On the complexity of minimizing probabilistic and quantum automata

Paulo Mateus[a,*], Daowen Qiu[a,b,c,†] and Lvzhou Li[b‡]

[a] *Dep. Matemática, Instituto Superior Técnico, Universidade Técnica de Lisboa, Portugal*
SQIG, Instituto de Telecomunicações,
Av. Rovisco Pais 1049-001, Lisbon, Portugal

[b] *Department of Computer Science, Sun Yat-sen University, Guangzhou 510006, China*

[c] *The State Key Laboratory of Computer Science, Institute of Software,*
Chinese Academy of Sciences, Beijing 100080, China

Abstract

Several types of automata, such as probabilistic and quantum automata, require to work with real and complex numbers. For such automata the acceptance of an input is quantified with a probability. There are plenty of results in the literature addressing the complexity of checking the equivalence of these automata, that is, checking whether two automata accept all inputs with the same probability. On the other hand, the critical problem of finding the minimal automata equivalent to a given one has been left open (C. Moore, J.P. Crutchfield, Theoretical Computer Science 237 (2000) 275-306, p. 304, Problem 5). In this work, we reduce the minimization problem of probabilistic and quantum automata to finding a solution of a system of algebraic polynomial (in)equations. An EXPSPACE upper bound on the complexity of the minimization problem is derived by applying Renegar's algorithm. More specifically, we show that the state minimization of probabilistic automata, measure-once quantum automata, measure-many quantum automata, measure-once generalized quantum automata, and measure-many generalized quantum automata is decidable and in EXPSPACE. Finally, we also solve an open problem concerning minimal covering of stochastic sequential machines (A. Paz, Introduction to Probabilistic Automata, Academic Press, 1971, Page 43).

Keywords: Probabilistic automata; Quantum automata; Decidability; Minimization.

[*] Corresponding author. *E-mail address:* pmat@math.ist.utl.pt. *Fax:* +351 218417048

[†] *E-mail address:* issqdw@mail.sysu.edu.cn.

[‡] *E-mail address:* lilvzhou@gmail.com.

1. Introduction

The seminal work by Tarski [22] opened a wide area of research on algorithms for algebraic geometry. Indeed, when Tarski showed that the first-order theory of real ordered fields was decidable (a somewhat opposite result to Godel's incompleteness theorem for Peano's arithmetic), the decision algorithm that was proposed was very inefficient. Throughout the last century several improvements were made [2] but, curiously, these results are fairly unknown, except by the core researchers of the field. In this work, we show that these algorithms can be used to solve several open problems concerning probabilistic and quantum automata.

Four decades ago, many computation models using probabilities flourished and are, presently, well accepted. Among the motivations for such models we can find fault modeling, environment modeling, quantification of non-determinism, randomization as a computational resource, and others. The literature on this topic is large (e.g. see [18]), and there is a full spectrum of models for all tastes, for instance, probabilistic automata (a generalization of finite state automata), stochastic sequential machines (a generalization of Mealy's machines), or probabilistic Turing machines (a generalization of Turing machines). Together with these models several problems were opened concerning the expressivity and complexity of such devices. Some of those questions remain open today, for instance determining whether there exists a minimal covering of *stochastic sequential machines* (SSM) (see [18], Page 43).

The practical interest for minimizing stochastic automata decreased with the engineering breakthrough that made memory very cheap. Presently, there is no interest in finding a classical automaton that can be modeled with less than 100 bits or so, as it is easy and cheap to produce devices with much more memory. However, quantum information emerged and this scenario changed. Indeed, it is still far beyond the reach of today's technology to make quantum memory with 100 entangled qubits and so, it is important to understand what tasks can be implemented using minimal quantum memory. Thus, minimizing quantum machines is a relevant problem in practice since we might not be able to have arbitrary large quantum memory in the near future.

One of the most appealing aspects for the introduction of quantum models is that we expect them to surpass their classical counterparts in time efficiency. Naturally, the study of theoretical models of computation complying with quantum mechanics has become an important research field. Quantum computers were first suggested by Benioff [3], and Feynman [8] and then further formalized by Deutsch [7]. Their power has been successfully shown by Shor's quantum algorithm for factoring integers in polynomial time [21], and afterwards, by Grover's algorithm for searching in a database of size n with only $O(\sqrt{n})$ memory accesses [9]. As we know, these algorithms are based on *quantum Turing machines* which seem complicated to implement using today's technology. Therefore, after it has turned out that building powerful quantum computers is still a long-term goal, it become clear that there is a need to

study "small-size" quantum processors using variations of the models that have shown their relevance in the classical cases [10], namely automata.

Quantum finite automata (QFA) were first studied by Moore and Crutchfield [16], Kondacs and Watrous [12], Ambainis and Freivalds [1], Brodsky and Pippenger [5], and other authors. The study of QFA is mainly divided into two areas: one is *one-way quantum finite automata* (1QFA) whose tape head moves one cell only to right at each evolution step; and the other is *two-way quantum finite automata* (2QFA), in which the tape head is allowed to move right or left, or remain stationary. There are two types of 1QFA: *measure-once* 1QFA (MO-1QFA) initiated by Moore and Crutchfield [16], and *measure-many* 1QFA (MM-1QFA) studied first by Kondacs and Watrous [12]. In MO-1QFA, there is only a measurement for computing each input string, performed after reading the last symbol; in contrast, in MM-1QFA, measurement is performed after reading each symbol.

Further generalizations of MO-1QFA and MM-1QFA are *measure-once one-way general quantum finite automata* (MO-1gQFA) and *measure-many one-way general quantum finite automata* (MM-1gQFA) [15], respectively. These generalizations are obtained by replacing pure states with mixed states and by replacing unitary operators with trace-preserving quantum operations. More specifically, in a 1gQFA, each symbol in the input alphabet induces a trace-preserving quantum operation, instead of a unitary transformation, and the states are mixed states instead of pure states. In an MO-1gQFA, as in an MO-1QFA, a measurement deciding to accept or reject is performed at the end of a computation, while in an MM-1gQFA, a similar measurement is performed after each trace-preserving quantum operation on reading each input symbol, as in an MM-1QFA.

The minimization of states for classical and probabilistic automata has been thoroughly studied [11, 19, 18, 4]. However, for probabilistic automata, the chief results focus only on reducing states or establishing sufficient and necessary conditions for an automaton to be minimized (see a systematical introduction by Paz [18] and by Bukharaev [4]). For quantum automata, the minimization problem has not been addressed yet. Here we recall an important problem regarding the minimization of MO-1QFA proposed by Moore and Crutchfield ([16], p. 304, Problem 5): *Is each quantum regular language (QRL) recognized by a unique QFA (up to isomorphism) with the minimal number of dimensions?* Here QFA means MO-1QFA, and the number of dimensions is the number of quantum basis states of QFA. Therefore, this problem is essentially that of minimizing the number of states of MO-1QFA.

In this work, we address this problem for several types of probabilistic and quantum finite automata and give an EXPSPACE algorithm to find a minimal automaton equivalent to a given one. More specifically, we show that the state minimization of probabilistic automata, MO-1QFA, MM-1QFA, MO-1gQFA, and MM-1gQFA, is decidable. Therefore, we solve the minimization problem for all these cases. In addition, our method for minimizing probabilistic automata is new and different from other approaches. Finally, we solve an open problem

proposed by Paz (see [18], page 43), that according to Paz, and as far as the authors know, is still open today. The problem consists of determining the decidability (and upper-bound for the complexity) of the minimal covering of SSM.

The paper is organized as follows. In Section 2, we recall the basic notions and relevant results that will be used in the sequel. There we also present the main idea of the minimization procedure and depict the general algorithm for minimizing automata. In Section 3 we present in detail the minimization of probabilistic automata. Subsequently, in Section 4 we address the minimization of MO-1QFA. For the sake of completeness, in Section 5 we solve the minimization of MM-1QFA and then, in Section 6, we solve the minimization of MO-1gQFA and MM-1gQFA. Although the minimization method for these models is more or less similar that of MO-1QFA, there are subtle differences and issues that deserve to be clarified. Finally, in Section 7 we solve an open problem concerning the covering minimization of SSM proposed by Paz [18]. To conclude, we summarize the main achievements in Section 8.

2. Preliminaries

In this section we introduce the basic notions and results concerning quantum information. Then, we present the main idea for the minimization procedure, which is adapted to several types of quantum and probabilistic finite automata throughout the rest of the paper.

2.1. Basic notions and relevant results

For a matrix or a linear operator A, we use A^*, $A^\top$ and $A^\dagger$ to denote its conjugate, transpose, and conjugate transpose, respectively. $\mathrm{Tr}(A)$ denotes the trace of matrix (operator) A. Generally, we use $\mathcal{H}$ to denote a finite-dimensional Hilbert space. Let $L(\mathcal{H})$ denote the set of all linear operators from $\mathcal{H}$ to itself. A mapping $\Phi : L(\mathcal{H}) \to L(\mathcal{H})$ is called a super-operator on $\mathcal{H}$.

A detailed overview of quantum information can be found in [17], and here we just present some relevant notions. According to the postulates of quantum mechanics, the state of a closed quantum system is represented by a unit vector $|\psi\rangle$ in a Hilbert space $\mathcal{H}$, and the state evolution of a closed quantum system is described by a unitary transformation on $\mathcal{H}$. A more general tool to describe the states of a quantum system is based on density operators. A density operator ρ on Hilbert space $\mathcal{H}$ is a linear operator satisfying the following conditions:

(1) (Trace condition) ρ has trace equal to 1, that is, $\mathrm{Tr}(\rho) = 1$;

(2) (Positivity condition) $\rho \geq 0$, that is, for any $|\psi\rangle \in \mathcal{H}$, $\langle\psi|\rho|\psi\rangle \geq 0$.

We denote the set of all density operators on Hilbert space $\mathcal{H}$ by $D(\mathcal{H})$.

In practice, absolutely closed systems do not exist, because every system interacts with its environment, and thus is open. In this case the state evolution of an open quantum system is characterized by a *quantum operation* [17]. A quantum operation, denoted by $\mathcal{E}$, has an *operator-sum representation* given by

$$\mathcal{E}(\rho) = \sum_k E_k \rho E_k^\dagger, \tag{1}$$

where ρ is a density operator on the input space $\mathcal{H}_{in}$, $\mathcal{E}(\rho)$ is a completely positive operator on the output space $\mathcal{H}_{out}$, and the set of $\{E_k\}$, known as *operation elements*, are linear operators from $\mathcal{H}_{in}$ to $\mathcal{H}_{out}$. Furthermore, $\mathcal{E}$ is said to be trace-preserving if the following holds:

$$\sum_k E_k^\dagger E_k = I, \tag{2}$$

where I is the identity operator on $\mathcal{H}_{in}$. Any physically allowed operation is a trace-preserving quantum operation (also called a completely positive trace-preserving mapping).

In the following, we introduce a useful linear mapping *vec* which maps a matrix $A \in \mathbb{C}^{n\times n}$ to an n^2-dimensional column vector, defined as follows:

$$vec(A)((i-1)n + j) = A(i, j). \tag{3}$$

In other words, $vec(A)$ is the vector obtained by taking the rows of A, transposing them to form column vectors, and stacking those column vectors on top of one another to form a single vector. For example, let

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}, \tag{4}$$

then we have that

$$vec(A) = \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix}. \tag{5}$$

If we let $|i\rangle$ be an n-dimensional column vector where the i'th entry is 1 and all the others are 0's, then $\{|i\rangle\langle j| : i, j = 1, \cdots, n\}$ form a basis of $\mathbb{C}^{n\times n}$. Therefore, the linear mapping *vec* can also be defined as follows:

$$vec(|i\rangle\langle j|) = |i\rangle|j\rangle. \tag{6}$$

Let A, B, C be $n \times n$ matrices, and let u, v be n-dimensional column vectors. Then the linear mapping *vec* satisfies the following properties [24, 26]:

$$vec(ACB) = (A \otimes B^\top) vec(C), \tag{7}$$

$$\mathrm{Tr}(AB) = vec(A^\top)^\top vec(B), \tag{8}$$

$$vec(uv^\dagger) = u \otimes v^*. \tag{9}$$

This minimization method is based, on one hand, on the decidability of the equivalence of the considered automata and, on the other hand, on the decidability of the theory of real ordered fields [2, 6, 20]. So, we further introduce the concepts and results concerning the decidability of the theory of real ordered fields [2, 6, 20].

The decision problem for the existential theory of the reals [20] is the problem of deciding if the set $\mathbb{S} = \{x \in \mathbb{R}^n : \mathbf{P}(x)\}$ is nonempty, where $\mathbf{P}(x)$ is a predicate defined as Boolean function of atomic predicates either of the form $f_i(x) \geq 0$ or $f_j(x) > 0$, f's being real polynomials (with rational coefficients). For this decision problem it is important to know three parameters: the number of atomic predicates m (i.e., the number of polynomials), the number of variables n, and the highest degree d among all atomic predicates forming $\mathbf{P}(x)$.

Canny [6] developed a PSPACE algorithm in n, m, d for the above problem, but its time complexity is very high. Later, Renegar [20] designed an asymptotically optimal algorithm of time complexity $(md)^{O(n)}$. Furthermore, to find a sample of $\mathbb{S}$ requires $\tau d^{O(n)}$ space if all coefficients of the atomic predicates use at most τ space (see [2], page 518). Here, to find a sample of $\mathbb{S}$ means to discover a solution of $\mathbf{P}(x)$. We summarize these results in the following theorem.

Theorem 1 ([2, 6, 20])**.** *Let $\mathbf{P}(x)$ be a predicate which is a Boolean function of atomic predicates either of the form $f_i(x) \geq 0$ or $f_j(x) > 0$, with f's being real polynomials. There is an algorithm to decide whether the set $\mathbb{S} = \{x \in \mathbb{R}^n : \mathbf{P}(x)\}$ is nonempty in PSPACE in n, m, d, where n is the number of variables, m is the number of atomic predicates, and d is the highest degree among all atomic predicates of $\mathbf{P}(x)$. Moreover, there is an algorithm of time complexity $(md)^{O(n)}$ for this problem. To find a sample of $\mathbb{S}$ requires $\tau d^{O(n)}$ space if all coefficients of the atomic predicates use at most τ space.*

We will use the above theorem to deal with the state minimization of QFA. However, since QFA are usually defined over the field of complex numbers, we need to transform a problem over the field of complex numbers to one over real numbers. This transformations will be based on the following observation.

Remark 2. *Any complex number $z = x + yi$ is determined by two reals x and y, and any complex polynomial $f(z)$ with $z \in \mathbb{C}^n$ can be equivalently written as $f(z) = f_1(x, y) + if_2(x, y)$ where $(x, y) \in \mathbb{R}^{2n}$ is the real representation of z, and f_1 and f_2 are real polynomials. Thus, the set $\mathbb{S}'$ defined over the field of complex numbers with n complex variables and m complex polynomials can be equivalently described by $\mathbb{S}$ over the field of real numbers with $2n$ real variables and $2m$ real polynomials.*

2.2. The main idea of state minimization

In this work we show that the state minimization problem of various types of quantum finite automata and probabilistic finite automata is decidable. As mentioned before, our results are based on the decidability of the equivalence of these automata and, moreover, on the decidability of the theory of real ordered fields [2, 6, 20]. Thus, we start by recalling the equivalence problem of various types of QFA's and probabilistic automata and then, we present the main idea for minimizing such automata.

Roughly speaking, two automata over the same input alphabet are said to be equivalent if they accept each input string with the same probability. For example, two probabilistic automata $\mathcal{A}_1$ and $\mathcal{A}_2$ on input alphabet Σ are said to be equivalent if they have the same accepting probability for each input $x \in \Sigma^*$. The equivalence problem of some type of automata is to determine whether any two given automata of this type are equivalent or not. So far, the equivalence problem has been proven to be decidable for probabilistic automata [18, 23], MO-1QFA [5, 14], MM-1QFA [13], and one-way QFA with mixed states [15]. Indeed, for each of these automata types, a certain bound on the word length has been derived such that two automata are equivalent if and only if they have the same accepting probability for all words with length less than this bound.

Based on the above results, in the subsequent sections we will prove in detail that the state minimization problem of all the above types of automata is decidable. Although the details are different for addressing different types of automata, they share the same essential idea. The main idea can be briefly depicted as follows.

1. Firstly, for a given automaton $\mathcal{A}$ of some type (say probabilistic, quantum, etc.) with n states, we define the set

$$\mathbb{S}_{\mathcal{A}}^{(n')} = \{\mathcal{A}' : \mathcal{A}' \text{ has } n' \text{ states, is of the same type of } \mathcal{A}, \text{ and is equivalent to } \mathcal{A}\}.$$

2. Next, we show that $\mathbb{S}_{\mathcal{A}}^{(n')}$ can be described as the solution of a system of polynomial equations and/or inequations. Then, by Theorem 1 there exists an algorithm to decide whether $\mathbb{S}_{\mathcal{A}}^{(n')}$ is nonempty or not, and furthermore, if it is nonempty, we can find a sample of it.

3. Now, the minimization algorithm can be depicted in Figure 1.

Input: an automaton $\mathcal{A}$ with n states
Output: a minimal automaton $\mathcal{A}^{'}$, of the same type of $\mathcal{A}$, and equivalent to $\mathcal{A}$
Step 1:

For $i = 1$ to $n - 1$
If ($\mathbb{S}_{\mathcal{A}}^{(i)}$ is not empty) Return $\mathcal{A}' =$ sample $\mathbb{S}_{\mathcal{A}}^{(i)}$

Step 2:

Return $\mathcal{A}' = \mathcal{A}$

Figure 1. The minimization algorithm

For each type of automaton we need to prove that $\mathbb{S}_{\mathcal{A}}^{(n')}$ can be described as the solution of a system of polynomial equations and/or inequations whose variables are the entries of the initial state, transition matrices, and final states of an automaton with n' states. Although there are significant differences when defining the systems of (in)equations for each type of automata, we stress that the definition of such systems shares the following characteristics:

a) The properties of the automata, such as "the initial vector is a probability distribution", "matrices are stochastic matrices", can be expressed as a system of polynomial equations/inequalities whose variables are the entries of the initial state, transition matrices, and final states.

b) The acceptance probability of a fixed automaton for a fixed input can be presented as a polynomial, whose variables are the entries of the initial state, transition matrices, and final states.

c) For each type of automaton to be handled, there is a bound on the word length such that two automata are equivalent if and only if they have the same accepting probability for all input words with length less than the known bound. In this way, the equivalence between two automata can be represented by a finite set of polynomial equations.

In the subsequent sections, we will adopt the above method to address the minimization problem of several types of automata, namely, probabilistic automata, MO-1QFA , MM-1QFA, and one-way QFA with mixed states.

3. Minimization of probabilistic automata

Recall that a probabilistic automaton is a tuple $\mathcal{A} = (S, \Sigma, \mu_0, \{M_\sigma\}_{\sigma\in\Sigma}, F)$ where:

- S is a finite set of *states*;
- Σ is the input *alphabet*;

- $\mu_0 : S \to \mathbb{R}$ is called the *initial probability distribution* and is a stochastic vector over S, that is, $\sum_{s \in S} \mu_0(s) = 1$ and $\mu_0(s) \geq 0$ for all $s \in S$;
- M_σ is the *transition induced by input symbol* σ and is a square stochastic matrix of dimension $|S|$;
- $F \subseteq S$ is called the *set of accepting states*.

The above machine $\mathcal{A}$ accepts an input string $\sigma_1\sigma_2\cdots\sigma_k$ with probability

$$P_{\mathcal{A}}(\sigma_1\sigma_2\cdots\sigma_k) = \sum_{s_i \in F} (\mu_0 M_{\sigma_1} M_{\sigma_2} \cdots M_{\sigma_k})_{s_i}. \tag{10}$$

Two probabilistic automata $\mathcal{A}_1$ and $\mathcal{A}_2$ over the same alphabet Σ are said to be equivalent (resp. k-equivalent) if $P_{\mathcal{A}_1}(w) = P_{\mathcal{A}_2}(w)$ for any $w \in \Sigma^*$ (resp. for any input string w with $|w| \leq k$). The following well-known result will be useful later on.

Theorem 3 ([18, 23, 13])**.** Two probabilistic automata $\mathcal{A}_1$ and $\mathcal{A}_2$ with n_1 and n_2 states, respectively, are equivalent if and only if they are $(n_1 + n_2 - 1)$-equivalent. Furthermore, there exists a polynomial-time algorithm running in time $O((n_1 + n_2)^4)$ that takes as input two probabilistic automata $\mathcal{A}_1$ and $\mathcal{A}_2$ and determines whether $\mathcal{A}_1$ and $\mathcal{A}_2$ are equivalent.

Now we are in the position to deal with the problem of state minimization of probabilistic automata. The following result is based on Theorems 1 and 3.

Theorem 4. *The state minimization problem of probabilistic automata is in EXPSPACE.*

Proof. Given a probabilistic automaton $\mathcal{A} = (S, \Sigma, \mu_0, \{M_\sigma\}_{\sigma\in\Sigma}, F)$, the goal is to find another probabilistic automaton $\mathcal{A}' = (S', \Sigma, \mu_0', \{M_\sigma'\}_{\sigma\in\Sigma}, F')$ that is equivalent to $\mathcal{A}$ and has the smallest number of states from all probabilistic automata equivalent to $\mathcal{A}$. Now, following the idea presented in Subsection 2.2. we present the proof as follows.

For the given probabilistic automaton $\mathcal{A}$ with $|S| = n$, define the set

$$\mathbb{S}_{\mathcal{A}}^{(n')} = \{\mathcal{A}' : \mathcal{A}' \text{ is a probabilistic automaton equivalent to } \mathcal{A} \text{ with } n' \text{ states}\}. \tag{11}$$

The minimization algorithm is now depicted in Figure 2 (which is a straightforward adaptation of that presented in Figure 1).

Input: a probabilistic automaton $\mathcal{A}$ with n states
Output: a minimal probabilistic automaton $\mathcal{A}^{'}$ equivalent to $\mathcal{A}$
Step 1:

For $n' = 1$ to $n - 1$
If ($\mathbb{S}_{\mathcal{A}}^{(n')}$ is not empty) Return $\mathcal{A}' =$ sample $\mathbb{S}_{\mathcal{A}}^{(n')}$

Step 2:

Return $\mathcal{A}' = \mathcal{A}$

Figure 2. Algorithm for the minimization of probabilistic automata.

To analyse the algorithm, we will use Theorem 1 and show that both problems: to decide whether $\mathbb{S}_{\mathcal{A}}^{(n')}$ is nonempty, and to find a sample of $\mathbb{S}_{\mathcal{A}}^{(n')}$, are decidable/computable. By Theorem 1, it is sufficient to show that $\mathbb{S}_{\mathcal{A}}^{(n')}$ is the solution of a system of polynomial equations and/or inequations.

Let $\mathcal{A}' = (S', \Sigma, \mu_0', \{M_\sigma'\}_{\sigma\in\Sigma}, F')$. Suppose $\mu_0' = (x_1, x_2, \cdots, x_{n'})$. Since μ_0' is a probability distribution, it satisfies

$$\sum_{i=1}^{n'} x_i = 1 \text{ and } x_i \geq 0 \text{ for } i = 1, 2, \cdots, n'. \tag{12}$$

Therefore, "μ_0' is a probability distribution over n' states" can be represented by n' variables with $n' + 1$ polynomial equations/inequations.

For any $\sigma \in \Sigma$, M_σ' is an $n' \times n'$ stochastic matrix. Suppose that $M_\sigma' = [m_{ij}(\sigma)]$. We have

$$\sum_{j=1}^{n'} m_{ij}(\sigma) = 1 \text{ and } m_{ij}(\sigma) \geq 0 \text{ for } i, j = 1, 2, \cdots, n'. \tag{13}$$

Thus "M_σ' is an $n' \times n'$ stochastic matrix" can be represented by ${n'}^2$ variables with ${n'}^2 + n'$ polynomial equations/inequations. Note that to present $\mathcal{A}'$ we should describe M_σ' for every $\sigma \in \Sigma$. Thus, the number of M_σ''s is $|\Sigma|$.

The accepting state set F' can be characterized by an n'-dimensional column vector $\eta' = (\eta_1, \eta_2, \ldots, \eta_{n'})^\top$ with entries 0 or 1, where $\eta_i = 1$ means that the state s_i is an accepting state, and $\eta_i = 0$ means that the state s_i is not an accepting state. Thus, the accepting set F' can be represented by n' variables with polynomial equations as

$$\eta_i = 1 \text{ or } \eta_i = 0 \text{ for } i = 1, 2, \cdots, n',$$

or, equivalently, as the following n' polynomial equations

$$\eta_i(\eta_i - 1) = 0 \text{ for } i = 1, 2, \cdots, n'. \tag{14}$$

Since $\mathcal{A}'$ is equivalent to $\mathcal{A}$, by Theorem 3 the following equation

$$P_{\mathcal{A}'}(x) = P_{\mathcal{A}}(x) \tag{15}$$

holds for each $x \in \Sigma^*$ with $|x| \leq (n + n' - 1)$. Equivalently, the accepting probability of $\mathcal{A}'$ can be represented as

$$P_{\mathcal{A}'}(x) = \mu_0 M_{x_1} M_{x_2} \cdots M_{x_{|x|}} \eta. \tag{16}$$

It is clear that the expression in (16) is a polynomial of degree $2 + |x|$ whose variables are the entries of μ_0', M_σ', and F'.

Thus, for each $x \in \Sigma^*$ with $|x| \leq (n+n'-1)$, Eq. (15) is a polynomial equation, since the left side is a polynomial whose variables are the entries of μ_0', M_σ', and F', and the right side is a fixed value for the given $\mathcal{A}$. Note that to describe the fact that $\mathcal{A}'$ and $\mathcal{A}$ are equivalent, the total number of polynomial equations like Eq. (15) needed is

$$P = 1 + |\Sigma|^1 + |\Sigma|^2 + \cdots + |\Sigma|^{n+n'-1}. \tag{17}$$

The above statements and analysis can now be summarized as follows: for a given probabilistic automaton $\mathcal{A}$ over an input alphabet Σ, any probabilistic automaton $\mathcal{A}' \in \mathbb{S}_{\mathcal{A}}^{(n')}$ that is equivalent to $\mathcal{A}$ can be represented by a vector $x \in \mathbb{R}^{|\Sigma|n'^2+2n'}$, satisfying the polynomial equations/inequations Eqs. (12), (13), (14), (15). The total number of polynomial equations/inequations needed is

$$M = (n' + 1) + |\Sigma|(n'^2 + n') + n' + P. \tag{18}$$

The highest degree in these polynomials is

$$d = 2 + (n + n' - 1). \tag{19}$$

Thus, by Theorem 1, for every $n' \leq n$ there exists an algorithm to decide if $\mathbb{S}_{\mathcal{A}}^{(n')}$ is nonempty and the time cost is

$$(Md)^{O(|\Sigma|n'^2+2n')} = \left(n^3|\Sigma| + n|\Sigma|^n\right)^{O\left(|\Sigma|n^2\right)}. \tag{20}$$

If we assume that $|\Sigma|$ is a constant c, then the time complexity is $2^{O(n^3)}$. Furthermore, if $\mathbb{S}_{\mathcal{A}}^{(n')}$ is nonempty, there exists an algorithm to find a sample of $\mathbb{S}_{\mathcal{A},}^{(n')}$ in space

$$\tau d^{O(|\Sigma|n'^2+2n')} = \tau n^{O\left(|\Sigma|n^2\right)}. \tag{21}$$

If we look $|\Sigma|$ as a constant, then the space complexity is $\tau 2^{O(n^3)}$.

Therefore, the procedures described in Figure 2 can be used to find a minimal probabilistic automaton equivalent to a given probabilistic automaton. □

4. Minimization of measure-once quantum automata

An MO-1QFA is defined as a quintuple $\mathcal{Q} = (Q, \Sigma, |\psi_0\rangle, \{U(\sigma)\}_{\sigma\in\Sigma}, Q_{acc})$, where Q is a set of finite states, $|\psi_0\rangle$ is the initial state that is a superposition of the states in Q, Σ is a finite input alphabet, $U(\sigma)$ is a unitary matrix for each $\sigma \in \Sigma$, and $Q_{acc} \subseteq Q$ is the set of accepting states.

As usual, we identify Q with an orthonormal base of a complex Euclidean space and every state $q \in Q$ is identified with a basis vector, denoted by Dirac symbol $|q\rangle$ (a column vector), and $\langle q|$ is the conjugate transpose of $|q\rangle$. We describe the computing process for any given input string $x = \sigma_1\sigma_2\cdots\sigma_m \in \Sigma^*$. At the beginning the machine $\mathcal{Q}$ is in the initial state $|\psi_0\rangle$, and upon reading σ_1, the transformation $U(\sigma_1)$ acts on $|\psi_0\rangle$. After that, $U(\sigma_1)|\psi_0\rangle$ becomes the current state and the machine reads σ_2. The process continues until the machine has read σ_m ending in the state $|\psi_x\rangle = U(\sigma_m)U(\sigma_{m-1})\cdots U(\sigma_1)|\psi_0\rangle$. Finally, a measurement is performed on $|\psi_x\rangle$ and the accepting probability $P_{\mathcal{A}}(x)$ is equal to

$$P_{\mathcal{A}}(x) = \langle\psi_x|P_a|\psi_x\rangle = \|P_a|\psi_x\rangle\|^2 \tag{22}$$

where $P_a = \sum_{q\in Q_{acc}} |q\rangle\langle q|$ is the projection onto the subspace spanned by $\{|q\rangle : q \in Q_{acc}\}$.

For the equivalence problem of MO-1QFA the following result holds:

Theorem 5 ([15]). *Two MO-1QFA $\mathcal{A}_1$ and $\mathcal{A}_2$ with n_1 and n_2 states, respectively, are equivalent if and only if they are $(n_1^2 + n_2^2 - 1)$-equivalent.*

Based on Theorems 1 and 5, we obtain the following result.

Theorem 6. *The state minimization problem of MO-1QFA is in EXPSPACE.*

Proof. Given an MO-1QFA $\mathcal{A} = (Q, \Sigma, |\psi_0\rangle, \{U(\sigma)\}_{\sigma\in\Sigma}, Q_{acc})$, the goal is to find another MO-1QFA $\mathcal{A}' = (Q', \Sigma, |\psi_0'\rangle, \{U'(\sigma)\}_{\sigma\in\Sigma}, Q'_{acc})$ that is equivalent to $\mathcal{A}$ and has the smallest number of states from all MO-1QFA equivalent to $\mathcal{A}$.

For the given MO-1QFA $\mathcal{A}$ with $|Q| = n$, define the set

$$\mathbb{S}_{\mathcal{A}}^{(n')} = \{\mathcal{A}' : \mathcal{A}' \text{ is an MO-1QFA equivalent to } \mathcal{A} \text{ with } n' \text{ states}\}. \tag{23}$$

The minimization algorithm is as depicted in Figure 1, except that the input and output are MO-1QFA. Now, the key step is to prove that $\mathbb{S}_{\mathcal{A}}^{(n')}$ can be represented by a set of polynomial equations and/or inequations.

Let $\mathcal{A}' = (Q', \Sigma, \{U'(\sigma)\}_{\sigma\in\Sigma}, |\psi_0'\rangle, Q'_{acc})$. Note that $|\psi_0'\rangle = [x_1, x_2, \cdots, x_{n'}]^\top$ is a unit vector in $\mathbb{C}^{n'}$. Then

$$\sum_{i=1}^{n'} x_i x_i^* = 1. \tag{24}$$

According to Remark 2, "$|\psi'_0\rangle$ is a unit vector in $\mathbb{C}^{n'}$" can be represented by two real polynomial equations with $2n'$ real variables.

For any $\sigma \in \Sigma$, $U'(\sigma)$ is an $n' \times n'$ unitary matrix. Suppose that $U'(\sigma) = [u_{ij}(\sigma)]$, and therefore

$$[u_{ij}(\sigma)] \times [u_{ij}(\sigma)]^\dagger = I. \tag{25}$$

Thus, "$U'(\sigma)$ is an $n' \times n'$ unitary matrix" can be represented by $2n'^2$ real polynomial equations with $2n'^2$ real variables. Note that to present $\mathcal{A}'$ we should describe $U'(\sigma)$ for every $\sigma \in \Sigma$. Thus, the number of $U'(\sigma)$'s is $|\Sigma|$.

The accepting state set Q'_{acc} can be characterized by an n'-dimensional vector $|\eta_{acc}\rangle = (\eta_1, \eta_2, \ldots, \eta_{n'})$ with entries 0 or 1, where $\eta_i = 1$ means that the state q_i is an accepting state, and $\eta_i = 0$ means that the state q_i is not an accepting state. Thus, the accepting set Q'_{acc} can be described by n' real variables with polynomial equations such that

$$\eta_i = 1 \text{ or } \eta_i = 0.$$

or, equivalently, as the following n' polynomial equations

$$\eta'_i(\eta'_i - 1) = 0 \text{ for } i = 1, 2, \cdots, n'. \tag{26}$$

Since $\mathcal{A}'$ is equivalent to $\mathcal{A}$, by Theorem 5 the following equation holds

$$P_{\mathcal{A}'}(x) = P_{\mathcal{A}}(x), \tag{27}$$

for each $x \in \Sigma^*$, with $|x| \leq (n^2 + n'^2 - 1)$. Equivalently, the accepting probability of $\mathcal{A}'$ on an input x can be represented as

$$\begin{aligned} P_{\mathcal{A}'}(x) &= \|P'_a U'(x)|\psi'_0\rangle\|^2 \\ &= \sum_{i=1}^{n'} |\langle\eta_i|U(x)|\psi_0\rangle|^2 \\ &= \sum_{i=1}^{n'} \langle\eta_i| \otimes \langle\eta_i|^* U(x) \otimes U(x)^* |\psi_0\rangle \otimes |\psi_0\rangle^*, \end{aligned} \tag{28}$$

where $U(x) = U(x_{|x|}) \cdots U(x_2)U(x_1)$ and $\langle\eta_i|$ is an n'-dimensional row vector with the i'th entry being the value of the i'th entry of $|\eta_{acc}\rangle$, and others being 0's.

It is clear that the probability given by Eq. (28) has always a real value and, furthermore, it can be computed by a real polynomial whose variables are the entries of $|\psi'_0\rangle$, $U'(\sigma)$ and Q'_{acc}. The degree of this polynomial is $2|x| + 4$.

Thus, for each $x \in \Sigma^*$ with $|x| \leq (n^2 + n'^2 - 1)$, the probabilities given by Eq. (27) can be represented by a real polynomial equation, since the left side, as we have shown, is a real polynomial, and the right side is a fixed value for the given MO-1QFA $\mathcal{A}$ (and can be

computed in polynomial time). Note that, by Theorem 5, to describe the fact that $\mathcal{A}'$ and $\mathcal{A}$ are equivalent, the total number of polynomial equations like Eq. (27) needed is

$$P = 1 + |\Sigma|^1 + |\Sigma|^2 + \cdots + |\Sigma|^{n^2+n'^2-1}. \tag{29}$$

The above statements and analysis can now be summarized as follows: for a given MO-1QFA $\mathcal{A}$ over an input alphabet Σ, any MO-1QFA $\mathcal{A}' \in \mathbb{S}_{\mathcal{A}}^{(n')}$ that is equivalent to $\mathcal{A}$ can be represented by a vector $x \in \mathbb{R}^{2|\Sigma|n'^2+3n'}$ satisfying the polynomial Eqs. (24), (25), (26), and (27). The total number of polynomial equations needed is

$$M = 2 + 2|\Sigma|n'^2 + 2n' + P. \tag{30}$$

The highest degree in these equations is

$$d = 2(n^2 + n'^2 - 1) + 4. \tag{31}$$

Thus, by Theorem 1, for every $n' \leq n$ there exists an algorithm to decide if $\mathbb{S}_{\mathcal{A},\Sigma}^{(n')}$ is nonempty with time cost

$$(Md)^{O(2|\Sigma|n'^2+3n')} = \left(n^4|\Sigma| + n^2|\Sigma|^{n^2}\right)^{O\left(|\Sigma|n^2\right)}. \tag{32}$$

If we assume $|\Sigma|$ to be constant the above time complexity becomes $2^{O(n^5)}$. Furthermore, if the set $\mathbb{S}_{\mathcal{A},\Sigma}^{(n')}$ is nonempty, then there exists an algorithm to find a sample of $\mathbb{S}_{\mathcal{A},\Sigma}^{(n')}$ in space

$$\tau d^{O(2|\Sigma|n'^2+3n')} = \tau\left(n^2|\Sigma|\right)^{O\left(|\Sigma|n^2\right)}. \tag{33}$$

Finally, if we consider $|\Sigma|$ to be constant the above space complexity becomes $\tau 2^{O(n^3)}$.

Therefore, the procedures described in Figure 1 can be used to find a minimal MO-1QFA equivalent to a given MO-1QFA. □

5. Minimization of measure-many quantum automata

The definition of MM-1QFA is similar to that of MO-1QFA, but an essential difference between them is that in an MO-1QFA only one measurement is allowed at the end of the input string, while in an MM-1QFA, the measurement is allowed after each symbol has been read. As a consequence of this difference, minimizing MM-1QFA is more complicated than minimizing MO-1QFA, and therefore, we devote this section to address this problem. First, we give a rigourous definition of MM-1QFA.

Definition 7. *An MM-1QFA is a 6-tuple* $\mathcal{A} = (Q, \Sigma, \{U(\sigma)\}_{\sigma\in\{\$\}\cup\Sigma}, |\psi_0\rangle, Q_{acc}, Q_{rej})$, *where*

- $Q = \{q_1, \ldots, q_n\}$ *is the basic state set; at any time, the state of* $\mathcal{M}$ *is a superposition of these basic states;*

- Σ *is a finite input alphabet, equipped with an end-marker symbol* $\$ \notin \Sigma$ *(denote* $\Gamma = \Sigma \cup \{\$\}$*);*
- $|\psi_0\rangle$ *with* $|||\psi_0\rangle|| = 1$ *is an* n*-dimensional vector, denoting the initial vector;*
- *for any* $\sigma \in \Gamma$*,* $U(\sigma)$ *is an* $n \times n$ *unitary matrix;*
- *the set* Q *is partitioned into three subsets:* Q_{acc} *is the set of accepting states,* Q_{rej} *is the set of rejecting states, and* Q_{non} *is the set of non-halting states.*

Remark 8. *Denote the state space of MM-1QFA* $\mathcal{A}$ *by* $\mathcal{H}_Q$*. Then the whole space* $\mathcal{H}_Q$ *is divided into three subspaces:* $E_{non} = \text{span}\{|q\rangle : q \in Q_{non}\}$*,* $E_{acc} = \text{span}\{|q\rangle : q \in Q_{acc}\}$*, and* $E_{rej} = \text{span}\{|q\rangle : q \in Q_{rej}\}$*. For these subspaces consider the corresponding projectors* P_{non}, P_{acc}*, and* P_{rej}*. Thus,* $M = \{P_{non}, P_{acc}, P_{rej}\}$ *is a projective measurement on* $\mathcal{H}_Q$*.*

The computing process of MM-1QFA is similar to that of MO-1QFA except that after each input symbol σ is read and the corresponding unitary operator $U(\sigma)$ is applied to the current quantum state of the system, the projection measurement M is applied to the state. If the measurement result is '*non*', then the computation continues; if the result is '*acc*', then $\mathcal{A}$ accepts, otherwise it rejects. After every measurement the state collapses into the subspace specified by the projector that has been applied.

Since there is a non-zero probability that the automaton $\mathcal{A}$ halts partway through the computation, it is useful to keep track of the cumulative accepting and rejecting probabilities. Thus, we can represent the current state of $\mathcal{A}$ as a triple $(|\psi\rangle, p_{acc}, p_{rej})$ where p_{acc}, p_{rej} are respectively the cumulative probabilities of accepting and rejecting. Then the initial state of $\mathcal{A}$ can be represented by $(|\psi_0\rangle, 0, 0)$, and the evolution of $\mathcal{A}$ upon reading a symbol σ is denoted by

$$(|\psi\rangle, p_{acc}, p_{rej}) \mapsto (P_{non}|\psi'\rangle, p_{acc} + \|P_{acc}|\psi'\rangle\|^2, p_{rej} + \|P_{rej}|\psi'\rangle\|^2), \tag{34}$$

where $|\psi'\rangle = U(\sigma)|\psi\rangle$. On an input string $\sigma_1\sigma_2\cdots\sigma_n\$$, the accepting probability of $\mathcal{A}$ is given by

$$P_{\mathcal{A}}(\sigma_1 \ldots \sigma_n) = \sum_{k=1}^{n+1} \|P_{acc}U(\sigma_k) \prod_{i=1}^{k-1} \big(P_{non}U(\sigma_i)\big)|\psi_0\rangle\|^2, \tag{35}$$

whith $\sigma_{n+1} = \$$, and $\prod_{i=1}^{n} A_i = A_n A_{n-1} \cdots A_1$, instead of $A_1 A_2 \cdots A_n$.

The equivalence problem for MM-1QFA has been solved in [13], and we summarize this result in the following theorem.

Theorem 9 ([13])**.** *Two MM-1QFA* $\mathcal{A}_1$ *and* $\mathcal{A}_2$ *with* n_1 *and* n_2 *states, respectively, are equivalent if and only if they are* $(3n_1^2 + 3n_2^2 - 1)$*-equivalent.*

Based on Theorem 1 and 9, we obtain the following result.

Theorem 10. *The state minimization problem of MM-1QFA is in EXPSPACE.*

Proof. Given an MM-1QFA $\mathcal{A} = (Q, \Sigma, \{U(\sigma)\}_{\sigma\in\{\$\}\cup\Sigma}, |\psi_0\rangle, Q_{acc}, Q_{rej})$, the goal is to find another MM-1QFA $\mathcal{A}' = (Q', \Sigma, \{U'(\sigma)\}_{\sigma\in\{\$\}\cup\Sigma}, |\psi_0'\rangle, Q'_{acc}, Q'_{rej})$ that is equivalent to $\mathcal{A}$ and has the smallest number of states from all MM-1QFA equivalent to $\mathcal{A}$.

For the given MM-1QFA $\mathcal{A}$ with $|Q| = n$, define the set

$$\mathbb{S}_{\mathcal{A}}^{(n')} = \{\mathcal{A}' : \mathcal{A}' \text{ is an MM-1QFA equivalent to } \mathcal{A} \text{ with } n' \text{ states}\}. \tag{36}$$

Again, the minimization algorithm is precisely that depicted in Figure 1, except that the input and output are MM-1QFA. To check the soundness and complexity of the algorithm it is sufficient, by Theorem 1, to show that $\mathbb{S}_{\mathcal{A}}^{(n')}$ can be characterized by a system of polynomial equations and/or inequations.

Let $\mathcal{A}' = (Q', \Sigma, \{U'(\sigma)\}_{\sigma\in\{\$\}\cup\Sigma}, |\psi_0'\rangle, Q'_{acc}, Q'_{rej})$. By a similar analysis performed for MO-1QFA, we know that: (i) $|\psi_0\rangle'$ can be represented by $2n'$ real variables with 2 real polynomial equations, (ii) each $U'(\sigma)$ can be represented by $2n'^2$ real variables with $2n'^2$ real polynomial equations, and (iii) the accepting state set Q'_{acc} can be characterized by an n'-dimensional vector $|\eta_{acc}\rangle = (\eta_1, \eta_2, \ldots, \eta_{n'})$ with $2n'$ polynomial equations.

Similarly, the non-halting set Q'_{non} can also be characterized by an n'-dimensional vector $|\tau_{non}\rangle = (\tau_1, \tau_2, \ldots, \tau_{n'})$ with entries 0 or 1, where $\tau_i = 1$ means that the state q_i is a non-halting state, and $\tau_i = 0$ means that the state q_i is a halting state. Thus, the non-halting set Q'_{non} can be represented by n' real variables with polynomial equations such that

$$\tau_i(\tau_i - 1) = 0. \tag{37}$$

Since $\mathcal{A}'$ is equivalent to $\mathcal{A}$, by Theorem 9 the following equation holds

$$P_{\mathcal{A}'}(x) = P_{\mathcal{A}}(x) \tag{38}$$

for each $x \in \Sigma^*$, with $|x| \leq (3n^2 + 3n'^2 - 1)$.

As known, the accepting probability of $\mathcal{A}'$ on an input x is given by

$$P_{\mathcal{A}'}(x) = \sum_{k=1}^{|x|+1} \|P'_{acc}|U'(x_k) \prod_{i=1}^{k-1} \left(P'_{non}U'(x_i)\right)|\psi_0'\rangle\|^2 \tag{39}$$

with $\sigma_{|x|+1} = \$$ and $\prod_{i=1}^{n} A_i = A_n A_{n-1} \cdots A_1$. Furthermore, this probability can be equivalently represented as

$$\begin{aligned} P_{\mathcal{A}'}(x) &= \sum_{k=1}^{|x|+1} \sum_{j=1}^{n'} \left| \langle\eta_j|U'(x_k) \prod_{i=1}^{k-1} A(x_i)|\psi_0'\rangle \right|^2 \\ &= \sum_{j=1}^{n'} \langle\eta_j| \otimes \langle\eta_j|^* \sum_{k=1}^{|x|+1} \left([U'(x_k) \prod_{i=1}^{k-1} A(x_i)] \otimes [(U'(x_k)^* \prod_{i=1}^{k-1} A^*(x_i)] \right) |\psi_0'\rangle \otimes |\psi_0'\rangle^* \end{aligned} \tag{40}$$

where:

- $\langle \eta_i| = |\eta_i\rangle^\dagger$, and $|\eta_i\rangle$ is an n'-dimensional column vector, with the i'th element being the value of the i'th element of $|\eta_{acc}\rangle = (\eta_1, \eta_2, \ldots, \eta_{n'})$, and others being 0.
- $A(x_i) = P'_{non}U'(x_i) = \mathrm{diag}[\tau_1, \tau_2, \cdots, \tau'_n]U'(x_i)$, where τ_i's are chosen from the characteristic vector $|\tau_{non}\rangle = (\tau_1, \tau_2, \ldots, \tau_{n'})$ defined above.

Now, it can be seen the probability given by Eq. (40) has always a real value and, furthermore, can be represented by a real polynomial whose variables are the entries of $|\psi'_0\rangle$, $U'(\sigma)$, Q'_{acc}, and Q'_{non}. Also note that the polynomial in Eq. (40) can be viewed as a sum of $|x|+1$ polynomials, each given by different $k = 1 \ldots |x|+1$, and that the highest degree of these polynomials is $4|x|+6$ when $k = |x|+1$.

Thus, for each $x \in \Sigma^*$ with $|x| \leq (3n^2 + 3n'^2 - 1)$, Eq. (38) can be represented by a real polynomial equation, since the left side, as we have shown, is a real polynomial, and the right side is a fixed value for the given MM-1QFA $\mathcal{A}$. Note that to describe the fact that $\mathcal{A}'$ and $\mathcal{A}$ are equivalent, the total number of polynomial equations like Eq. (38) needed is

$$P = |\Sigma|^1 + |\Sigma|^2 + \cdots + |\Sigma|^{3n^2+3n'^2-1}. \tag{41}$$

The above statements and analysis can now be summarized as follows: for a given MM-1QFA $\mathcal{A}$ over an input alphabet Σ, any MM-1QFA $\mathcal{A}' \in \mathbb{S}_{\mathcal{A}}^{(n')}$ that is equivalent to $\mathcal{A}$ can be represented by a vector $x \in \mathbb{R}^{2(|\Sigma|+1)n'^2+4n'}$ satisfying polynomial equations Eqs. (24), (25), (26), (37), and (38). The total number of polynomial equations needed is

$$M = 2 + 2(|\Sigma|+1)n'^2 + 2n' + 2n' + P. \tag{42}$$

The highest degree in these equations is

$$d = 4(3n^2 + 3n'^2 - 1) + 6. \tag{43}$$

Thus, by the discussion above and by Theorem 1, for every $n' \leq n$ there exists an algorithm to decide if $\mathbb{S}_{\mathcal{A}}^{(n')}$ is nonempty with time cost

$$(Md)^{O(2(|\Sigma|+1)n'^2+4n')} = \left(n^4|\Sigma| + n^2|\Sigma|^{n^2}\right)^{O\left(|\Sigma|n^2\right)}. \tag{44}$$

If we assume that $|\Sigma|$ is constant, then the time complexity becomes $O(2^{n^5})$. Furthermore, if the set $\mathbb{S}_{\mathcal{A},\Sigma}^{(n')}$ is nonempty, then there exists an algorithm to find a sample of $\mathbb{S}_{\mathcal{A}}^{(n')}$ in space

$$\tau d^{O(2(|\Sigma|+1)n'^2+4n')} = \tau\left(n^2|\Sigma|\right)^{O\left(|\Sigma|n^2\right)}, \tag{45}$$

and, if we consider $|\Sigma|$ to be constant, the space complexity becomes $\tau 2^{O(n^3)}$.

Therefore, the procedures described in Figure 1 can be used to find a minimal MM-1QFA equivalent to a given MM-1QFA. □

6. Minimization of generalized quantum automata

In this section, our purpose is to prove that the state minimization problem for QFA with mixed states and general operations is solvable as well. The model to be studied is named *one-way general quantum finite automata* (1gQFA), and there are two types: measure-once 1gQFA (MO-1gQFA) and measure-many 1gQFA (MM-1gQFA) [15].

6.1. Definition of 1gQFA and relevant results

According to the times of measurement performed in the computation, there are two kinds of 1gQFA: measure-once 1gQFA (MO-1gQFA) and measure-many 1gQFA (MM-1gQFA) [15]. The definition of MO-1gQFA is as follows.

Definition 11. *An MO-1gQFA $\mathcal{M}$ is a five-tuple $\mathcal{M} = \{Q, \Sigma, \rho_0, \{\mathcal{E}_\sigma\}_{\sigma\in\Sigma}, Q_{acc}\}$, where*

- *$Q = \{q_1, q_2, \cdots, q_n\}$ is a finite set of states, of which each state q_i can be presented by an n-dimensional vector $|q_i\rangle = (0, 0, \cdots, 1, \cdots, 0)^\top$ with the i'th entry being 1 and else 0's; $\mathcal{H} = span\{|q_1\rangle, |q_2\rangle, \cdots, |q_n\rangle\}$ is the state space of $\mathcal{M}$;*
- *Σ is a finite input alphabet;*
- *ρ_0, the initial state of $\mathcal{M}$, is a density operator on $\mathcal{H}$; generally, we assume that ρ_0 is a pure state, that is, $\rho_0 = |\psi_0\rangle\langle\psi_0|$, where $|\psi_0\rangle$ is a superposition of states from Q;*
- *$\mathcal{E}_\sigma$ corresponding to $\sigma \in \Sigma$ is a trace-preserving quantum operation acting on $\mathcal{H}$;*
- *$Q_{acc} \subseteq Q$ is the set of accepting states, and it is associated with a projector $P_{acc} = \sum_{q_i\in Q_{acc}} |q_i\rangle\langle q_i|$; denote $P_{rej} = I - P_{acc}$, then $\{P_{acc}, P_{rej}\}$ form a projective measurement on $\mathcal{H}$.*

On input word $\sigma_1\sigma_2\ldots\sigma_n \in \Sigma^*$, the above MO-1gQFA $\mathcal{M}$ proceeds as follows: the quantum operations $\mathcal{E}_{\sigma_1}, \mathcal{E}_{\sigma_2}, \ldots, \mathcal{E}_{\sigma_n}$ are performed on ρ_0 in succession, and then the projective measurement $\{P_{acc}, P_{rej}\}$ is performed on the final state, obtaining the accepting result with a certain probability. Thus, MO-1gQFA $\mathcal{M}$ defined above induces a function $P_{\mathcal{M}} : \Sigma^* \to [0, 1]$ as

$$P_{\mathcal{M}}(\sigma_1\sigma_2\ldots\sigma_n) = \mathrm{Tr}(P_{acc}\mathcal{E}_{\sigma_n} \circ \cdots \circ \mathcal{E}_{\sigma_2} \circ \mathcal{E}_{\sigma_1}(\rho_0)), \tag{46}$$

where $\mathcal{E}_2 \circ \mathcal{E}_1(\rho)$ stands for $\mathcal{E}_2(\mathcal{E}_1(\rho))$. In fact, for every $x \in \Sigma^*$, $P_{\mathcal{M}}(x)$ represents the probability that $\mathcal{M}$ accepts x.

The definition of MM-1gQFA is as follows.

Definition 12. *An MM-1gQFA $\mathcal{M}$ is a six-tuple $\mathcal{M} = \{Q, \Sigma, \rho_0, \{\mathcal{E}_\sigma\}_{\sigma\in\Sigma\cup\{¢,\$\}}, Q_{acc}, Q_{rej}\}$, where all the elements are almost the same as the ones of MO-1gQFA except that the input alphabet Σ is additionally equipped with two symbols: the left end-marker ¢ and the right end-marker \$, and the state set Q is divided into three parts: Q_{acc}, Q_{rej}, and Q_{non}, which respectively denote accepting state set, rejecting state set, and non-halting state set.*

For MM-1gQFA $\mathcal{M}$, the whole state space $\mathcal{H}$ should be divided into three subspaces, that is, $\mathcal{H} = \mathcal{H}_{acc} \oplus \mathcal{H}_{rej} \oplus \mathcal{H}_{non}$, where $\mathcal{H}_{acc}$, $\mathcal{H}_{rej}$, $\mathcal{H}_{non}$ are subspaces spanned by states from Q_{acc}, Q_{rej}, and Q_{non}, respectively. There is a measurement $\{P_{non}, P_{acc}, P_{rej}\}$, of which the elements in turn are the projectors onto the subspaces $\mathcal{H}_{non}, \mathcal{H}_{acc}$, and $\mathcal{H}_{rej}$, respectively.

The input string of MM-1gQFA $\mathcal{M}$ has the form ¢x\$ where $x \in \Sigma^*$, and ¢ and \$ are the left and right end-maker, respectively. The behavior of MM-1gQFA is similar to that of MM-1QFA. Reading each symbol σ in the input string, the machine has two actions: (i) first $\mathcal{E}_\sigma$ is performed, such that the current state ρ evolves into $\mathcal{E}_\sigma(\rho)$; (ii) the measurement $\{P_{non}, P_{acc}, P_{rej}\}$ is performed on the state $\mathcal{E}_\sigma(\rho)$. If the result "acc" (or "rej") is observed, the machine halts in an accepting (or rejecting) state with a certain probability. Otherwise, with probability $\mathrm{Tr}(P_{non}\mathcal{E}_\sigma(\rho))$, the machine continues to read the next symbol.

Define $\mathcal{V} = L(\mathcal{H}) \times \mathbb{R} \times \mathbb{R}$. The elements of $\mathcal{V}$ represent states of $\mathcal{M}$ as follows: a machine described by $(\rho, p_{acc}, p_{rej}) \in \mathcal{V}$ has accepted with probability p_{acc}, rejected with probability p_{rej}, and continued with probability $\mathrm{Tr}(\rho)$, in which case the current density operator is $\frac{1}{\mathrm{Tr}(\rho)}\rho$. The evolution of $\mathcal{M}$ upon reading symbol $\sigma \in \Sigma \cup \{¢, \$\}$, can be described by an operator $\mathcal{T}_\sigma$ on $\mathcal{V}$ as follows:

$$\mathcal{T}_\sigma : (\rho, p_{acc}, p_{rej}) \to (P_{non}\mathcal{E}(\rho)P_{non}, \mathrm{Tr}(P_{acc}\mathcal{E}(\rho)) + p_{acc}, \mathrm{Tr}(P_{rej}\mathcal{E}(\rho)) + p_{rej}). \tag{47}$$

We use $P_{\mathcal{M}}(x)$ to denote the probability that MM-1gQFA $\mathcal{M}$ accepts $x \in \Sigma^*$. Then $P_{\mathcal{M}}(x)$ accumulates all the accepting probabilities produced on reading each symbol in the input string ¢x\$. Concretely, $P_{\mathcal{M}}(x)$ can be represented as follows:

$$P_{\mathcal{A}}(x_1 \ldots x_n) = \sum_{k=1}^{n+2} \mathrm{Tr}\big(P_{acc}\mathcal{E}_{x_k} \circ \prod_{i=1}^{k-1} \tilde{\mathcal{E}}_{x_i}(\rho_0)\big), \tag{48}$$

where $x_1 = ¢$, $x_{n+2} = \$$, $\mathcal{E}_2 \circ \mathcal{E}_1(\rho)$ stands for $\mathcal{E}_2(\mathcal{E}_1(\rho))$, and

$$\prod_{i=1}^{n} \tilde{\mathcal{E}}_{x_i} = \tilde{\mathcal{E}}_{x_n} \circ \cdots \circ \tilde{\mathcal{E}}_{x_1}, \tag{49}$$

$$\tilde{\mathcal{E}}_{\sigma_i}(\rho) = P_{non}\mathcal{E}_{\sigma_i}(\rho)P_{non}. \tag{50}$$

In Ref. [15] it was shown that the equivalence problem of both MO-1gQFA and MM-1gQFA is decidable within polynomial time. This result plays a crucial role to solve the state minimization problem of 1gQFA. We recall the main results in the following.

Theorem 13 ([15], Theorem 9)**.** *Two MO-1gQFA $\mathcal{M}_1$ and $\mathcal{M}_2$ are equivalent if and only if they are $(n_1+n_2)^2$-equivalent, where n_1 and n_2 are the numbers of states of $\mathcal{M}_1$ and $\mathcal{M}_2$, respectively.*

Theorem 14 ([15], Theorem 18)**.** *Two MM-1gQFA $\mathcal{M}_1$ and $\mathcal{M}_2$ are equivalent if and only if they are $(n_1+n_2)^2$-equivalent, where n_1 and n_2 are numbers of states of $\mathcal{M}_1$ and $\mathcal{M}_2$, respectively.*

As mentioned in [15], the above equivalence criteria can be slightly improved to $n_1^2+n_2^2-1$ using results from [26], but this is not an essential improvement, and has no influence on the state minimization problem to be discussed later on.

6.2. State minimization of MO-1gQFA

The minimization process of MO-g1QFA is similar to that of MO-1QFA and MM-1QFA except for the details concerning trace-preserving quantum operations. In the interest of completeness, we prove the following theorem.

Theorem 15. *The state minimization problem of MO-1gQFA is in EXPSPACE.*

Proof. Given an MO-1gQFA $\mathcal{A}=\{Q,\Sigma,\rho_0,\{\mathcal{E}_\sigma\}_{\sigma\in\Sigma},Q_{acc}\}$, the goal is to find another MO-1gQFA $\mathcal{A}'=\{Q',\Sigma,\rho_0',\{\mathcal{E}_\sigma'\}_{\sigma\in\Sigma},Q_{acc}'\}$ such that $\mathcal{A}'$ is equivalent to $\mathcal{A}$ and has the smallest number of states from all MO-1gQFA equivalent to $\mathcal{A}$.

For the given MO-1gQFA $\mathcal{A}$ with $|Q|=n$, define the set

$$\mathbb{S}_{\mathcal{A}}^{(n')}=\{\mathcal{A}':\mathcal{A}' \text{ is an MO-1gQFA equivalent to } \mathcal{A} \text{ with } n' \text{ states}\}. \tag{51}$$

The minimization algorithm is as depicted in Figure 1, except that the input and output are MO-1gQFA. To verify the algorithm, by Theorem 1, it is sufficient to show that $\mathbb{S}_{\mathcal{A}}^{(n')}$ can be represented by some polynomial equations and/or inequations.

Let $\mathcal{A}'=\{Q',\Sigma,\rho_0',\{\mathcal{E}_\sigma'\}_{\sigma\in\Sigma},Q_{acc}'\}$. As mentioned in the definition of MO-1gQFA, it is generally assumed that ρ_0' is a pure state, and thus $\rho_0'=|\psi_0'\rangle\langle\psi_0'|$ for a normalized state $|\psi_0'\rangle$. Suppose that $|\psi_0'\rangle=(x_1,x_2,\cdots,x_{n'})^\top$. Then

$$\sum_{i=1}^{n'}x_ix_i^*=1. \tag{52}$$

Thus, according to Remark 2, we can use two real polynomial equations equipped with $2n'$ real variables to describe that $|\psi_0'\rangle$ is a unit vector in $\mathbb{C}^{n'}$. Namely, ρ_0' can be represented by two real polynomial equations equipped with $2n'$ real variables.

For any $\sigma\in\Sigma$, $\mathcal{E}_\sigma'$ is a trace-preserving quantum operation acting on $\mathbb{C}^{n'}$. It is well known that each trace-preserving quantum operation $\mathcal{E}$ has an operation-sum representation, and

the number of operator elements in this representation will not surpass the square of the dimension of the space that $\mathcal{E}$ acts on. Thus, for each $\sigma \in \Sigma$, we suppose that $\mathcal{E}'_\sigma$ has the following form:

$$\mathcal{E}'_\sigma(\rho) = \sum_{k=1}^{n'^2} E_k \rho E_k^\dagger,$$

where $E_k = [e_{ij}^k]$ are $n' \times n'$ matrices, and they must satisfy the following condition:

$$\sum_{k=1}^{n'^2} E_k E_k^\dagger = I. \tag{53}$$

Therefore, to describe that $\mathcal{E}'_\sigma$ is a trace-preserving quantum operation, we need n'^4 complex variables (i, j in e_{ij}^k count from 1 to n', and k counts from 1 to n'^2), and n'^2 complex polynomials equation derived from Eq. (53). By Remark 2, we thus need $2n'^2$ real polynomial equations, equipped with $2n'^4$ real variables, to characterize $\mathcal{E}'_\sigma$ for each $\sigma \in \Sigma$. Note that the number of $\mathcal{E}_\sigma$'s is $|\Sigma|$.

Similar to the case of MO-1QFA, the accepting state set Q'_{acc} can be characterized by an n'-dimensional vector $|\eta_{acc}\rangle = (\eta_1, \eta_2, \ldots, \eta_{n'})$ with polynomial equations

$$\eta_i(\eta_i - 1) = 0, \tag{54}$$

where $\eta_i = 1$ means q_i is an accepting state, and $\eta_i = 0$ means q_i is not an accepting state.

The next key step is to show that the assertion $\mathcal{A}'$ is equivalent to $\mathcal{A}$ can be reduced to solving a system of polynomial equations. First, since $\mathcal{A}'$ is equivalent to $\mathcal{A}$, by Theorem 13 it is required that

$$P_{\mathcal{A}'}(x) = P_{\mathcal{A}}(x) \tag{55}$$

holds for any $x \in \Sigma^*$ with $|x| \leq (n + n')^2$. As we know, the accepting probability of $\mathcal{A}'$ on an input x is

$$P_{\mathcal{A}'}(x_1 x_2 \ldots x_m) = \mathrm{Tr}(P'_{acc} \mathcal{E}'_{x_m} \circ \cdots \circ \mathcal{E}'_{x_2} \circ \mathcal{E}'_{x_1}(\rho'_0)).$$

This probability can be rewritten in another equivalent form by using the mapping *vec* introduced in Section 2. For $\sigma \in \Sigma$, suppose that $\mathcal{E}'_\sigma(\rho) = \sum_{k=1}^{n'^2} E_k^\sigma \rho E_k^{\sigma\dagger}$, and denote

$$A'_\sigma = \sum_k E_k^\sigma \otimes E_k^{\sigma *}.$$

Then, by Eq. (7), we have

$$\begin{aligned} vec(\mathcal{E}_{\sigma_1}(\rho)) &= A'_{\sigma_1} vec(\rho), \\ vec(\mathcal{E}_{\sigma_2} \circ \mathcal{E}_{\sigma_1}(\rho)) &= A'_{\sigma_2} A'_{\sigma_1} vec(\rho). \end{aligned}$$

As a result, the probability of $\mathcal{A}'$ accepting $x \in \Sigma^*$ can be rewritten in the following way:

$$\begin{aligned} P_{\mathcal{A}'}(x_1x_2\ldots x_m) &= vec(P'_{acc})^\top A'_{x_m}\cdots A'_{x_2}A'_{x_1}vec(\rho'_0) \\ &= \sum_{q_i\in Q_{acc}} \langle q_i|\otimes\langle q_i|A'_{x_m}\cdots A'_{x_2}A'_{x_1}|\psi'_0\rangle\otimes|\psi'_0\rangle^* \\ &= \sum_{i=1}^{n'} \eta_i\langle q_i|\otimes\langle q_i|A'_{x_m}\cdots A'_{x_2}A'_{x_1}|\psi'_0\rangle\otimes|\psi'_0\rangle^* \end{aligned} \tag{56}$$

where η_i is the i'th entry of $|\eta_{acc}\rangle = (\eta_1, \eta_2, \ldots, \eta_{n'})$, which is a characteristic vector of Q_{acc}.

Now, it can be seen that $P_{\mathcal{A}'}(x_1x_2\ldots x_m)$ from Eq. (56) has always a real value and, furthermore, it can be represented by a real polynomial whose variables are the entries of ρ'_0, $\mathcal{E}'_\sigma$, and Q'_{acc}. Also note that the degree of the polynomial is $2|x|+3$. Thus, for each $x \in \Sigma^*$ with $|x| \leq (n'+n)^2$, Eq. (55) corresponds to a real polynomial equation, since the left side, as we have shown, is a real polynomial, and the right side is a fixed value for the given MO-1gQFA $\mathcal{A}$. Note that to describe the fact that $\mathcal{A}'$ and $\mathcal{A}$ are equivalent, the total number of polynomial equations needed is

$$P = |\Sigma|^1 + |\Sigma|^2 + \cdots + |\Sigma|^{(n'+n)^2}. \tag{57}$$

The above statements and analysis can now be summarized as follows: for a given MO-1gQFA $\mathcal{A}$ over an input alphabet Σ, any MO-1gQFA $\mathcal{A}' \in \mathbb{S}^{(n')}_{\mathcal{A},\Sigma}$ that is equivalent to $\mathcal{A}$ can be represented by a vector $x \in \mathbb{R}^{2|\Sigma|n'^4+3n'}$ satisfying the real polynomial equations from Eqs. (52), (53), (54), (55). The total number of polynomial equations needed is

$$M = 2 + 2|\Sigma|n'^2 + 2n' + P. \tag{58}$$

The highest degree in these equations is

$$d = 2(n'+n)^2 + 3. \tag{59}$$

Thus, by Theorem 1, for every $n' \leq n$ there exists an algorithm to decide if $\mathbb{S}^{(n')}_{\mathcal{A}}$ is nonempty and, moreover, its time cost is

$$(Md)^{O(2|\Sigma|n'^4+3n')} = \left(n^4|\Sigma| + n^2|\Sigma|^{n^2}\right)^{O\left(|\Sigma|n^4\right)}. \tag{60}$$

If we assume Σ to be constant, then time complexity of the above algorithm becomes $2^{O(n^7)}$. Furthermore, if the set $\mathbb{S}^{(n')}_{\mathcal{A},\Sigma}$ is nonempty, then there exists an algorithm to find a sample of $\mathbb{S}^{(n')}_{\mathcal{A},\Sigma}$ in space

$$\tau d^{O(2|\Sigma|n'^4+3n')} = \tau\left(n^2\right)^{O\left(|\Sigma|n^4\right)}. \tag{61}$$

If we take Σ to be constant, then the space complexity of the above algorithm becomes $\tau 2^{O(n^5)}$. □

6.3. State minimization of MM-1gQFA

In the previous section, we have solved the minimization problem of MO-1gQFA. Similarly, we can address the minimization problem of MM-1gQFA. The method is similar to the one in the case of MO-1gQFA, but we need some additional technical treatment on MM-1gQFA, since the computing process of an MM-1gQFA is more complicated than that of MO-1gQFA.

Theorem 16. *The state minimization problem of MM-1gQFA is in EXPSPACE.*

Proof. Given an MM-1gQFA $\mathcal{A} = (Q, \Sigma, \rho_0, \{\mathcal{E}_\sigma\}_{\sigma\in\{\text{¢},\$\}\cup\Sigma}, Q_{acc}, Q_{rej})$, the goal is to find another MM-1gQFA QFA $\mathcal{A}' = (Q', \Sigma, \rho'_0, \{\mathcal{E}'_\sigma\}_{\sigma\in\{\text{¢},\$\}\cup\Sigma}, Q'_{acc}, Q'_{rej})$ that is equivalent to $\mathcal{A}$ and has the smallest number of states from all MM-1gQFA equivalent to $\mathcal{A}$.

For the given MM-1QFA $\mathcal{A}$ with $|Q| = n$ we define the set

$$\mathbb{S}_{\mathcal{A}}^{(n')} = \{\mathcal{A}' : \mathcal{A}' \text{ is an MM-1gQFA equivalent to } \mathcal{A} \text{ with } n' \text{ states}\}. \tag{62}$$

To show that the algorithm in Figure 1 works for this model, by Theorem 1 it is sufficient to show that $\mathbb{S}_{\mathcal{A}}^{(n')}$ can be represented as a solution of a system of polynomial equations and/or inequations.

Let $\mathcal{A}' = (Q', \Sigma, \rho'_0, \{\mathcal{E}'_\sigma\}_{\sigma\in\{\text{¢},\$\}\cup\Sigma}, Q'_{acc}, Q'_{rej})$. From a similar discussion on MO-1gQFA, we know that: (i) ρ'_0 can be represented by $2n'$ real variables with 2 real polynomial equations, (ii) each $\mathcal{E}_\sigma$ can be represented by $2n'^4$ real variables with $2n'^2$ real polynomial equations, and (iii) the accepting state set Q'_{acc} can be characterized by an n'-dimensional vector $|\eta_{acc}\rangle = (\eta_1, \eta_2, \ldots, \eta_{n'})$ with $2n'$ polynomial equations like $\eta_i(\eta_i - 1) = 0$.

Similarly, the non-halting set Q'_{non} can also be characterized by an n'-dimensional vector $|\tau_{non}\rangle = (\tau_1, \tau_2, \ldots, \tau_{n'})$ with polynomial equations such as

$$\tau_i = 1 \text{ or } \tau_i = 0, \tag{63}$$

or equivalently, by

$$\tau_i(\tau_i - 1) = 0, \tag{64}$$

where $\tau_i = 1$ means q_i is a non-halting state and $\tau_i = 0$ means q_i is a halting state.

The next key step is to show that the equivalence between $\mathcal{A}'$ and $\mathcal{A}$ can be restated as a solution of a system of polynomial equations. Firstly, since $\mathcal{A}'$ is equivalent to $\mathcal{A}$, by Theorem 14 it is required that

$$P_{\mathcal{A}'}(x) = P_{\mathcal{A}}(x) \tag{65}$$

holds for any $x \in \Sigma^*$ with $|x| \leq (n + n')^2$.

As we know, the accepting probability of $\mathcal{A}'$ on an input x is

$$P_{\mathcal{A}'}(x_1 \ldots x_m) = \sum_{k=1}^{m+2} \mathrm{Tr}\big(P'_{acc}\mathcal{E}'_{x_k} \circ \prod_{i=1}^{k-1} \tilde{\mathcal{E}}'_{x_i}(\rho_0)\big),$$

where $x_1 = ¢$, $x_{m+2} = \$$, $\mathcal{E}_2 \circ \mathcal{E}_1(\rho)$ stands for $\mathcal{E}_2(\mathcal{E}_1(\rho))$, and

$$\prod_{i=1}^{n} \tilde{\mathcal{E}'}_{x_i} = \tilde{\mathcal{E}'}_{x_n} \circ \cdots \circ \tilde{\mathcal{E}'}_{x_1},$$
$$\tilde{\mathcal{E}'}_{\sigma_i}(\rho) = P'_{non}\mathcal{E}'_{\sigma_i}(\rho)P'_{non}.$$

Using the mapping *vec* introduced in Section 2, we can rewrite the above probability in the following way:

$$\begin{aligned} P_{\mathcal{A}'}(x_1x_2\ldots x_m) &= \sum_{k=1}^{m+2} vec(P'_{acc})^\top vec\big(\mathcal{E}'_{x_k} \circ \prod_{i=1}^{k-1} \tilde{\mathcal{E}'}_{x_i}(\rho'_0)\big) \\ &= \sum_{k=1}^{m+2} vec(P'_{acc})^\top A'_{x_k} vec\big(\prod_{i=1}^{k-1} \tilde{\mathcal{E}'}_{x_i}(\rho'_0)\big) \\ &= \sum_{k=1}^{m+2} vec(P'_{acc})^\top A'_{x_k} \prod_{i=1}^{k-1} \tilde{A'}_{x_i} vec(\rho'_0) \\ &= \sum_{i=1}^{n'} \eta_i \langle q_i| \otimes \langle q_i| \sum_{k=1}^{m+2} A'_{x_k} \prod_{i=1}^{k-1} \tilde{A'}_{x_i} |\psi'\rangle \otimes |\psi'\rangle^* \end{aligned} \tag{66}$$

where:

- $A'_\sigma = \sum_k E_k^\sigma \otimes E_k^{\sigma *}$, associated with quantum operation $\mathcal{E}'_\sigma$ defined as $\mathcal{E}'_\sigma(\rho) = \sum_{k=1}^{n'^2} E_k^\sigma \rho E_k^{\sigma\dagger}$.
- $\tilde{A'_\sigma} = (P'_{non} \otimes P'^\top_{non})A'_\sigma$, and $P'_{non} = \mathrm{diag}[\tau_1, \tau_2, \cdots, \tau_{n'}]$ where τ_i's are chosen from $|\tau_{non}\rangle = (\tau_1, \tau_2, \ldots, \tau_{n'})$, the characteristic vector of Q'_{non}.
- η_i's are chosen from $|\eta_{acc}\rangle = (\eta_1, \eta_2, \ldots, \eta_{n'})$, the characteristic vector of Q_{acc}.

Now, it can be seen that $P_{\mathcal{A}'}(x_1x_2\ldots x_m)$ from Eq. (66) has always a real value and, furthermore, can be represented by a real polynomial whose variables are the entries of ρ'_0, $\mathcal{E}'_\sigma$, Q'_{acc}, and Q'_{non}. Also note that the polynomial in Eq. (66) can be viewed as the sum of $|x|+2$ polynomials by taking value from $k = 1$ to $k = |x|+2$, and the degree of polynomials attains the highest value $4|x| + 9$ when $k = |x| + 2$.

Thus, for each $x \in \Sigma^*$ with $|x| \leq (n + n')^2$, Eq. (65) can be represented by a real polynomial equation, since the left side, as we have shown, is a real polynomial, and the right side is a fixed value for the given MM-1gQFA $\mathcal{A}$. Note that to describe the fact that $\mathcal{A}'$ and $\mathcal{A}$ are equivalent, The total number of polynomial equations needed is

$$P = |\Sigma|^1 + |\Sigma|^2 + \cdots + |\Sigma|^{(n'+n)^2}. \tag{67}$$

The above statements and analysis can now be summarized as follows: for a given MM-1gQFA $\mathcal{A}$ over an input alphabet Σ, any MM-1gQFA $\mathcal{A}' \in \mathbb{S}_{\mathcal{A}}^{(n')}$ that is equivalent to $\mathcal{A}$ can

be represented by a vector $x \in \mathbb{R}^{2(|\Sigma|+2)n'^4+4n'}$, satisfying the real polynomial equations (52), (53), (54), (64), (65). The total number of polynomial equations needed is

$$M = 2 + 2(|\Sigma| + 2)n'^2 + 4n' + P. \tag{68}$$

The highest degree in these equations is

$$d = 4(n + n')^2 + 9. \tag{69}$$

Thus, by Theorem 1, for every $n' \leq n$ there exists an algorithm to decide if $\mathbb{S}_{\mathcal{A}}^{(n')}$ is nonempty and the time cost is

$$(Md)^{O(2(|\Sigma|+2)n'^4+4n')} = \left(n^4|\Sigma| + n^2|\Sigma|^{n^2}\right)^{O\left(|\Sigma|n^4\right)}. \tag{70}$$

If we consider m a constant c, then the time complexity becomes $2^{O(n^7)}$. Furthermore, if the set $\mathbb{S}_{\mathcal{A}}^{(n')}$ is nonempty, then there exists an algorithm to find a sample of $\mathbb{S}_{\mathcal{A}}^{(n')}$ in space

$$\tau d^{O(2(|\Sigma|+2)n'^4+4n')} = \tau\left(n^2\right)^{O\left(|\Sigma|n^4\right)}. \tag{71}$$

If we consider m a constant, then the space complexity becomes $\tau 2^{O(n^5)}$.

Therefore, the procedures described in Figure 1 can be used to find a minimal MM-1gQFA equivalent to a given MM-1gQFA. □

7. Covering minimization of stochastic sequential machines

Stochastic sequential machines (SSM) are an important and historical model for probabilistic computation [18]. However, there are still some basic problems regarding state minimization to be solved. Indeed, Paz proposed two open problems (see [18], Page 43, *Open Problems*) of whether the reduction of the number of states for any given SSM is decidable, and how to construct a finite algorithm for finding a reduced SSM. In order to present these two problems clearly, we begin with recalling some definitions and notations related.

Definition 17 ([18])**.** A *stochastic sequential machine* (SSM) is a quadruple

$$\mathcal{M} = (S, I, O, \{A(y|x)\})$$

where S, I and O are finite sets (the internal states, inputs, and outputs, respectively), and $\{A(y|x)\}$ is a finite set containing $|I| \times |O|$ square matrices of order $|S|$ such that $a_{ij}(y|x) \geq 0$ for all i and j, and

$$\sum_{y \in O} \sum_{j=1}^{|S|} a_{ij}(y|x) = 1 \tag{72}$$

where $A(y|x) = [a_{ij}(y|x)]$, and $|I|$, $|O|$, and $|S|$ mean the cardinality of sets I, O, and S, respectively.

Let $\mathcal{M} = (S, I, O, \{A(y|x)\})$ be an SSM and let π be an initial stochastic distribution (i.e., an $|S|$-dimensional stochastic row vector). Then the accepting probability $p_\pi^{\mathcal{M}}(v|u)$ for inputting string $u = x_1x_2\cdots x_m$ and outputting string $v = y_1y_2\cdots y_m$ is defined as follows:

$$p_\pi^{\mathcal{M}}(v|u) = \pi A(v|u)\eta \tag{73}$$

where $A(v|u) = A(x_1|y_1)A(x_2|y_2)\cdots A(x_m|y_m)$, and η is an $|S|$-dimensional column vector with all entries being 1.

For a given SSM $\mathcal{M}$, $\mathbb{F}^{\mathcal{M}}$ denotes the set of all functions $\mathbb{F}^{\mathcal{M}} = \{p_\pi^{\mathcal{M}} : \pi \in \mathbb{P}_n\}$, where $\mathbb{P}_n$ denotes the set of all n-dimensional stochastic row vectors.

Definition 18 ([18])**.** Let $\mathcal{M}$ and $\mathcal{M}'$ be two SSM. The machine $\mathcal{M}'$ covers the machine $\mathcal{M}$ (denoted by $\mathcal{M}' \geq \mathcal{M}$) if $\mathbb{F}^{\mathcal{M}'} \supseteq \mathbb{F}^{\mathcal{M}}$.

Now we are ready to introduce the open problems proposed by Paz [18] as follows:

1. Answer the following problem, or prove that it is not decidable:

   Given an SSM $\mathcal{M}$, does there exist an SSM $\mathcal{M}'$ with fewer states than SSM $\mathcal{M}$ and such that $\mathcal{M}' \geq \mathcal{M}$.

2. If the problem under 1 is decidable, then construct a finite algorithm for finding a machine $\mathcal{M}' \geq \mathcal{M}$ with $|S^{\mathcal{M}'}| < |S^{\mathcal{M}}|$, whenever such a machine $\mathcal{M}'$ exists, where $|S^{\mathcal{M}'}|$ and $|S^{\mathcal{M}}|$ denote the numbers of states of M' and M, respectively.

We address the above open problems and show that Question 1 is decidable. Moreover, we give an EXPSPACE upper bound (on the number of states) for the algorithm finding a minimal covering SSM. The idea is similar to that shown in Section 2. We show that the set of all SSM's $\mathcal{M}'$ with n' states covering some SSM $\mathcal{M}$ with n states such that $n' < n$ can be represented by a solution of a system of polynomial equations and/or inequations. Moreover, we show that the emptiness of this set can be checked in EXPSPACE in n and that it can also be sampled in EXPSPACE (in n). To obtain this result we need the following well-known result:

Theorem 19 ([18])**.** Let $\mathcal{M}$ and $\mathcal{M}'$ be SSM's with n and n' states respectively. Then $\mathcal{M}' \geq \mathcal{M}$ iff for all $i = 1, \ldots, n$, there exists a stochastic row π_i' with dimension n' such that for all inputting string u and outputting string v with $|u| = |v| \leq n + n' - 1$,

$$P_{e_i}^{\mathcal{M}}(v|u) = P_{\pi_i'}^{\mathcal{M}'}(v|u) \tag{74}$$

where $e_i = (0, \ldots, 0, 1, 0, \ldots, 0)$ with the i'th entry being 1 and else 0's.

As we shall see, an immediate corollary of the above theorem is that the set of SSM's with a fixed dimension that cover another SSM is represented by a set of polynomial equations and/or inequations. Then, by Theorem 1, we obtain the following result.

Theorem 20. Given an SSM M with n states, checking whether there exists an SSM $\mathcal{M}'$ covering M with n' states and $n' < n$ is decidable and finding such SSM is EXPSPACE in n.

Proof. Here we depict the main idea. Given an SSM $\mathcal{M} = (S, I, O, \{A(y|x)\})$ with n' states, define the set

$$\mathbb{S}_{\mathcal{M}}^{(n')} = \{\mathcal{M}' : \mathcal{M}' \text{ covers } \mathcal{M} \text{ with } n' \text{ states}\}. \tag{75}$$

Then $\mathcal{M}' \in \mathbb{S}_{\mathcal{M}}^{(n')}$ can be presented by a vector $\mathbb{R}^{n'^2(|I|\times|O|)+n'n}$ having the form

$$(w'_{11x_1y_1}, \ldots, w'_{n'n'x_ky_m}, \pi_1^1, \ldots, \pi_{n'}^1, \ldots, \pi_1^n, \ldots, \pi_{n'}^n)$$

where

- $A' = (w'_{11x_1y_1}, \ldots, w'_{n'n'x_ky_m})$ consists of real variables representing the transition matrices of $\mathcal{M}'$; these variables must satisfy the polynomial Eqs. (72).
- $\pi^\ell = (\pi_1^\ell, \ldots, \pi_{n'}^\ell)$ represents a stochastic vector with dimension n' and $\ell = 1, \ldots, n$; as shown in Section 3, "a vector is stochastic vector" can be represented by a set of polynomial equations and inequations.
- In addition, the accepting probability $p_\pi^{\mathcal{M}'}(v|u)$ of $\mathcal{M}'$ can be represented by a polynomial as shown in Eq. (73); furthermore, " $\mathcal{M}'$ covers $\mathcal{M}$" can be represented by a set of polynomial equations that follow from Eq. (74).

Shortly, $\mathbb{S}_{\mathcal{M}}^{(n')}$ can be presented by a set of polynomial equations and inequations with $n'^2(|I| \times |O|) + n'n$ variables. Thus, by checking the emptiness of the set we know if there exists an SSM with dimension n' covering $\mathcal{M}$. According to Theorem 1 this can be done in EXPSPACE in n. Here, we omit the complexity analysis, which is in fact similar to the cases from the previous sections. □

Corollary 21. Finding a minimal covering SSM can be performed in EXPSPACE on the number of states.

Proof. The following result follows straightforwardly from Theorem 20. Given an SSM $\mathcal{M}$ with dimension n, search a covering SSM with dimension ranging from 1 to $n-1$, and output (if any is found) the first SSM. Since each step can be achieved in EXPSPACE, so does the full search. □

8. Conclusion

In this work we presented a method to minimize several types of quantum and probabilistic finite automata. We proved that the state minimization of these models is decidable and that

its complexity is at most exponential in space. The proposed technique can be employed to minimize any kind of finite automata that is able to be bilinearized. In particular, we have shown that the minimization of probabilistic automata and measure-once quantum finite automata is decidable, solving an open problem proposed by Moore and Crutchfield ([16], p. 304, Problem 5). Furthermore, we proved that the reduction of the number of states for any given SSM is decidable by presenting and EXPSPACE algorithm for finding a reduced SSM. With this result we solved an open problem proposed by Paz (see [18], page 43). Finally, we also addressed the minimization problem for many classes of quantum automata, namely MM-1QFA, MO-1gQFA, and MM-1gQFA.

Acknowledgments

The authors are grateful to the anonymous reviewers for their invaluable suggestions that help to improve the presentation of the paper. We thank Amílcar Sernadas for several helpful suggestions, and with whom we started the discussion about the minimization problem for quantum automata during the QuantLog project. We thank Azia Paz for several comments and clarifications concerning probabilistic automata. Moreover, we thank John Watrous that suggested we considered quantum automata with general evolutions and observations. Finally, we also thank Nikola Paunkovic that helped to improve the readability of the paper. This work is supported in part by the National Natural Science Foundation (Nos. 60873055, 61073054, 61100001), the Natural Science Foundation of Guangdong Province of China (No. 10251027501000004), the Fundamental Research Funds for the Central Universities (Nos. 10lgzd12, 11lgpy36), the Research Foundation for the Doctoral Program of Higher School of Ministry of Education of China (Nos. 20100171110042, 20100171120051), the China Postdoctoral Science Foundation project (Nos. 20090460808, 201003375), and the project of SQIG at IT, funded by FCT and EU FEDER projects Quantlog POCI/MAT/55796/2004 and QSec PTDC/EIA/67661/2006, IT Project QuantTel, NoE Euro-NF, and the SQIG LAP initiative.

References

[1] A. Ambainis, R. Freivalds, One-way quantum finite automata: strengths, weaknesses and generalizations, in: Proceedings of the 39th Annual Symposium on Foundations of Computer Science, IEEE Computer Society Press, Palo Alfo, California, USA, 1998, pp. 332-341.

[2] S. Basu, R. Pollack, and R. M.-F. Coise, Algorithms in Real Algebraic Geometry, 2nd Edition, Springer, 2006.

[3] P. Benioff, The computer as a physical system: a microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines, Journal of Statistic Physics 22 (1980) 563-591.

[4] R. G. Bukharaev, Probabilistic Automata, Translated from Itogi Nauki i Tekhniki. Teoriya Veroyatnostei, Matematicheskaya Statistika, Teoreticheskaya Kibernetika, Vol. 15, pp. 79-122, 1978.

[5] A. Brodsky, N. Pippenger, Characterizations of 1-way quantum finite automata, SIAM Journal on Computing 31 (2002) 1456-1478.

[6] J. Canny, Some algebraic and geometric computations in PSPACE. In: Proceedings of the twentieth annual ACM Symposium on Theory of Computing, ACM, New York, USA, 1988, pp. 460–469.

[7] D. Deutsh, Quantum theory, the Church-Turing principle and the universal quantum computer, Proceedings of the Royal Society of London Series A 400 (1985) 97-117.

[8] R.P. Feynman, Simulating physics with computers, International Journal of Theoretical Physics 21 (1982) 467-488.

[9] L. Grover, A fast quantum mechanical algorithms for datdbase search, in: Proceedings of the 28th Annual ACM Symposium on Theory of Computing, ACM, Philadelphia, Pennsylvania, 1996, pp. 212-219.

[10] J. Gruska, Quantum Computing, McGraw-Hill, London, 1999.

[11] J.E. Hopcroft, J.D. Ullman, Introduction to Automata Theory, Languages, and Computation, Addision-Wesley, New York, 1979.

[12] A. Kondacs, J. Watrous, On the power of finite state automata, in: Proceedings of the 38th Annual Symposium on Foundations of Computer Science, IEEE Computer Society Press, Miami Beach, Florida, USA, 1997, pp. 66-75.

[13] L.Z. Li, D.W. Qiu, Determining the equivalence for one-way quantum finite automata, Theoretical Computer Science 403 (2008) 42-51.

[14] L.Z. Li, D.W. Qiu, A note on quantum sequential machines, Theoretical Computer Science 410 (2009) 2529-2535.

[15] L.Z. Li, D.W. Qiu, *et al*, Characterizations of one-way general quantum finite automata, Theoretical Computer Science (2011), doi:10.1016/j.tcs.2011.10.021 (in press), also see arXiv:0911.3266,.

[16] C. Moore, J.P. Crutchfield, Quantum automata and quantum grammars, Theoretical Computer Science 237 (2000) 275-306.

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

[18] A. Paz, Introduction to Probabilistic Automata, Academic Press, New York, 1971.

[19] M. O. Rabin, Probabilistic Automata, Information and Control 6 (3) (1963) 230-245.

[20] J. Renegar, A faster PSPACE algorithm for deciding the existential theory of the reals, in: Proceedings of the 29th Annual Symposium on Foundations of Computer Science, IEEE Computer Society Press, 1988, pp. 291-295.

[21] P.W. Shor, Algorithm for quantum computation: discrete logarithms and factoring, in: Proceedings of the 37th Annual Symposium on Foundations of Computer Science, IEEE Computer Society Press, 1994, pp. 124-134.

[22] A. Tarski, A Decision Method for Elementary Algebra and Geometry, Univ. of California Press, 1951.

[23] W.G. Tzeng, A Polynomial-time Algorithm for the Equivalence of Probabilistic Automata, SIAM Journal on Computing 21 (2) (1992) 216-227.

[24] J. Watrous, Lecture notes in University of Waterloo: Theory of Quantum Information (fall 2008), http://www.cs.uwaterloo.ca/ watrous/quant-info/.

[25] A. Yakaryilmaz, A.C.C. Say, Languages recognized with unbounded error by quantum finite automata, In Proceedings of the 4th Computer Science Symposium in Russia, Lecture Notes in Comput. Sci. 5675, Springer-Verlag, Berlin, 2009, pp. 356-367.

[26] A. Yakaryilmaz, A.C.C. Say, Unbounded-error quantum computation with small space bounds, Information and Computation 209 (2011) 873-892.