RAIRO-Inf. Theor. Appl. **40** (2006) 315-332
DOI: 10.1051/ita:2006007

QUANTUM FINITE AUTOMATA WITH CONTROL LANGUAGE*

Carlo Mereghetti[1] and Beatrice Palano[1]

Abstract. Bertoni *et al.* introduced in *Lect. Notes Comput. Sci.* **2710** (2003) 1–20 a new model of 1-way quantum finite automaton (1qfa) called *1qfa with control language (1qfc)*. This model, whose recognizing power is exactly the class of regular languages, generalizes main models of 1qfa's proposed in the literature. Here, we investigate some properties of 1qfc's. In particular, we provide algorithms for constructing 1qfc's accepting the inverse homomorphic images and quotients of languages accepted by 1qfc's. Moreover, we give instances of binary regular languages on which 1qfc's are proved to be more succinct (*i.e.*, to have less states) than the corresponding classical (deterministic) automata.

Mathematics Subject Classification. 68Q10, 68Q19, 68Q45.

Introduction

Quantum finite automata (qfa's, for short) [5, 12] are computational devices particularly interesting since they represent a theoretical model for a quantum computer with finite memory. We can hardly expect to see a fully quantum computer [11] in the near future, while it is reasonable to think of classical computers incorporating small quantum components such as, *e.g.*, a qfa. Qfa's exhibit both advantages and disadvantages with respect to their classical (deterministic or probabilistic) counterpart. Basically, quantum superposition offers some computational

Keywords and phrases. Quantum computing, quantum finite automata.

* *Partially supported by M.I.U.R. under the projects "COFIN: Automi e linguaggi formali: aspetti matematici e applicativi." and "FIRB: Complessità descrizionale di automi e strutture correlate".*

[1] Dipartimento di Scienze dell'Informazione, Università degli Studi di Milano, via Comelico 39, 20135 Milano, Italy; {mereghetti,palano}@dsi.unimi.it

advantages on probabilistic superposition. On the other hand, quantum dynamics are reversible: because of limitation of memory, it is sometimes impossible to simulate deterministic automata by quantum automata.

Originally, two models of qfa's have been proposed and investigated. The simplest model is represented by *measure-once* 1-way qfa's (mo-1qfa's) [9,15]. In this model, the probability of accepting strings is evaluated by "observing" just once, at the end of input processing. The computational power of mo-1qfa's is weaker than that of classical automata. In fact, in [4,9] it is proved that they recognize exactly the class of group languages [21], a proper subclass of regular languages. In *measure-many* 1qfa's (mm-1qfa's) [2,14], instead, such an observation is performed after each move. Mm-1qfa's are proved to have a computational power stronger than mo-1qfa's, but still weaker than classical automata [1,3,14].

Several modifications to these original models of qfa's have then been proposed in order to gain more computational power, but still retaining the "quantum nature" of computing. Thus, "enhanced" [20], reversible [10], Latvian [1] qfa's have been introduced. The computational power of these models lies between that of mo-1qfa's and that of classical automata.

Along this line of research, Bertoni *et al.* proposed in [5] the model of *1-way quantum finite automata with control language (1qfc's for short)*. This model is particularly interesting since it generalizes several models of 1qfa's. A 1qfc $\mathcal{A}$ can be regarded to as a computational device having a quantum processor controlled by a classical automaton C. The state of $\mathcal{A}$ is observed after each move by an observable with a fixed, but arbitrary, set of possible outcomes. On any given input word x, a sequence y of outcomes is generated with a certain probability; the computation of $\mathcal{A}$ on x is accepting whenever y belongs to the regular language (the control language) recognized by C. In [5], several closure properties for the class of stochastic events realized by 1qfc's are investigated. Yet, it is proved that a language accepted with isolated cut point by a 1qfc is regular.

Here, we continue the investigations on 1qfc's by studying both their computational power and their descriptional complexity (*i.e.*, the size of their classical and quantum components). In Section 2, we begin by establishing the exact computational power of 1qfc's. In fact, we prove that the class of languages recognized with isolated cut point by 1qfc's coincides with the class of regular languages. We obtain this result by designing an algorithm which, having a given deterministic automaton as input, constructs an equivalent 1qfc (although with a "large" classical component). In Section 3.1, we provide algorithms for constructing 1qfc's that accept quotients and inverse homomorphic images of languages accepted by 1qfc's. This enables us to study the cost, in terms of quantum basis states and classical states, of these operations on 1qfc's (for other types of 1qfa's, these operations have been investigated, *e.g.*, in [1]). Moreover, the construction for the inverse homomorphic images is used in Section 3.2 to build 1qfc's recognizing a family of binary regular languages. On this family, we are able to obtain 1qfc's more

succinct than equivalent classical automata, thus showing that, in some cases, the general construction in Section 2 can be improved.

1. Basic definitions

We begin by quickly recalling some notations of linear algebra. For more details, we refer the reader to, *e.g.*, [16,17].

We denote by $\mathbf{N}$ the set of non negative integers and $\mathbf{C}$ the set of complex numbers. Given a complex number $z \in \mathbf{C}$, its *conjugate* is denoted by $\overline{z}$, and its *modulus* is $|z| = \sqrt{z\overline{z}}$. We denote by $\mathbf{C}^{n\times m}$ the set of $n \times m$ matrices with entries in $\mathbf{C}$. The *adjoint* of a matrix $M \in \mathbf{C}^{n\times m}$ is the matrix $M^\dagger \in \mathbf{C}^{m\times n}$, where $M^\dagger_{ij} = \overline{M_{ji}}$. For matrices $A \in \mathbf{C}^{n\times m}$ and $B \in \mathbf{C}^{p\times q}$, their direct sum and Kronecker's product are the $(n+p)\times(m+q)$ and $np \times mq$ matrices defined, respectively, as

$$A \oplus B = \begin{pmatrix} A & 0_{n\times q} \\ 0_{p\times m} & B \end{pmatrix}, \qquad A \otimes B = \begin{pmatrix} A_{11}B & \cdots & A_{1m}B \\ \vdots & \ddots & \vdots \\ A_{n1}B & \cdots & A_{nm}B \end{pmatrix},$$

where $0_{n\times m}$ denotes the $n \times m$ zero matrix. As a shortcut, we let $0_n = 0_{n\times n}$. For vectors $\pi \in \mathbf{C}^{1\times n}$ and $\eta \in \mathbf{C}^{1\times m}$, their direct sum is the $1 \times (n+m)$ vector $\pi \oplus \eta = (\pi_1, \ldots, \pi_n, \eta_1, \ldots, \eta_m)$.

An Hilbert space of dimension n is the linear space $\mathbf{C}^{1\times n}$ equipped with sum and product by elements in $\mathbf{C}$, in which the *inner product* $\langle \pi, \xi \rangle = \pi\xi^\dagger$ is defined. If $\langle \pi, \xi \rangle = 0$ we say that π is *orthogonal* to ξ. The *norm* of vector $\pi \in \mathbf{C}^{1\times n}$ is defined as $\| \pi \| = \sqrt{\langle \pi, \pi \rangle}$. Two subspaces X, Y are orthogonal if any vector in X is orthogonal to any vector in Y; in this case, the linear space generated by $X \cup Y$ is denoted by $X \dotplus Y$.

A matrix $M \in \mathbf{C}^{n\times n}$ is said to be *unitary* whenever $MM^\dagger = I_n = M^\dagger M$, where I_n is the $n \times n$ identity matrix; moreover, a matrix is unitary if and only if it preserves the norm, *i.e.*, $\| \pi M \| = \| \pi \|$ for each vector $\pi \in \mathbf{C}^{1\times n}$. M is said to be *Hermitian* whenever $M = M^\dagger$. Given an Hermitian matrix $\mathcal{O} \in \mathbf{C}^{n\times n}$, let $c_1, \ldots, c_s$ be its eigenvalues and $E_1, \ldots, E_s$ the corresponding eigenspaces. It is well known that each eigenvalue c_k is real, that E_i is orthogonal to E_j, for any $i \neq j$, and that $E_1 \dotplus \cdots \dotplus E_s = \mathbf{C}^{1\times n}$. Each vector $\pi \in \mathbf{C}^{1\times n}$ can be uniquely decomposed as $\pi = \pi_1 + \cdots + \pi_s$, where $\pi_j \in E_j$. The linear transformation $\pi \mapsto \pi_j$ is the *projector* P_j on the subspace E_j. It is easy to see that $\sum_{j=1}^s P_j = I$. The Hermitian matrix $\mathcal{O}$ is biunivocally determined by its eigenvalues and its eigenspaces or, equivalently, by its projectors: in fact, we have that $\mathcal{O} = c_1P_1 + \cdots + c_sP_s$.

Let us now use this formalism to describe quantum systems.

Given a set $Q = \{q_1, \ldots, q_m\}$, every q_i can be represented by its characteristic vector $e_i = (0, \ldots, 1, \ldots, 0)$. A *quantum state* on Q is a superposition $\pi = \sum_{k=1}^m \alpha_k e_k$, where the coefficients α_k are complex *amplitudes* and $\| \pi \| = 1$. Every e_k is called *quantum basis state*. Given alphabet $\Sigma = \{\sigma_1, \ldots, \sigma_l\}$, with every symbol σ_i we associate a unitary transformation $U(\sigma_k) : \mathbf{C}^{1\times m} \to \mathbf{C}^{1\times m}$.

An *observable* is described by an Hermitian matrix $\mathcal{O} = c_1P_1 + \cdots + c_sP_s$. Suppose that a quantum system is described by the quantum state π. Then, we can operate

(1) *Evolution* $U(\sigma_j)$. In this case, the new state $\xi = \pi U(\sigma_j)$ is reached; this dynamics is reversible, since $\pi = \xi U^{\dagger}(\sigma_j)$.
(2) *Measurement of* $\mathcal{O}$. In this case, every result in $\{c_1, \ldots, c_s\}$ can be obtained; c_j is obtained with probability $\| \pi P_j \|^2$ and the state after such a measurement is $\pi P_j / \| \pi P_j \|$. The state transformation induced by a measurement is typically irreversible.

1.1. Quantum finite automata with control language

Let us now recall the model of 1-way quantum finite automata with control language as stated in [5]. In this model, an input word x from a given input alphabet Σ is placed onto an input tape with a special character $\sharp \notin \Sigma$ as right endmarker. The state of the system can be observed after each symbol of x is processed. An observable $\mathcal{O}$ with a fixed, but arbitrary, set of possible results $\mathcal{C} = \{c_1, \ldots, c_s\}$ is considered. On x, the computation displays a sequence $y \in \mathcal{C}^*$ of results of measurements of $\mathcal{O}$ with a certain probability $p(y; x)$: the computation is "accepting" if and only if y belongs to a fixed regular control language $\mathcal{L} \subseteq \mathcal{C}^*$. More formally:

Definition 1.1. Given an alphabet Σ and an endmarker symbol $\sharp \notin \Sigma$, an m-state *1-way quantum finite automaton with control language* (1qfc, for short) is a system $\mathcal{A} = (\pi, \{U(\gamma)\}_{\gamma \in \Gamma}, \mathcal{O}, \mathcal{L})$, for $\Gamma = \Sigma \cup \{\sharp\}$, where

- $\pi \in \mathbf{C}^{1 \times m}$ is the initial amplitude vector satisfying $\| \pi \| = 1$;
- $U(\gamma) \in \mathbf{C}^{m \times m}$ is a unitary matrix, for all $\gamma \in \Gamma$;
- $\mathcal{O}$ is an observable on $\mathbf{C}^{1 \times m}$; if $\mathcal{C} = \{c_1, \ldots, c_s\}$ is the class of all possible results of measurements of $\mathcal{O}$, $P(c_i)$ denotes the projector on the eigenspace corresponding to c_i, for all $c_i \in \mathcal{C}$;
- $\mathcal{L} \subseteq \mathcal{C}^*$ is a regular language (the control language).

Now, we define the behavior of $\mathcal{A}$ on a word $x_1 \cdots x_n\sharp \in \Sigma^*\sharp$. At any time, the state of $\mathcal{A}$ is a vector $\xi \in \mathbf{C}^{1 \times m}$ with $\| \xi \| = 1$. The computation starts in the state π, then transformations associated with the symbols in the word $x_1 \cdots x_n\sharp$ are applied in succession. The transformation corresponding to a symbol $\gamma \in \Gamma$ consists of two steps:

(1) First, $U(\gamma)$ is applied to the current state ξ of the automaton yielding the new state ξ'.
(2) Then, the observable $\mathcal{O}$ is measured on ξ'. According to quantum mechanics principles above recalled, the result of measurement is c_k with probability $\| \xi' P(c_k) \|^2$, and the state of the automaton "collapses" to $\xi' P(c_k) / \| \xi' P(c_k) \|$.

Thus, a computation on $x_1 \cdots x_n \sharp$ leads to a sequence $y_1 \cdots y_n y_\sharp$ of results of the measurements of $\mathcal{O}$ with probability $p_{\mathcal{A}}(y_1 \cdots y_n y_\sharp; x_1 \cdots x_n \sharp)$ given by

$$p_{\mathcal{A}}(y_1 \cdots y_n y_\sharp; x_1 \cdots x_n \sharp) = \left|\left| \pi \left(\prod_{i=1}^{n} U(x_i) P(y_i) \right) U(\sharp) P(y_\sharp) \right|\right|^2 .$$

A computation leading to the word $y_1 \cdots y_n y_\sharp$ is *accepting* if $y_1 \cdots y_n y_\sharp \in \mathcal{L}$, otherwise it is rejecting. Hence, the probability that, on input $x_1 \cdots x_n \sharp$, the automaton generates an accepting computation is

$$\mathcal{E}_{\mathcal{A}}(x_1 \cdots x_n) = \sum_{y_1 \cdots y_n y_\sharp \in \mathcal{L}} p_{\mathcal{A}}(y_1 \cdots y_n y_\sharp; x_1 \cdots x_n \sharp). \tag{1}$$

The function $\mathcal{E}_{\mathcal{A}} : \Sigma^* \to [0,1]$ is the *stochastic event induced by* $\mathcal{A}$. The *language accepted by* $\mathcal{A}$ *with cut point* $\lambda \in (0,1]$ is the set

$$L_{\mathcal{A},\lambda} = \{x \in \Sigma^* \mid \mathcal{E}_{\mathcal{A}}(x) > \lambda\}.$$

A language L is said to be accepted by $\mathcal{A}$ *with isolated cut point* λ, if there exists $\varepsilon > 0$ such that, for any $x \in L$ ($x \notin L$), we have $\mathcal{E}_{\mathcal{A}}(x) \geq \lambda + \varepsilon$ ($\mathcal{E}_{\mathcal{A}}(x) \leq \lambda - \varepsilon$).

In what follows, we say that $\mathcal{A}$ has q quantum basis states and k classical states whenever it is a q-state 1qfc and the control language $\mathcal{L}$ can be recognized by a k-state 1-way deterministic finite state automaton (1dfa). In other words q (resp., k) measures the size of the quantum (resp., classical) component of our device.

2. The computational power of 1qfc's

In [5], Bertoni *et al.* proved that: (1) the stochastic events induced by 1qfc's are bounded rational formal series, and (2) the languages defined with isolated cut point by bounded real valued rational formal series are regular. These two facts lead to the following

Theorem 2.1. *The languages recognized with isolated cut point by 1qfc's are regular.*

We are now going to show that the converse holds true as well, and hence the class of languages accepted by 1qfc's coincides with the class of regular languages. In what follows, a 1dfa will be denoted by a 5-tuple $(K, \Sigma, \delta, q, F)$ where K is the set of states, Σ the input alphabet, δ the transition function, q the initial state, and F the set of accepting states.

Theorem 2.2. *For any regular language $L \subseteq \{a,b\}^*$ accepted by a k-state 1dfa A, there exists a 1qfc $\mathcal{A}$ recognizing deterministically L with 3 quantum basis states and $3k$ classical states.*

Proof. We define $\mathcal{A} = (\pi_0, \{U(\gamma)\}_{\gamma\in\{a,b,\sharp\}}, \mathcal{O}, \mathcal{L})$ as follows:

- $\pi_0 = (1, 0, 0)$;
- $U(a) = I_3$, $U(b) = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$, $U(\sharp) = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$;
- $\mathcal{O} = 0 \cdot P(0) + 1 \cdot P(1) + 2 \cdot P(2)$, where

$$P(0) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \; P(1) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \; P(2) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix};$$

- the control language $\mathcal{L}$ is accepted by the 1dfa C constructed as follows. Let $A = (K, \{a, b\}, \delta, q, F)$ be the k-state 1dfa recognizing L, then

$$C = (K' = K \times \{0, 1, 2\}, \{0, 1, 2\}, \delta', (q, 0), F' = \{(p, i) \in K' \mid p \in F\})$$

where, for any $(p, i) \in K'$ and $j \in \{0, 1, 2\}$:

$$\delta'((p, i), j) = \begin{cases} (\delta(p, a), j) & \text{if } j = i \\ (\delta(p, b), j) & \text{if } j = (i + 1) \bmod 3 \\ (p, j) & \text{if } j = (i + 2) \bmod 3. \end{cases} \tag{2}$$

Let us now briefly explain how the 1qfc $\mathcal{A}$ works. First of all, it is easy to see that, at any time during a computation, $\mathcal{A}$ is exactly in one of the following states: $\pi_0 = (1, 0, 0)$, $\pi_1 = (0, 1, 0)$, $\pi_2 = (0, 0, 1)$. Moreover, the observation performed in these states leads with certainty to the outcome 0, 1, 2, respectively, thus leaving $\mathcal{A}$ in the same state. Suppose that $\mathcal{A}$ is in π_i (hence the outcome of the observation has been i) and reads the input symbol $\gamma \in \{a, b, \sharp\}$. Then:

- if $\gamma = a$, $\mathcal{A}$ remains in π_i and then the observation outputs i with certainty, as the outcome at the previous step;
- if $\gamma = b$, $\mathcal{A}$ moves to state $\pi_{(i+1) \bmod 3}$ and then the observation outputs $(i + 1) \bmod 3$ with certainty;
- if $\gamma = \sharp$, $\mathcal{A}$ moves to state $\pi_{(i+2) \bmod 3}$ and then the observation outputs $(i + 2) \bmod 3$ with certainty.

In other words, the dynamics of $\mathcal{A}$ is deterministic and, on any input string $x\sharp$, it returns with certainty a unique string $\Psi_{\mathcal{A}}(x\sharp) \in \{0, 1, 2\}^*$ of outcomes of the observations. Yet, this transduction $\Psi_{\mathcal{A}} : \{a, b, \sharp\}^* \to \{0, 1, 2\}^*$ is bijective, as the reader may easily verify. Thus, for any $x \in \{a, b\}^*$ and $y \in \{0, 1, 2\}^*$, we get

$$p_{\mathcal{A}}(y, x\sharp) = \begin{cases} 1 \text{ if } y = \Psi_{\mathcal{A}}(x\sharp) \\ 0 \text{ otherwise.} \end{cases} \tag{3}$$

If we set the control language as $\mathcal{L} = \{\Psi_{\mathcal{A}}(x\sharp) \mid x \in L\}$, by (3), we would obtain

$$\mathcal{E}_{\mathcal{A}}(x) = \sum_{y\in\mathcal{L}} p_{\mathcal{A}}(y; x\sharp) = \begin{cases} 1 \text{ if } \Psi_{\mathcal{A}}(x\sharp) \in \mathcal{L} \text{ (iff } x \in L) \\ 0 \text{ otherwise,} \end{cases}$$

whence the correctness of our construction. Thus, it is enough to show that the 1dfa C given in the definition of $\mathcal{A}$ recognize $\{\Psi_{\mathcal{A}}(x\sharp) \mid x \in L\}$. We informally describe how C works. By definition, the states of C consists of two components: the first is a state of A (the 1dfa accepting L), the second records the outcome of the last observation of $\mathcal{A}$. According to the definition (2), the transition of C takes place by comparing the second component i of the state (p, i) with the input symbol j. If these two values coincide, then the 1qfc has read an a; if $j = (i+1) \bmod 3$ ($j = (i+2) \bmod 3$) then the 1qfc has read a b (the endmarker $\sharp$). The transition leads to a state where the second component stores j, while in the first component we operate as follows: if the symbol read by the 1qfc is a (b) then we evolve in $\delta(p, a)$ ($\delta(p, b)$), otherwise, on input $\sharp$, we remain in the current state p. It is easy to see that, at the end of the computation of $\mathcal{A}$ on $x\sharp$, C reaches the state $(\delta(q, x), h)$, for some $h \in \{0, 1, 2\}$. Hence, $\Psi_{\mathcal{A}}(x\sharp)$ is accepted by C if and only if $(\delta(q, x), h) \in F'$ if and only if $\delta(q, x) \in F$ if and only if $x \in L$. □

3. On the descriptional complexity of 1qfc's

As we have seen in the previous section, the class of languages accepted by 1qfc's coincides with the class of regular languages. This directly implies that the class of languages accepted by 1qfc's is closed under inverse homomorphic images and quotients. In what follows, we are going to design algorithms to explicitly construct 1qfc's that accept quotients and inverse homomorphic images of regular languages (defined by 1qfc's). This will enable to study the cost, in terms of quantum basis states and classical states, of these operations on 1qfc's.

3.1. Building 1qfc's

First, let us recall the operations on languages we are interested in.

Definition 3.1. Given a language $L \subseteq \Sigma^*$ and two words $v, w \in \Sigma^*$, the quotient of L with respect to v, w is the language $v^{-1}Lw^{-1} = \{x \in \Sigma^* \mid vxw \in L\}$.

Definition 3.2. Given two alphabets Σ, Δ, a language $L \subseteq \Delta^*$, and an homomorphism $\varphi : \Sigma^* \to \Delta^*$, the inverse homomorphic image of L is the language $\varphi^{-1}(L) = \{x \in \Sigma^* \mid \varphi(x) \in L\}$. Given a word $y \in \Delta^*$, we also set $\varphi^{-1}(y) = \{x \in \Sigma^* \mid \varphi(x) = y\}$; thus, $\varphi^{-1}(L) = \cup_{y \in L} \varphi^{-1}(y)$.

We begin by approaching the construction of 1qfc's for quotients. We will construct 1qfc's for accepting $\sigma^{-1}L$ and $L\sigma^{-1}$, for $\sigma \in \Sigma$ and language $L \subseteq \Sigma^*$ accepted by a 1qfc. By iterating these constructions, one obtains a 1qfc for $v^{-1}Lw^{-1}$, for given $v, w \in \Sigma^*$.

Theorem 3.3. *Let $L \subseteq \Sigma^*$ be a language recognized with isolated cut point by a 1qfc $\mathcal{A}$ with q quantum basis states, k classical states and a control language alphabet consisting of $s > 1$ symbols. Then, there exists a 1qfc $\mathcal{B}$ with $(s^2 + s)q$ quantum states and k^s classical states that recognizes $\sigma^{-1}L$ with isolated cut point.*

Proof. For the sake of simplicity, we begin by proving the result for binary control languages. Then, we generalize our reasoning to arbitrary observables. Thus, let $\mathcal{A} = (\pi, \{U(\gamma)\}_{\gamma\in\Gamma}, \mathcal{O}, \mathcal{L})$, with $\Gamma = \Sigma \cup \{\sharp\}$, $\mathcal{O} = 0\cdot P(0) + 1\cdot P(1)$ and $\mathcal{L} \subseteq \{0,1\}^*$, be the 1qfc accepting the language L with isolated cut point. We construct the 1qfc $\mathcal{B} = (\pi', \{U'(\gamma)\}_{\gamma\in\Gamma}, \mathcal{O}', \mathcal{L}')$ inducing the stochastic event $\mathcal{E}_{\mathcal{B}}(x) = \mathcal{E}_{\mathcal{A}}(\sigma x)$, for any $x \in \Sigma^*$, thus recognizing $\sigma^{-1}L$ with isolated cut point. Roughly speaking, on input $x \in \Sigma^*$, $\mathcal{B}$ "carries on the possible computations" of $\mathcal{A}$ on input σx. By the choice of a suitable observable, these computations are correctly taken into account upon evolving and observing on $\sharp$.

The initial state of $\mathcal{B}$ is the $1 \times 6q$ unit vector $\pi' = (\pi U(\sigma)P(0), \pi U(\sigma)P(1), 0_{1\times 4q})$. The evolution on $\gamma \in \Sigma$ is represented by the $6q \times 6q$ unitary matrix

$$U'(\gamma) = U(\gamma) \oplus U(\gamma) \oplus I_{4q},$$

while the evolution on $\sharp$ is defined as

$$U'(\sharp) = \begin{pmatrix} 0_q & 0_q & U(\sharp)P(0) & U(\sharp)P(1) & 0_q & 0_q \\ 0_q & 0_q & 0_q & 0_q & U(\sharp)P(0) & U(\sharp)P(1) \\ 0_q & 0_q & U(\sharp)P(1) & U(\sharp)P(0) & 0_q & 0_q \\ 0_q & 0_q & 0_q & 0_q & U(\sharp)P(1) & U(\sharp)P(0) \\ I_q & 0_q & 0_q & 0_q & 0_q & 0_q \\ 0_q & I_q & 0_q & 0_q & 0_q & 0_q \end{pmatrix}.$$

The reader may verify that $U'(\sharp)$ is unitary.

The observable is $\mathcal{O}' = 0\cdot P'(0) + 1\cdot P'(1) + a\cdot P'(a) + b\cdot P'(b) + c\cdot P'(c) + d\cdot P'(d)$, where the projectors are as follows:

$$\begin{array}{ll} P'(0) = P(0) \oplus P(0) \oplus 0_{4q} & P'(1) = P(1) \oplus P(1) \oplus 0_{4q} \\ P'(a) = 0_{2q} \oplus I_q \oplus 0_{3q} & P'(b) = 0_{3q} \oplus I_q \oplus 0_{2q} \\ P'(c) = 0_{4q} \oplus I_q \oplus 0_q & P'(d) = 0_{5q} \oplus I_q. \end{array}$$

The control language is $\mathcal{L}' = \Lambda_0 \cup \Lambda_1$, where $\Lambda_0 = 0^{-1}\mathcal{L}0^{-1}a \cup 0^{-1}\mathcal{L}1^{-1}b$ and $\Lambda_1 = 1^{-1}\mathcal{L}0^{-1}c \cup 1^{-1}\mathcal{L}1^{-1}d$. Supposing that $\mathcal{L}$ is recognized by the k-state 1dfa $C = (K, \{0,1\}, \delta, q_0, F)$, it is not hard to see that Λ_0 can be recognized by the k-state 1dfa $C_0 = (K, \{0,1,a,b\}, \delta_0, \delta(q_0,0), F)$. The transition function δ_0 is defined as δ except for transitions leading to a final state: for any $p \in K$ such that $\delta(p,0) = s \in F$ ($\delta(p,1) = s \in F$), we let $\delta_0(p,a) = s$ ($\delta_0(p,b) = s$). In a similar way, a k-state 1dfa C_1 recognizing Λ_1 can be built from C. Finally, a k^2-state 1dfa accepting $\mathcal{L}' = \Lambda_0 \cup \Lambda_1$ can be built from C_1 and C_2 by using the standard Cartesian product construction (see, *e.g.*, [13]).

Let us now evaluate the stochastic event realized by $\mathcal{B}$ on the word $x = x_1 \cdots x_n \in \Sigma^*$. In what follows, we let $y = y_1 \cdots y_n \in \{0,1\}^*$ be a sequence of observation outcomes. By definition, we have

$$\mathcal{E}_{\mathcal{B}}(x) = \sum_{yy_\sharp \in \mathcal{L}'} p_{\mathcal{B}}(yy_\sharp; x\sharp). \tag{4}$$

By the definition of $U'(\sharp)$ and the observable $\mathcal{O}'$, one can see that $y_\sharp \in \{a, b, c, d\}$. Thus, we can rewrite (4) as

$$\mathcal{E}_{\mathcal{B}}(x) = \sum_{ya\in\mathcal{L}'} p_{\mathcal{B}}(ya; x\sharp) + \sum_{yb\in\mathcal{L}'} p_{\mathcal{B}}(yb; x\sharp) + \sum_{yc\in\mathcal{L}'} p_{\mathcal{B}}(yc; x\sharp) + \sum_{yd\in\mathcal{L}'} p_{\mathcal{B}}(yd; x\sharp). \quad (5)$$

Let us focus on the first sum in (5). First of all, by the definition of $\mathcal{L}'$, we observe that $ya \in \mathcal{L}'$ if and only if $0y0 \in \mathcal{L}$. Moreover, by the definition of the 1qfc $\mathcal{B}$, we have

$$\begin{aligned} p_{\mathcal{B}}(ya; x\sharp) &= \left\| \pi' \left(\prod_{i=1}^{n} U'(x_i)P'(y_i) \right) U'(\sharp)P(a) \right\|^2 \\ &= \left\| \pi U(\sigma)P(0) \left(\prod_{i=1}^{n} U(x_i)P(y_i) \right) U(\sharp)P(0) \right\|^2 = p_{\mathcal{A}}(0y0; x\sharp). \end{aligned}$$

This enables us to rewrite the first sum in (5) as $\sum_{0y0\in\mathcal{L}} p_{\mathcal{A}}(0y0; \sigma x\sharp)$. With analogous reasonings, we can replace the other three sums in (5), respectively, with $\sum_{0y1\in\mathcal{L}} p_{\mathcal{A}}(0y1; \sigma x\sharp)$, $\sum_{1y0\in\mathcal{L}} p_{\mathcal{A}}(1y0; \sigma x\sharp)$, $\sum_{1y1\in\mathcal{L}} p_{\mathcal{A}}(1y1; \sigma x\sharp)$. Thus, we obtain

$$\begin{aligned} \mathcal{E}_{\mathcal{B}}(x) = &\sum_{0y0\in\mathcal{L}} p_{\mathcal{A}}(0y0; \sigma x\sharp) + \sum_{0y1\in\mathcal{L}} p_{\mathcal{A}}(0y1; \sigma x\sharp) + \sum_{1y0\in\mathcal{L}} p_{\mathcal{A}}(1y0; \sigma x\sharp) \\ &+ \sum_{1y1\in\mathcal{L}} p_{\mathcal{A}}(1y1; \sigma x\sharp) = \sum_{y_0 y y_\sharp\in\mathcal{L}} p_{\mathcal{A}}(y_0 y y_\sharp; \sigma x\sharp) = \mathcal{E}_{\mathcal{A}}(\sigma x). \end{aligned}$$

If $\mathcal{A}$ presents a general observable $\mathcal{O}$ with outcomes $\{c_1, \ldots, c_s\}$, the construction of $\mathcal{B}$ generalizes as follows. The initial superposition is represented by the $1\times(s+s^2)q$ unit vector $\pi' = (\bigoplus_{i=1}^{s} \pi U(\sigma)P(c_i)) \oplus 0_{1\times s^2 q}$. The evolution on $\gamma \in \Sigma$ is given by the unitary matrix $U'(\gamma) = (\bigoplus_{i=1}^{s} U(\gamma)) \oplus I_{s^2 q}$, while the evolution on $\sharp$ is represented by

$$U'(\sharp) = \begin{pmatrix} 0_{sq} & \oplus_{j=1}^{s} H_1 \\ 0_{sq} & \oplus_{j=1}^{s} H_2 \\ \vdots & \vdots \\ 0_{sq} & \oplus_{j=1}^{s} H_s \\ I_{sq} & 0_{sq\times s^2 q} \end{pmatrix},$$

where, for $1 \le i \le s$, we set $H_i = H(\Pi \otimes I_q)^{i-1}$, with H being the $q \times sq$ block matrix $H = (U(\sharp)P(c_1), U(\sharp)P(c_2), \ldots, U(\sharp)P(c_s))$, and Π the circular permutation matrix on s symbols. The reader may verify that $U'(\sharp)$ is unitary. The observable is $\mathcal{O}' = \sum_{i=1}^{s} c_i P'(c_i) + \sum_{i,j=1}^{s} d_{ij} P'(d_{ij})$, where, for $1 \le i, j \le s$,

$$P'(c_i) = \left(\bigoplus_{h=1}^{s} P(c_i) \right) \oplus 0_{s^2 q}, \;\; P'(d_{ij}) = 0_{sqi+(j-1)q} \oplus I_q \oplus 0_{sq(s-i)+(s-j)q}.$$

The control language $\mathcal{L}' \subseteq \{c_1, \ldots, c_s, d_{11}, \ldots, d_{i,j}, \ldots, d_{ss}\}^*$ is defined as $\mathcal{L}' = \bigcup_{i=1}^{s} \Lambda_{c_i}$, where $\Lambda_{c_i} = \bigcup_{j=1}^{s} c_i^{-1}\mathcal{L}c_j^{-1}d_{ij}$. Every Λ_{c_i} is easily seen to be recognized by a k-state 1dfa, and hence $\mathcal{L}'$ turns out to be recognized by a k^s-state 1dfa. □

Theorem 3.4. *Let $L \subseteq \Sigma^*$ be a language recognized with isolated cut point by a 1qfc $\mathcal{A}$ with q quantum basis states, k classical states and a control language alphabet consisting of $s > 1$ symbols. Then, there exists a 1qfc $\mathcal{B}$ with $(s^2+1)q$ quantum states and k classical states that recognizes $L\sigma^{-1}$ with isolated cut point.*

Proof. The technique is similar to that used in the proof of the previous theorem. We directly give the construction for general observables; its correctness may be easily verified by the reader. Let $\mathcal{A} = (\pi, \{U(\gamma)\}_{\gamma\in\Gamma}, \mathcal{O}, \mathcal{L})$, with $\Gamma = \Sigma \cup \{\sharp\}$, $\mathcal{O} = \sum_{i=1}^{s} c_i P(c_i)$ and $\mathcal{L} \subseteq \{c_1, \ldots, c_s\}^*$, be the 1qfc accepting the language L with isolated cut point. We construct the 1qfc $\mathcal{B} = (\pi', \{U'(\gamma)\}_{\gamma\in\Gamma}, \mathcal{O}', \mathcal{L}')$ inducing the stochastic event $\mathcal{E}_{\mathcal{B}}(x) = \mathcal{E}_{\mathcal{A}}(x\sigma)$, for any $x \in \Sigma^*$, thus recognizing $L\sigma^{-1}$ with isolated cut point. This time, on input x, $\mathcal{B}$ simulates $\mathcal{A}$ on input $x\sigma$. By the choice of a suitable observable, the evolution and observation on σ is taken into account upon evolving and observing on $\sharp$.

The initial state of $\mathcal{B}$ is represented by the $1 \times (s^2+1)q$ unit vector $\pi' = \pi \oplus 0_{1\times s^2 q}$. The evolution on $\gamma \in \Sigma$ is given by the unitary matrix $U'(\gamma) = U(\gamma) \oplus I_{s^2 q}$, while the evolution on $\sharp$ is represented by

$$U'(\sharp) = \begin{pmatrix} 0_q & R_1 \\ 0_q & R_2 \\ \vdots & \vdots \\ 0_q & R_{s^2} \\ I_q & 0_{q\times s^2 q} \end{pmatrix},$$

where, for $1 \leq i \leq s^2$, we set $R_i = R(\Pi \otimes I_q)^{i-1}$, with R being the $q \times s^2 q$ block matrix

$$\begin{aligned} R = {} & (U(\sigma)P(c_1)U(\sharp)P(c_1), U(\sigma)P(c_1)U(\sharp)P(c_2), \ldots, U(\sigma)P(c_1)U(\sharp)P(c_s), \\ & U(\sigma)P(c_2)U(\sharp)P(c_1), U(\sigma)P(c_2)U(\sharp)P(c_2), \ldots, U(\sigma)P(c_2)U(\sharp)P(c_s), \\ & \ldots, \\ & U(\sigma)P(c_s)U(\sharp)P(c_1), U(\sigma)P(c_s)U(\sharp)P(c_2), \ldots, U(\sigma)P(c_s)U(\sharp)P(c_s)), \end{aligned}$$

and Π the circular permutation matrix on s^2 symbols. The reader may verify that $U'(\sharp)$ is unitary. The observable is $\mathcal{O}' = \sum_{i=1}^{s} c_i P'(c_i) + \sum_{i,j=1}^{s} d_{ij} P'(d_{ij})$, where, for $1 \leq i, j \leq s$,

$$P'(c_i) = P(c_i) \oplus 0_{s^2 q}, \;\; P'(d_{ij}) = 0_{sq(i-1)+jq} \oplus I_q \oplus 0_{sq(s-i+1)-jq}.$$

The control language $\mathcal{L}' \subseteq \{c_1, \ldots, c_s, d_{11}, \ldots, d_{ij}, \ldots, d_{ss}\}^*$ is the language $\mathcal{L}' = \bigcup_{i,j=1}^{s} \mathcal{L}(c_i c_j)^{-1} d_{ij}$. Supposing that $\mathcal{L}$ is recognized by the k-state 1dfa

$(K, \{c_1, \ldots, c_s\}, \delta, q_0, F)$, it is not hard to see that $\mathcal{L}'$ can be recognized by the k-state 1dfa $(K, \{c_1, \ldots, c_s, d_{11}, \ldots, d_{ij}, \ldots, d_{ss}\}, \delta', q_0, F)$. The transition function δ' is defined as δ except for transitions leading to a final state in two moves: for any $p \in K$ such that $\delta(p, c_i c_j) = s \in F$, we let $\delta'(p, d_{ij}) = s$. □

Let us now focus on constructing 1qfc's for inverse homomorphic images. Here, we state our construction for homomorphisms from binary alphabets. Yet, for avoiding too many technicalities, we will assume that 1qfc's do not work with the endmarker ♯ at the end of input strings. These assumptions do not substantially influence the generality of our construction.

Theorem 3.5. *Let $L \subseteq \Delta^*$ be a language recognized with isolated cut point by a 1qfc $\mathcal{A}$ with q quantum states, k classical states and a control language alphabet consisting of $s > 1$ symbols. Then, given an homomorphism $\varphi : \{a, b\} \to \Delta^*$ with $m = \max\{|\varphi(a)|, |\varphi(b)|\}$, there exists a 1qfc $\mathcal{B}$ with $2s^m q$ quantum basis states and $2s^m k$ classical states that recognizes $\varphi^{-1}(L)$ with isolated cut point.*

Proof. We begin by exhibiting our construction for an homomorphism $\varphi : \{a, b\} \to \{\alpha, \beta\}^*$ defined as $\varphi(a) = \alpha\beta$ and $\varphi(b) = \beta$, so that $m = 2$. Yet, we start from a language L accepted by a 1qfc with a binary observable. We omit the proof of the correctness of the construction which is quite technical but not hard to provide. Instead, we explain how the resulting 1qfc for $\varphi^{-1}(L)$ works. Then, we sketch the construction for arbitrary homomorphisms and 1qfc's.

We assume that $\mathcal{A} = (\pi, \{U(\gamma)\}_{\gamma \in \Gamma}, \mathcal{O}, \mathcal{L})$, with $\Gamma = \{\alpha, \beta, \sharp\}$, $\mathcal{O} = 0 \cdot P(0) + 1 \cdot P(1)$ and $\mathcal{L} \subseteq \{0, 1\}^*$, is the 1qfc accepting the language L with isolated cut point. We construct the 1qfc $\mathcal{B} = (\pi', \{U'(\sigma)\}_{\sigma \in \{a, b, \sharp\}}, \mathcal{O}', \mathcal{L}')$ inducing the stochastic event $\mathcal{E}_{\mathcal{B}}(x) = \mathcal{E}_{\mathcal{A}}(\varphi(x))$, for any $x \in \{a, b\}^*$, thus recognizing $\varphi^{-1}(L)$ with isolated cut point. Roughly speaking, on reading the input symbol a (b), $\mathcal{B}$ "carries on" the possible computations of $\mathcal{A}$ on input $\alpha\beta$ (β). By the choice of a suitable observable, these computations are correctly taken into account at each evolution step plus observation.

The initial superposition of $\mathcal{B}$ is the $1 \times 8q$ unit vector $\pi' = \pi \oplus 0_{1 \times 7q}$. The evolution is represented by the following $8q \times 8q$ unitary matrices:

$$U'(a) = \begin{pmatrix} U_a & 0_{4q} \\ 0_{4q} & U_a \end{pmatrix} \text{ where } U_a \text{ is the } 4q \times 4q \text{ matrix}$$

$$\begin{pmatrix} U(\alpha)P(0)U(\beta)P(0) & U(\alpha)P(0)U(\beta)P(1) & U(\alpha)P(1)U(\beta)P(0) & U(\alpha)P(1)U(\beta)P(1) \\ U(\alpha)P(0)U(\beta)P(1) & U(\alpha)P(0)U(\beta)P(0) & U(\alpha)P(1)U(\beta)P(1) & U(\alpha)P(1)U(\beta)P(0) \\ U(\alpha)P(1)U(\beta)P(0) & U(\alpha)P(1)U(\beta)P(1) & U(\alpha)P(0)U(\beta)P(0) & U(\alpha)P(0)U(\beta)P(1) \\ U(\alpha)P(1)U(\beta)P(1) & U(\alpha)P(1)U(\beta)P(0) & U(\alpha)P(0)U(\beta)P(1) & U(\alpha)P(0)U(\beta)P(0) \end{pmatrix},$$

$$U'(b) = \begin{pmatrix} 0_{4q} & U_b \\ U_b & 0_{4q} \end{pmatrix} \text{ with } U_b = \begin{pmatrix} U(\beta)P(0) & 0_q & U(\beta)P(1) & 0_q \\ 0_q & U(\beta)P(0) & 0_q & U(\beta)P(1) \\ U(\beta)P(1) & 0_q & U(\beta)P(0) & 0_q \\ 0_q & U(\beta)P(1) & 0_q & U(\beta)P(0) \end{pmatrix}.$$

The observable is $\mathcal{O}' = \sum_{i=1}^{4} c_i \cdot P'(c_i) + \sum_{i=1}^{4} \hat{c}_i \cdot P'(\hat{c}_i)$, where the projectors are as follows, for $1 \leq i \leq 4$:

$$P'(c_i) = 0_{(i-1)q} \oplus I_q \oplus 0_{(8-i)q}$$
$$P'(\hat{c}_i) = 0_{(3+i)q} \oplus I_q \oplus 0_{(4-i)q}.$$

Let us now describe the 1dfa C that recognizes the control language $\mathcal{L}'$, supposing that $\mathcal{L}$ is recognized by the k-state 1dfa $(K, \{0,1\}, \delta, q_0, F)$. We begin by setting $\mathrm{T} = \{c_1, \ldots, c_4\}$, $\hat{\mathrm{T}} = \{\hat{c_1}, \ldots, \hat{c_4}\}$, and letting $K_{\mathrm{T}} = K \times \mathrm{T}$, $K_{\hat{\mathrm{T}}} = K \times \hat{\mathrm{T}}$. Then, we define $C = (K_{\mathrm{T}} \cup K_{\hat{\mathrm{T}}}, \mathrm{T} \cup \hat{\mathrm{T}}, \delta', (q_0, c_1), F' = \{(p, \tau) \mid p \in F\})$. The transition function δ' is defined according these two rules:

(1) if $(p, \tau) \in K_{\mathrm{T}}$ and $x \in \mathrm{T}$ or $(p, \tau) \in K_{\hat{\mathrm{T}}}$ and $x \in \hat{\mathrm{T}}$, then $\delta'((p, \tau), x) = (\delta(p, W(\tau, x)), x)$, where W is the function displayed in the following table:

W	$c_1/\hat{c_1}$	$c_2/\hat{c_2}$	$c_3/\hat{c_3}$	$c_4/\hat{c_4}$
$c_1/\hat{c_1}$	00	01	10	11
$c_2/\hat{c_2}$	01	00	11	10
$c_3/\hat{c_3}$	10	11	00	01
$c_4/\hat{c_4}$	11	10	01	00

(2) if $(p, \tau) \in K_{\mathrm{T}}$ and $x \in \hat{\mathrm{T}}$ or $(p, \tau) \in K_{\hat{\mathrm{T}}}$ and $x \in \mathrm{T}$, then $\delta'((p, \tau), x) = (\delta(p, V(\tau, x)), x)$, where V is the function displayed in the following table ($\perp$ denotes a situation that never occurs):

V	$c_1/\hat{c_1}$	$c_2/\hat{c_2}$	$c_3/\hat{c_3}$	$c_4/\hat{c_4}$
$c_1/\hat{c_1}$	0	$\perp$	1	$\perp$
$c_2/\hat{c_2}$	$\perp$	0	$\perp$	1
$c_3/\hat{c_3}$	1	$\perp$	0	$\perp$
$c_4/\hat{c_4}$	$\perp$	1	$\perp$	0

Let us now explain how the 1qfc $\mathcal{B}$ works on an input in $\{a, b\}^*$.

First of all, at any time during a computation, the state of $\mathcal{B}$ has one of the following form: $(\xi, 0_{1\times 4q})$ or $(0_{1\times 4q}, \xi)$, for some $\xi \in \mathbf{C}^{1\times 4q}$. We swap from a form to the other upon reading b, while reading a does not change the form of the state. The outcomes of observations on states of the first (second) form belong to T ($\hat{\mathrm{T}}$). The initial superposition is $(\pi, 0_{1\times 7q})$, *i.e.*, a state in the first form. Suppose that $\mathcal{B}$ reads a, then the state remains in the first form with $\xi = (\pi U(\alpha)P(0)U(\beta)P(0), \pi U(\alpha)P(0)U(\beta)P(1), \pi U(\alpha)P(1)U(\beta)P(0), \pi U(\alpha)P(1)U(\beta)P(1))$ and the observation leaves $\mathcal{B}$ in one of these four components, say $(0_{1\times 3q}, \eta)$. Thus, the outcome of the observation on $\mathcal{B}$ encompasses two observation outcomes of $\mathcal{A}$ (in this case, 11). Suppose now that $\mathcal{B}$ reads b, then the state assumes the second form with $\xi = (0_{1\times q}, \eta U(\beta)P(1), 0_{1\times q}, \eta U(\beta)P(0))$ and the observation leaves $\mathcal{B}$ in one of the two nonzero components, say $(0_{1\times 3q}, \eta')$. Now, the outcome of the observation on $\mathcal{B}$ represents only one observation outcome of $\mathcal{A}$ (in this case, 0). This fact is correctly taken into account by the alternation between T and $\hat{\mathrm{T}}$ in the symbols

of the string of outcomes, witnessing that b has been read. In fact, let us see how the 1dfa C recognizing the control language $\mathcal{L}'$ works. The states of C consist of two components: the first is a state of the 1dfa recognizing the control language $\mathcal{L}$ of $\mathcal{A}$, and the second is an observation outcome of $\mathcal{B}$. The transition function δ' simulates the computation of the 1dfa for $\mathcal{L}$ in the first component by following two different rules:

(1) two moves are performed according to the function W, for outcomes after reading a;
(2) one move is performed according to the function V, for outcomes after reading b.

Yet, an outcome of $\mathcal{B}$ does not always represents the same sequence of observations of $\mathcal{A}$, but it depends on the position of the nonzero component of the previous state (the one in which $\mathcal{B}$ has collapsed after the previous observation). As we have already seen in the other constructions of 1qfc's, this problem is solved by storing the result of the previous observation in the second component of the states of C.

As a final observation on this example, we notice that the construction may be easily modified in case the homomorphism maps the symbol b into the empty word. In this case, $U_b = I_{4q}$ and the rule (2) for the construction of the 1dfa C recognizing $\mathcal{L}'$ modifies as follows: if $(p, c_i) \in K_{\mathrm{T}}$ and $\hat{c_j} \in \hat{\mathrm{T}}$, then $\delta'((p, c_i), \hat{c_j}) = (p, \hat{c_i})$; if $(p, \hat{c_i}) \in K_{\hat{\mathrm{T}}}$ and $c_j \in \mathrm{T}$, then $\delta'((p, \hat{c_i}), c_j) = (p, c_i)$. Basically, this rule enables C to ignore the observation after reading b.

Let us now generalize our reasoning for a general homomorphism $\varphi : \{a, b\} \to \Delta^*$ defined as $\varphi(a) = x_1 \cdots x_m$ and $\varphi(b) = y_1 \cdots y_r$, with $x_i, y_i \in \Delta$ an $m \geq r > 1$. We notice that the case in which φ maps a symbol into the empty string can be dealt with as before.

We now assume that the 1qfc $\mathcal{A}$ accepting L has a general observable $\mathcal{O} = \sum_{i=1}^{s} i \cdot P(i)$. The 1qfc $\mathcal{B}$ modifies as follows. The initial superposition of $\mathcal{B}$ is the $1 \times 2s^m q$ unit vector $\pi' = \pi \oplus 0_{1\times(2s^m-1)q}$. The evolution is represented by the following $2s^m q \times 2s^m q$ unitary matrices:

$$U'(a) = \begin{pmatrix} U_a & 0_{s^m q} \\ 0_{s^m q} & U_a \end{pmatrix},$$

with U_a built according to the following rule. With each symbol x_i, on which the evolution of $\mathcal{A}$ is $U(x_i)$, we associate the matrix

$$M_i = \bigoplus_{h=1}^{s^{i-1}} \begin{pmatrix} N_1 \\ \vdots \\ N_s \end{pmatrix},$$

where $N_j = N(\Pi \otimes I_{s^{m-i}q})^{j-1}$, with Π the circular permutation matrix on s symbols, and $N = (\oplus_{t=1}^{s^{m-i}} U(x_i)P(1), \ldots, \oplus_{t=1}^{s^{m-i}} U(x_i)P(s))$ a $s^{m-i}q \times s^{m-i+1}q$ block matrix. Then, we let $U_a = \prod_{i=1}^{m} M_i$. In a similar way, we define the

evolution on b:

$$U'(b) = \begin{pmatrix} 0_{s^m q} & U_b \\ U_b & 0_{s^m q} \end{pmatrix},$$

with $U_b = \prod_{i=1}^{r} S_i$ and S_i built as M_i by substituting $U(x_i)$ with $U(y_i)$.

The observable is $\mathcal{O}' = \sum_{i=1}^{s^m} i \cdot P'(i) + \sum_{i=1}^{s^m} \hat{\imath} \cdot P'(\hat{\imath})$, where the projectors are an easy generalization of those presented before. Moreover, the reader may easily argue that the definition of the 1dfa C accepting $\mathcal{L}'$ is analogous to that above shown, except for the tables V and W used in the definition of the transition function. These two tables must now be derived from the definition of U_a and U_b, respectively. For instance, given two observation $\tau, x \in \{1, \ldots, s^m\}$, $W(\tau, x)$ is now the word consisting of the outcomes of the observations in the (τ, x)th entry of U_a. □

3.2. 1QFC'S *VS.* CLASSICAL AUTOMATA

In Section 2, we have provided an algorithm to construct an 1qfc $\mathcal{A}$ recognizing a given regular language L with only 3 quantum basis states and (due to the generality of our construction) three times the number of states of the 1dfa for L for the classical component. Hence, the size of the classical component of $\mathcal{A}$ turns out to be quite expensive. However, this must be considered together with the fact that $\mathcal{A}$ recognizes L deterministically.

Here, we show instances of regular languages for which we are able to consistently decrease the size of the classical component — which becomes less than the size of minimal 1dfa's for these languages — paying by a certain error probability of acceptance.

We define a family of binary regular languages as follows. Any $x \in \{0,1\}^*$ can be clearly written as $x = 1^{s_1}01^{s_2}0\cdots01^{s_t}$, for suitable $t \geq 0$ and $s_i \geq 0$. For a fixed $m > 0$, we define a function $\phi_m(x) = \phi_m(1^{s_1}01^{s_2}0\cdots01^{s_t}) = |\{i \in \{1, \ldots, t-1\} : s_i \bmod m \neq 0\}|$. In other words, $\phi_m(1^{s_1}01^{s_2}0\cdots1^{s_t})$ returns the number of blocks of consecutive 1's ending with a 0, whose length is not a multiple of m. We call these blocks *bad*, *good* otherwise. Thus, for $m > 0$, $h \geq 0$, our family of languages is defined as

$$L_{m,h} = \{x \in \{0,1\}^* \mid \phi_m(x) \leq h\},$$

i.e., $L_{m,h}$ consists of words containing no more than h bad blocks ended by 0.

It is not hard to see that $L_{m,h}$ can be recognized by a 1dfa consisting of a sequence of $h+1$ disjoint deterministic cycles each one containing m states, plus a trap state at the end. Each cycle counts the length of blocks of consecutive 1's modulo m. We start from the first cycle and, each time we discover a bad block, we jump into the next cycle upon reading 0. So, if we reach the ith cycle, we have already found $i-1$ bad blocks. If we never reach the last cycle we accept. Otherwise, if we have reached the last cycle, there are two possibilities. Either we never find a bad block ending with 0 and hence we accept (the possible last 1's not ending with 0 are simply ignored), or we find the $(h+1)$th bad block ending

with 0 and then we reject by entering the trap state upon reading 0. It is not hard to prove that this $(m(h+1)+1)$-state 1dfa is minimal for $L_{m,h}$.

We are now going to construct a succinct 1qfc recognizing $L_{m,h}$ with isolated cut point *where the number of quantum basis states is a constant, and the number of classical states is $O(h)$ and does not depend on m.*

To this regard, we first slightly modify $L_{m,h}$ by introducing the homomorphism $\varphi : \{0,1\}^* \to \{a,b,\$\}^*$ such that $\varphi(1) = a$ and $\varphi(0) = b\$$, and letting $L^{\$}_{m,h} = \varphi(L_{m,h})$.

We say that a string in $\{a,b,\$\}^*$ is well formed if, whenever b ($\$$) appears in the string, the next (previous) symbol must be $\$$ (b). For instance, $b\$$ is well formed, while b or $\$$ are not well formed. We now design a 1qfc $\mathcal{A}$ that accepts $L^{\$}_{m,h}$ *whenever input strings are restricted to be well formed.* Next, by applying the construction in Theorem 3.5, we will obtain a 1qfc for $\varphi^{-1}(L^{\$}_{m,h}) = L_{m,h}$ which simulates $\mathcal{A}$ working on well formed inputs.

The quantum component of our 1qfc is basically a 2-state measure-once automaton described in [19] for accepting the unary language $L_m = \{a^n \mid n \bmod m = 0\}$:

$$M = \left((1,0), U = \begin{pmatrix} \cos\frac{\pi}{m} & i\sin\frac{\pi}{m} \\ i\sin\frac{\pi}{m} & \cos\frac{\pi}{m} \end{pmatrix}, P_e = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \right).$$

This automaton processes the whole string and it is observed only once, at the end of computation. The observable has only two outcomes: e for *accept* (with projector P_e) and r for *reject* (with projector $P_r = I_2 - P_e$). One can easily verify that the probability that M accepts the string a^n amounts to $p_M(a^n) = ||(1,0)U^nP_e||^2 = \cos^2\left(\frac{\pi n}{m}\right)$. Hence, M accepts with certainty the strings whose length is multiple of m, while the acceptance probability for the other strings is bounded above by $\rho = p_M(a) < 1$ (maximum error probability). We can also state that M recognizes L_m with cut point $(1+\rho)/2$ isolated by $(1-\rho)/2$.

Notice that the same event p_M could be realized by M also by having $(0,1)$ as initial superposition and $P_r = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$ as projector. Clearly, in this case, the role of outcomes e and r are swapped: e stands for *reject* and r for *accept*.

Theorem 3.6. *There exists a 1qfc $\mathcal{A}$ with 4 quantum basis states and $h+2$ classical states recognizing $L^{\$}_{m,h}$ with isolated cut point when inputs are restricted to be well formed strings.*

Proof. Let us provide the 1qfc $\mathcal{A} = (\pi, \{U(\gamma)\}_{\gamma\in\{a,b,\$,\sharp\}}, \mathcal{O}, \mathcal{L})$. The initial superposition is the 1×4 unit vector $\pi = (1,0,0,0)$. The evolution on $\{a,b,\$,\sharp\}$ are represented by the following 4×4 unitary matrices:

$$U(a) = U \oplus I_2, \text{ where } U \text{ is the evolution of the measure-once automaton } M,$$

$$U(b) = U(\$) = \begin{pmatrix} 0_2 & I_2 \\ I_2 & 0_2 \end{pmatrix},\ U(\sharp) = I_4.$$

The observable is $\mathcal{O} = g \cdot P(g) + e \cdot P(e) + r \cdot P(r)$, where the projectors are as follows:

$$P(g) = I_2 \oplus 0_2,\ P(e) = 0_2 \oplus P_e,\ P(r) = 0_2 \oplus P_r,$$

where P_e, P_r are the projectors of the measure-once automaton M.

The control language is defined as $\mathcal{L} = \cup_{j=0}^{h} E_j$, where E_j's enjoy the following recursive definition: $E_0 = \{e, g\}^*$, $E_{2i+1} = E_{2i} r \{r, g\}^*$, $E_{2i} = E_{2i-1} e \{e, g\}^*$. A 1dfa for $\mathcal{L}$ has $h + 1$ accepting states $\{q_0, \ldots, q_h\}$ and a trap state q_{h+1}. These states are organized into a path with the trap state at the end. Each state has a loop on the symbol g. Except for the trap state, each q_{2i} (q_{2i+1}) has also a loop on e (r) and a transition labeled r (e) leading to q_{2i+1} (q_{2i+2}). The trap state has a loop also on the symbols e, r.

Let us see how the 1qfc $\mathcal{A}$ works on a well formed input $a^{s_1} b \$ a^{s_2} b \$ \cdots a^{s_t} \{b\$\sharp, \sharp\}$. The idea is that $\mathcal{A}$ on input $a^{s_i} b$ simulates M on input a^{s_i}. The scanning of the symbol \$ restarts $\mathcal{A}$ on the next block.

First of all, at any time during a computation, the state of $\mathcal{A}$ has one of the following form: $(\xi, 0, 0)$ or $(0, 0, \xi)$, for some $\xi \in \mathbf{C}^{1\times 2}$. The first form is assumed as long as the symbol a is read. The observation in this phase always yields the outcome g without modifying the quantum state. Hence, after processing the first block a^{s_1}, $\mathcal{A}$ reaches the state $((1, 0)U^{s_1}, 0, 0)$. When reading b, the quantum states assumes the second form $(0, 0, (1, 0)U^{s_1})$. The observation makes $\mathcal{A}$ collapsing to $(0, 0, 1, 0)$ with probability $||(1, 0)U^{s_1} P_e||^2$ giving the outcome e. This happens when a^{s_1} is a good block. Otherwise, $\mathcal{A}$ collapses to $(0, 0, 0, 1)$ with probability $||(1, 0)U^{s_1} P_r||^2$ giving to the outcome r. This happens when a^{s_1} is a bad block. Then, $\mathcal{A}$ reads \$, the state becomes $(1, 0, 0, 0)$ or $(0, 1, 0, 0)$, and the observation gives g. The computation on the next block then starts. If a^{s_2} is a good block the outcome after reading b is e in case $\mathcal{A}$ started from $(1, 0, 0, 0)$, r in case $\mathcal{A}$ started from $(0, 1, 0, 0)$ (*i.e.*, the meaning of observation results are swapped in this latter case). If $\mathcal{A}$ started from $(0, 1, 0, 0)$ and a^{s_2} is bad, than e and r swap their role again, reassuming their original meanings. Thus, the reader may easily verify that the number of bad blocks in the input string is given by the number of times e and r alternate in the string of observation outcomes. Yet, it is easy to see that the control language $\mathcal{L}$ consists exactly of those strings in $\{g, e, r\}$ where the symbols e and r alternates at most h times.

Let us now evaluate $p_{\mathcal{A}}(y, x\sharp)$, for any well formed $x \in \{a, b, \$\}^*$ and $y \in \{g, e, r\}^*$. We observe that each good block of x is correctly classified with certainty while bad blocks can be wrongly classified as good with probability not exceeding ρ, as pointed out in the description of the measure-once automaton M. It is clear that if $x \in L^{\$}_{m,h}$, then $\mathcal{A}$ accepts x with certainty since classifying bad blocks as good leaves the number of bad blocks less than or equal to h. On the contrary, an error may occur on $x \notin L^{\$}_{m,h}$ since the number of wrong block classifications could reduce the number of bad blocks thus leading $\mathcal{A}$ to accept. As an example, suppose that x has $h + z$ bad blocks, with $z > 0$. It is not hard to verify that the

probability that $\mathcal{A}$ wrongly accepts x is bounded above by

$$1 - \sum_{i=0}^{z-1} \binom{h+z}{i} \rho^i (1-\rho)^{h+z-i}.$$

The maximum value of this probability is given by $z = 1$, *i.e.*, $1 - (1-\rho)^{h+1}$.

In conclusion $\mathcal{A}$, restricted on well formed input, recognizes $L^{\$}_{m,h}$ with cut point $1 - \frac{(1-\rho)^{h+1}}{2}$ isolated by $\frac{(1-\rho)^{h+1}}{2}$. $\mathcal{A}$ has 4 quantum basis states and $h+2$ classical states. □

In conclusion, by applying Theorem 3.5 on $L^{\$}_{m,h}$, we get

Corollary 3.7. *There exists a 1qfc with a constant number of quantum basis states and $O(h)$ classical states recognizing $L_{m,h}$ with isolated cut point.*

As a final observation, we point out that the quantum component we used in our 1qfc $\mathcal{A}$ — the measure-once automaton M — is very small but not very accurate on the error probability which approaches to 1 for m becoming large. However, Bertoni *et al.* provide a $O(1/\varepsilon^3 \log m)$-state measure-once automaton that accepts the words in L_m with certainty, and the others with arbitrary small probability ε [7, 8]. This latter automaton can replace M in our construction to enhance acceptance precision, paying by only $O(\log m)$ quantum basis states plus $O(h)$ classical states. This should be compared with the size of minimal 1dfa's for $L_{m,h}$ which we recalled to be $O(mh)$.

As future investigations, it could be interesting to replace M with a measure-once automaton recognizing a given unary periodic language with isolated cut point (see, *e.g.*, [6, 18]) or a given binary group language (see, *e.g.*, [4]). Thus, our construction (or a suitable generalization) could be useful for producing succinct 1qfc's for other families of languages.

References

[1] A. Ambainis, M. Beaudry, M. Golovkins, A. Kikusts, M. Mercer and D. Thérien, Algebraic Results on Quantum Automata. *Theory Comput. Syst.* **39** (2006) 165–188.

[2] A. Ambainis and R. Freivalds, 1-way Quantum Finite Automata: Strengths, Weaknesses and Generalizations, in *Proc. 39th Annual Symposium on Foundations of Computer Science*. IEEE Computer Society Press (1998) 332–342.

[3] A. Ambainis, A. Kikusts and M. Valdats, On the class of languages recognizable by 1–way quantum finite automata, in *Proc. 18th Annual Symposium on Theoretical Aspects of Computer Science. Lect. Notes Comput. Sci.* **2010** (2001) 305–316.

[4] A. Bertoni and M. Carpentieri, Regular languages accepted by quantum automata. *Inform. Comput.* **165** (2001) 174–182.

[5] A. Bertoni, C. Mereghetti and B. Palano, Quantum computing: 1-way quantum automata, in *Proc. 7th International Conference on Developments in Language Theory. Lect. Notes Comput. Sci.* **2710** (2003) 1–20.

[6] A. Bertoni, C. Mereghetti and B. Palano, Golomb Rulers and Difference Sets for Succinct Quantum Automata. *Int. J. Found. Comp. Sci.* **14** (2003) 871–888.

[7] A. Bertoni, C. Mereghetti and B. Palano, Small size quantum automata recognizing some regular languages. *Theoret. Comput. Sci.* **340** (2005) 394–407.
[8] A. Bertoni, C. Mereghetti and B. Palano, Some formal tools for analyzing quantum automata. *Theoret. Comput. Sci.* **356** (2006) 14–25.
[9] A. Brodsky and N. Pippenger, Characterizations of 1-Way Quantum Finite Automata. *SIAM J. Comput.* **5** (2002) 1456–1478.
[10] M. Golovkins and M. Kravtsev, Probabilistic Reversible Automata and Quantum Automata, in *Proc. 8th International Computing and Combinatorics Conference. Lect. Notes Comput. Sci.* **2387** (2002) 574–583.
[11] J. Gruska, *Quantum Computing.* McGraw-Hill (1999).
[12] J. Gruska, Descriptional complexity issues in quantum computing. *J. Automata, Languages and Combinatorics* **5** (2000) 191–218.
[13] J. Hopcroft and J. Ullman, *Formal Languages and Their Relation to Automata.* Addison-Wesley (1969).
[14] A. Kondacs and J. Watrous, On the Power of Quantum Finite State Automata, in *Proc. 38th Annual Symposium on Foundations of Computer Science.* IEEE Computer Society Press (1997) 66–75.
[15] C. Moore and J. Crutchfield, Quantum automata and quantum grammars. *Theoret. Comput. Sci.* **237** (2000) 275–306.
[16] M. Marcus and H. Minc, *Introduction to Linear Algebra.* The Macmillan Company (1965). Reprinted by Dover (1988).
[17] M. Marcus and H. Minc, *A Survey of Matrix Theory and Matrix Inequalities.* Prindle, Weber & Schmidt (1964). Reprinted by Dover (1992).
[18] C. Mereghetti and B. Palano, On the Size of One-way Quantum Finite Automata with periodic behaviors. *RAIRO-Inf. Theor. Appl.* **36** (2002) 277–291.
[19] C. Mereghetti, B. Palano and G. Pighizzini, Note on the Succinctness of Deterministic, Nondeterministic, Probabilistic and Quantum Finite Automata. *RAIRO-Inf. Theor. Appl.* **35** (2001) 477–490.
[20] A. Nayak, Optimal lower bounds for quantum automata and random access codes, in *Proc. 40th Symposium on Foundations of Computer Science* (1999) 369–376.
[21] J.E. Pin, On Languages Accepted by finite reversible automata, in *Proc. 14th Int. Coll. Automata, Languages and Programming. Lect. Notes Comput. Sci.* **267** (1987) 237–249.