ does not contain $|s_{j,0}\rangle$.

At the second split, the QFT further generates $N$ parallel branches $\frac{1}{N} \sum_{j=1}^{N} \exp(2\pi i jl/N) \langle t,0| |\#\rangle \psi_j + |\Delta'_j\rangle$, which equals $\frac{1}{N} \langle t_N,0| |\#\rangle \psi_j + |\Delta''_j\rangle$, where $|\Delta''_j\rangle$ is obtained from $|\Delta'_j\rangle$ by the QFT and $|\Delta''\rangle$ is an appropriate quantum state not containing $|t_N\rangle$. Thus, at most one of the computation paths can reach $|t_N,0\rangle |\#\rangle$. Hence, the probability of $V$ reaching such an acceptance configuration is no more than $1/N^2$. Since there are $N$ computation paths $\{p_j\}_{1 \leq j \leq N}$ generated at the first split, the overall acceptance probability is at most $N \times (1/N^2) = 1/N$. Since $V$’s computation paths always end with certain halting states, it follows that $V$ rejects $x$ with probability $1 - 1/N \geq 1 - \epsilon$.

\[\square\]

### 4 What If a Verifier Reveals Private Information?

In Dwork-Stockmeyer IP systems [6], the prover’s view of the verifier’s computation is limited to a small window (i.e., a communication cell) and the strength of a prover’s strategy hinges on the amount of the information that a verifier is willing to reveal to the prover through this window. Let us consider a situation, in their IP system, that a verifier always alters the communication cell. Since the behavior of a 2pfa verifier depends on not only messages from a prover but also its internal random choices (or its coin flips), no prover can gain more than the information on the number of the verifier’s moves, and therefore any prover knows little of the verifier’s actual configurations.

In Babai’s Arthur-Merlin proof systems [3] (also known as “public-coin” IP systems [7]), on the contrary, the verifier must always pass the information on his next move resulted by his internal random choices, and such information suffices for the mighty prover to keep track of the verifier’s configurations. Dwork and Stockmeyer [6] defined AM(\text{restriction}) as a variant of their original IP systems by requiring their verifiers to reveal publicly next inner states and tape head directions determined by internal coin flips.

Here, we shall consider a straightforward quantum analogy of the above “public-coin” IP systems and investigate their language recognition power. In our QIP system, we demand the verifier to reveal through the communication cell his choice of non-halting inner state as well as his tape head direction at every step. Formally, we define a public QIP system as follows, whereas we sometimes call the original QIP systems private QIP systems in comparison.

**Definition 4.1** A 2qfa-verifier QIP system $(P, V)$ is called \textit{public\$^\ast$} if the verifier’s linear operators $\{U^F_{q,k}\}_{q,k \in \Sigma^N}$ induced by $\delta$ satisfy the following publicness condition: for any tuple $(x, q, k, \gamma)$, if $q$ is a non-halting state, then $U^F_{q,k,\gamma} = \sum q', \xi, d \delta(q, x(k), \gamma, q', \xi, d)|q', k + d (mod |x| + 2), \xi\rangle$, where $\xi = (q', d)$, $x_0 = \xi$, and $x_{i+1} = x_i + 2$.

In particular, when the verifier $V$ is a 1qfa, we omit the information on tape head direction $d$ from the communication symbol $\xi = (q', d)$ in Definition 4.1 since $V$ always moves its tape head to the right (i.e., $d = +1$) and the information on $d$ is obviously redundant. To emphasize the “publicness” of this

\$^\ast$Another variant of public QIP system may require that $U^F_{q,k,\gamma}$ satisfies the same equality only for non-halting states $q'$, instead of $q$. See [13]
new system, we use the specific notation \langle public \rangle. For instance, QIP(2qfa,public) indicates a collection of languages recognized by public QIP systems with 2qfa verifiers. By direct analogy with AM(2qfa), however, we might possibly write QAM(2qfa) for QIP(2qfa,public). In Definition 4.1 since there is no restriction on provers, all public QIP systems with 1qfa verifiers are also private QIP systems with the same verifiers. It therefore holds that, for example, QIP(1qfa,public) ⊆ QIP(1qfa) and QIP(2qfa,public) ⊆ QIP(2qfa).

We shall further demonstrate the power of public QIP systems. Now, we shall concentrate on the language class QIP(1qfa,public). Unlike 1QFA ⊆ QIP(1qfa), the containment 1QFA ⊆ QIP(1qfa,public), which seems to hold naturally at a quick glance, is not yet known. The difficulty of proving this containment is caused by the publicness condition of the public QIP systems. The verifier must announce its next move to a prover, but he may allow the prover to unintentionally make the verifier’s local system entangled with the prover’s local system. We do not know how to cope with this entanglement.

Despite the publicness condition, we can still demonstrate the power of QIP(1qfa,public) beyond 1QFA. Let us consider the language Zero = \{w0 | w ∈ \{0,1\}∗\} is in QIP(1qfa,public), which is known to reside outside of 1QFA \[11\]. In the next lemma, we shall prove that Zero has a public QIP system with 1qfa verifiers.

**Lemma 4.2** Zero ∈ QIP_{1,1}(1qfa,public).

The following proof exploits the prover’s ability to inform the location of the rightmost bit 0 of an instance in Zero. To simplify the description of 1qfa verifiers \(\{\#\}\)-transitions. Let \(V\) be its input alphabet and let \(Q_{non} = \{q_0,q_1\}\), \(Q_{acc} = \{q_{accept,0},q_{accept,1},q_{accept,-1}\}\) and \(Q_{rej} = \{q_{reject,i},q_{reject,i}‘ | i ∈ \{0,±1\}\}\) be respectively the sets of non-halting inner states, of accepting inner states, and of rejecting inner states of \(V\). Using he above-mentioned abbreviation, our communication alphabet \(Γ\) can be defined as \(\{\#,q_0,q_1\}\).

The protocol of \(V\) is described in the following. See Table 2 for the formal description of \(V\)’s \((Q × Γ)\)-transitions. Let \(x = yb\) be any input string, where \(b ∈ \{0,1\}\). The verifier \(V\) stays in the initial state \(q_0\) by publicly announcing \(q_0\) (i.e., sending the communication symbol \(q_0\) to a prover) until the prover returns \#. Whenever \(V\) receives \#, he immediately rejects \(x\) by entering \(q_{reject,-1}\) (after applying either \(V_0[<q_0,\#>|q_1,1]\) or \(V_1[<q_0,\#>|\#]\)) 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 by entering \(q_1\). If the next symbol is \$, then he accepts \(x\) after applying \(V_0[<q_1,|q_1]\); otherwise, he rejects \(x\) by entering \(q_{reject,i}‘\) (after applying either \(V_0[<q_1,|q_i]\) or \(V_1[<q_1,|q_i]\)). Our honest prover \(P\) does not alter the communication cell until \(V\) reaches the right end of \(\$y\) and \(P\) must return \# just before \(V\) reads the symbol \(b\) so that \(V\) can apply \(V_0[<q_1,|\#]\).

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 \(x\) is in Zero, the honest prover \(P\) returns \# just after \(V\) reads the rightmost symbol of \(\$y\). This information helps \(V\) locate the end of \(y\). Moving its tape head rightward, \(V\) confirms that the next scanning symbols are 0\$ and then enters an accepting inner state (either \(q_{accept,0},q_{accept,1},\) or \(q_{accept,-1}\)) with probability 1. 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 essentially deterministic, this strategy only decreases the chance of cheating by \(P^*\). To make the best of the adversary’s strategy, \(P^*\) must return the communication symbol \# just before \(V\) scans 0. Nonetheless, when \(P^*\) returns \#, \(V\) applies \(V_0[<q_0,\#]\)
and then applies $V_0(q_0)|q_i\rangle$ or $V_1(q_1)|q_i\rangle$, where $i \in \{0, \pm 1\}$ and $q_{-1} = \#$. Obviously, this leads to a rejecting inner state of $V$ with certainty. Therefore, the QIP system $(P, V)$ recognizes $\text{Zero}$ with certainty. \hfill \square

It follows from Lemma 4.2 that QIP(1qfa, public) is powerful enough to contain certain languages that cannot be recognized by 1qfa's alone. It is also possible to show that QIP(1qfa, public) contains all languages recognized by 1qfa's whose transition amplitudes are limited to $[0, 1]$. Those 1qfa's are known as $1$-way (deterministic) reversible finite automaton (1rfa, in short) \cite{1}. Let 1RFA denote the collection of all languages recognized by such 1rfa's. As Ambainis and Freivalds \cite{1} showed, 1RFA is characterized exactly as the

\begin{theorem}
1RFA $\subseteq$ QIP$_{1,1}(1qfa, public) \nsubseteq$ 1QFA.
\end{theorem}

\begin{proof}
Firstly, we shall show that 1RFA is contained within QIP$_{1,1}(1qfa, public)$. Take an arbitrary set $L$ recognized by a 1rfa $M = (Q, \Sigma \cup \{\emptyset, \langle\rangle\}, q_0, \delta_M, Q_{\text{acc}}, Q_{\text{rej}})$. Henceforth, we shall construct a public $(1,1)$-QIP system $(P, V)$ that “mimics” a computation of $M$. The desired 1qfa verifier $V = (Q', \Sigma \cup \{\emptyset, \langle\rangle\}, \Gamma, q_0, Q'_{\text{acc}}, Q'_{\text{rej}})$ behaves as follows. Let $Q'_{\text{acc}} = Q_{\text{acc}}, Q'_{\text{rej}} = Q_{\text{rej}} \cup \{q_{\text{rej}, p, q} \mid p \in Q_{\text{non}}, q \in Q, p \neq q\}$, and $Q' = \Gamma = Q \cup Q'_{\text{rej}}$, provided that $q_{\text{rej}, p, q}$'s are all fresh symbols not in $Q$. Assume that $V$ is in inner state $p$, scanning symbol $b$ on an input tape. Whenever $M$ changes its inner state from $p$ to $q$ after scanning $b$, $V$ does so by revealing its next inner state $q$ to a prover. As soon as $V$ finds that the communication symbol has been altered intentionally by the prover, $V$ immediately rejects the input. This process forces any prover to unalter the content of the communication cell. Table 3 gives a list of $(Q \times \Gamma)$-transitions that induces $V$’s strategy $\delta$. It is clear from the list that $\delta$ is well-formed and also the publicness condition for $V$ is met. Finally, the honest prover $P$ is a prover who does not alter any communication symbol; that is, $P$ applies only the identity operator at every step.

| $V_{\emptyset}(q_0)\#$ | $|q\rangle|q\rangle$ if $\delta_M(q_0, \emptyset) = q$ |
|------------------------|-----------------------------------------|
| $V_{\emptyset}(p)p$   | $|q\rangle|q\rangle$ if $\delta_M(p, b) = q$ |
| $V_{\emptyset}(p)|q\rangle$ | $|q_{\text{rej}, p, q}\rangle|q_{\text{rej}, p, q}\rangle$ if $p \neq q$ and $p \in Q_{\text{non}}$ |

Table 3: $(Q \times \Gamma)$-transitions $\{V_\sigma\}_{\sigma \in \Sigma}$ of $V$ for $L$ with $b \in \Sigma \cup \{\emptyset\}$ and $p, q \in Q$. All inner states $q_{\text{rej}, p, q}$ are rejecting states.

On input $x \in \Sigma^*$, the QIP system $(P, V)$ accepts $x$ with certainty if $x \in L$, since $V$ exactly simulates $M$ by the help of the honest prover $P$. Let us consider the opposite case where $x \not\in L$. It is easy to see that the best strategy for a dishonest prover $P^*$ is to keep any communication symbol unchanged because any alteration of the communication symbols causes $V$ to reject $x$ immediately and lowers the acceptance probability of $V$. Against such a prover $P^*$, $V$ obviously enables to reject $x$ with certainty because, in this case, $V$’s final decision is not influenced by the communication symbols. Therefore, $(P, V)$ recognizes $L$ with certainty. Since $L$ is arbitrary, we obtain the desired containment 1RFA $\subseteq$ QIP$_{1,1}(1qfa, public)$.

Secondly, the separation between 1QFA and QIP$_{1,1}(1qfa, public)$ immediately follows from Lemma 4.2, together with the fact that Zero is not in 1QFA \cite{1}. Moreover, since 1RFA $\subseteq$ 1QFA and QIP$_{1,1}(1qfa, public) \nsubseteq$ 1QFA, we can conclude that 1RFA $\nsubseteq$ QIP(1qfa, public). This completes the proof. \hfill \square

Next, we shall examine public QIP systems whose verifiers are 2qfa’s. Similar to Theorem 3\cite{2}, we can claim the following two separations.

\begin{theorem}
1. QIP(2qfa, public, poly-time) $\nsubseteq$ AM(2pfa, poly-time).
2. QIP(2qfa, public, poly-time, c-prover) $\nsubseteq$ AM(2pfa, poly-time).
\end{theorem}

A language that separates the public QIP systems with 2qfa verifiers from AM(2pfa, poly-time) is $Upal = \{0^n1^n \mid n \in \mathbb{N}\}$. Since $Upal$ resides outside of AM(2pfa, poly-time) \cite{1} and $Upal$ belongs to 2QFA(poly-time) \cite{1}, the separation 2QFA(poly-time) $\nsubseteq$ AM(2qfa, poly-time) follows immediately. This separation, however, does not directly imply Theorem 1\cite{1} because, for a technical reason similar to the case of 1qfa verifiers, it is not known whether 2QFA(poly-time) is included in QIP(2qfa, public, poly-time) or even in QIP(2qfa, public, poly-time, c-prover). Therefore, we still need to prove in the next lemma that $Upal$ indeed belongs to both QIP(2qfa, public, poly-time) and QIP(2qfa, public, poly-time, c-prover).
Lemma 4.5 For any constant $\epsilon$ in $(0, 1]$, $Upal \in QIP_{1,1-}(2qfa, public, poly-time) \cap QIP_{1,1-}(2qfa, public, poly-time, c-prover)$.

Proof. In what follows, we shall prove that $Upal$ belongs to $QIP_{1,1-}(2qfa, public, poly-time)$. The proof for $Upal \in QIP_{1,1-}(2qfa, public, poly-time, c-prover)$ is similar. Let $N = \lceil 1/\epsilon \rceil$. Let us define our public QIP system $(P, V)$. The honest prover $P$ always applies the identity operation at every step. The verifier $V$ acts as follows. In the first phase, it determines whether an input $x$ is of the form $0^n1^n$. The rest of the verifier’s algorithm is similar to the one given in the proof of Lemma 4.3. In the second phase, $V$ generates $N$ parallel branches with equal amplitude $1/\sqrt{N}$ by entering $N$ different inner states, say, $r_1, r_2, \ldots, r_N$. In the third phase, along the $j$th branch starting with $r_j$ ($j \in [1, N]$), the tape head idles for $N - j$ steps at each tape cell containing 0 and idles for $j$ steps at each cell containing 1 until the tape head finishes reading 1s. In the fourth phase, $V$ applies the QFT to collapse all computation paths to a single accepting inner state if $m = n$. Otherwise, all the computation paths do not interfere with each other since the tape head reaches $\$$ at different times along different computation paths. 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.

An analysis of the QIP protocol $(P, V)$ for its completeness and soundness conditions is essentially the same as in the proof of Lemma 4.3. In conclusion, $Upal$ is in $QIP_{1,1-}(2qfa, public, poly-time)$, as requested. $\square$

5 How Many Interactions are Necessary or Sufficient?

In the previous two sections, despite heavy restrictions on QIP systems, we have witnessed that quantum interactions between a prover and a qfa verifier remarkably enhance the qfa’s ability to recognize certain types of languages. Since our basic QIP model forces a verifier to communicate with a prover at every move, it is natural to ask whether such interactions are truly necessary. To answer this question, we shall remodel QIP systems so that verifiers are allowed to communicate with provers only at the time when the verifiers need any help from the provers. Throughout this section, we shall shed new light on the number of interactions between a prover and a verifier in our new QIP systems, and we shall carefully examine how many interactions are necessary or sufficient to conduct a given task of language recognition.

5.1 Interaction-Bounded QIP Systems

To study the number of interactions between a prover and a verifier, we want to modify our basic QIP systems so that a prover should alter a communication symbol in the communication cell exactly when the verifier asks the prover to do so. To make such a modification, we first look into the IP systems of Dwork and Stockmeyer [6]. In their system, a verifier is allowed to do computation silently at any chosen time with no communication with a prover; in other words, the verifier interacts with the prover only when the help of the prover is needed and the prover patiently awaits for next interactions without conducting any computation.

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 willfully tampering with the symbol $\#$ in the communication cell. Since the verifier is governed by quantum mechanics, if a malicious prover willfully modifies $\#$, the verifier’s computation may be significantly hampered and the verifier may have no means to prevent such an action of the prover because of the unitarity requirement of the verifier’s strategy $\delta$. To describe a “valid and legitimate” prover $P$ independent of the choice of verifiers, we require the prover’s strategy $P_\pi = \{U_{P,\pi i}\}_{i \in \mathbb{N}^+}$ acting on the prover’s visible configuration space $M \otimes P$ on each input $x$ to do nothing (namely, apply the identity operator). To allow a prover $P$ to maintain the unitarity of his strategy $U_{P,\pi i}$, we also permit the prover to modify his private information $\gamma$ (including a content of the communication cell) when $\gamma$ never appears in a computation with non-zero amplitudes. To formulate this condition independent of the verifier, we need to introduce a series $\{S_i\}_{i \in \mathbb{N}}$ of elements in $\Delta_{fin}^\infty$. This series $\{S_i\}_{i \in \mathbb{N}}$ is defined recursively as $S_0 = \{\#^\infty\}$ and $S_i$ ($i \in \mathbb{N}^+$) is
a collection of all elements $y \in \Delta_{fin}^\infty$ such that, for a certain element $z \in S_{i-1}$ and certain communication
symbols $\sigma, \tau \in \Gamma$, the superposition $U_{P,i}^z|\sigma|z\rangle$ contains the visible configuration $|\tau|y\rangle$ of non-zero amplitude,
namely, $|\langle y| (U_{P,i}^z|\sigma|z\rangle) z\rangle| > 0$. Now, our requested condition is expressed as follows.

(*) For every $i \in \mathbb{N}^+$ and every $y \in S_{i-1}$, $U_{P,i}^y|\#\rangle y\rangle = |\#\rangle y\rangle$.

Any prover $P$ who satisfies Condition (*) is briefly referred to as committed. A trivial example of a committed prover is the prover $P_I$, who always applies the identity operator. A committed prover lets the verifier safely make a number of moves without any “direct” interaction with him. Observe that this new QIP model with committed provers is in essence closer to a circuit-based QIP model of Watrous [17] than the original QIP model is. For convenience, we name our new model an interaction-bounded QIP system and use the new notation QIP$^#(1qfa)$ for the class of all languages recognized with bounded error by such interaction-bounded QIP systems with 1qfa verifiers. Note that standard QIP systems can be naturally transformed into interaction-bounded QIP systems by (possibly) modifying the blank symbol appropriately to a fresh non-blank symbol. This simple fact implies that QIP$^#(1qfa)$ contains QIP(1qfa), which equals REG [10].

**Lemma 5.1** REG $\subseteq$ QIP$^#(1qfa)$.

We are now ready to clarify the meaning of the number of interactions in an interaction-bounded QIP system $(P, V)$. Let us 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 such a query configuration, $V$ is said to query a symbol to a prover. Recall from Section 2.1 the definition of global computation paths. The number of interactions in a given computation means the maximum number, over all global computation paths $\chi$ of the computation of $(P, V)$, of all query configurations of non-zero amplitudes along $\chi$. Let $L$ be any language and assume that $(P, V)$ recognizes $L$. 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 the QIP system $(P, V)$ $k$-interaction bounded if (i) for every $x \in L$, the protocol $(P, V)$ makes at most $k$ interactions on the input $x$ and (ii) for every $x \notin L$ and for every committed prover $P^*$, the protocol $(P^*, V)$ makes at most $k$ interactions on the input $x$. At last, let QIP$^k_qfa$ denote the class of all languages recognized with bounded error by $k$-interaction bounded QIP systems with 1qfa verifiers. Since verifiers can control the number of queries, it is not difficult to show that 1QFA $\subseteq$ QIP$^#(1qfa)$ $\subseteq$ QIP$^#_{k+1}(1qfa)$ $\subseteq$ QIP$^#(1qfa)$ for any constant $k \in \mathbb{N}$. In particular, QIP$^#(1qfa)$ = 1QFA holds.

As the main theorem of this section, we want to show in Theorem 5.2 that (i) 1-interaction helps a verifier but (ii) 1-interaction does not achieve the full power of QIP$^#(1qfa)$.

**Theorem 5.2** QIP$^#_1(1qfa)$ $\subseteq$ QIP$^#(1qfa)$ $\subseteq$ QIP$^#(1qfa)$.

Theorem 5.2 is a direct consequence of Lemma 5.3 and Proposition 5.4 and its proof proceeds as follows. For the first inequality of the theorem, we take 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 $\notin$ QIP$^#_1(1qfa)$, it suffices to show in Lemma 5.3 that Odd belongs to QIP$^#_1(1qfa)$. For the second inequality of the theorem, recall the regular language Zero = $\{x0 \mid x \in \{0,1\}^*\}$. We shall demonstrate in Proposition 5.4 that QIP$^#_1(1qfa)$ does not include Zero. Since REG $\subseteq$ QIP$^#(1qfa)$ by Lemma 5.1, Zero belongs to QIP$^#(1qfa)$, and therefore we obtain the desired separation of QIP$^#(1qfa)$ from QIP$^#_1(1qfa)$. It therefore suffices to prove Lemma 5.3 and Proposition 5.4.

As the first step, we prove Lemma 5.3 that asserts Odd $\in$ QIP$^#_1(1qfa)$.

**Lemma 5.3** Odd $\in$ QIP$^#_1(1qfa)$.

**Proof.** We shall design a 1-interaction bounded QIP system $(P, V)$ that recognizes Odd. Now, let $\Sigma = \{0,1\}$ and $\Gamma = \{\#, a\}$ be respectively an input alphabet and a communication alphabet for $(P, V)$. Let

11 There are a number of possible variants, one of which requires that, for every $i \in \mathbb{N}^+$ and for every $y \in \Delta_{fin}^\infty$, $U_{P,i}^y|\#\rangle y\rangle = |\#\rangle y\rangle$ holds for a certain quantum state $|\psi_{y,i}\rangle$.

12 When $x \in L$ and $P^*$ is a malicious prover, Condition (i) does not impose any restriction on the number of interactions for $(P^*, V)$ on $x$. Instead of Conditions (i)-(ii), we could take a much stronger condition; for example, for every $x$ and every committed prover $P^*$, $(P^*, V)$ makes at most $k$ interactions. Such a stronger condition actually makes simpler the proof of, say, Proposition 5.4.
Let $Q = \{ q_0, q_1, q_2, q_{acc}, q_{rej}, 0, q_{rej, 1} \}$ be a 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 described as follows. Table 4 gives a formal description of $V$’s $(Q, \Gamma)$-transitions. Making no query to a committed prover, $V$ continues to read input symbols until its tape head scans 1 in the input tape. When $V$ reads 1, he queries the symbol $a$ to the committed prover by applying $V_1|q_0\rangle|\#\rangle = |q_0\rangle|a\rangle$. If the prover returns $a$, then $V$ immediately rejects the input. Otherwise, $V$ checks whether the substring of the input after 1 includes an odd number of 0s. This check can be done by $V$ alone by applying $V_b|q_j\rangle|\#\rangle$ and $V_b|q_k\rangle|\#\rangle$ for $b \in \Sigma$. The role of the honest prover $P$ is to work as an eraser, which erases any non-blank symbol written in the communication cell, to help the verifier safely make a transition from the inner state $q_0$ to $q_1$. Note that, without the eraser, $V$ alone cannot make such a transition because of the unitarity requirement of $V$’s strategy. To be more precise, whenever receiving the symbol $a$ from $P$, $V$ returns the symbol $\#$ and copies $a$ into the first blank cell of his private tape.

Technically speaking, to make $P$ unitary, we need to map other visible configurations $|\#\rangle|y\rangle$ for certain $y$’s not having appeared in $P$’s private tape to superpositions of the form $|a\rangle|\phi_{x,y}\rangle$ with an appropriate vector $|\phi_{x,y}\rangle$. By right implementation, we can make $P$ a committed prover.

| $V_1|q_0\rangle|\#\rangle$ = $|q_0\rangle|\#\rangle$ | $V_0|q_0\rangle|\#\rangle$ = $|q_0\rangle|\#\rangle$ | $V_1|q_0\rangle|\#\rangle$ = $|q_1\rangle|a\rangle$ |
| $V_2|q_0\rangle|\#\rangle$ = $|q_{rej, 0}\rangle|\#\rangle$ | $V_0|q_1\rangle|\#\rangle$ = $|q_2\rangle|\#\rangle$ | $V_1|q_1\rangle|\#\rangle$ = $|q_1\rangle|\#\rangle$ |
| $V_3|q_1\rangle|\#\rangle$ = $|q_{rej, 1}\rangle|\#\rangle$ | $V_0|q_2\rangle|\#\rangle$ = $|q_1\rangle|\#\rangle$ | $V_1|q_2\rangle|\#\rangle$ = $|q_2\rangle|\#\rangle$ |
| $V_3|q_2\rangle|\#\rangle$ = $|q_{acc}\rangle|\#\rangle$ | $V_0|q_1\rangle|a\rangle$ = $|q_{rej, 0}\rangle|\#\rangle$ | $V_1|q_1\rangle|a\rangle = |q_{rej, 0}\rangle|\#\rangle$ |

Table 4: $(Q \times \Gamma)$-transitions $\{ V_\sigma \}_{\sigma \in \Sigma}$ of $V$ for Odd

Next, we shall show that $(P, V)$ recognizes $Odd$ with probability 1. Let $x$ be any binary input. First, consider the case where $x$ is in $Odd$. 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 $V$ when $V$ reads 1. This helps $V$ shift to the next mode of checking an odd number of 0s. Since $V$ can check whether $y$ includes an odd number of 0s without any communication with the prover, $V$ eventually accepts $x$ with certainty. Now, 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, focus our attention to 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 essentially deterministic. To maximize the acceptance probability of $V$, a malicious 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 made by the protocol is obviously at most 1, Odd therefore belongs to $QIP^\#_1(1qfa)$, as requested.

As the second step, we need to prove Proposition 5.4 regarding the language $Zero = \{ \alpha \in \{0, 1\}^* \}$. This regular language $Zero$ is known to be outside of IQFA [11]; in other words, $Zero \notin QIP^\#_0(1qfa)$ since $QIP^\#_0(1qfa) = 1QFA$. Proposition 5.4 expands this impossibility result and shows that $Zero$ is not even in $QIP^\#_1(1qfa)$.

**Proposition 5.4** $Zero \notin QIP^\#_1(1qfa)$.

Since the proof of Proposition 5.4 is quite involved, it will be given in the subsequent subsection.

### 5.2 Proof of Proposition 5.4

Our proof of Proposition 5.4 proceeds by way of contradiction. Towards a contradiction, we start with assuming $Zero \in QIP^\#_1(1qfa)$ and take a 1-interaction bounded QIP system $(P, V)$ with 1qfa verifier $V$ that recognizes $Zero$ with error probability at most $1/2 - \eta$ for some constant $\eta > 0$. Our goal is to pick a suitable string $\gamma 0$ and an appropriate prover $P^*$ and to prove that its associated protocol $(P^*, V)$ accepts $\gamma 01^m$ with probability at least 1/2, because this contradicts our assumption that $Zero \in QIP^\#_1(1qfa)$. For this purpose, we shall employ a technical tool, called “query weight”, which is the sum of all the squared magnitudes of query configurations appearing in a computation of the protocol $(P^*, V)$ on an input. However, a different choice of provers may result in different query weights, as seen in the second and the
third computation trees shown in Figure 2. To cope with this unfavorable situation, we shall introduce another computation model, which is not dependent on the choice of provers, and we shall prove that this model gives an upper-bound of the query weight induced by any committed prover. Using this model and its query weight, we shall finally select the desired string \( \tilde{y} \) and the desired prover \( P'_y \).

Now, let \( Q \) be a set of \( V \)'s inner states and let \( \Gamma \) be a communication alphabet. Write \( \Sigma \) for our input alphabet \( \{0, 1\} \). For technicality, we assume without loss of generality that \( V \) never queries at the very time when it enters a certain halting inner state, that is, the time when \( V \)'s tape head scans \( \$ \).

First, we introduce two useful notions: “1-interaction condition” and “query weight.” Fix an input \( x \in \Sigma^* \) and let \( P' \) be any committed prover. For readability, we use the notation \( \text{Comp}_V(P', x) \) to denote a computation of the QIP protocol \( (P', V) \) on input \( x \) when \( P' \) takes a strategy \( P'_x \).

A committed prover \( P' \) is said to satisfy the 1-interaction condition at \( x \) with \( V \) if the corresponding protocol \( (P', V) \) makes at most 1 interaction. Note that, when \( P' \) satisfies the 1-interaction condition at \( x \) with \( V \), for any query configuration \( \xi \) of non-zero amplitude along computation path \( \chi \) in \( \text{Comp}_V(P', x) \), there exists no other query configuration in \( \text{Comp}_V(P', x) \) between the initial configuration and this given configuration \( \xi \) along the computation path \( \chi \). Let \( C^{(1)}_{x,V} \) be the collection of all committed provers \( P' \) who satisfy the 1-interaction condition at \( x \) with \( V \). It is important to note that, whenever a prover in \( C^{(1)}_{x,V} \) answers to \( V \) with non-blank communication symbols with non-zero amplitude, \( V \) must change these symbols back to the blank symbol immediately since, otherwise, \( V \) is considered to make the second query in the next round, according to the definition of our interaction-bounded QIP model.

Now, we choose any prover \( P' \) in \( C^{(1)}_{x,V} \) and consider its computation \( \text{Comp}_V(P', x) \). By introducing an extra projection, we modify \( \text{Comp}_V(P', x) \) as follows. Whenever \( V \) performs a measurement onto \( (W_{\text{non}}, W_{\text{acc}}, W_{\text{reg}}) \), we then apply to the communication cell an additional projection that maps onto the Hilbert space spanned by |\#\>. This projection makes all non-blank symbols collapse. The protocol \( (P, V) \) then continues to the next step. By Condition (*) in Section 5.1, observe that the computation obtained by inserting an extra projection operator at every step of \( V \) is independent of the choice of committed provers. To express this modified computation of \( V \) on \( x \), we use another notation \( M\text{Comp}_V(x) \). Figure 2 illustrates the difference between such a modified computation and two computations generated by two different provers \( P_1 \) and \( P_2 \).

For two strings \( x, y \in \Sigma^* \), the query weight \( wt_V(x, y) \) of \( V \) at \( y \) conditional to \( x \) is the sum of all the squared magnitudes of the amplitudes of query configurations appearing in \( M\text{Comp}_V(xy) \) while \( V \) is reading \( y \). For brevity, let \( wt_V(y) = wt_V(\lambda)(y) \), where \( \lambda \) is the empty string. Note that a query weight \( wt_V(x, y) \) ranges between 0 and 1 and satisfies that \( wt_V(x) + wt_V(y) = wt_V(xy) \) for any strings \( x, y \in \Sigma^* \).

Figure 2: Example of a modified computation. The leftmost graph depicts the modified computation of \( V \) on input \( x \). Two remaining 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. Let query configuration \( \xi \) marked black have zero amplitude. The double 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 our assumption that \((P, V)\) is a 1-interaction-bounded QIP system recognizing Zero with success probability at least \( 1/2 + \eta \). The following lemma shows two properties of the query weights of \( V \).
Lemma 5.5 Let $P'$ be any committed prover and let $x, y$ be any strings.
1. If $P' \in C^{(1)}_{x,y,V}$ then, for every query configuration $\xi$ of non-zero amplitude in $\text{Comp}_P(P', x)$, any computation path $\chi$ in $\text{Comp}_P(P', x)$ ending with $\xi$ appears in $M\text{Comp}_P(x)$ ending with $\xi$ of the same amplitude.
2. If $P' \in C^{(1)}_{x,y,V}$, then $w_{\chi}^{(2)}(y)$ is greater than or equal to the sum of all the squared magnitudes of amplitudes of query configurations in $\text{Comp}_P(P', xy)$ while V’s tape head is reading $y$.

Proof. (1) Take any committed prover $P'$ in $C^{(1)}_{x,y,V}$. Let $\xi$ be any query configuration in $\text{Comp}_P(P', x)$ with non-zero amplitude, say, $\alpha_\xi$. Since $\alpha_\xi$ is not zero in $\text{Comp}_P(P', x)$, there must exist at least one computation path in $\text{Comp}_P(P', x)$ ending with $\xi$. Let us consider such a computation path, say, $\chi$. Since $P'$ satisfies the 1-interaction condition, $\chi$ cannot contain any query configuration of non-zero amplitude except for the last configuration $\xi$. By the definition of $M\text{Comp}_P(x)$, no projection measurement on the communication cell is performed along $\chi$. Hence, all configurations inside $\chi$ must be present also in $M\text{Comp}_P(x)$. Thus, $\chi$ appears in $M\text{Comp}_P(x)$. Since $\chi$ is arbitrary, all the computation paths in $\text{Comp}_P(P', x)$ ending with $\xi$, which contribute to the amplitude $\alpha_\xi$, must appear in $M\text{Comp}_P(x)$. Therefore, the amplitude of $\xi$ in $M\text{Comp}_P(x)$ equals $\alpha_\xi$, as requested.

(2) Assume that $P' \in C^{(1)}_{x,y,V}$. Recall that the query weight $w_{\chi}^{(2)}(y)$ is the sum of all the squared magnitudes of amplitudes of query configurations in $M\text{Comp}_P(xy)$ during V’s reading $y$. By (1), for every query configuration $\xi$ of non-zero amplitude in $\text{Comp}_P(P', xy)$, the squared magnitude of the amplitude of $\xi$ in $\text{Comp}_P(P', xy)$ is equal to that of $\xi$ in $M\text{Comp}_P(xy)$. Note that the converse in general may not be true; that is, there may be a query configuration of non-zero amplitude in $M\text{Comp}_P(x)$ that never appears in $\text{Comp}_P(P', x)$. Figure 2 illustrates such a case. By summing up the squared magnitudes over all query configurations $\xi$ in $\text{Comp}_P(P', xy)$, we immediately obtain (2).

We continue the proof of Proposition [5, 4]. Let us consider a value $\nu$ that is the supremum, over all strings $w$ in Zero, of the query weight of $V$ at $w$; namely, $\nu = \sup_{w \in \text{Zero}} \{w_{\chi}(w)\}$. Observe that $0 \leq \nu \leq 1$ since any query weight is in the real interval $[0, 1]$. For readability, we omit the letter $V$ whenever it is clear from the context. Let $P_t$ denote a committed prover applying only the identity operator at every step.

Claim 1 $\nu > 0$.

Proof. Let us assume that $\nu = 0$. From this assumption, it follows that $w_{\chi}(x) = 0$ for all $x \in \text{Zero}$. To obtain a contradiction, we aim at constructing an appropriate bounded-error 1qfa $M$ that recognizes Zero. Recall that $P$ is a honest committed prover that makes the 1-interaction bounded QIP system $(P, V)$ recognize Zero. Henceforth, we want to assert that even a simple protocol $(P_t, V)$ can recognize Zero. For each input $x \in \text{Zero}$, since $w_{\chi}(x) = 0$, Lemma 5.5 implies that all query configurations in $\text{Comp}_P(P, x)$ must have zero amplitudes. This situation implies that, in the superposition of global configurations at each step of the computation of $(P_t, V)$ on $x$, the verifier V’s next moves cannot be affected by any messages sent out by $P$. Hence, we can replace $P$ by $P_t$ without changing the outcome of $V$.

Let $M$’s inner state be of the form $(q, \sigma)$, which directly reflects both V’s inner state $q$ and a symbol $\sigma$ in the communication cell. The desired 1qfa $M$ behaves as follows. On input $x$, $M$ simulates $V$ on $x$ using the “imaginary” prover $P_t$ by maintaining the content of the communication cell as an integrated part of $M$’s inner states.

Now, we claim that $M$ recognizes Zero with bounded error. Let $x$ be any input string. If $x$ is in $\text{Zero}$, then, since any query configuration with the prover $P_t$ has the zero amplitude, $M$ correctly accepts $x$ with probability $\geq 1/2 + \eta$. Likewise, if $x$ is not in $\text{Zero}$, then $(P_t, V)$ rejects $x$ with probability $\geq 1/2 + \eta$; thus, $M$ also rejects $x$ with the same probability. Therefore, $M$ recognizes Zero with error probability $\leq 1/2 - \eta$, as requested. Since Zero $\notin$ 1QFA, we obtain a contradiction, and therefore $\nu > 0$ follows.

Next, we shall construct a committed prover $P'$ and a string $z \notin \text{Zero}$ that force the protocol $(P', V)$ to accept $z$ with probability at least $1/2$. Let us recall from Section 2.1 the notation $P_w$, which refers to a strategy $(U_\gamma)^{n_{\text{Pos}}} \in \mathbb{P}$ on input $w$. Since $\nu > 0$ by Claim 1, for every real number $\gamma' \in (0, \nu]$, there exists a string $w$ in Zero such that $w_{\chi}(w) \geq \nu - \gamma$. Given any $y \in \Sigma^*$, we set $\gamma_y = \min\{\gamma^2/16(|y| + 1)^2, \nu\}$ and choose the lexicographically minimal string $w_y \in \text{Zero}$ satisfying $w_{\chi}(w_y) \geq \nu - \gamma_y$. For readability, we abbreviate the string $w_y$ as $\tilde{y}$.

Moreover, we define a new committed prover $P'_y$ that behaves on input $\tilde{y}01^m = w_y01^m$, where $m \in \mathbb{N}^+$, in the following fashion: $P'_y$ follows the strategy $P_{50}$ while V’s tape head is reading $\tilde{y}w_y$ and then $P'_y$ behaves
as \( P \) while \( V \) is reading the remaining portion \( y01^m \). Since \( P \) satisfies the 1-iteration condition, we obtain \( P_{y0} \in C_{\bar{g}0}^{(1)}V \). By its definition, \( P_{y0} \) also belongs to \( C_{\bar{g}0}^{(1)}V \). Regarding the notation \( p_{\text{acc}}(x, P, V) \) in Section 2.1, we simply drop \( \langle V' \rangle \) and write \( p_{\text{acc}}(x, P) \) instead. We then claim the following.

**Claim 2** Let any string \( y \in \Sigma^* \), \( p_{\text{acc}}(\bar{g}0, P_{y0}) \geq 1/2 + \eta/2 \).

**Proof.** Let \( y \) be an arbitrary input string. Since \( \bar{g}0 \in Zero \), it must hold that \( p_{\text{acc}}(\bar{g}0, P_{y0}) \geq 1/2 + \eta \) since \( (P, V) \) recognizes Zero. Note that, on the same input \( \bar{g}0 \), the protocol \((P_{y0}, V)\) works in the same way as \((P_{y0}, V)\) while \( V \) is reading \( \bar{g}w_y \). Consider the query weight \( wt(\bar{g}0) \). Since \( wt(\bar{g}0) \leq \nu \), we obtain \( wt(w_y) + wt(w_y)(y0) = wt(\bar{g}0) \leq \nu \), from which it follows that \( wt(w_y)(y0) \leq \gamma_y \) using the inequality \( wt(w_y) \geq \nu - \gamma_y \). Lemma 5.5.1(2) implies that, for any committed prover \( P^* \) in \( C_{\bar{g}0}^{(1)}V \), \( wt(w_y)(y0) \) upper-bounds the sum of all the squared amplitudes of the magnitudes of query configurations in \( \text{Comp}_{P^*}(\bar{g}0, V) \) while the tape head is reading \( y08 \). Notice that \( P_{y0} \) and \( P_{y0}' \) are in \( C_{\bar{g}0}^{(1)}V \). Taking \( P_{y0} \) and \( P_{y0}' \) as \( P^* \), a simple calculation shows that

\[
|p_{\text{acc}}(\bar{g}0, P_{y0}') - p_{\text{acc}}(\bar{g}0, P_{y0})| \leq 2 \left( \frac{\eta}{16(|y| + 1)^2} \right) \leq \frac{\eta}{2}.
\]

where the first inequality is shown as in, e.g., [18, Lemma 9]. Since \( p_{\text{acc}}(\bar{g}0, P_{y0}) \geq 1/2 + \eta \), it follows that \( p_{\text{acc}}(\bar{g}0, P_{y0}') \geq p_{\text{acc}}(\bar{g}0, P_{y0}) - \eta/2 \geq 1/2 + \eta/2 \geq 1/2 + \eta/2 \).

Henceforth, we shall focus our attention to the protocol \((P_{y0}', V)\) working on input \( \bar{g}01^m \) for any number \( m \in \mathbb{N}^+ \) and we shall estimate its acceptance probability \( p_{\text{acc}}(\bar{g}01^m, P_{y0}') \).

The initial superposition of global configurations is \( |q_0||\#|^\#_{\infty} \), where we omit the qubits representing the tape head position of \( V \), because the tape head moves only in one direction without stopping. Let \( V = \text{span}\{|q| \mid q \in Q\}, M = \text{span}\{|\sigma| \mid \sigma \in \Gamma\}, \) and let \( \mathcal{P} \) be a Hilbert space representing the prover’s private tape. For each number \( m \in \mathbb{N} \), we denote by \( |\psi_{y,m}\rangle \) a superposition in the global configuration space \( V \otimes M \otimes \mathcal{P} \) obtained by \((P_{y0}', V)\) on input \( \bar{g}01^m \) just after \( V \)’s tape head moves off the right end of \( \bar{g}0 \) and \( P_{y0}' \) then applies the identity operator while reading the remaining \( 1^m \). Since \( P_{y0}' \) basically does nothing after \( V \) has read \( \bar{g}w_y \), \( |\psi_{y,m}\rangle \) does not depend on the choice of \( m \). We therefore write \( |\psi_y\rangle \) instead of \( |\psi_{y,m}\rangle \), for simplicity. Moreover, we set \( \mu = \inf_{y \in \Sigma^*} \{||\psi_y||\} \).

**Claim 3** \( \mu > 0 \).

**Proof.** Assume that \( \mu = 0 \). Since \( \eta > 0 \), there is a string \( y \in \Sigma^* \) such that \( ||\psi_y||^2 < \eta \). Let us consider the input string \( \bar{g}01 \). Since \( \bar{g}0 \notin Zero \), it follows that \( p_{\text{acc}}(\bar{g}01, P_{y0}') \leq 1/2 - \eta \) by the definition of the QIP system \( (P, V) \). Notice that \( |\psi_y\rangle \) contains only non-halting configurations of non-zero amplitude. For convenience, let \( \alpha_y \) denote the total acceptance probability of \( V \) obtained during \( V \)’s reading \( \bar{g}0 \) (not including \( \bar{g} \)). It holds that \( p_{\text{acc}}(\bar{g}0, P_{y0}') \leq \alpha_y + ||\psi_y||^2 \). While \( V \) is reading \( \bar{g}0 \) on \( \bar{g}01 \), \( P_{y0}' \) takes exactly the same strategy as \( P \) does on the input \( \bar{g}0 \). We thus conclude that \( p_{\text{acc}}(\bar{g}01, P_{y0}') \) is at least \( \alpha_y \). By combining those inequalities, it follows that \( p_{\text{acc}}(\bar{g}0, P_{y0}') \leq p_{\text{acc}}(\bar{g}01, P_{y0}') + ||\psi_y||^2 \). Since \( p_{\text{acc}}(\bar{g}01, P_{y0}') \geq 1/2 + \eta/2 \) by Claim 2, \( ||\psi_y||^2 \geq p_{\text{acc}}(\bar{g}01, P_{y0}') - p_{\text{acc}}(\bar{g}01, P_{y0}') \geq 1/2 + \eta/2 - (1/2 - \eta) > \eta \), a contradiction.

Let \( \epsilon \) be any sufficiently small positive real number and choose the lexicographically first string \( y \) for which \( ||\psi_y||^2 \in [\mu, \mu + \epsilon] \). Let \( I_1 \) and \( I_2 \) denote the identity operators acting on \( \mathcal{P} \) and \( V \), respectively, and write \( W \) for \((I_2 \otimes P_{y0}')(E_{\text{non}} \otimes I_1)(V_1 \otimes I_1)\). For any integer \( j \geq 1 \), let \( \mu \leq ||W^j|\psi_y|| < \mu + \epsilon \) by the definition of \( \mu \). Note that, since \( y \) is fixed, the prover \( P_{y0}' \) uses only a finite number of tape cells on his private tape. Let \( d \) denote the maximal number of cells in use by \( P_{y0}' \). Without loss of generality, we can assume that the Hilbert space \( \mathcal{P} \) is of dimension \( d \). Thus, a series \( \{W^j|\psi_y| \mid j \in \mathbb{N}^+ \} \) is a Hilbert space of dimension \( d/\Gamma \). Since \( \mu > 0 \) by Claim 3, there are two numbers \( j_1, j_2 \in \mathbb{N} \) with \( j_1 < j_2 \) such that \( ||W^{j_1}|\psi_y| - W^{j_2}|\psi_y|| < \epsilon \). By an analysis similar to [11] (see also [8, Lemma 4.1.10]), there exist a constant \( \epsilon > 0 \), which is independent of the value of \( \epsilon \), satisfying that \( ||W^{j_1}|\psi_y| - W^{j_2}|\psi_y|| \leq ||\psi_y| - W^m|\psi_y|| < \epsilon \cdot 1/4 \), where \( m = j_2 - j_1 \geq 1 \). From this inequality follows

\[
|p_{\text{acc}}(\bar{g}0, P_{y0}') - p_{\text{acc}}(\bar{g}01^m, P_{y0}')| \leq ||V_k|\psi_y| - V_k W^m|\psi_y|| = ||\psi_y| - W^m|\psi_y|| < \epsilon \cdot 1/4,
\]

where the first inequality is a folklore (see [18, Lemma 8] for its proof). Since \( \epsilon \) is arbitrary, we can set
\[ \epsilon = (\eta/2c)^4. \]
Because \( p_{acc}(\tilde{y}0, P'_y) \geq 1/2 + \eta/2 \) by Claim 2, it follows that \( p_{acc}(\tilde{y}01^m, P'_y) \geq (1/2 + \eta/2) - \epsilon c^{1/4} = 1/2. \) This contradicts our assumption that, for any committed prover \( P^*, (P^*, V) \) accepts \( \tilde{y}01^m \) with probability \( \leq 1/2 - \eta/2 < 1/2. \) Therefore, \( \text{Zero} \) does not belong to \( \text{QIP}^\#_1(1qfa), \) as requested.

We have finally completed the proof of Proposition 5.4.

6 Challenging Open Questions

Throughout Sections 3–5, we have placed various restrictions on the behaviors of verifiers and provers in our qfa-verifier QIP systems and we have studied how those restrictions affect the language recognition power of the systems. The restricted models that we have considered in Sections 3–5 include: classical-prover QIPs, public QIPs, and interaction-bounded QIPs. After an initial study of this paper, nonetheless, there still remain numerous unsolved questions concerning those QIP systems. Hereafter, we shall give a short list of the important open questions as a guide to future research.

(1) The relationships between quantum provers and classical provers are still not entirely clear in the context of qfa-verifier QIP systems, simply because of the soundness condition imposed on the systems. In particular, we expect to see a fundamental separation between \( \text{QIP}(1qfa) \) and \( \text{QIP}(1qfa,c-prover) \) as well as between \( \text{QIP}(2qfa,\text{poly-time}) \) and \( \text{QIP}(2qfa,\text{poly-time},c-prover). \)

(2) In general, we need to discover precise relationships between “public-coin” IP systems (i.e., AM systems) and public QIP systems beyond Theorem 4.4. Moreover, associated with 2qfa verifiers, we may ask whether \( 2\text{QFA}(\text{poly-time}) \) is properly included in \( \text{QIP}(2qfa,\text{public}) \) and whether \( \text{QIP}(2qfa,\text{public},\text{poly-time}) \) is different from \( \text{QIP}(2qfa,\text{poly-time}). \) Similarly, a separation between \( \text{QIP}(2qfa) \) and \( \text{QIP}(2qfa,\text{public}) \) is also unknown.

(3) A new interaction-bounded QIP system is of special interest in analyzing the roles of interactions between provers and verifiers. For this model, we hope to see that the equality \( \text{QIP}^\#(1qfa) = \text{QIP}(1qfa) \) indeed holds. Unsolvable so far is a general question of whether \( k+1 \) interactions are more powerful than \( k \) interactions. Since a 1qfa verifier is unable to count the number of interactions (or queries), we may not directly generalize the proof of Theorem 5.2 to assert that \( \text{QIP}^\#_{k+1}(1qfa) \neq \text{QIP}^\#_k(1qfa) \) for any constant \( k \) in \( \mathbb{N}^+. \) Nevertheless, we still conjecture that this assertion is true.

(4) It is of great interest to seek an algebraic characterization of our qfa-verifier QIP systems. Such a characterization may shed new light on a nature of quantum interactions between two parties.

References

[1] A. Ambainis and R. Freivalds. 1-way quantum finite automata: strengths, weaknesses and generalizations. In Proc. 39th IEEE Symposium on Foundations of Computer Science, pp.332–341, 1998.
[2] A. Ambainis, A. Kikusts, and M. Valdats. On the class of languages recognizable by one-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.
[3] L. Babai. Trading group theory for randomness. In Proc. 17th ACM Symposium on Theory of Computing, pp.421–429, 1985.
[4] 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.
[5] 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.
[6] C. Dwork and L. Stockmeyer. Finite state verifiers I: the power of interaction. J. ACM 39 (1992) 800–828.
[7] S. Goldwasser and M. Sipser. Private coins versus public coins in interactive proof systems. In Proc. 18th Annual Symposium on Theory of Computing, pp.59–68, 1986.
[8] J. Gruska. Quantum Computing. McGraw Hill, 1999.
[9] J. Hopcroft, R. Motwani, and J. Ullman. Introduction to Automata Theory, Languages, and Computation, Second Edition. Addison-Wesley, 2001.
[10] H. Kobayashi and K. Matsumoto. Quantum multi-prover interactive proof systems with limited prior entanglement. *J. Comput. Syst. Sci.* **66** (2003) 429–450.

[11] 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.

[12] R. J. Lipton. Recursively enumerable languages have finite state interactive proofs. Technical Report, CS-TR-213-89, Computer Science Department, Princeton University, 1989.

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

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

[15] H. Nishimura and T. Yamakami. An application of quantum finite automata to quantum interactive proof systems (extended abstract). In *Proc. 9th International Conference on Implementation and Application of Automata*, Lecture Notes in Comput. Sci., Vol.3317, pp.225–236, 2004.

[16] H. Nishimura and T. Yamakami. An application of quantum finite automata to quantum interactive proof systems. *J. Comput. Syst. Sci.* **75** (2009) 255–269. This extends only the first half portion of [15].

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

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