Non-Interactive Proof Verification Procedures of Reversible and Quantum Finite Automata

Marcos Villagra[1] Tomoyuki Yamakami[1]

Abstract: We discuss the computational complexity of non-interactive verification procedures of finite automata that determine whether given proofs are correct. First, we show that languages admitting non-interactive proof systems by one-way quantum or reversible finite automata are exactly regular languages. When those automata are forced to read their entire input (referred to as a classical acceptance model), there are regular languages that do not admit those systems. In this model, we further show that quantum proofs, even in the form of entangled quantum states, are no more powerful than non-entangled proofs. We also give a semigroup-theoretic sufficient condition for languages characterized by non-interactive proof systems with reversible finite automata. From this condition, we draw a conclusion, that no finite language admits non-interactive proof systems with reversible finite automata, and moreover, the non-closure under complementation of the corresponding class of languages.

Keywords: verification procedure, proof, Merlin-Arthur proof system, quantum automaton, reversible automaton, nondeterminism

1. Introduction

1.1 Background

A computational verification procedure of a "proof" has been discussed for decades using various mathematical models of two-party communication and computation, where a proof is a piece of information that may contain sufficient data to help verify the correctness of a certain assertion. We shall study in this work a situation in which a prover presents a proof, either correct or erroneous, for its verification by a weak verifier, who runs a one-way finite (state) automaton.

A more general interactive model has been studied in the past literature [4], [9]. Dwork and Stockmeyer [4] conducted an initial study on the power and limitations of interactive proof (IP) systems whose verifiers are limited to 2-way probabilistic finite automata (or 2pfa's, in short). A good survey of this model of "classical" interactive proof systems can be found in [3]. More recently, Nishimura and Yamakami [9] studied "quantum" interactive proof (QIP) systems.

The number of interactions in a proof system can be seen as a computational resource which could affect the recognition of languages. Results in polynomial-time computation state that a three-message QIP is sufficient to recognize any language that has a QIP with a polynomial number of messages [7]. This is not believed to be true for a classical IP because it would entail a collapse of the polynomial hierarchy.

In this work, we aim at a better understanding of non-interactive proof systems with finite-state verifiers as a continuation of previous work by Nishimura and Yamakami [9]. A non-interactive proof system is an interactive proof system in which a prover presents a proof and a verifier checks its validity. Classically, this is known as a Merlin-Arthur (MA) proof system. We say that a language L has an MA proof system if there exists a one-way finite automaton (called *Arthur*) such that, for every input $x \in L$ Arthur accepts x if a prover (called Merlin) provides a concrete proof and Arthur verifies that the proof is correct; on the contrary, for every input $x \notin L$ and any proof Arthur rejects x.

Merlin can take various forms. We consider three forms of Merlins. In a deterministic Merlin model, Merlin deterministically chooses a string as a proof and sends it to Arthur, who runs a 1-way reversible finite automaton (or 1rfa). This model corresponds to nondeterministic computation. For this reason, we call it a 1-way nondeterministic reversible automaton or 1nrfa. We also consider a model in which Merlin applies any quantum operation to generate a "quantum proof" (a pure quantum state) and sends it to Arthur who runs a 1-way

[1] Department of Information Science, University of Fukui, 3-9-1 Bunkyo, Fukui 910-8507, Japan. The first author's research is supported by a JSPS Research Fellowship.

quantum finite-state automaton (1qfa). We refer to this model as a Quantum Merlin-Arthur system (QMA). In this model, we differentiate two forms of quantum Merlin. In the first model, Merlin generates the proof, sends it to Arthur and does nothing afterwards; in the second model, Merlin generates the proofs, sends it to Arthur and possibly do some computation on its inner states while Arthur is occupied verifying the proof. If Merlin entangles its inner states with the proof, any local operation on Merlin's side can potentially affect Arthur's verification procedure.

1.2 Contributions

In this work, we shall study the power and limitations of the aforementioned proof systems. To make our notations readable, we say that a machine is nondeterminstic to indicate the deterministic Merlin model and QMA for the quantum Merlin model. Using these notations, we define the following language classes: 1NRFA, and QMA(1qfa).

First, we start by studying some basic properties of the classes 1NRFA and QMA(1qfa). We show that 1NRFA and QMA(1qfa) coincide exactly with the class of regular languages REG. On the contrary, if we demand our machines to read the entire input (a model sometimes called *classical acceptance*), then they are strictly included inside REG. To emphasize classical acceptance, we add the suffix "-CLA" to each language class. Thus, we introduce the language classes 1NRFA-CLA and QMA(1qfa)-CLA. The latter actually agrees with QMA(mo-1qfa), i.e., QMA($1qfa$)-CLA = QMA(mo-1qfa) where mo-1qfa is a 1qfa tha makes only one measurement at the end of its computation. One key technical challenge to show these containments is to argue that once a proof was given, Merlin cannot cheat Arthur by making some prior entanglement with the provided proof. This is achieved by showing that the acceptance probability of Arthur is independent of any changes Merlin does to its inner states, even in the presence of entanglement.

For the class 1NRFA-CLA, we introduce a sufficient condition based on semigroup theory for languages in the class. This result allow us to show several impossibility results on specific languages. Relying on that result, we can show that any finite language cannot be in 1NRFA-CLA, and furthermore, we show the non-closure under complementation property of 1NRFA-CLA.

1.3 Outline of the Paper

This paper is organized as follows. In Section 2 we introduce the new concepts which includes Merlin-Arthur proof systems and nondeterminism for reversible and quantum finite-state automata. In Section 3 we examine the power of a deterministic Merlin. In Section 4 we show our semigroup-theoretic sufficient condition for proving lower-bounds on nondeterministic reversible automata with classical acceptance. Section 5 explores the power of having a quantum Merlin and presents the main argument for a cheating Merlin.

2. Merlin-Arthur Proof Systems and Finite Automata

In this section, we describe the process of the verification of proofs by finite automata. Let $\mathcal{A}$ be a finite automaton. $\mathcal{A}$ will receive, beside its input denoted x, another string w which we will call the *proof*. During its computation, $\mathcal{A}$ will use w in order to accept or reject x. If $x \in L$ then there always exists a proof that makes $\mathcal{A}$ accept x; otherwise, if $x \notin L$ then does not matter which proof is given to $\mathcal{A}$, it will always reject x. This is similar to an IP; however, since the proof is received by Arthur in its entirety at the beginning of the computation, this proof system model is normally referred to as *non-interactive*.

The proof is normally assumed to come from an all-powerful computer traditionally called Merlin (the verifier is Arthur). Merlin will decide what proof he will give to Arthur based only on the input at the beginning of the computation. This is typically called a *Merlin-Arthur proof system*. In this paper we will consider two types of Merlin: 1) deterministic, and 2) quantum.

Now we proceed with a formal definition of the proof verification procedure. We always assume that Σ is the finite input alphabet and the proof is always defined over a (possibly different) finite proof alphabet Γ.

2.1 Verification by Reversible Automata with Deterministic Merlin

Let $\mathcal{A}$ be a deterministic automaton defined as $\mathcal{A} = (Q, \Sigma, \delta, q_0, Q_{acc}, Q_{rej}, \Gamma)$. We also include the proof alphabet Γ in the definition. As explained in the previous paragraph, the change of state of $\mathcal{A}$ depends on the input and the proof; therefore, the transition function is defined as $\delta : Q \times \Sigma \times \Gamma \to Q$. If $x \in L$ then there exists a proof $w \in \Gamma^{|x|}$ such that $\mathcal{A}$ on input (x, w) arrives to an accepting state. If $x \notin L$ then for every proof $w \in \Gamma^{|x|}$, on input (x, w), $\mathcal{A}$ arrives at a rejecting state. Since Merlin is deterministic, there exist a function $\eta : \Sigma^* \to \Gamma^*$ that on input x generates the proof $w = \eta(x)$ where $|w| = |x|$. The function η is not necessarily recursive.

Given our definitions above, it is not hard to see that if $\mathcal{A}$ is a 1dfa, the implementation of a proof yields exactly the well-known 1-way nondeterministic finite state automata (or simply 1nfa). We can further define a 1-way nondeterministic reversible automaton (or 1nrfa)

by allowing $\mathcal{A}$ to a 1rfa. Nondeterminism is the case of a deterministic Merlin, and it is normally called that way in the literature.

In order for δ to be reversible, we need to define it in the following way. Let $q, q' \in Q$ and $\gamma, \gamma' \in \Gamma$. For any $\sigma \in \Sigma$, the transition function is injective with respect to the state and proof, i.e., if $\delta(q, \sigma, \gamma) = \delta(q', \sigma, \gamma')$ then $(q, \gamma) = (q', \gamma')$ whenever both are defined (if both are undefined it does not matter). We define the class of languages recognized by 1nrfa as 1NRFA.

If we require classical-acceptance, then $\mathcal{A}$ is a permutation automaton which we denote 1nrfa-cla. The class of languages recognized by 1nrfa-cla is 1NRFA-CLA.

2.2 Verification by Quantum Automata with Quantum Merlin

Here we explain how a quantum automaton receives and verifies a proof. This also requires an explanation of how a quantum Merlin produces the proof, which is strikingly different than classical Merlin.

Let $\mathcal{H}_A$ be the Hilbert space of the verifier, $\mathcal{H}_M$ be the Hilbert space which corresponds to Merlin's inner workspace, and let $\mathcal{H}_p$ be the Hilbert space containing the proof. We will also need a Hilbert space $\mathcal{H}_{input}$ that will contain the input string to the system. Thus, the Hilbert space of the entire system is $\mathcal{H}_M \otimes \mathcal{H}_{input} \otimes \mathcal{H}_p \otimes \mathcal{H}_M$. For convenience, we will omit the register that corresponds to the input string. Nevertheless, we should remember that it is there.

If Merlin provides the proof without any entanglement, the initial state of the system is $|z\rangle|\phi\rangle|q_0\rangle$, where $|z\rangle$ is the state of Merlin's inner workspace, $|\phi\rangle$ is the proof, and $|q_0\rangle$ is the initial state of Arthur. In general, however, Merlin can provide the proof with entanglement between subspaces $\mathcal{H}_p$ and $\mathcal{H}_M$. Thus, the initial state should be

$$|\psi_0\rangle = \sum_{z,w \in \Gamma^{|x|}} \alpha_{z,w} |z, w\rangle |q_0\rangle. \qquad (1)$$

2.2.1 Quantum Merlin with mo-1qfa Verifier

First we explain the computation of an mo-1qfa $\mathcal{A}$ on input $|x\rangle$ and proof $|\phi\rangle$. Similarly as in the case of interactive proofs, the verifier is a set of unitaries $\{U_{\delta,i}\}_{i \in [0,n+1]}$ where each $U_{\delta,i}$ acts on $\mathcal{H}_A$ conditioned on $\mathcal{H}_{input} \otimes \mathcal{H}_p$. The subscript in each unitary indicates that we apply U_σ (where $\sigma = x_i$) to the internal states conditioned on the i-th position of the input string and the i-th qubit of the proof.

The computation of an mo-1qfa QMA system at the i-th step is

$$|\psi_i\rangle = U_{\delta,i} \cdots U_{\delta,1} |\psi_0\rangle. \qquad (2)$$

Thus, the probabilities of accepting or rejecting an input x given proof $|\phi\rangle$, denoted $p_{acc}(x, \phi)$ and $p_{rej}(x, \phi)$, is $p_{acc}(x, \phi) = \langle \psi_{n+1} | \Pi_{acc} | \psi_{n+1} \rangle$ and $p_{rej}(x, \phi) = \langle \psi_{n+1} | \Pi_{rej} | \psi_{n+1} \rangle$, where Π_{acc} and Π_{rej} are projectors onto the subspaces spanned by the accepting and rejecting states of $\mathcal{A}$ respectively.

Note that in Eq. (2) Merlin gives the proof only at the beginning of the computation and does nothing afterwards. Quantum communication, however, allows a different way of non-interactive system using entanglement. Merlin can provide a proof at the start that is entangled with its own internal states. Then during the computation, Merlin can alter the proof by applying only local operations on his inner-workspace. We call this non-interactive proof computation a QMA system.

In a QMA system with an mo-1qfa verifier, we start with the same initial state $|\psi_0\rangle$ and define the state at the i-th step as

$$|\psi_i\rangle = P_i U_{\delta,i} \cdots P_1 U_{\delta,1} |\psi_0\rangle, \qquad (3)$$

where each P_i is a unitary operator that acts on $\mathcal{H}_M$ conditioned on $\mathcal{H}_p$. Note that Merlin cannot change the proof register directly because of the non-interactive model we want to achieve. Thus, the accepting and rejecting probabilities are defined as before using instead the time evolution given in Eq. (3).

Now we can define the acceptance and rejection conditions of a QMA system with mo-1qfa verifier.

Definition 2.1 (QMA system with mo-1qfa) Let ε be any constant in the interval $[0, 1/2)$. A QMA system with mo-1qfa verifier with error ϵ recognizes language L if it satisfies the following two conditions.

(1) (completeness) For all $x \in L$ there exists $|\phi\rangle \in \mathbb{C}^{\Gamma^{|x|}}$, $p_{acc}(x, \phi) \geq 1 - \varepsilon$.

(2) (soundness) For all $x \notin L$ and any proof $|\phi\rangle \in \mathbb{C}^{\Gamma^{|x|}}$, $p_{rej}(x, \phi) \geq 1 - \varepsilon$.

If the computation of $\mathcal{A}$ is given by Eq. (2) then we say that $\mathcal{A}$ is a 1-way nondeterministic mo-1qfa (mo-1nqfa). If the computation of $\mathcal{A}$ is given by Eq. (3) then we say that $\mathcal{A}$ is a QMA system.

Thus, the class of languages recognized by mo-1nqfa is denoted MO-1NQFA. The class of languages recognized by QMA systems with mo-1qfa verifiers is denoted QMA(mo-1qfa).

2.2.2 Quantum Merlin with 1qfa Verifier

In this section we will explain the time evolution of a 1qfa, i.e., a quantum automaton that measures its states at each step.

The best way to describe the global evolution of the computation of a 1qfa is to keep track of accumulated accepting/rejecting probabilities at each step [6]. Let $\mathcal{V} = \ell_2(Q) \times \mathbb{R} \times \mathbb{R}$. A vector $\Psi = (\psi, p_{acc}, p_{rej})^T \in \mathcal{V}$ means that the 1qfa has accepted thus far with probility p_{acc}, rejecting with p_{rej}, and neither with

probability $\|\psi\|^2$. A norm on $\mathcal{V}$ can be defined as $\|(\psi, p_{acc}, p_{rej})^T\| = \frac{1}{2}(\|\psi\| + |p_{acc}| + |p_{rej}|)$. The time evolution $T_{\delta,i}$ is given for each $i \in \{0, \ldots, n+1\}$ as

$$T_{\delta,i}(\psi, p_{acc}, p_{rej})^T = \begin{bmatrix} \Pi_{non} U_{\delta,i}|\psi\rangle \\ p_{acc} + \|\Pi_{acc} U_{\delta,i}|\psi\rangle\|^2 \\ p_{rej} + \|\Pi_{rej} U_{\delta,i}|\psi\rangle\|^2 \end{bmatrix}.$$

The initial state is $\Psi_0 = (\psi_0, 0, 0)$. Thus, on input $x = x_1 \cdots x_n$ the state at step i is $\Psi_i = T_{\delta,i}\Psi_{i-1}$.

The acceptance of a QMA system with 1qfa verifier can be defined the same way is in the previous section. Here, however, a 1qfa can stop its computation without reading the entire input. Similarly, we define the classes of languages 1NQFA and QMA(1qfa) as before depending on the way Merlin behaves during the computation.

3. The Power of Classical Merlin

In this section, we explore the power of a deterministic Merlin with Arthur running a 1rfa. The main result in this section is that the class of languages recognized by 1nrfa's are exactly the regular languages. However, if we require the 1nrfa to read its entire input (classical acceptance), then it is equivalent to an IP system with 1nrfa-cla verifier.

Theorem 3.1
(1) 1NRFA = REG.
(2) 1NRFA-CLA = IP(1nrfa-cla).
(3) 1NRFA-CLA $\subseteq$ 1NRFA.

Proof. Statement 3 is trivial. To prove statement 2, first note that the containment 1NRFA $\subseteq$ REG can be shown by a standard simulation of a 1nfa by a 1dfa. For the containment REG $\subseteq$ 1NRFA, we proceed by showing that IP(1rfa) $\subseteq$ 1NRFA. The equality IP(1rfa) = REG is implicit in [9].

Let (P, V) be an IP system with 1rfa verifier recognizing L. Let $V = (Q, \Sigma, \delta, q_0, Q_{acc}, Q_{rej}, \Gamma)$. We will define a 1nrfa $\mathcal{A}$ recognizing L.

Let $\mathcal{A} = (Q, \Sigma, \delta', q_0, Q_{acc}, Q_{rej}, \Gamma \times \Gamma)$. Thus, the machine $\mathcal{A}$ have the same states with the same initial state and halting states. The transition function is defined as

$$\delta'(q, \sigma, (\gamma, \gamma')) = (q', \gamma') \text{ if } \delta(q, \sigma, \gamma) = (q', \gamma'), \quad (4)$$

and undefined otherwise. Intuitively, at each step, we need to guess what the prover sent and what the verifiers sends. In this case γ is the symbol received from the prover, and γ' is the symbol sent from the verifier.

We claim that δ' is reversible. Let q and $\check{q}$ be two states. Assume $\delta'(q, \sigma, (\gamma, \gamma')) = \delta'(\check{q}, \sigma, (\check{\gamma}, \check{\gamma}'))$, and say they map to $(\bar{q}, \bar{\gamma})$. Then by our definition we have that $\gamma' = \check{\gamma}' = \bar{\gamma}$ and $\delta(q, \sigma, \gamma) = \delta(\check{q}, \sigma, \check{\gamma}) = (\bar{q}, \bar{\gamma})$. By the reversibility of δ we have that $\gamma = \check{\gamma}$ and $q = \check{q}$, and hence, δ' is reversible.

It is easy to see that for $x \in L$ there exists a proof that makes $\mathcal{A}$. This is because there exists a conversation between V and a honest-prover P. For any $x \notin L$ all proofs makes $\mathcal{A}$ to reject, because no prover can cheat V. This proves statement 1.

In statement 2, note that IP(1rfa-cla) $\subseteq$ 1NRFA-CLA holds, because in Eq. (4), if δ is totally defined for its entire domain then δ' is also totally defined.

To finish the proof, we show that 1NRFA-CLA $\subseteq$ IP(1rfa-cla). Let $L \in$ 1NRFA-CLA and let $M = (Q, \breve{\Sigma}, q_0, \delta, Q_{acc}, Q_{rej}, \Gamma)$ be the 1nrfa recognizing L. We construct an IP system (P, V) recognizing L where V is a p-aut. Let $V = (Q, \breve{\Sigma}, q_0, \delta', Q_{acc}, Q_{rej}, \Gamma)$. The transition function is defined as $\delta'(q, \sigma, \gamma) = (p, \gamma)$ where $\delta(q, \sigma, \gamma) = p$. Thus, the verifier never modifies the communication cell and the prover provides the proof. Thus, δ' is reversible because δ is reversible. If an input x is in L, then there exists a prover that can provide the proof. However, if x is not in L, no prover can ever come up with a proof. □

4. Limitations of Classical Merlin with Classical Acceptance

In this section, we show that if an 1rfa is required to scan the entire input (classical acceptance) then its recognition power is strictly weaker than that of a finite-state automaton.

To prove that a language cannot be recognized by a 1nrfa-cla with classical Merlin, we will show in Section 4.1 a sufficient condition that all languages that are recognized must obey. In Section 4.2, we will show how this criterion can be used to prove impossibility results for specific regular languages.

To read through the subsequent sections, we assume the reader's familiarity with semigroup theory. A thorough treatment of the subject can be found in [10]. Briefly, a semigroup is a set with a binary operation on the same set. A *monoid* is a semigroup containing an identity element. A finite monoid $\mathcal{M}$ recognizes a language L if there exists a semigroup homomorphism $\varphi : \Sigma^* \to M$ and a subset $N \subseteq \mathcal{M}$ such that $L = \varphi^{-1}(N)$. An *idempotent* is an element e such that $ee = e$.

4.1 A Sufficient Condition for Languages Recognized by 1nrfa-cla

Lemma 4.1 *If $\mathcal{A}$ is a 1nrfa-cla recognizing a language L, then there exists a finite monoid $\mathcal{M}$ recognizing L where each element $m \in \mathcal{M}$ is a finite set of one-to-one total relations over the states of $\mathcal{A}$. Furthermore, for any state q and any idempotent $e \in \mathcal{M}$ there exists a relation $r \in e$ such that $(q, q) \in r$.*

Proof. Let $\mathcal{A}$ be a 1nrfa-cla and let $\mathcal{R}(Q)$ be a set of relations on the set of states of $\mathcal{A}$. Define a mapping $\mu : \Sigma^* \times \Gamma^* \to \mathcal{R}(Q)$ such that $(p, q) \in \mu(x, w)$ if and only if there is a path between states p and q on input x and proof w. In particular, each relation $\mu(x, w)$ is a total one-to-one relation because $\mathcal{A}$ is reversible with respect to its input and proof.

Let $\mathcal{S}(Q)$ be the set of all total one-to-one relations on Q and let $\mathcal{M} = 2^{\mathcal{S}(Q)} - \{\emptyset\}$. We define a new mapping $\varphi : \Sigma^* \to \mathcal{M}$ as $\varphi(x) = \bigcup_{w \in \Gamma^{|x|}} \{\mu(x, w)\}$. The function φ maps each word to a finite set of total injective (one-to-one) relations.

Given two subsets $\mathcal{R}, \mathcal{T} \subseteq \mathcal{S}(Q)$ we define the product $\mathcal{R} \bullet \mathcal{T} \stackrel{def}{=} \{r \circ t : r \in \mathcal{R}, t \in \mathcal{T}\}$ where $\circ$ denotes the functional composition.

Claim 1 *$(\mathcal{M}, \bullet)$ is a finite monoid.*

Claim 2 *The function φ is a monoid morphism.*

The recognition of L by $\mathcal{M}$ follows from Claims 1 and 2. We need to find a set $N \subseteq \mathcal{M}$ such that $L = \varphi^{-1}(N)$. Let $N = \{m : \exists r \in m \quad \exists q_{acc} \in Q_{acc} \quad (q_0, a_{acc}) \in r\}$. Clearly, $L = \varphi^{-1}(N)$ because only strings in L have paths from q_0 to certain accepting states.

All what remains to prove the lemma is the last property on the idempotents. First, however, is convenient to define a semigroup action on $\mathcal{M}$. For any $m \in \mathcal{M}$ we define a *right action* of m on Q as $q \cdot m = q'$ if and only if there exists $r \in m$ such that $(q, q') \in r$. Any action requires the following distributivity property.

Claim 3 *For any $m, s \in \mathcal{M}$ and any $q \in Q$, $(q \cdot m) \cdot s = q \cdot (m \bullet s)$.*

Now we are ready to prove the property on the idempotents, which relies on the following claim.

Claim 4 *For any $r \in e$ and any $(q, p) \in r$ there exists $r' \in e$ such that $(q, p) \in r \circ r'$.*

Proof. Let $e \in \mathcal{M}$ be and idempotent and let $r \in m$. For any $(q, p) \in r$, since e is an idempotent, $q \cdot e^2 = p \cdot e = p$. Hence, there exists a relation $r' \in e$ such that $(p, p) \in r'$, and therefore, $(q, p) \in r \circ r'$ and $r \circ r' \in e^2 = e$. □

Claim 5 *For any $q \in Q$ and any idempotent $e \in \mathcal{M}$, $q \cdot e = q$.*

Proof. Any $r \in e$ is a permutation on Q. Hence, by Claim 4, there always exists a relation r such that $(q, q) \in r$. □

□

4.2 Impossibility Results for Regular Languages

In this section, we will show some applications of Lemma 4.1. In particular, we show that there exists regular languages that cannot be recognized by any 1nrfa-cla.

Theorem 4.2 1NRFA-CLA $\subsetneq$ REG.

The proof of the theorem will trivially follows from the examples we present next. As a first application of Lemma 4.1 we show the following.

Proposition 4.3 *If L is a finite language then $L \notin$* 1NRFA-CLA.

Proof. Assume by contradiction that $L \in$ 1NRFA-CLA. Then by Lemma 4.1 the monoid $\mathcal{M}$ recognizes L. Now let $L = \{x_1, \ldots, x_k\}$ be an enumeration of L, i.e., $|L| = k$. Say that $m_i = \varphi(x_i)$. Since $\mathcal{M}$ is finite, each element always have an idempotent power, i.e., for each $m \in \mathcal{M}$ there exists a positive integer ℓ such that m^ℓ is an idempotent.

For any $x_i \in L$, let $q_0 \cdot (\varphi(¢) \bullet m_i) = q$ for some state q. Since $x_i \in L$ we have that $q_0 \cdot (\varphi(¢) \bullet m_i \bullet \varphi(\$)) = q_{acc}$ for some accepting state. Now, m^ℓ is an idempotent and, by Lemma 4.1, we have that $q_0 \cdot (\varphi(¢) \bullet m_i \bullet m_i^\ell) = q$ also holds. Hence, $q_0 \cdot (\varphi(¢) \bullet m_i \bullet m_i^\ell \bullet \varphi(\$)) = q_{acc}$. This implies that $x_i x_i^\ell \in L$ which is a contradiction. □

Next, we give some examples of infinite regular languages. Let $One = \{w1 : w \in \{0,1\}^*\}$ defined over the binary alphabet $\{0, 1\}$ and let $Two = \{w2 : w \in \{0,1\}^*\}$ defined over the ternary alphabet $\{0, 1, 2\}$.

Proposition 4.4 *$One \notin$* 1NRFA-CLA.

Proof. Assume by contradiction that $One \in$ 1NRFA-CLA. Then by Lemma 4.1 there exists a monoid $\mathcal{M}$ recognizing One. Let $x1 \in One$ where $x \in \{0,1\}^*$ is some arbitrary string. Then $q_0 \cdot (\varphi(¢) \bullet \varphi(x1) \bullet \varphi(\$)) = q_{acc}$ for some accepting state q_{acc}. Let $q_0 \cdot (\varphi(¢) \bullet \varphi(x1)) = q$ for some state q and let $\varphi(x0)^k$ be an idempotent power for some positive integer k. Again by Lemma 4.1 $(q, q) \in \varphi(x2)^k$. This implies that $q_0 \cdot \left(\varphi(¢) \bullet \varphi(x1) \bullet \varphi(x0)^k \bullet \varphi(\$)\right) = q_{acc}$, and hence, $x1(x0)^k \in L$ which is a contradiction. □

Corollary 4.5 1QFA $\neq$ 1NRFA-CLA.

Corollary 4.5 follows from the fact that One cannot be recognized by any 1qfa [6].

Proposition 4.6 *$Two \notin$* 1NRFA-CLA.

The proof is omitted due to its similarity to the proof

of Proposition 4.4.

Corollary 4.7 1RFA-CLA $\neq$ 1NRFA-CLA.

It is easy to show that Two cannot be recognized by any 1rfa-cla; thus, Corollary 4.7 follows.

Proposition 4.8 *The class* 1NRFA-CLA *is not closed under complement.*

This can proved by constructing a 1nrfa-cla for the unary language $\{0\}^+$. Since its complement is finite the proposition follows.

5. The Power of Quantum Merlin

We have shown in Theorem 3.1 that the class of languages recognized by 1nrfa is composed exactly of regular languages. Analogously, in this section, we show that all languages in QMA(1qfa) are also regular. An important point is to show that QMA(1qfa) includes 1NRFA. This latter result requires a key technical observation that a quantum Merlin cannot cheat Arthur even in the case of an entangled proof.

Theorem 5.1 QMA(1qfa) = REG.

The proof of this theorem is divided in two subsections. In Section 5.1, we will show that quantum Merlin helps to recognize more languages than classical Merlin. From that result and Theorem 3.1 it follows that QMA(1qfa) contains all regular languages and that IP(1rfa-cla) $\subseteq$ QMA(mo-1qfa). Then in Section 5.2 we will show that quantum Merlin, with mo-1qfa or 1qfa as verifiers, is not more powerful than a finite-state automaton. The main technical contribution of this latter section is that a quantum Merlin that creates entanglement between its inner-states and the proof, it has exactly the same recognition power of a quantum Merlin that does not create any entanglement.

5.1 Quantum Merlin cannot Cheat

We start by showing that a 1nrfa (1nrfa-cla) can be simulated by a 1nqfa (mo-1nqfa), i.e., quantum nondeterminism is more powerful than classical nondeterminism.

Lemma 5.2
(1) 1NRFA-CLA $\subseteq$ MO-1NQFA.
(2) 1NRFA $\subseteq$ 1NQFA.

Proof. We only prove statement (1). Statement (2) is a simple modification of the same argument. Let $\mathcal{A} = (Q, \Sigma, \delta, q_0, Q_{acc}, Q_{rej}, \Gamma)$ be a 1nrfa with classical acceptance and let L be the language recognized by $\mathcal{A}$. We construct a mo-1nqfa machine M recognizing L.

Let $M = (\check{Q}, \Sigma, \check{\delta}, \check{q}_0, \check{Q}_{acc}, \check{Q}_{rej}, \check{\Gamma})$. The idea is to simulate exactly the behavior of $\mathcal{A}$. To that end, let $\check{Q} = Q, \check{q}_0 = q_0, \check{Q}_{acc}, \check{Q}_{rej}, \check{\Gamma} = \Gamma$. The transition function is $\check{\delta}(q, \sigma, \gamma, q') = 1$ if $\delta(q, \sigma, \gamma) = q'$ and $\check{\delta}(q, \sigma, \gamma, q') = 0$ otherwise. $\check{\delta}$ is unitary because δ defines a permutation on Q given the input and proof.

Let $x \in L$. Then there exists a proof w that makes $\mathcal{A}$ accept. Thus, the same proof w makes M accept. Now let $x \notin L$. Let $|\phi\rangle = \sum_{w \in \Gamma^{|x|}} \alpha_w |w\rangle$ and let $|w\rangle = |w_0\rangle \cdots |w_{n+1}\rangle$. Also let $k_{rej} = |Q_{rej}|$. On input x and proof $\alpha_w|w\rangle$, since $\mathcal{A}$ rejects on all proofs, M will arrive on a rejecting state $\alpha_w|q_{rej}\rangle$. The initial configuration of $\mathcal{A}$ is $|x, \phi, q_0\rangle$. After step i the state is $|\psi_i\rangle = U_{\delta,i} \cdots U_{\delta,0} |x, \phi, q_0\rangle = |x, \phi\rangle \sum_{w \in \Gamma^*} \alpha_w |q_{w,i}\rangle$. where $\hat{\delta}(q_0, x_0 \ldots x_i, w_0 \ldots w_i) = q_{w,i}$ Hence, $U_{\delta,n+1} \cdots U_{\delta,0} \alpha_w |x, w, q_0\rangle = \alpha_w |x, w, q_{rej,w}\rangle$. Note that each pair of vectors $|x, w, q\rangle$ for any $w \in \Gamma^{|x|}$ are pair-wise orthogonal, and hence, each path in $|\psi_{n+1}\rangle$ for each proof w does not interfere. Thus, $|\psi_{n+1}\rangle = U_{\delta,n+1} \cdots U_{\delta,0} \sum_{w \in \Gamma^{|x|}} \alpha_w |x\rangle|w\rangle|q_0\rangle = \sum_{w \in \Gamma^{|x|}} \alpha_w |x, w\rangle|q_{rej,w}\rangle$ because $\mathcal{A}$ rejects all proofs on x. Thus, after the measurement, M will end its computation on a rejecting state with probability 1. □

As we saw in Section 2.2, we have two different types of quantum Merlin, one that makes no local operations during the computation (nondeterminism) and one that does. The following proposition shows that in fact both models are equivalent.

Proposition 5.3
(1) MO-1NQFA = QMA(mo-1qfa).
(2) 1NQFA = QMA(1qfa).

The proof of the proposition relies on the following key lemma. Intuitively, the statement says that the acceptance and rejecting probabilities are not affected by the local operations of a quantum Merlin.

Lemma 5.4 *Let $\mathcal{A}$ be a QMA system with mo-1qfa verifier. The acceptance/rejection probability of $\mathcal{A}$ is independent of any changes made by Merlin on its workspace during the computation. The same holds if $\mathcal{A}$ is a 1qfa.*

Proof. When $\mathcal{A}$ applies its first unitary $U_{\delta,1}$ only acts on its inner-states (conditioned on the proof register) and as the identity anywhere else. Hence,

$$U_{\delta,1}|\psi_0\rangle = U_{\delta,1}\left(\sum_{z,w} \alpha_{z,w}|z, w\rangle \otimes |q_0\rangle\right) = \sum_{z,w} \alpha_{z,w}|z, w\rangle \otimes \sum_{q_w^{(1)} \in Q} \beta_{q_w^{(1)}}|q_w^{(1)}\rangle.$$

When the prover acts it only do so on its private workspace. Hence we obtain

$$
\begin{aligned}
&P_1U_{\delta,1}|\psi_0\rangle \\
&\quad=\sum_{z,w}\alpha_{z,w}P_1|z,w\rangle\otimes\sum_{q_w^{(1)}\in Q}\beta_{q_w^{(1)}}|q_w^{(1)}\rangle \\
&\quad=\sum_{z,w,z_w^{(1)}}\alpha_{z,w}\alpha_{z_w^{(1)}}|z_w^{(1)}\rangle|w\rangle\otimes\sum_{q_w^{(1)}\in Q}\beta_{q_w^{(1)}}|q_w^{(1)}\rangle.
\end{aligned}
$$

Then $\mathcal{A}$ applies its next unitary and so on. In general,

$$
\begin{aligned}
|\psi_i\rangle &= P_iU_{\delta,i}\cdots P_1U_{\delta,1}|\psi_0\rangle \\
&=\sum_{z,w,z_w^{(1)},\ldots,z_w^{(i)}}\alpha_{z,w}\alpha_{z_w^{(1)}}\cdots\alpha_{z_w^{(i)}}|z_w^{(i)},w\rangle \\
&\qquad\otimes\sum_{q_w^{(i)}\in Q}\beta_{q_w^{(i)}}|q_w^{(i)}\rangle.
\end{aligned}
$$

After reading the entire input $\mathcal{A}$ measures his halting states. The probability of accepting is

$$
\begin{aligned}
&\langle\psi_{n+2}|\Pi_{acc}|\psi_{n+2}\rangle \\
&\quad=\sum_{z,w,z_w^{(1)},\ldots,z_w^{(n+2)}}|\alpha_{z,w}|^2|\alpha_{z_w^{(1)}}|^2\cdots|\alpha_{z_w^{(n+2)}}|^2 \\
&\qquad\left(\sum_{q_w^{(n+2)}\in Q_{acc}}|\beta_{q_w^{(n+2)}}|^2\right) \\
&\quad=\sum_{z,w}|\alpha_{z,w}|^2\left(\sum_{q_w^{(n+2)}\in Q_{acc}}|\beta_{q_w^{(n+2)}}|^2\right).
\end{aligned}
$$

The second equality follows because for each w and each i we have $\sum_{z_w^{(i)}}|\alpha_{z_w^{(i)}}|^2=1$. Thus, the acceptance probability depends only on the initial amplitudes of the proof and evolution on the verifiers side.

In a similar manner the rejection probability can be shown to be independent of the prover changes to its workspace.

To see that the statement holds for a 1qfa it is sufficient to note that after each unitary $U_{\delta,i}$ is applied a measurement is made solely on the states of $\mathcal{A}$. By arguing in the same way as above, the probabilities of halting or continuing the computation are independent of what the prover does during the computation. □

Lemma 5.5 *(1)* MO-1NQFA $\subseteq$ QMA(mo-1qfa). *(2)* 1NQFA $\subseteq$ QMA(1qfa).

Proof. Let $\mathcal{A}=(Q,\Sigma,\delta,q_0,Q_{acc},Q_{rej},\Gamma)$ be a mo-1qfa and let L be the language recognized by $\mathcal{A}$ with error ϵ. We argue that the same machine $\mathcal{A}$ cannot be cheated by a prover P that perform local operations only on his own workspace. From Lemma 5.4 we know that the accepting/rejecting probabilities are independent of any changes the prover does during the computation. Hence, it does not matter what changes any cheating-prover mades to the amplitudes of any proof, it will fail to fool $\mathcal{A}$. □

Lemma 5.6 *(1)* QMA(mo-1qfa) $\subseteq$ MO-1NQFA. *(2)* QMA(1qfa) $\subseteq$ 1NQFA

Proof. Let $\mathcal{A}=(Q,\Sigma,\delta,q_0,Q_{acc},Q_{rej},\Gamma)$ be a QMA mo-1qfa system recognizing L. We argue that for the same machine $\mathcal{A}$ there exists a honest-prover that does not change its inner-workspace during the computation.

The soundness condition is trivial, because the prover of a mo-1nqfa only provides the proof at the beginning and does nothing afterwards.

The completeness condition also holds because any proof given at the beginning is also a valid proof by Lemma 5.4. □

Thus, from lemmas 5.5-5.6 Proposition 5.3 follows.

5.2 Quantum Merlin does not Help

The main goal of this section is to show that quantum Merlin with an mo-1qfa or 1qfa verifier is not more powerful than a deterministic finite-state automaton.

Proposition 5.7 QMA(mo-1qfa) $\subseteq$ QMA(1qfa) $\subseteq$ REG.

The first containment in Proposition 5.7 is easy to see because a 1qfa can simulate an mo-1qfa. To prove the second containment, we make use of generating functions. However, before making our argument with generating functions, we need to simplify our model of QMA system with 1qfa verifier.

Definition 5.8 Let $\mathcal{A}$ be a QMA 1qfa with error ϵ. Let $\epsilon'<\epsilon$. We define the finite ϵ'-approximation of $\mathcal{A}$, denoted $\mathcal{A}_{\epsilon'}$, as a 1qfa that behaves exactly as $\mathcal{A}$ where Merlin is limited to choose proofs from a finite set for all input lengths. More formally, let x be any string ($n=|x|$). Define $P_n\subseteq\mathcal{H}_M\otimes\mathbb{C}^{\Gamma^n}$ to be a finite set such that for all proofs $|\phi\rangle\in\mathcal{H}_M\otimes\mathbb{C}^{\Gamma^n}$ there exists $|\phi'\rangle\in P_n$ satisfying $\|\phi-\phi'\|\leq\epsilon'$ (this is equivalent to say that P_n is a dense subset). A language L is recognized by $\mathcal{A}_{\epsilon'}$ if and only if (1) for all $x\in L$ there exists $|\phi\rangle\in P_n$ where $\mathcal{A}_{\epsilon'}(x,|\phi\rangle)$ accepts with probability at least $1-\epsilon-\epsilon'$; (2) for all $x\notin L$ for all $|\phi\rangle\in P_n$ where $\mathcal{A}_{\epsilon'}(x,|\phi\rangle)$ rejects with probability at least $1-\epsilon-\epsilon'$.

Next, we show that, for any 1qfa $\mathcal{A}$, there exists an ϵ-approximation $\mathcal{A}_\epsilon$ recognizing the same language. To that end, we make use of the following lemma proven by Nishimura and Yamakami [9].

Lemma 5.9 *[9] For any sufficiently large number $k\in\mathbb{N}^+$, any k-qubit unitary operator U_k, and any real number $\epsilon>0$, there exists a quantum circuit C of size at most $2^{3k}\log^3(1/\epsilon)$ acting on k quits satisfying $\|U_C-U_k\|\leq\epsilon$, where U_C is the unitary operator*

computing C *and* $\|A\| = \sup_{|\psi\rangle\neq 0} \|A|\psi\rangle\|/\||\psi\rangle\|$.

Lemma 5.10 *Let* $\mathcal{A}$ *be a QMA 1qfa with success probability at most* $1/2+\epsilon$ *recognizing a language* L*. Then, for some* $\epsilon' < \epsilon$*, there exists a finite* ϵ'*-approximation* $\mathcal{A}_{\epsilon'}$ *that recognizes* L*.*

Proof. Let x be any string. We need to construct a finite set P of proofs that makes $\mathcal{A}_\epsilon$ recognize L. Let $|\phi\rangle$ be any proof given by Merlin to $\mathcal{A}$. By Lemma 5.4, without loss of generality, we will only consider the case where the proof is given at the beginning and there are no other actions later on Merlin's side.

Assume that when Merlin provides the proof he do so by applying a unitary U_ϕ acting on k qubits, i.e., $|\phi\rangle = U_\phi|0\rangle$.

Fix any finite universal basis for quantum computation $\mathcal{U}$. Let $\mathcal{C}$ be the set of all quantum circuits of size at most $2^{3k}\log^3(1/\epsilon')$ made with gates from $\mathcal{U}$ for some $\epsilon' < \epsilon$. Define $P = \{|\psi\rangle = U|0\rangle : U \in \mathcal{C}\}$. By Lemma 5.9 there exists an unitary U_k of size $2^{3k}\log^3(1/\epsilon')$ such that $\|U_\phi - U_k\| < \epsilon'$. Hence, for any proof $|\phi\rangle$ supplied by Merlin there exists $|\phi'\rangle \in P$ such that $\|\phi - \phi'\| < \epsilon'$. Furthermore, P is finite because there are at most $|\mathcal{U}|^{2^{3k}\log^3(1/\epsilon)}$ many circuits in $\mathcal{C}$.

The soundness follows trivially from the soundness of $\mathcal{A}$ because P is a finite subset of all possible proofs. To show the completeness, let $|\phi\rangle$ be a valid proof for some $x \in L$. Then there exists $|\phi'\rangle \in P$ such that $\||\phi\rangle - |\phi'\rangle\| < \epsilon'$. Hence, the probability of acceptance of $\mathcal{A}_\epsilon$ is at least $1-\epsilon-\epsilon'$ (similarly for rejection). □

Given a QMA system with 1qfa $\mathcal{A}$, by Lemma 5.10, in order to prove that its language is regular, it is sufficient to show that the language recognized by $\mathcal{A}_\epsilon$ for some ϵ is regular. This finishes the proof of Proposition 5.7. We argue by computing the generating function corresponding to $\mathcal{A}_\epsilon$.

We make a brief review of generating functions of languages. We refer the interested reader to the book by Sakarovitch [10] for a thorough treatment.

Let L be a language and let s_n is the number of all words in L of length n. The generating function of L is a formal series $G_L(z) = \sum_{n=0}^{\infty} s_n z^n = \sum_{w\in L} z^{|w|}$.

Note that a language is regular if and only if its generating function is a rational polynomial.

Lemma 5.11 *Let* $\mathcal{A}_\epsilon$ *be the finite* ϵ*-approximation of some 1qfa* $\mathcal{A}$*. If* L *is the language recognized by* $\mathcal{A}_\epsilon$*, then* L *is regular.*

Proof. To show that L is regular, it is sufficient to show that its generating function is a rational polynomial. We define the generating function of $\mathcal{A}_\epsilon$ as

$$\begin{aligned} G_{\mathcal{A}_\epsilon} &= \sum_{x\in\Sigma^*}\left(\sum_{|\phi\rangle\in P} p_{acc}(x,\phi)\right)x \\ &= \sum_{|\phi\rangle\in P}\sum_{x\in\Sigma^*} p_{acc}(x,\phi)x. \end{aligned}$$

The acceptance probability is $p_{acc}(x,\phi) = \langle(0,1,0), T_x(|q_0,\phi\rangle,0,0)\rangle$. Thus, $G_{\mathcal{A}_\epsilon} = (0,1,0)\sum_{|\phi\rangle\in P}\sum_{x\in\Sigma^*} T_x(|q_0,\phi\rangle,0,0)^T x$. If we define $T = \sum_{\sigma\in\Sigma}\sigma T_\sigma$, we have that

$$\begin{aligned} G_{\mathcal{A}_\epsilon} &= (0,1,0)\sum_{|\phi\rangle\in P}\sum_{x\in\Sigma^*} T_x(|q_0,\phi\rangle,0,0)^T x \\ &= (0,1,0)\sum_{|\phi\rangle\in P}\sum_{n\in\mathbb{N}} T^n(|q_0,\phi\rangle,0,0)^T \\ &= (0,1,0)\sum_{|\phi\rangle\in P}(I-T)^{-1}(|q_0,\phi\rangle,0,0)^T. \end{aligned}$$

Let each $\sigma\in\Sigma$ be $\sigma = z$, where z is a variable. The term $\left\langle(0,1,0),(I-T)^{-1}(|q_0,\phi\rangle,0,0)\right\rangle$ is a polynomial on z because the acceptance probability is given by matrix multiplication of each T_σ. Finally, since the class of rational polynomials is close under finite addition, $G_{\mathcal{A}_\epsilon}$ must be a rational polynomial. □

References

[1] A. Ambainis and R. Freivalds. 1-way quantum finite automata: strengths, weaknesses, and generalizations. In *Proc. of the 39th Annual Symposium on Foundations of Computer Science*, pp.332–342, 1998.

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

[3] 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.

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

[5] J. Gruska. *Quantum Computing*. McGraw Hill, 2000.

[6] A. Kondacs and J. Watrous. On the power of quantum finite state automata. In *Proc. 3FOCS 1997*, pp.66–75, 1997.

[7] C. Marriott and J. Watrous. Quantum Arthur-Merlin Games. *Computational Complexity* 14 (2005) 122–152.

[8] C. Moore and J. Crutchfield. Quantum automata and quantum languages. *Theoret. Comput. Sci.* 237 (2000) 275–306.

[9] H. Nishimura and T. Yamakami. An application of quantum finite automata to interactive proof systems. *J. Comput. System Sci.*, 75 (2009) 255-269. Extended abstract in Proc. of CIAA 2004.

[10] J. Sakarovitch. *Elements of Automata Theory*. Cambridge University Press, 2009.