applied sciences

IMPACT FACTOR 2.7

CITESCORE 4.5

Article

Unary Quantum Finite State Automata with Control Language

Carlo Mereghetti, Beatrice Palano and Priscilla Raucci

MDPI

https://doi.org/10.3390/app14041490

applied sciences

MDPI

Article

Unary Quantum Finite State Automata with Control Language

Carlo Mereghetti *, **Beatrice Palano** and **Priscilla Raucci**

Dipartimento di Informatica "Giovanni Degli Antoni", Università degli Studi di Milano, Via Celoria 18, 20133 Milano, Italy; beatrice.palano@unimi.it (B.P.); priscilla.raucci@unimi.it (P.R.)
* Correspondence: carlo.mereghetti@unimi.it

Abstract: We study quantum finite automata with control language (QFCs), a theoretical model for finite memory hybrid systems coupling a classical computational framework with a quantum component. We constructively show how to simulate measure-once, measure-many, reversible, and Latvian QFAs by QFCs, emphasizing the size cost of such simulations. Next, we prove the decidability of testing the periodicity of the stochastic event induced by a given QFC. Thanks to our QFA simulations, we can extend such a decidability result to measure-once, measure-many, reversible, and Latvian QFAs as well. Finally, we focus on comparing the size efficiency of quantum and classical finite state automata on unary regular language recognition. We show that unary regular languages can be recognized by isolated cut point QFCs for which the size is generally quadratically smaller than the size of equivalent DFAs.

Keywords: quantum finite state automata; periodic stochastic events; unary regular languages

Citation: Mereghetti, C.; Palano, B.; Raucci, P. Unary Quantum Finite State Automata with Control Language. *Appl. Sci.* **2024**, *14*, 1490. https://doi.org/10.3390/app14041490

Academic Editor: Yurii K. Gun'ko

Received: 26 November 2023
Revised: 7 February 2024
Accepted: 9 February 2024
Published: 12 February 2024

Correction Statement: This article has been republished with a minor change. The change does not affect the scientific content of the article and further details are available within the backmatter of the website version of this article.

1. Introduction

Several impressive theoretical results, Shor and Grover's algorithms [1,2], above all, address the potential of the quantum over classical computational paradigm and motivate great efforts towards actually building quantum computational devices. From this latter viewpoint, especially in the realms of photonics (see, e.g., [3–9]) and condensed matter physics (see, e.g., [10–15]), relevant steps have been made to engineer basic quantum components such as qubits, quantum gates, and quantum communication devices.

From a physical realization point of view, *hybrid* computational architectures featuring "small" and dedicated quantum components hardwired into and cooperating with a classical computational environment have proved to be quite promising. In order to precisely assess their computational power and emphasize various advantages of adopting quantum hardware, several theoretical models for hybrid architectures with *finite* (constant, not depending on the input length) memory have been proposed in the literature. Among others, we recall *quantum finite automata* (QFAs) with open time evolution [16], QFAs with quantum and classical states [17–20], and semi-quantum finite state automata [21–23].

In this paper, we focus on the hybrid model of QFAs *with control language (*QFCs*)* introduced in [24,25]. From an architectural point of view, a QFC A consists of:

- A *quantum* component, represented by a *pure* QFA M that can be seen as the quantum counterpart of a classical deterministic finite state automaton (DFA); this component is "controlled" by
- A *classical* component, which is a DFA D.

The aim of A is to process strings on a given input alphabet Σ. For each input string in Σ^*, A returns an "accept" outcome with a certain probability. The language recognized by A is the set of those strings in Σ^* for which the acceptance probability exceeds a fixed threshold. Let us overview the behavior of A in more detail. At any time during the computation of A on a given input string $x \in \Sigma^*$, the state of the quantum component M is represented by a *superposition* of basis states (formally, a norm 1 vector in the Hilbert space spanned by the basis states of the QFA M). The computation step of A consists of the following two phases:

- *Evolution*: the currently scanned symbol of x is consumed; such a symbol determines a unitary operator (formally, a unitary matrix mapping norm 1 vectors into norm 1 vectors) to be applied to the current superposition of M, yielding a new superposition upon which
- *Observation*: a fixed observable is measured, yielding an outcome with a certain probability; such an outcome, represented by a symbol, is passed to and processed by the classical component, i.e., the DFA D.

The observation makes the superposition of M collapse to a superposition (which, in general, does not have norm 1) coherent with the observed outcome. The normalization of the collapsed superposition is clearly a norm 1 vector and is the superposition A starts from in order to process the next input symbol.

Thus, the computation of A on the input string x produces a sequence y of observable outcomes with a certain probability. We say that the computation is *accepting* whenever y is accepted by D, or equivalently, whenever y belongs to the language recognized by D, called the *control language*. The probability that A accepts x is the probability $\mathscr{E}_A(x)$ for A to display an accepting computation on x, and we call $\mathscr{E}_A : \Sigma^* \to [0,1]$ the stochastic event induced by A. Fixing a cut point $\lambda \in [0,1]$, the *language recognized by A with cut point λ* is defined to be the set of strings $L_{A,\lambda} = \{x \in \Sigma^* \mid \mathscr{E}_A(x) > \lambda\}$. Particularly interesting for both theoretical and practical reasons (see [26,27]) is the case in which λ is *isolated*, i.e., whenever $\varrho \in (0, \frac{1}{2}]$ exists, such that $|\mathscr{E}_A(x) - \lambda| \geq \varrho$ holds true for every $x \in \Sigma^*$.

Concerning the computational capabilities, we have that isolated cut point QFCs reach the same computational power as classical finite memory devices (e.g., DFAs): the class of languages recognized by isolated cut point QFCs coincides with the class of regular languages [24,25]. This computational equivalence (holding for the other above-quoted hybrid models as well) is in sharp contrast with what happens when pure quantum finite state devices are considered. In fact, several models of pure QFAs have been defined, motivated by different possible physical implementations. Thus, e.g., we have: measure-once and measure-many QFAs [28–31], enhanced QFAs [32], reversible QFAs [24], and Latvian QFAs [33,34]. For all these (and other) isolated cut point models, we have a recognition power that strictly lays within the class of regular languages [24,33,34]. This computational weakness is basically due to the fact that, within a finite amount of memory, it is generally impossible to guarantee computation reversibility, which is a fundamental trait of quantum models.

However, it is important to notice that QFAs may exhibit a higher *descriptional* power, i.e., their sizes can be significantly smaller than the sizes of equivalent classical devices (see, e.g., [28,35–39]). This fact makes QFAs very well worth considering as quantum components in a hybrid architecture such as QFCs: on the one hand, we can benefit from their higher descriptional power guaranteeing more hardware size efficient systems; on the other hand, the classical environment within which they are embedded can compensate for their lower computational power (see, e.g., [20,22,40]). In addition, from a physical realization viewpoint, it is worth remarking that QFAs turn out to be quite promising (see, e.g., [6,9]) due to their architectural "simplicity".

Besides addressing and suggesting practical issues in quantum technology realization, QFCs are also of theoretical interest since they may represent a unifying theoretical framework within which to investigate many types of finite memory quantum devices. In fact, several models of QFAs can be suitably simulated by QFCs without significantly increasing the size of simulated machines. As a consequence, some computational and descriptional properties shown for simulating QFCs hold true for simulated types of QFAs as well. This QFC capability of providing a unifying investigation framework has been variously and fruitfully exploited in the literature:

- As recalled above, in [24,25], it is proved that *(i)* isolated cut point QFCs recognize all and only regular languages and that *(ii)* several QFA models previously introduced in the literature can be simulated by QFCs. These two facts together imply that the recognition power of all the simulated isolated cut point QFA models cannot go beyond regular languages. This latter limitation was previously proved in the literature model-by-model using ad hoc techniques.
- In [41], conditions are established for a pair of QFCs to be equivalent with respect to induced stochastic events. In addition, efficient algorithms are provided for checking such equivalence conditions. Even in this case, QFC simulation capabilities enable to suitably transfer results to other QFA models at once.
- In the same spirit of the two lines of investigations quoted above, state lower bounds are provided in [21,36] for several isolated cut point QFA models owing to the simulation capabilities of QFCs.

More generally, we feel that the QFC model has partially inspired the introduction in the literature of several paradigms (see, e.g., [42–45]) aiming to provide both theoretical models for physically plausible hybrid quantum/classical architectures and general frameworks to uniformly cope with different types of computational devices.

1.1. Main Contributions

In this paper, we start from this simulation ability. We construct size efficient QFCs that simulate measure-once, measure-many, reversible, and Latvian QFAs having q basis states. In particular, the simulating QFCs feature at most $2q$ quantum basis states for the first three models and $2q^2$ basis state for the last model. In addition, the control languages turn out to be very simple and require at most 3 states for the first three models, and q states for the last model to be recognized by the classical control unit (DFA component).

Next, we study the decidability of the problem PERIODICITY: given a QFC A working on a unary (i.e., with a single-letter) alphabet $\{\sigma\}$ and an integer $d \geq 0$, decide whether or not the stochastic event $\mathscr{E}_A : \sigma^* \to [0,1]$ is d-periodic (i.e., whether $\mathscr{E}_A(\sigma^k) = \mathscr{E}_A(\sigma^{k+d})$ holds true for any $k \geq 0$). By using a suitable linear representation of QFCs and generating function arguments, we prove the decidability of PERIODICITY. This, together with the constructive QFA simulation results quoted above, shows that PERIODICITY is decidable on measure-once, measure-many, reversible, and Latvian QFAs as well.

It is important to remark that the ability of inducing periodic stochastic events by using a very restricted amount of basis states is at the core of the construction of extremely succinct QFAs for language recognition (see, e.g., [28,35–39]). Therefore, from this point of view, our algorithm to decide PERIODICITY may represent a relevant diagnostic tool in the actual project of size efficient QFAs.

Finally, we show that isolated cut point QFCs are generally more size efficient than equivalent DFAs when recognizing *unary regular languages*. It is well known (see, e.g., [46,47]) that any given unary regular language $L \subseteq \sigma^*$ is recognized by a minimal DFA whose state diagram consists of an initial path of T states joined to a simple cycle of P states for suitable integers $T \geq 0$ and $P > 0$; accepting states are suitably located on both the initial path and the cycle. This clearly enables us to view L as the disjoint union of the finite language L_T consisting of the strings in L of length less than T with the ultimately P-periodic language L_P of the strings in L of length greater than or equal to T.

Following this view, we design a QFC in which, very roughly speaking, the classical component takes care of L_T while its quantum component accounts for membership in L_P. In particular, this latter component can fully exploit the ability of QFAs to induce periodic stochastic events with a very restricted amount of quantum basis states. The resulting QFC recognizes L with an isolated cut point, uses a DFA with $T + 3$ states for its control language, and features a quantum component with only $O(\sqrt{P})$ states. The global amount of $T + O(\sqrt{P})$ states is to be compared with the $T + P$ states required for equivalent DFAs.

We feel it interesting to point out that our construction scheme for unary QFCs is fully modular and naturally allows a "plug-and-play" approach by which more size efficient classical or quantum components can be easily hardwired in order to obtain even more size efficient QFCs for certain language families.

1.2. Paper Organization

The paper is organized as follows. In Section 2, we begin by quickly overviewing the classical model of a DFA and providing its linear representation. We focus on the class of unary regular languages and on the form of unary DFAs. Next, we recall those basics in linear algebra, which are useful to describe quantum systems. Several models of QFAs are then presented together with their language recognition and descriptional power under the isolated cut point acceptance mode. Finally, we expand on the model of a QFC we are mainly interested in, addressing its computational and descriptional power.

In Section 3, we construct QFCs simulating measure-once, measure-many, reversible, and Latvian QFAs, emphasizing the cost in terms of quantum and classical states of these simulations.

In Section 4, we define the problem PERIODICITY of deciding the periodicity of the stochastic event induced by a given QFC and prove its decidability. This result, together with the QFA simulations presented in Section 3, extends the decidability of PERIODICITY to the simulated QFA models.

In Section 5, we build size efficient isolated cut point QFCs for unary regular languages. We propose a modular construction scheme for such QFCs, within which different types of quantum components can be easily inserted and coupled with the classical control unit. To show the versatility of this modular framework, we construct two versions of isolated cut point QFCs for unary regular languages differing from the adopted quantum component. In particular, the resulting second version proves that on unary regular languages, isolated cut point QFCs generally use a number of quantum and classical states that is quadratically smaller than the number of states of equivalent DFAs.

Finally, in Section 6, we draw some concluding remarks and research outlooks.

2. Preliminaries

2.1. Formal Languages and Classical Automata

For the basics of formal language theory, we refer the reader to, e.g., [48]. The set of natural numbers, including zero, is denoted by $\mathbb{N}$. Given a finite alphabet Σ, we let Σ^* denote the set of strings (or words) on Σ, including the empty string ε. For a string $w \in \Sigma^*$, we let $|w|$ denote its length, $|w|_\sigma$ the number of occurrences of the symbol $\sigma \in \Sigma$ in w, and w_i its ith symbol. The set of all strings of length $k \in \mathbb{N}$ on Σ is denoted by Σ^k, with $\Sigma^0 = \{\varepsilon\}$. We let $\Sigma^{<k} = \bigcup_{j=0}^{k-1} \Sigma^j$ and $\Sigma^{\geq k} = \Sigma^* \setminus \Sigma^{<k}$. For a symbol $\sigma \in \Sigma$ and $k \in \mathbb{N}$, we let σ^k denote the string consisting of k consecutive copies of the symbol σ, with σ^0 being ε; we let $\sigma^* = \bigcup_{k \geq 0} \sigma^k$, we let $\sigma^{<k}$ denote the set of strings $\{\varepsilon, \sigma, \sigma^2, \ldots, \sigma^{k-1}\}$, and we let $\sigma^{\geq k} = \sigma^* \setminus \sigma^{<k}$. A language on the alphabet Σ is any subset $L \subseteq \Sigma^*$.

A *deterministic finite state automaton* (DFA) is a 5-tuple $D = (Q, \Sigma, q_{\mathrm{I}}, \delta, F)$, where Q is the finite set of states, Σ the finite input alphabet, $q_{\mathrm{I}} \in Q$ is the initial state, $F \subseteq Q$ is the set of accepting states, and $\delta : Q \times \Sigma \to Q$ is the transition function. By letting $\delta^* : Q \times \Sigma^* \to Q$ be the canonical extension of δ to Σ^* as

$$\delta^*(q, w) = \begin{cases} q & \text{if } w = \varepsilon \\ \delta(\delta^*(q, x), \sigma) & \text{if } w = x\sigma, \text{ for } x \in \Sigma^* \text{ and } \sigma \in \Sigma, \end{cases}$$

we have that D accepts an input string $w \in \Sigma^*$ if and only $\delta^*(q_{\mathrm{I}}, w) \in F$, i.e., whenever the computation of D on w ends in an accepting state. The language recognized by D is the set of strings $L_D = \{w \in \Sigma^* | \delta^*(q_{\mathrm{I}}, w) \in F\}$.

A convenient representation of a DFA is by means of its *state diagram*. The state diagram of our DFA D is the labeled digraph with the state set Q as the set of vertexes and for which an arc labeled $\sigma \in \Sigma$ joins the vertex p to the vertex q if and only if $\delta(p, \sigma) = q$. Therefore, the computation of D on input $x \in \Sigma^*$ can be tracked down in the state diagram by following the path labeled x from the vertex corresponding to the initial state of D. Such a path ends up in a final state if and only if D accepts x. Figure 1 displays an example of a DFA state diagram.

Another useful description of finite state automata is by their *linear representation*. Let us explain such a representation by considering the DFA D. Suppose $Q = \{q_1, \ldots, q_m\}$ is its finite set of states. We represent each state $q_i \in Q$ by its characteristic m-dimensional boolean row vector e_i, which has 1 as the ith component and 0 elsewhere. Instead, the transition function $\delta : Q \times \Sigma \to Q$ is represented by the family $\{M(\sigma)\}_{\sigma\in\Sigma}$ of $m \times m$ boolean matrices for which for any $\sigma \in \Sigma$, the (i, j)th component of $M(\sigma)$ is 1 if and only if $\delta(q_i, \sigma) = q_j$. Notice that, D being deterministic, each matrix $M(\sigma)$ has exactly one 1 per each row. The set $F \subseteq Q$ of accepting states is represented by its characteristic m-dimensional boolean column vector β having 1 at the ith component if and only if $q_i \in F$. So the linear representation of the DFA D is the triple $D = (\alpha, \{M(\sigma)\}_{\sigma\in\Sigma}, \beta)$, where $\alpha = \mathrm{e}_t$ such that q_t is the initial state (in other words, α is the characteristic vector of the initial state).

The computation of $D = (\alpha, \{M(\sigma)\}_{\sigma\in\Sigma}, \beta)$ on the input string $w \in \Sigma^*$ starts from the initial state represented by α, and after consuming the whole input from left to right one symbol per step, it reaches the state in Q represented by the characteristic vector

$$\alpha \cdot M(w_1) \cdot M(w_2) \cdot \cdots \cdot M(w_{|w|}) = \alpha \cdot \prod_{i=1}^{|w|} M(w_i). \tag{1}$$

(The operator $\cdot$ in Equation (1) is the usual row–column product between vectors and matrices or between matrices; more details on linear algebra will be recalled in the next section). Therefore, D accepts w if and only if $\alpha \cdot (\prod_{i=1}^{|w|} M(w_i)) \cdot \beta = 1$, and the language recognized by D can now be written as $L_D = \{w \in \Sigma^* \mid \alpha \cdot (\prod_{i=1}^{|w|} M(w_i)) \cdot \beta = 1\}$.

Concerning the language recognition capabilities, a seminal result in formal language theory states that the class of languages recognized by DFAs coincides with the class of *regular languages*. Indeed, e.g., in [48], the reader may find other relevant characterizations of the regular languages by machines, grammars, and other formal tools.

A *unary language* is any language built over a *single-letter* alphabet. In what follows, we will be using the single-letter alphabet $\Sigma = \{\sigma\}$, so that a unary language will be any set $L \subseteq \sigma^*$. *Unary regular languages* form *ultimately periodic* sets:

Theorem 1 ([47])**.** *For any given unary regular language $L \subseteq \sigma^*$, two integers $T \geq 0$ and $P > 0$ exist such that for any $k \geq T$, we have $\sigma^k \in L$ if and only if $\sigma^{k+P} \in L$.*

According to Theorem 1, it is easy to see that the unary regular language L can be recognized by a minimal DFA whose state diagram, depicted in Figure 1, consists of an initial path of T states joined to a cycle of P states; accepting states are suitably settled on both the path and the cycle.

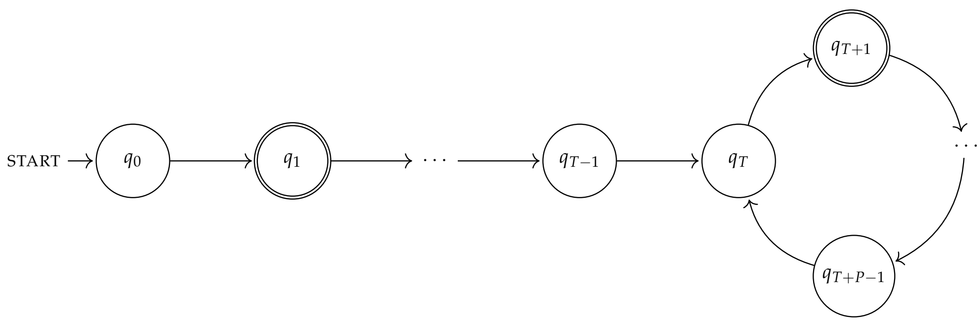

Figure 1. The state diagram of a DFA for a unary regular language. The leftmost vertex with an incoming edge from the label START is the initial state, while double-circled vertexes correspond to the accepting states.

Those unary regular languages satisfying Theorem 1 with $T = 0$ are called *P-periodic languages*. Thus, for P-periodic languages, it is easy to see that the state diagram in Figure 1 reduces to a simple cycle of P states.

2.2. Linear Algebra

As seen in the previous section, few elementary concepts from linear algebra may suitably describe the computation of a classical device such as a DFA. When it comes to the quantum world, additional and deeper notions of linear algebra are needed to properly describe and analyze quantum computational devices. So we are now going to quickly recall such notions, and we refer the reader to, e.g., [49–52], for in-depth presentations.

The fields of real and complex numbers are, respectively, denoted by $\mathbb{R}$ and $\mathbb{C}$. Given a complex number $z = a + ib$, with $a, b \in \mathbb{R}$, its *conjugate* is denoted by $z^* = a - ib$ and its *modulus* is $|z| = \sqrt{z \cdot z^*}$. By $\mathbb{C}^{n\times m}$, we denote the set of $n \times m$ matrices with entries in $\mathbb{C}$. Given a matrix $M \in \mathbb{C}^{n\times m}$, we denote: by M_{ij}, its (i, j)th entry; by $M^* \in \mathbb{C}^{n\times m}$, the matrix satisfying ${M^*}_{ij} = (M_{ij})^*$; by $M^T \in \mathbb{C}^{m\times n}$, its *transpose*, i.e., the matrix with ${M^T}_{ij} = M_{ji}$; and by $M^\dagger = (M^T)^*$, its *adjoint*.

For matrices $A, B \in \mathbb{C}^{n\times m}$, their *sum* is the $n \times m$ matrix $(A + B)_{ij} = A_{ij} + B_{ij}$. For matrices $C \in \mathbb{C}^{n\times m}$ and $D \in \mathbb{C}^{m\times r}$, their *product* is the $n \times r$ matrix $(C \cdot D)_{ij} = \sum_{k=1}^{m} C_{ik} \cdot D_{kj}$. For matrices $A \in \mathbb{C}^{n\times m}$ and $B \in \mathbb{C}^{p\times q}$, their *direct sum* and *direct (or tensor or Kronecker) product* are the $(n + p) \times (m + q)$ and $n \cdot p \times m \cdot q$ matrices defined, respectively, as

$$A \oplus B = \begin{pmatrix} A & \mathbf{0} \\ \mathbf{0} & B \end{pmatrix}, \quad A \otimes B = \begin{pmatrix} A_{11} \cdot B & \cdots & A_{1m} \cdot B \\ \vdots & \ddots & \vdots \\ A_{n1} \cdot B & \cdots & A_{nm} \cdot B \end{pmatrix},$$

where $\mathbf{0}$ denotes zero-matrices of suitable dimensions. When operations are allowed to take place by matrix dimensions, the following properties hold:

$$(A \oplus B) \cdot (C \oplus D) = (A \cdot C) \oplus (B \cdot D) \quad \text{and} \quad (A \otimes B) \cdot (C \otimes D) = (A \cdot C) \otimes (B \cdot D).$$

A *Hilbert space* of dimension n is the linear space $\mathbb{C}^n$ of n-dimensional complex row vectors equipped with the sum and product by elements in $\mathbb{C}$ where for vectors $\varphi, \psi \in \mathbb{C}^n$, the *inner product* $\langle \varphi, \psi \rangle = \varphi \cdot \psi^\dagger$ is defined. The ith component of vector $\varphi \in \mathbb{C}^n$ is denoted by φ_i, and the *norm* of φ is given by $\|\varphi\| = \sqrt{\langle \varphi, \varphi \rangle} = \sqrt{\sum_{i=1}^{n} |\varphi_i|^2}$. For a complex number $z \in \mathbb{C}$, we let $z \cdot \varphi \in \mathbb{C}^n$ be the vector satisfying $(z \cdot \varphi)_i = z \cdot \varphi_i$. Clearly, we have $\|z \cdot \varphi\| = |z| \cdot \|\varphi\|$. For vectors $\varphi, \psi \in \mathbb{C}^n$, the triangular inequality states that $\|\varphi + \psi\| \leq \|\varphi\| + \|\psi\|$. Vectors φ and ψ are *orthogonal (orthonormal)* whenever $\langle \varphi, \psi \rangle = 0$ (and $\|\varphi\| = 1 = \|\psi\|$).

An orthonormal basis of $\mathbb{C}^n$ is any set of n orthonormal vectors in $\mathbb{C}^n$. In particular, the *canonical basis* of $\mathbb{C}^n$ is the set $\{e_1, e_2, \ldots, e_n\}$ of orthonormal vectors, where $e_i \in \mathbb{C}^n$ is the vector having 1 as its ith component and 0 elsewhere. The canonical basis *spans* $\mathbb{C}^n$ in that any $\varphi \in \mathbb{C}^n$ can be obtained as the (unique) linear combination $\varphi = \sum_{i=1}^{n} \varphi_i \cdot e_i$. More generally: a *subspace* is any subset of $\mathbb{C}^n$ that is a vector space. The subspace *spanned* by a set of vectors $X \subseteq \mathbb{C}^n$ is the linear space $\{\sum_{\varphi \in S} \alpha_\varphi \cdot \varphi \mid \alpha_\varphi \in \mathbb{C}\}$. Two subspaces $X, Y \subseteq \mathbb{C}^n$ are orthogonal if any vector in X is orthogonal to any vector in Y. In this case, we denote by $X \dotplus Y$ the linear space spanned by $X \cup Y$.

The *direct sum (direct, tensor, or Kronecker product)* of vectors $\varphi \in \mathbb{C}^n$ and $\psi \in \mathbb{C}^m$ is the vector $\varphi \oplus \psi = (\varphi_1, \ldots, \varphi_n, \psi_1, \ldots, \psi_m) \in \mathbb{C}^{n+m}$ ($\varphi \otimes \psi = (\varphi_1 \cdot \psi, \ldots, \varphi_n \cdot \psi) \in \mathbb{C}^{n\cdot m}$). It is easy to see that $\|\varphi \otimes \psi\| = \|\varphi\| \cdot \|\psi\|$, and $\|\varphi \oplus \psi\|^2 = \|\varphi\|^2 + \|\psi\|^2$.

Given a vector $\varphi \in \mathbb{C}^n$, we let $\mathrm{diag}(\varphi) \in \mathbb{C}^{n\times n}$ be the *diagonal matrix* having φ on its main diagonal and 0 elsewhere, i.e., $\mathrm{diag}(\varphi)_{ij} = \varphi_i$ if $i = j$ and 0 otherwise. Given the vector $\mathbf{1} = (1, \ldots, 1) \in \mathbb{C}^n$, we let $I^{(n)} = \mathrm{diag}(\mathbf{1})$ be the $n \times n$ identity matrix. The *inverse* of a matrix $M \in \mathbb{C}^{n\times n}$ is the unique matrix $M^{-1} \in \mathbb{C}^{n\times n}$ (if it exists) such that $M \cdot M^{-1} = I^{(n)} = M^{-1} \cdot M$. Let us quickly overview two matrix families playing a crucial role in the mathematical foundations of quantum mechanics and computing:

- *Unitary matrices*: A matrix $M \in \mathbb{C}^{n\times n}$ is *unitary* whenever $M \cdot M^\dagger = I^{(n)} = M^\dagger \cdot M$; thus, $M^{-1} = M^\dagger$ for M being unitary. Equivalently, M is unitary if and only if it preserves the norm, i.e., $\|\varphi \cdot M\| = \|\varphi\|$ for any vector $\varphi \in \mathbb{C}^n$. Direct sums and products of unitary matrices are unitary as well.
- *Hermitian matrices*: A matrix $M \in \mathbb{C}^{n\times n}$ is *Hermitian (or self-adjoint)* whenever $M = M^\dagger$. Let $\mathscr{O} \in \mathbb{C}^{n\times n}$ be an Hermitian matrix, $\nu_1, \nu_2, \ldots, \nu_s$ its eigenvalues, and $E_1, E_2, \ldots, E_s \subseteq \mathbb{C}^n$ the corresponding eigenspaces. It is well known that each eigenvalue ν_k is real, that E_i is orthogonal to E_j for every $1 \leq i \neq j \leq s$, and that $E_1 \dotplus E_2 \dotplus \cdots \dotplus E_s = \mathbb{C}^n$. So any vector $\varphi \in \mathbb{C}^n$ has a unique decomposition as $\varphi = \varphi_{(1)} + \varphi_{(2)} + \cdots + \varphi_{(s)}$ for unique $\varphi_{(j)} \in E_j$. The linear transformation mapping φ to $\varphi_{(j)}$ is the *projector* $P_j \in \mathbb{C}^{n\times n}$ onto the subspace E_j. Actually, the Hermitian matrix $\mathscr{O}$ is biunivocally determined by its eigenvalues and projectors as $\mathscr{O} = \sum_{i=1}^{s} \nu_i \cdot P_i$. We recall that a matrix $P \in \mathbb{C}^{n\times n}$ is a projector if and only if P is Hermitian and idempotent, i.e., $P^2 = P$.

As we will see in the next section, all linear algebra concepts so far recalled will provide a suitable mathematical description of a quantum finite state automaton and its computation. In fact, in accordance with quantum mechanics principles:

- The state of a quantum finite state automaton A at any given time during its computation is represented by a norm 1 vector from the Hilbert space spanned by the basis states of A; such a norm 1 vector is called *superposition of basis states*.
- The state evolution of A in a computation step is modeled by unitary matrices.
- Information on certain characteristics of A are probabilistically extracted by measuring some "observables" represented by Hermitian matrices.

The reader is referred to [50] for a gentle introduction to the mathematical foundations and interpretation of quantum mechanics.

2.3. Models of Quantum Finite State Automata and Quantum Automata with Control Language

Let us review some of the main models of quantum finite state automata (QFAs) introduced in the literature and which will be considered in our investigations. We refer the reader to, e.g., [53], for a general introduction to QFAs.

Measure-once QFAs: We start from the original, simplest, and most investigated model of a QFA, namely, the *measure-once* QFA model (MO-QFA) [29,31]. Let the alphabet $\Gamma = \Sigma \cup \{\sharp\}$, with Σ being an input alphabet and $\sharp \notin \Sigma$ being an endmarker symbol. A MO-QFA on Γ with m basis states is a triple $A = (\varphi, \{U(\sigma)\}_{\sigma\in\Gamma}, \eta)$ where:

- $\varphi \in \mathbb{C}^m$, with $\|\varphi\| = 1$, is the initial *superposition* of the basis states; the component φ_i, with $|\varphi_i| \leq 1$, is the *amplitude* of the ith basis state, while $|\varphi_i|^2$ is the *probability* of observing A being in the ith basis state.
- $U(\sigma) \in \mathbb{C}^{m\times m}$ is the unitary evolution matrix on $\sigma \in \Gamma$; $U(\sigma)_{ij}$, with $|U(\sigma)_{ij}| \leq 1$, is the amplitude of transitioning from the ith to jth basis state upon reading σ, while $|U(\sigma)_{ij}|^2$ is the related probability.
- $\eta \in \{0,1\}^m$ is the characteristic vector of the accepting states, i.e., $\eta_i = 1$ if and only if the ith basis state is accepting.

At any given time along its computation, the state of A is described by a superposition $\xi \in \mathbb{C}^m$, with $\|\xi\| = 1$, carrying the following probabilistic meaning: by observing A in ξ, we find A to be in its ith basis state with probability $|\xi_i|^2$. So the computation of A on an input word $w\sharp \in \Sigma^*\sharp$ starts from the initial superposition φ by scanning the leftmost input symbol. Then, the evolutions related to input symbols are applied in succession starting from $U(w_1)$, which acts on φ. Figure 2 depicts the first steps of this dynamic.

$$\begin{array}{rl} & \varphi \\ U(w_1) & \big| \\ & \varphi \cdot U(w_1) \\ U(w_2) & \big| \\ & \cdots \end{array}$$

Figure 2. First steps of the computation of the MO-QFA A on the input string $w_1 w_2 \cdots w_n \sharp$: the resulting superposition dynamic.

After the first k input symbols have been processed, A reaches the superposition $\varphi \cdot (\prod_{i=1}^{k} U(w_i)) \in \mathbb{C}^m$, which is again a norm 1 vector since $\|\varphi\| = 1$ and the matrices $U(w_i)$s are unitary. After processing the whole input word, A enters the final superposition $\varphi \cdot (\prod_{i=1}^{|w|} U(w_i)) \cdot U(\sharp)$. At this point, we evaluate the probability of observing A in an *accepting* basis state, yielding the probability $p_A(w)$ that A accepts the string $w \in \Sigma^*$. Such a probability is easily seen to be obtained as

$$p_A(w) = \sum_{\{i \mid \eta_i = 1\}} \left| \left(\varphi \cdot \left(\prod_{i=1}^{|w|} U(w_i) \right) \cdot U(\sharp) \right)_i \right|^2. \tag{2}$$

Equivalently, we can define $p_A(w)$ by introducing the *observable* described by the Hermitian matrix $\mathscr{O} = a \cdot P(a) + r \cdot P(r)$, with $P(a) = \mathrm{diag}(\eta)$ and $P(r) = I^{(m)} - P(a)$ being the projectors onto the two orthogonal subspaces of $\mathbb{C}^m$ spanned, respectively, by the accepting and rejecting (i.e., nonaccepting) basis states. By measuring this observable$\mathscr{O}$, we get two possible outcomes:

- a, which stands for "accepting": getting this outcome means seeing A in an *accepting* basis state;
- r, which stands for "rejecting": getting this outcome means seeing A in a *nonaccepting* basis state.

Each of these two outcomes shows up with a certain probability, which is computed as follows. Suppose at a given time A is in the norm 1 superposition $\xi \in \mathbb{C}^m$, and we measure the observable$\mathscr{O}$. Then, we see either the outcome a or the outcome r, with the probability equal to the square norm of the projection of ξ onto the subspace spanned by, respectively, the accepting or nonaccepting basis states:

$$\begin{aligned} \text{probability of seeing } a \quad &\mapsto \quad \|\xi \cdot P(a)\|^2 = \sum_{\{i \mid \eta_i = 1\}} |(\xi \cdot P(a))_i|^2, \\ \text{probability of seeing } r \quad &\mapsto \quad \|\xi \cdot P(r)\|^2 = \sum_{\{i \mid \eta_i = 0\}} |(\xi \cdot P(r))_i|^2. \end{aligned} \tag{3}$$

Thus, the probability $p_A(w)$ for A to accept $w \in \Sigma^*$ is given by the probability of obtaining the outcome a from measuring $\mathscr{O}$ at the end of the computation of A on w, i.e., when A is in the final superposition $\varphi \cdot (\prod_{i=1}^{|w|} U(w_i)) \cdot U(\sharp)$. According to (3), this is written as

$$p_A(w) = \left\| \varphi \cdot \left(\prod_{i=1}^{|w|} U(w_i) \right) \cdot U(\sharp) \cdot P(a) \right\|^2 = \sum_{\{i \mid \eta_i = 1\}} \left| \left(\varphi \cdot \left(\prod_{i=1}^{|w|} U(w_i) \right) \cdot U(\sharp) \right)_i \right|^2, \tag{4}$$

in accordance with Equation (2). When adopting the observable $\mathscr{O}$ instead of the characteristic vector η of the accepting basis states, we will use the notation $A = (\varphi, \{U(\sigma)\}_{\sigma \in \Gamma}, \mathscr{O})$ for an MO-QFA. The *stochastic event induced by* A is the function $p_A : \Sigma^* \to [0,1]$.

Measure-many QFAs: Let us now turn to the *measure-many* QFA model (MM-QFA) [28,30]. In an MM-QFA A, the m basis states are partitioned into *halting* states—which in turn are partitioned into *accepting* and *rejecting* states—and *non-halting* states, the latters being called *go* states. We let $\eta(a), \eta(r), \eta(g) \in \{0,1\}^m$ be the characteristic vectors of, respectively, accepting, rejecting, and go basis states. Again, we let $\Gamma = \Sigma \cup \{\sharp\}$. Then, we define the MM-QFA A on Γ with m basis states as the triple $A = (\varphi, \{U(\sigma)\}_{\sigma\in\Gamma}, \mathscr{O})$, where:

- $\varphi \in \mathbb{C}^m$ is the initial superposition and satisfies $\|\varphi\| = 1$.
- $U(\sigma) \in \mathbb{C}^{m\times m}$ is the unitary evolution matrix on $\sigma \in \Gamma$.
- $\mathscr{O} = a \cdot P(a) + r \cdot P(r) + g \cdot P(g)$, with projectors $P(c) = \mathrm{diag}(\eta(c))$ for $c \in \{a, r, g\}$, is the $m \times m$ Hermitian matrix representing an observable with three possible outcomes: a stands for "accept", r for "reject", and g for "go". Correspondingly, the projectors $P(a)$, $P(r)$, and $P(g)$ project onto the subspaces of $\mathbb{C}^m$ spanned by, respectively, the accepting, rejecting, and go basis states.

The computation of A on an input word $w\sharp \in \Sigma^*\sharp$ starts from the initial superposition φ by scanning the leftmost input symbol. Then, superposition transformations associated with input symbols are applied in succession. More precisely, the transformation triggered by a symbol $\sigma \in \Gamma$ consists of two phases:

1. *Evolution*: $U(\sigma)$ acts on the current superposition $\xi \in \mathbb{C}^m$ of A, with $\|\xi\| = 1$, to yield the next superposition $\xi' = \xi \cdot U(\sigma)$.
2. *Observation*: The observable $\mathscr{O}$ is measured on ξ', and the outcome is $c \in \{a, r, g\}$ with probability $\|\xi' \cdot P(c)\|^2$. If the outcome is either a or r, the computation of A halts and the input word is accepted or rejected, respectively. Otherwise, in case of outcome g, the superposition ξ' "collapses" to the norm 1 superposition $\xi' \cdot P(g)/\|\xi' \cdot P(g)\|$ from which A continues its computation.

Figure 3 exemplifies such an evolution/observation dynamic starting from $U(w_1)$, which acts on φ.

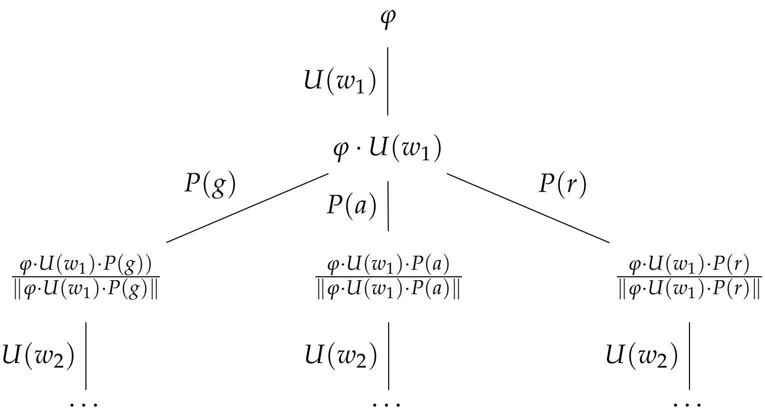

Figure 3. First steps of the computation of the MM-QFA A on the input string $w_1w_2\cdots w_n\sharp$: the resulting evolution/observation dynamic.

We remark that, differently from the MO-QFA model, the MM-QFA A can halt and accept also in the middle of the input string. The probability $p_A(w)$ for A to accept $w \in \Sigma^*$ accumulates step by step and is finally computed as

$$p_A(w) = \sum_{k=1}^{|w|+1} \left\| \varphi \cdot \left(\prod_{i=1}^{k-1} U(w_i) \cdot P(g) \right) \cdot U(w_k) \cdot P(a) \right\|^2, \tag{5}$$

where, for ease of notation, we let $w_{|w|+1} = \sharp$ and let $I^{(m)}$ be the result of the matrix product for $k = 1$. The function $p_A : \Sigma^* \to [0,1]$ is the *stochastic event induced by A*.

Latvian QFAs: We now present the *Latvian* QFA model (LQFA) [33,34]. As usual, we let $\Gamma = \Sigma \cup \{\sharp\}$. An LQFA on Γ with m basis states is a triple $A = (\varphi, \{U(\sigma)\}_{\sigma\in\Gamma}, \{\mathcal{O}_\sigma\}_{\sigma\in\Gamma})$, where:

- $\varphi \in \mathbb{C}^m$ is the initial superposition and satisfies $\|\varphi\| = 1$.
- $U(\sigma) \in \mathbb{C}^{m\times m}$ is the unitary evolution matrix on $\sigma \in \Gamma$.
- For any $\sigma \in \Sigma$, we let $\mathcal{O}_\sigma = \sum_{i=0}^{o_\sigma - 1} c_i(\sigma) \cdot P_i(\sigma)$ be an $m \times m$ Hermitian matrix representing an observable, where $\{c_0(\sigma), \ldots, c_{o_\sigma-1}(\sigma)\}$ is the set of all possible outcomes (eigenvalues) from measuring $\mathcal{O}_\sigma$, and $\{P_0(\sigma), \ldots, P_{o_\sigma-1}(\sigma)\}$ are the projectors onto the corresponding eigenspaces.
- For the final observable $\mathcal{O}_\sharp$, basis states are assumed to be partitioned into accepting and rejecting. So $\mathcal{O}_\sharp = a \cdot P(a) + r \cdot P(r)$, where $P(a)$ (respectively, $P(r) = I^{(m)} - P(a)$) is the projector onto the subspace of $\mathbb{C}^m$ spanned by the accepting (respectively, rejecting) basis states.

The computation of A on an input word $w\sharp \in \Sigma^*\sharp$ starts from the initial superposition φ by scanning the leftmost input symbol. Then, superposition transformations associated with input symbols are applied in succession. More precisely, the transformation triggered by a symbol $\sigma \in \Gamma$ consists of two phases:

1. *Evolution:* $U(\sigma)$ acts on the current superposition $\xi \in \mathbb{C}^m$ of A, with $\|\xi\| = 1$, yielding the next superposition $\xi' = \xi \cdot U(\sigma)$.
2. *Observation:* The observable $\mathcal{O}_\sigma$ is measured on ξ', and the outcome is $c_i(\sigma)$ with probability $\|\xi' \cdot P_i(\sigma)\|^2$. Upon getting the outcome $c_i(\sigma)$, the superposition ξ' "collapses" to the norm 1 superposition $\xi' \cdot P_i(\sigma) / \|\xi' \cdot P_i(\sigma)\|$ from which A continues its computation, unless we are processing the endmarker $\sharp$.

Figure 4 exemplifies such an evolution/observation dynamic starting from $U(w_1)$, which acts on φ.

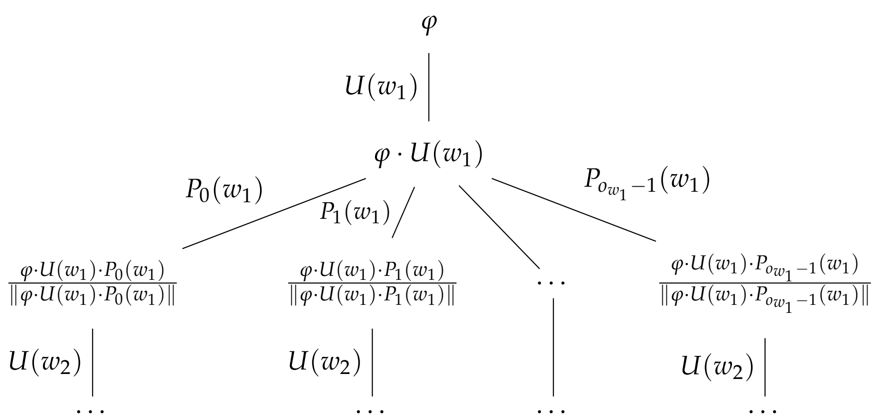

Figure 4. First steps of the computation of the LQFA A on the input string $w_1 w_2 \cdots w_n\sharp$: the resulting evolution/observation dynamic.

Upon processing the endmarker $\sharp$, the final observable $\mathcal{O}_\sharp$ is measured, yielding the probability of seeing A in an accepting basis state (i.e., of getting the outcome a). The probability for A to accept $w \in \Sigma^*$ is given by

$$p_A(w) = \sum_{i_1=0}^{o_{w_1}-1} \cdots \sum_{i_{|w|}=0}^{o_{w_{|w|}}-1} \left\| \varphi \cdot \left(\prod_{j=1}^{|w|} U(w_j) \cdot P_{i_j}(w_j) \right) \cdot U(\sharp) \cdot P(a) \right\|^2. \tag{6}$$

We emphasize that, similarly to the MO-QFA model and differently from the MM-QFA model, a LQFA cannot halt accepting or rejecting in the middle of input strings. The function $p_A : \Sigma^* \to [0,1]$ is the *stochastic event induced by A*.

Reversible QFAs: Finally, we overview the *reversible* QFA model (QRA) [24], which can be easily explained by starting from the LQFA model. In fact, a QRA A on $\Gamma = \Sigma \cup \{\sharp\}$ with m basis states is a LQFA $A = (\varphi, \{U(\sigma)\}_{\sigma\in\Gamma}, \{\mathscr{O}_\sigma\}_{\sigma\in\Gamma})$ on Γ with basis states from the set Q satisfying $|Q| = m$, where for *each* $\sigma \in \Sigma$, we have that the observable $\mathscr{O}_\sigma$ is fixed to be the *canonical observable* $\mathscr{O}$ defined as follows: For any basis state $q \in Q$, we let $\eta(q) \in \{0,1\}^m$ be its characteristic vector and $P_q = \mathrm{diag}(\eta(q))$ the corresponding projector onto the one-dimensional subspace of $\mathbb{C}^m$ spanned by q. Then, we define the canonical observable as $\mathscr{O} = \sum_{q\in Q} q \cdot P_q$. By measuring $\mathscr{O}$ when A is in the superposition $\xi \in \mathbb{C}^m$, we observe A in the basis state q with probability $\|\xi \cdot P_q\|^2$. Clearly, such a probability is equal to the square modulus of the amplitude in ξ relative to the basis state q. Together with the canonical observable $\mathscr{O}$ for every $\sigma \in \Sigma$, we have the usual final observable $\mathscr{O}_\sharp = a \cdot P(a) + r \cdot P(r)$ to be measured at the end of the computation of A on any given input word $w\sharp \in \Sigma^*\sharp$. Therefore, we will sometimes specify the QRA A as $A = (\varphi, \{U(\sigma)\}_{\sigma\in\Gamma}, \{\mathscr{O}, \mathscr{O}_\sharp\})$. The probability for the QRA A to accept $w \in \Sigma^*$ can be easily obtained from Equation (6) as

$$p_A(w) = \sum_{q_1\in Q} \cdots \sum_{q_{|w|}\in Q} \left\| \varphi \cdot \left(\prod_{j=1}^{|w|} U(w_j) \cdot P_{q_j} \right) \cdot U(\sharp) \cdot P(a) \right\|^2 . \tag{7}$$

The function $p_A : \Sigma^* \to [0,1]$ is the *stochastic event induced by A*.

2.4. Language Recognition Capabilities and Descriptional Power

Let A be any of the QFA models described in the previous section, with $p_A : \Sigma^* \to [0,1]$ being the corresponding induced stochastic event. Fixing a cut point $\lambda \in [0,1]$, the *language recognized by A with cut point* λ is the set of strings $L_{A,\lambda} = \{w \in \Sigma^* \mid p_A(w) > \lambda\}$. If, in addition, there exists $\varrho \in (0, \frac{1}{2}]$ satisfying $|p_A(w) - \lambda| \geq \varrho$ for any $w \in \Sigma^*$, then we say that the cut point λ is *isolated*, and ϱ is the *radius of isolation*.

More generally, a language $L \subseteq \Sigma^*$ is said to be recognized with isolated cut point by a QFA of a certain model whenever there exists a QFA A of this model such that

$$(\inf\{p_A(w) \mid w \in L\} - \sup\{p_A(w) \mid w \notin L\}) > 0.$$

In this case, we can set the cut point an the radius of isolation, respectively, to

$$\lambda = (\inf\{p_A(w) \mid w \in L\} + \sup\{p_A(w) \mid w \notin L\})/2,$$
$$\varrho = (\inf\{p_A(w) \mid w \in L\} - \sup\{p_A(w) \mid w \notin L\})/2.$$

Throughout this paper, for the sake of conciseness, we will sometimes write "isolated cut point QFA for a language" instead of "QFA recognizing a language with an isolated cut point".

The isolated cut point mode turns out to be one of the main language recognition policies within the literature of probabilistic devices. Its practical relevance in the realm of finite state automata is given by the possibility of efficiently using classical amplification techniques to enhance accuracy: on an isolated cut point finite state device, we can arbitrarily decrease the probability of wrongly classifying an input string w by repeating a constant number of times (i.e., not depending on $|w|$) the parsing of w and accepting w if and only if the relative frequency of acceptance exceeds the cut point. We refer the reader to Section 5 of [27], where the notion of isolated cut point recognition is first introduced and carefully analyzed (see also [26] for an extensive presentation of probabilistic finite state automata).

From a theoretical perspective, it is well known that isolated cut point language recognition maintains the computational power of classical probabilistic and quantum devices within the realm of regular languages. We point out that by dropping cut point isolation, even nonregular languages can be recognized by both classical probabilistic and quantum finite memory devices. For these and other recognition issues, the reader is referred to, e.g., [26,27,54,55] for classical probabilistic automata and to [24,28,56,57] and the discussion below for quantum devices.

2.4.1. Isolated Cut Point QFAs Recognition Power

We quickly review the language recognition power of the QFA models described above and compare it with that of classical finite state automata. While for general computation devices (e.g., Turing machines), the quantum and the classical paradigms share the same recognition power [58–61], for constant memory bounded devices, this is not the case. In fact, several results in the literature show that *pure* quantum models of finite state automata, as those so far considered, are strictly less powerful than classical models. Let us summarize some known results witnessing such a quantum recognition weakness.

As recalled, DFAs (and isolated cut point probabilistic finite state automata as well [26,27]) recognize all and only regular languages. It is well known (see, e.g., [29,31,57]) that the class of languages recognized by isolated cut point MO-QFAs coincides with the class of group languages [62], which is a proper subclass of regular languages.

Isolated cut point LQFAs are proved in [33,34] to be strictly more powerful than isolated cut point MO-QFAs since their recognition power coincides with the class of block group languages. An equivalent characterization states that a language is recognized by an isolated cut point LQFA if and only if it belongs to the boolean closure of languages of the form $L_1 a_1 L_2 a_2 \cdots a_k L_{k+1}$ for group languages $L_i \subseteq \Sigma^*$, $a_i \in \Sigma$ and $|\Sigma| > 1$. So even isolated cut point LQFAs turn out to be strictly less powerful than classical finite state automata: for instance, both the regular languages $a\Sigma^*$ and $\Sigma^* a$, with $a \in \Sigma$ and $|\Sigma| > 1$, cannot be recognized by isolated cut point LQFAs [33,34].

The recognition power of isolated cut point MM-QFAs still remains an open question. However, it is known that isolated cut point MM-QFAs are strictly more powerful than isolated cut point LQFAs but strictly less powerful than DFAs. In particular, isolated cut point MM-QFAs can recognize the regular language $a\Sigma^*$, but still they cannot recognize the language $\Sigma^* a$ [30,33,34]. It is worth remarking that when restricted to unary inputs, the recognition power of isolated cut point MM-QFAs reverts to that of DFAs: any unary regular language can be accepted by an isolated cut point MM-QFA [33,34,38]. This latter result clearly does not hold for isolated cut point MO-QFAs.

Even for isolated cut point QRAs, the recognition power has not been characterized yet, although, being a particular case of LQFAs, they are strictly less powerful than DFAs. From [24], we have that isolated cut point QRAs recognize the class of group languages. However, they are not able to accept the regular languages $a\Sigma^*$ and $\Sigma^* a$.

For the reader's ease of mind, the Venn diagram displayed in Figure 5 shows at a glance the recognition power of the models of isolated cut point QFAs so far considered.

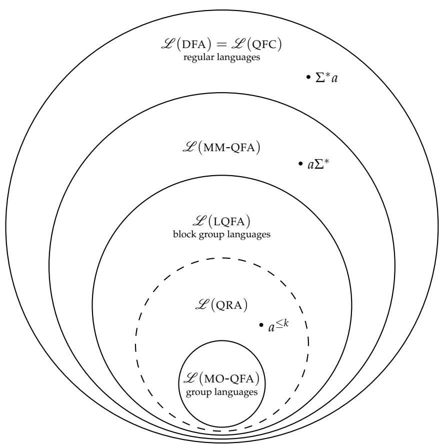

Figure 5. The language hierarchy for the automaton models considered in this paper. In the picture, we let $\mathscr{L}(X)$ denote the class of languages recognized by isolated cut point automata of type $X \in \{$MO-QFA, QRA, LQFA, MM-QFA, DFA, QFC$\}$; QFAs with control language (QFCs) will be explained later in Section 2.5 and are here reported for the sake of completeness. Solid line language classes imply proper containments, while the dashed line language class indicates a containment still not known to be proper. The languages displayed near dots • witness proper containments.

2.4.2. Isolated Cut Point QFAs Descriptional Power

Let us now quickly expand on the descriptional power of QFAs. Descriptional complexity (see, e.g., [63–67] for surveys and examples of results) investigates formal systems on the basis of their *size*. In the case of finite state automata, a natural size measure is represented by the number of states. Studying the descriptional power of models of finite state automata basically means to study their ability to recognize languages using a very limited amount of states, this having practical impacts in the physical construction of classical and quantum finite memory devices (see [6] for a discussion).

A lot of results in the literature show that the quantum descriptional power may greatly outperform the classical one when it comes to finite state devices. To address this quantum superiority, we provide only a few examples of the size economy for recognizing specific families of regular languages.

The periodic language family $L_d = \{\sigma^k \mid k \bmod d = 0\}$ for integers $d > 0$ has been widely studied. In a classical setting, we have that d states are necessary and sufficient to recognize L_d by deterministic and nondeterministic finite state automata. Moreover, for prime d, we have also that two-way nondeterministic and isolated cut point probabilistic finite state automata require not less than d states (see [6] for an overview of descriptional complexity issues related to the language family L_d).

With the quantum paradigm, only two basis states are sufficient to recognize L_d by an isolated cut point MO-QFA. However, it should be stressed that the radius of isolation of this simple MO-QFA tends to zero as d grows. To fix this accuracy issue, some statistical frameworks were designed in [37,68] in order to show the existence of MO-QFAs recognizing L_d with only $O(\log d)$ basis states and arbitrarily large isolation around the cut point. Such an exponential size gain has then been extended from the unary case to more general languages on arbitrary alphabets, such as multiperiodic languages [37].

Other relevant examples of isolated cut point MO-QFAs that are significantly smaller than the equivalent classical finite state automata will be tackled in more detail in Section 5. There, we will use such small MO-QFAs as quantum components for the design of size efficient isolated cut point QFCs for the *whole* class of unary regular languages.

In conclusion, from the overview provided in this section, we can motivate the design and analysis of hybrid computational devices as follows: although pure QFAs do not reach the computational power of classical finite memory devices, they may result in very small quantum components performing specific tasks to be embedded in a classical control environment. This architecture can be modeled by the notion of a QFA with control language, which we are going to present in the next section.

2.5. Quantum Finite State Automata with Control Language

As usual, we let $\Gamma = \Sigma \cup \{\sharp\}$. A QFA *with control language* (QFC) [24,25] on Γ with q quantum basis states is a 4-tuple $A = (\varphi, \{U(\sigma)\}_{\sigma\in\Gamma}, \mathscr{O}, \mathscr{L})$, where:

- $\varphi \in \mathbb{C}^q$ is the initial superposition and satisfies $\|\varphi\| = 1$.
- $U(\sigma) \in \mathbb{C}^{q\times q}$ is the unitary evolution matrix on $\sigma \in \Gamma$.
- $\mathscr{O} = \sum_{c\in C} c \cdot P(c)$ is a $q \times q$ Hermitian matrix representing an observable, with C being the set of all possible outcomes (eigenvalues) of measuring $\mathscr{O}$ and $P(c)$ the projector onto the eigenspace corresponding to $c \in C$.
- $\mathscr{L} \subseteq C^*$ is a *regular* language, called the *control language*.

We briefly describe the behavior of A on the input string $x\sharp \in \Sigma^*\sharp$. The computation starts from the initial superposition φ by scanning the leftmost input symbol. Then, superposition transformations associated with each input symbol are applied in succession. More precisely, the transformation triggered by a symbol $\sigma \in \Gamma$ consists of two phases:

1. *Evolution*: $U(\sigma)$ acts on the current superposition $\xi \in \mathbb{C}^q$ of A, with $\|\xi\| = 1$, yielding the next superposition $\xi' = \xi \cdot U(\sigma)$.

2. *Measuring*: the observable $\mathscr{O}$ is measured on ξ', and the outcome c is seen with probability $\|\xi' \cdot P(c)\|^2$. Upon getting the outcome c, the superposition ξ' "collapses" to the norm 1 superposition $\xi' \cdot P(c)/\|\xi' \cdot P(c)\|$, from which A continues its computation.

So the computation of A on $x\sharp = x_1 \cdots x_n\sharp$ produces the sequence $y_1 \cdots y_n y_\sharp \in C^*$ of outcomes of measuring $\mathscr{O}$ at each step with probability $p_A(y_1 \cdots y_n y_\sharp; x_1 \cdots x_n\sharp)$ computed as

$$p_A(y_1 \cdots y_n y_\sharp; x_1 \cdots x_n\sharp) = \left\| \varphi \cdot \left(\prod_{i=1}^{n} U(x_i) \cdot P(y_i) \right) \cdot U(\sharp) \cdot P(y_\sharp) \right\|^2. \tag{8}$$

A computation of A yielding the outcome sequence $y_1 \cdots y_n y_\sharp \in C^*$ is said to be *accepting* (respectively, *rejecting*) if and only if $y_1 \cdots y_n y_\sharp \in \mathscr{L}$ (respectively, $y_1 \cdots y_n y_\sharp \notin \mathscr{L}$). Therefore, we define the *probability* $\mathscr{E}_A(x)$ *that* A *accepts the word* $x \in \Sigma^*$ as the probability for A to exhibit an accepting computation upon processing the input string $x\sharp$, namely,

$$\mathscr{E}_A(x) = \sum_{y_1 \cdots y_n y_\sharp \in \mathscr{L}} p_A(y_1 \cdots y_n y_\sharp; x_1 \cdots x_n\sharp). \tag{9}$$

The function $\mathscr{E}_A : \Sigma^* \to [0,1]$ is the *stochastic event induced by* A.

Intuitively, a QFC can be regarded as a hybrid system where a quantum component—a QFA—and a classical component—a DFA—cooperate. The quantum component provides the system evolution together with an observable to be measured at each step. The classical component processes the sequence of observable outcomes one outcome at a time by checking whether such a sequence leads to acceptance according to membership in the regular control language $\mathscr{L}$. From this viewpoint, the language $\mathscr{L}$ "controls" the final acceptance/rejection outcome in the computation of the QFC. Indeed, with $\mathscr{L}$ being a regular language, the classical component checking membership in $\mathscr{L}$ is well suited to be a DFA.

When expressing the *size* of a QFC A, we must take into account the size of both its quantum and its classical component. Therefore, we say that *A has q quantum basis states and m classical states* whenever its quantum component features q basis states, and its control language is recognized by a DFA, the classical component, having m states.

2.5.1. Isolated Cut Point QFCs Recognition Power

Concerning the language recognition power, isolated cut point QFCs recognize *all and only regular languages* (see also Figure 5 for relationships with the recognition power of other models of QFAs). The fact that the language recognized by an isolated cut point QFC A is regular is proved in [24] by showing that: *(i)* the formal power series related to A is real-valued bounded rational (see Section 4 for the notion of a formal power series/generating function), and *(ii)* the language defined with isolated cut point by a real-valued bounded rational formal power series is regular. Conversely, a construction is given in [25] by which for any given regular language $L \subseteq \Sigma^*$ recognized by a DFA with m states, we can build an isolated cut point QFC with $|\Sigma| + 1$ quantum basis states and $(|\Sigma| + 1) \cdot m$ classical states. It should be pointed out that the resulting QFC actually recognizes L "deterministically", i.e., it always exhibits the correct accept/reject outcome with probability 1.

2.5.2. Isolated Cut Point QFCs Descriptional Power

Concerning hardware succinctness capabilities, the higher descriptional power of QFCs vs. classical DFAs is pointed out in [25] on a particular family $L_{d,h}$ of regular languages on a binary alphabet. Precisely, *(i)* $d \cdot (h+1) + 1$ states are necessary and sufficient for any DFA to recognize $L_{d,h}$, while *(ii)* an isolated cut point QFCs for $L_{d,h}$ exists and features a classical component with $O(h)$ states and a quantum component with only a constant number of quantum basis states. As a matter of fact, the accuracy of this QFC can be arbitrarily enhanced by replacing its quantum component with a more precise one having $O(\log d)$ quantum basis states.

3. QFA Models Simulations by QFCs

One of the main capabilities of QFCs is to provide a unifying framework within which to represent several QFA models. Here, we show how to simulate MO-QFAs, MM-QFAs, QRAs, and LQFAs by QFCs, emphasizing the cost in terms of quantum and classical states of such simulations. The simulation of LQFAs is devised in [36], while the other simulations are quickly addressed in [24]. For the sake of completeness, we display all these simulations in full detail. To this regard, to study simulation correctness, we will make use of the following technical lemma on the dynamics of QFCs, the proof of which can be given by induction on the length of the input strings:

Lemma 1. *Let $A = (\varphi,\ \{U(\sigma)\}_{\sigma\in\Gamma},\ \mathscr{O} = \sum_{c\in C} c \cdot P(c),\ \mathscr{L})$ be a QFC with q quantum basis states. Then for any vector $\xi \in \mathbb{C}^q$ and any string $x_1 \cdots x_n \in \Gamma^*$, we have*

$$\sum_{y_1\cdots y_n\in C^n} \left\| \xi \cdot \left(\prod_{i=1}^{n} U(x_i) \cdot P(y_i) \right) \right\|^2 = \|\xi\|^2.$$

We are now ready to show the main result of this section:

Theorem 2. *Let A be a MO-QFA, MM-QFA, QRA, or LQFA with q basis states (Table 1). Then, there exists a QFC B satisfying $\mathscr{E}_B = p_A$ with the following number of quantum basis states and classical states:*

Table 1. The costs in terms of quantum and classical states of simulating models of QFAs by QFCs.

Simulated QFA Model A with q Basis States	Quantum Basis States for Simulating QFC B	Classical States for Simulating QFC B
MO-QFA	$2 \cdot q$	2
MM-QFA	q	3
QRA	q	2
LQFA	$2 \cdot q^2$	q

Proof. Throughout the following simulations, for a given $t \in \mathbb{N}$, we let $\mathbf{0}_t$ denote the t-dimensional row zero-vector and $[\mathbf{0}]_t$ denote the $t \times t$ zero-matrix.

MO-QFA simulation: Let $A = (\varphi,\ \{U(\sigma)\}_{\sigma\in\Gamma},\ \mathscr{O})$ be an MO-QFA with q basis states and observable $\mathscr{O} = a \cdot P(a) + r \cdot P(r)$. We define the simulating QFC $B = (\phi,\ \{Y(\sigma)\}_{\sigma\in\Gamma},\ \Omega,\ \mathscr{L})$ with $2 \cdot q$ quantum basis states, where:

- $\phi = \varphi \oplus \mathbf{0}_q$ is the initial superposition; we have $\phi \in \mathbb{C}^{2\cdot q}$ and $\|\phi\| = \|\varphi\| = 1$.
- $Y(\sigma) = U(\sigma) \oplus I^{(q)}$ for any $\sigma \in \Sigma$, and $Y(\sharp) = \begin{pmatrix} [\mathbf{0}]_q & U(\sharp) \\ U(\sharp) & [\mathbf{0}]_q \end{pmatrix}$ are the evolution matrices; we have that $Y(\sigma), Y(r) \in \mathbb{C}^{2\cdot q\times 2\cdot q}$ are unitary matrices.
- $\Omega = a \cdot \Lambda(a) + r \cdot \Lambda(r)$, with projectors $\Lambda(a) = I^{(q)} \oplus P(a)$ and $\Lambda(r) = [\mathbf{0}]_q \oplus P(r)$, is a $2 \cdot q \times 2 \cdot q$ Hermitian matrix representing an observable with the two outcomes a (accept) and r (reject).
- $\mathscr{L} = a^* \subseteq \{a, r\}^*$ is the regular control language.

To have an idea of the behavior of B, we can think of each possible superposition of B as being divided into two halves. Along its computation, *before* processing the input endmarker, B replicates the dynamic of the simulated MO-QFA A into the left half and leaves the right half untouched. As the reader may easily verify, during this phase any observation by Ω yields the outcome a with certainty. Thus, by setting the control language to $\mathscr{L} = a^*$, we are basically letting B reach with certainty a superposition featuring in the left half the superposition assumed by A when the endmarker is about to be read. Upon reading the endmarker, the left half of this superposition is swapped with the right half while evolving on $\sharp$ as in A. After the evolution on $\sharp$, the final observation on B for acceptance takes place exactly in the same context as in A.

Let us formally show that $\mathscr{E}_B = p_A$. Let $y_1 \cdots y_n y_\sharp \in \{a,r\}^*$ be a sequence of outcomes of measuring Ω along the computation of B on an input word $x\sharp = x_1 \cdots x_n \sharp \in \Sigma^*\sharp$. As pointed out above, we can only have $y_i = a$ for every $1 \le i \le n$. In addition, this computation is accepting if and only if $y_\sharp = a$ as well. So by Equation (9) and recalling that the control language is defined as $\mathscr{L} = a^*$, we get that the probability for B to accept $x \in \Sigma^*$ is

$$\mathscr{E}_B(x) = \sum_{y_1 \cdots y_n y_\sharp \in \mathscr{L}} p_B(y_1 \cdots y_n y_\sharp; x_1 \cdots x_n \sharp) = p_B(a^{n+1}; x\sharp).$$

Indeed, by Equation (8), the probability $p_B(a^{n+1}; x\sharp)$ of observing the outcome sequence a^{n+1} along the computation of B on the input word $x\sharp$ is

$$\begin{aligned}
\mathscr{E}_B(x) = p_B(a^{n+1}; x\sharp) &= \left\| \phi \cdot \left(\prod_{i=1}^{n} Y(x_i) \cdot \Lambda(a) \right) \cdot Y(\sharp) \cdot \Lambda(a) \right\|^2 \\
&= \left\| (\varphi \oplus \mathbf{0}_q) \cdot \left(\prod_{i=1}^{n} \begin{pmatrix} U(x_i) & [\mathbf{0}]_q \\ [\mathbf{0}]_q & P(a) \end{pmatrix} \right) \cdot \begin{pmatrix} [\mathbf{0}]_q & U(\sharp) \\ U(\sharp) & [\mathbf{0}]_q \end{pmatrix} \cdot \Lambda(a) \right\|^2 \\
&= \left\| \left(\mathbf{0}_q \oplus \left(\varphi \cdot \left(\prod_{i=1}^{n} U(x_i) \right) \cdot U(\sharp) \right) \right) \cdot \Lambda(a) \right\|^2 \\
&= \left\| \varphi \cdot \left(\prod_{i=1}^{n} U(x_i) \right) \cdot U(\sharp) \cdot P(a) \right\|^2 = p_A(x),
\end{aligned}$$

where the last equality follows from Equation (4). The smallest DFA recognizing the control language $\mathscr{L} = a^* \subseteq \{a,r\}^*$ clearly has two states.

MM-QFA simulation: Let $A = (\varphi, \{U(\sigma)\}_{\sigma\in\Gamma}, \mathscr{O})$ be an MM-QFA with q basis states and observable $\mathscr{O} = a \cdot P(a) + r \cdot P(r) + g \cdot P(g)$. We define the simulating QFC $B = (\varphi, \{U(\sigma)\}_{\sigma\in\Gamma}, \mathscr{O}, \mathscr{L})$ with q basis states and where the control language $\mathscr{L} \subseteq \{a,r,g\}^*$ is set to $\mathscr{L} = g^* a \{a,r,g\}^*$. According to Equations (8) and (9), the stochastic event induced by B on $x \in \Sigma^*$ is obtained as follows (we let $x_{n+1} = \sharp$ and $y_{n+1} = a$ and let $I^{(q)}$ be the result of the matrix products below whenever the lower limit exceeds the upper limit):

$$\begin{aligned}
\mathscr{E}_B(x) &= \sum_{y_1 \cdots y_n y_\sharp \in \mathscr{L}} \left\| \varphi \cdot \left(\prod_{i=1}^{n} U(x_i) \cdot P(y_i) \right) \cdot U(\sharp) \cdot P(y_\sharp) \right\|^2 \\
&= \sum_{k=1}^{n+1} \sum_{y_{k+1} \cdots y_n y_\sharp \in C^{n-k+1}} \left\| \varphi \cdot \left(\prod_{i=1}^{k-1} U(x_i) \cdot P(g) \right) \cdot U(x_k) \cdot P(a) \cdot \left(\prod_{j=k+1}^{n+1} U(x_j) \cdot P(y_j) \right) \right\|^2 \\
&= \sum_{k=1}^{n+1} \left\| \varphi \cdot \left(\prod_{i=1}^{k-1} U(x_i) \cdot P(g) \right) \cdot U(x_k) \cdot P(a) \right\|^2 = p_A(x),
\end{aligned}$$

where the last two equalities follow, respectively, from Lemma 1 and Equation (5). The smallest DFA recognizing the control language $\mathscr{L} = g^* a \{a,r,g\}^*$ clearly has three states.

QRA simulation: Let $A = (\varphi, \{U(\sigma)\}_{\sigma\in\Gamma}, \{\mathscr{O}, \mathscr{O}_\sharp\})$ be a QRA with Q being the set of basis states satisfying $|Q| = q$, $\mathscr{O} = \sum_{q\in Q} q \cdot P_q$ being the canonical observable to be used all along processing symbols in Σ, and with $\mathscr{O}_\sharp = a \cdot P(a) + r \cdot P(r)$ being the usual final accept/reject observable. We define the simulating QFC $B = (\varphi, \{U(\sigma)\}_{\sigma\in\Gamma}, \mathscr{O}, \mathscr{L})$ with q basis states, where the control language $\mathscr{L} \subseteq Q^*$ is set to $\mathscr{L} = Q^* F$, and with $F \subseteq Q$ being the set of accepting basis states (i.e, those states spanning the subspace of $\mathbb{C}^q$ that $P(a)$

projects onto). Then, by Equations (8) and (9), the probability that B accepts the input word $x = x_1 \cdots x_n \in \Sigma^*$ is written as

$$\begin{aligned}
\mathscr{E}_B(x) &= \sum_{q_1\cdots q_n q_\sharp \in \mathscr{L}} p_B(q_1 \cdots q_n q_\sharp; x_1 \cdots x_n \sharp) \\
&= \sum_{q_1\cdots q_n q_\sharp \in \mathscr{L}} \left\| \varphi \cdot \left(\prod_{i=1}^{n} U(x_i) \cdot P_{q_i} \right) \cdot U(\sharp) \cdot P_{q_\sharp} \right\|^2 \\
&= \sum_{q_1\cdots q_n \in Q^*} \left\| \varphi \cdot \left(\prod_{i=1}^{n} U(x_i) \cdot P_{q_i} \right) \cdot U(\sharp) \cdot P(a) \right\|^2 \\
&= \sum_{q_1 \in Q} \cdots \sum_{q_n \in Q} \left\| \varphi \cdot \left(\prod_{j=1}^{n} U(x_j) \cdot P_{q_j} \right) \cdot U(\sharp) \cdot P(a) \right\|^2 = p_A(x),
\end{aligned}$$

where the last equality follows from Equation (7). The smallest DFA recognizing the control language $\mathscr{L} = Q^* F$ clearly has two states.

LQFA simulation: Let $A = (\varphi,\ \{U(\sigma)\}_{\sigma\in\Gamma},\ \{\mathscr{O}_\sigma\}_{\sigma\in\Gamma})$ be an LQFA with q basis states. For ease of notation, in the final observable $\mathscr{O}_\sharp$, we rename with $P_0(\sharp)$ (respectively, $P_1(\sharp)$) the projector $P(a)$ (respectively, $P(r)$) onto the subspace of $\mathbb{C}^q$ spanned by the accepting (respectively, rejecting) basis states. By recalling the form of the observable $\mathscr{O}_\sigma = \sum_{i=0}^{o_\sigma - 1} c_i(\sigma) \cdot P_i(\sigma)$ associated with $\sigma \in \Gamma$, we set $\theta = \max\{o_\sigma \mid \sigma \in \Gamma\}$; clearly, $\theta \le q$ holds true. Then, for each $\sigma \in \Gamma$, we let the $\theta \cdot q \times \theta \cdot q$ matrix

$$H(\sigma) = \begin{pmatrix}
U(\sigma)\cdot P_0(\sigma) & U(\sigma)\cdot P_1(\sigma) & \cdots & U(\sigma)\cdot P_{\theta-2}(\sigma) & U(\sigma)\cdot P_{\theta-1}(\sigma) \\
U(\sigma)\cdot P_1(\sigma) & U(\sigma)\cdot P_2(\sigma) & \cdots & U(\sigma)\cdot P_{\theta-1}(\sigma) & U(\sigma)\cdot P_0(\sigma) \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
U(\sigma)\cdot P_{\theta-1}(\sigma) & U(\sigma)\cdot P_0(\sigma) & \cdots & U(\sigma)\cdot P_{\theta-3}(\sigma) & U(\sigma)\cdot P_{\theta-2}(\sigma)
\end{pmatrix},$$

where for every $o_\sigma \le j \le \theta - 1$, we let $P_j(\sigma) = [\mathbf{0}]_q$. The reader may easily verify that $H(\sigma)$ is a unitary matrix. So we define the simulating QFC $B = (\phi,\ \{Y(\sigma)\}_{\sigma\in\Gamma},\ \Omega, \mathscr{L})$ with $2\theta \cdot q$ quantum basis states equivalent to the LQFA $A = (\varphi,\ \{U(\sigma)\}_{\sigma\in\Gamma},\ \{\mathscr{O}_\sigma\}_{\sigma\in\Gamma})$ as:

- $\phi = \varphi \oplus \mathbf{0}_{(2\theta-1)\cdot q}$ is the initial superposition, which can be regarded as a row vector of 2θ blocks, each of dimension q; we have $\phi \in \mathbb{C}^{2\theta\cdot q}$ and $\|\phi\| = \|\varphi\| = 1$.
- $Y(\sigma) = \begin{pmatrix} H(\sigma) & [\mathbf{0}]_{\theta\cdot q} \\ [\mathbf{0}]_{\theta\cdot q} & I^{(\theta\cdot q)} \end{pmatrix}$ for any $\sigma \in \Sigma$, and $Y(\sharp) = \begin{pmatrix} [\mathbf{0}]_{\theta\cdot q} & H(\sharp) \\ I^{(\theta\cdot q)} & [\mathbf{0}]_{\theta\cdot q} \end{pmatrix}$ are the evolution matrices; we have that $Y(\sigma), Y(\sharp) \in \mathbb{C}^{2\theta\cdot q \times 2\theta\cdot q}$ are unitary matrices.
- $\Omega = \sum_{i=0}^{\theta-1} a_i \cdot \Lambda_i + \sum_{i=0}^{\theta-1} b_i \cdot \Lambda_{\theta+i}$, with projectors $\Lambda_j = [\mathbf{0}]_{j\cdot q} \oplus I^{(q)} \oplus [\mathbf{0}]_{(2\theta-j-1)\cdot q}$ for $0 \le j \le 2\theta - 1$ is a $2\theta \cdot q \times 2\theta \cdot q$ Hermitian matrix representing an observable with the set of outcomes $C = \bigcup_{i=0}^{\theta-1} \{a_i, b_i\}$; we notice that the projector Λ_j resets to zero all blocks in a superposition of B with the exception of the $(j+1)$th block.
- $\mathscr{L} \subseteq C^*$ is the regular control language recognized by the DFA

  $$D = (\{s_0,\ \dots,\ s_{\theta-1}\},\ C,\ s_0,\ \delta, \{s_0\})$$

  for which the transition function is defined for $0 \le i, j \le \theta - 1$ as

  $$\delta(s_i, c) = \begin{cases} s_j & \text{if } c = a_j \\ s_{(i+j) \bmod \theta} & \text{if } c = b_j. \end{cases}$$

Let us overview the dynamics of B. A superposition of B is a norm 1 vector in $\mathbb{C}^{2\theta\cdot q}$, which can be seen to be divided into a left half and a right half. Each half consists of θ consecutive q-dimensional blocks. On the input symbol $\sigma \ne \sharp$, the matrix $Y(\sigma)$ replicates in the left half the corresponding evolution step of the simulated LQFA A together with all the o_σ many projections arising from measuring $\mathscr{O}_\sigma$ and stores all such resulting projection vectors into the θ many blocks of the left half of the superposition. At the same time,

$Y(\sigma)$ leaves the right half of the superposition unchanged. The endmarker evolution $Y(\sharp)$ behaves similarly plus swaps the left half with the right half of the superposition.

It is not hard to see that measuring the observable Ω along processing *before* $\sharp$ basically takes place on the left half of superpositions, and outcomes of the form a_i are always returned. On the other hand, measuring Ω after evolving on $\sharp$ basically takes place on the right half of superpositions, and always outcomes of the form b_i are returned. Clearly, this latter fact enables the classical component of B, i.e., the DFA D, to detect the moment in which the quantum component of B has parsed the endmarker $\sharp$.

We now formally show that $\mathscr{E}_B = p_A$. By Equation (9), the probability for B to accept the word $x = x_1 \cdots x_n \in \Sigma^*$ is

$$\mathscr{E}_B(x) = \sum_{a_{i_1} \cdots a_{i_n} b_f \in \mathscr{L}} \left\| \varphi \cdot \left(\prod_{j=1}^{n} Y(x_j) \cdot \Lambda_{i_j} \right) \cdot Y(\sharp) \cdot \Lambda_f \right\|^2. \tag{10}$$

To process Equation (10), we observe that:

- The operator $Y(\sigma) \cdot \Lambda_s$ in the product of (10) acts on the $(r+1)$th q-dimensional block of the current superposition of B and *(i)* transforms the block by the matrix $U(\sigma) \cdot P_{(r+s) \bmod \theta}(\sigma)$, and then *(ii)* it moves the transformed block to the $(s+1)$th block position. To formally explain the action of the operator $Y(\sigma) \cdot \Lambda_s$, assume that the $(r+1)$th q-dimensional block is the row vector $\xi \in \mathbb{C}^q$. Then, we have

$$(\mathbf{0}_{r\cdot q} \oplus \xi \oplus \mathbf{0}_{(2\theta - r - 1)\cdot q}) \cdot Y(\sigma) \cdot \Lambda_s = (\mathbf{0}_{s\cdot q} \oplus \xi \cdot U(\sigma) \cdot P_{(r+s) \bmod \theta}(\sigma) \oplus \mathbf{0}_{(2\theta - s - 1)\cdot q}). \tag{11}$$

- Given that the initial superposition ϕ of the QFC B has a single nonzero block and considering Equation (11), we get that at any step along the computation of B, the nonzero entries of the superposition are always in a single q-dimensional block. So when evaluating the event $\mathscr{E}_B$ in (10), we can focus only on the single nonzero q-dimensional block.
- By model definition, the computation of B on input $x\sharp$ featuring the outcome sequence $a_{i_1} \cdots a_{i_n} b_f$ is accepting if and only if $a_{i_1} \cdots a_{i_n} b_f \in \mathscr{L}$. In turn, $a_{i_1} \cdots a_{i_n} b_f \in \mathscr{L}$ if and only if $(i_n + f) \bmod \theta = 0$, as the reader may easily verify by checking the transition function δ of the DFA D for $\mathscr{L}$. The sum in (10) returns the global probability of having an accepting computation on $x\sharp$, i.e., the probability $\mathscr{E}_B(x)$ that B accepts x.

  By these observations, we can rewrite Equation (10) as

$$\begin{aligned} \mathscr{E}_B(x) &= \sum_{i_1=0}^{\theta-1} \sum_{i_2=0}^{\theta-1} \cdots \sum_{i_n=0}^{\theta-1} \sum_{\{0 \le f \le \theta-1 \,\mid\, (i_n+f) \bmod \theta = 0\}} \\ &\quad \left\| \varphi \cdot \left(\prod_{j=1}^{n} U(x_j) \cdot P_{(i_{j-1}+i_j) \bmod \theta}(x_j) \right) \cdot U(\sharp) \cdot P_{(i_n+f) \bmod \theta}(\sharp) \right\|^2 \\ &= \sum_{i_1=0}^{\theta-1} \sum_{i_2=0}^{\theta-1} \cdots \sum_{i_n=0}^{\theta-1} \left\| \varphi \cdot \left(\prod_{j=1}^{n} U(x_j) \cdot P_{(i_{j-1}+i_j) \bmod \theta}(x_j) \right) \cdot U(\sharp) \cdot P_0(\sharp) \right\|^2, \end{aligned} \tag{12}$$

where in (12) we stipulate that $i_0 = 0$. Now, let us fix a $i_{j-1} \in \{0, 1, \ldots, \theta - 1\}$: it is not hard to see that

$$\left\{ (i_{j-1} + i_j) \bmod \theta \;\mid\; 0 \le i_j \le \theta - 1 \right\} = \{0, 1, \ldots, \theta - 1\}.$$

We can use this set equality to manipulate the subscript of the projector P in (12) and get

$$\mathscr{E}_B(x) = \sum_{i_1=0}^{\theta-1} \sum_{i_2=0}^{\theta-1} \cdots \sum_{i_n=0}^{\theta-1} \left\| \varphi \cdot \left(\prod_{j=1}^{n} U(x_j) \cdot P_{i_j}(x_j) \right) \cdot U(\sharp) \cdot P_0(\sharp) \right\|^2. \tag{13}$$

By recalling that $P_j(\sigma) = [\mathbf{0}]_q$ for $o_\sigma \leq j \leq \theta - 1$, we can bound the upper limits of the sums in (13) and obtain

$$\mathscr{E}_B(x) = \sum_{i_1=0}^{o_{x_1}-1} \cdots \sum_{i_n=0}^{o_{x_n}-1} \left\| \varphi \cdot \left(\prod_{j=1}^{n} U(x_j) \cdot P_{i_j}(x_j) \right) \cdot U(\sharp) \cdot P_0(\sharp) \right\|^2 = p_A(x),$$

in accordance with Equation (6). Therefore, we conclude that the QFC B simulates the LQFA A using $2\,\theta \cdot q \leq 2 \cdot q^2$ quantum basis states and $\theta \leq q$ classical states. $\square$

4. Testing Periodicity on Quantum Finite State Automata

In general, a *stochastic event* on an alphabet Σ is a function $p : \Sigma^* \to [0,1]$. In Section 2.3, for instance, we introduced stochastic events induced by QFA models as means of defining languages (by isolated cut points).

Let $p : \sigma^* \to [0,1]$ be a stochastic event on a *unary* alphabet $\{\sigma\}$. Since the unary string σ^k is basically the unary representation of the number $k \in \mathbb{N}$, for ease of notation, we will be simply writing $p(k)$ instead of $p(\sigma^k)$. Therefore, from this point of view, a unary stochastic event is a function $p : \mathbb{N} \to [0,1]$. We say that p is *d-periodic* for a given integer $d > 0$ (called the *period*) whenever $p(k) = p(k+d)$ holds true for any $k \geq 0$.

Here, we study the problem of deciding whether or not a given *unary* QFA, i.e., a QFA working on a unary alphabet, induces a periodic stochastic event. We formally state such a decision problem for *unary* QFCs:

PERIODICITY
INPUT: A unary QFC A and an integer $d \geq 0$.
QUESTION: Is $\mathscr{E}_A$ a d-periodic stochastic event?

We are going to prove that PERIODICITY *is decidable*, i.e., there exists a deterministic algorithm for its solution. We tackle this problem inspired by [38], where the same question has been posed for the more restricted model of MM-QFAs. First of all, we present some formal tools useful in our investigation. From now on, we will be using the following notations introduced in Section 2.2 and here recalled: for a row vector $\varphi \in \mathbb{C}^m$, we let $\varphi^* \in \mathbb{C}^m$ be the vector satisfying ${\varphi^*}_i = (\varphi_i)^*$ and $\varphi^T \in \mathbb{C}^{m\times 1}$ be the column vector obtained by transposing φ; for a matrix $J \in \mathbb{C}^{m\times m}$, we let $J^* \in \mathbb{C}^{m\times m}$ be the matrix satisfying ${J^*}_{ij} = (J_{ij})^*$.

Definition 1. *Let $A = (\varphi,\ \{U(\sigma)\}_{\sigma\in\Gamma},\ \mathscr{O} = \sum_{c\in C} c \cdot P(c),\ \mathscr{L})$ be a* QFC *with q quantum basis states and with $D = (\alpha, \{M(c)\}_{c\in C}, \beta)$ being the* DFA *with m classical states recognizing the control language $\mathscr{L}$ (see Section 2.1 for the linear representation of a* DFA*). The linear representation of A is defined as $\mathrm{Li}(A) = (\pi, \{J(\sigma)\}_{\sigma\in\Gamma}, \eta)$, where:*

- *$\pi = \varphi \otimes \varphi^* \otimes \alpha$ is a row vector in $\mathbb{C}^{q^2\cdot m}$.*
- *$J(\sigma) = (U(\sigma) \otimes U^*(\sigma) \otimes I^{(m)}) \cdot \sum_{c\in C} P(c) \otimes P^*(c) \otimes M(c)$ for any $\sigma \in \Gamma$ is a matrix in $\mathbb{C}^{q^2\cdot m\times q^2\cdot m}$.*
- *$\eta = \sum_{i=1}^{q} {\mathrm{e}_i}^T \otimes {\mathrm{e}_i}^T \otimes \beta$ is a column vector in $\{0,1\}^{q^2\cdot m\times 1}$, with $\{\mathrm{e}_1, \ldots, \mathrm{e}_q\}$ being the canonical basis of $\mathbb{C}^q$ and the column vector $\beta \in \{0,1\}^{m\times 1}$ being the boolean characteristic vector of the accepting states of the* DFA *D.*

$\mathrm{Li}(A)$ enables us to give an alternative and more manageable representation of the stochastic event $\mathscr{E}_A$ induced by the QFC A, as shown in the following:

Proposition 1. *Let $A = (\varphi, \{U(\sigma)\}_{\sigma\in\Gamma}, \mathscr{O} = \sum_{c\in C} c \cdot P(c), \mathscr{L})$ be a* QFC *for which the classical component is the* DFA *$D = (\alpha, \{M(c)\}_{c\in C}, \beta)$ that recognizes the control language $\mathscr{L}$. Let $\mathrm{Li}(A) = (\pi, \{J(\sigma)\}_{\sigma\in\Gamma}, \eta)$ be the linear representation of A. Then for any $x \in \Sigma^*$, we have*

$$\mathscr{E}_A(x) = \pi \cdot \left(\prod_{i=1}^{|x|} J(x_i)\right) \cdot J(\sharp) \cdot \eta.$$

Proof. By Definition 1 and the linear algebra properties in Section 2.2, we can write

$$\begin{aligned}
\pi\cdot\left(\prod_{i=1}^{|x|} J(x_i)\right) \cdot J(\sharp) \cdot \eta &= (\varphi \otimes \varphi^* \otimes \alpha) \cdot \left(\prod_{i=1}^{|x|}(U(x_i) \otimes U^*(x_i) \otimes I^{(k)}) \cdot \sum_{c\in C} P(c) \otimes P^*(c) \otimes M(c)\right) \\
&\quad \cdot \left((U(\sharp) \otimes U^*(\sharp) \otimes I^{(k)}) \cdot \sum_{c\in C} P(c) \otimes P^*(c) \otimes M(c)\right) \cdot \left(\sum_{j=1}^{q} \mathrm{e}_j^T \otimes \mathrm{e}_j^T \otimes \beta\right) \\
&= \sum_{j=1}^{q} \sum_{y_1\ldots y_{|x|}y_\sharp \in C^*} \left(\varphi \cdot \left(\prod_{i=1}^{|x|} U(x_i) \cdot P(y_i)\right) \cdot U(\sharp) \cdot P(\sharp)\right)_j \cdot \left(\varphi^* \cdot \left(\prod_{i=1}^{|x|} U^*(x_i) \cdot P^*(y_i)\right) \cdot U^*(\sharp) \cdot P^*(\sharp)\right)_j \\
&\quad \cdot \left(\alpha \cdot \left(\prod_{i=1}^{|x|} M(y_i)\right) \cdot M(y_\sharp) \cdot \beta\right) \\
&= \sum_{y_1\ldots y_{|x|}y_\sharp \in C^*} \left(\alpha \cdot \left(\prod_{i=1}^{|x|} M(y_i)\right) \cdot M(y_\sharp) \cdot \beta\right) \cdot \sum_{j=1}^{q} \left| \left(\varphi \cdot \left(\prod_{i=1}^{|x|} U(x_i) \cdot P(y_i)\right) \cdot U(\sharp) \cdot P(\sharp)\right)_j \right|^2 \\
&= \sum_{y_1\ldots y_{|x|}y_\sharp \in C^*} \left(\alpha \cdot \left(\prod_{i=1}^{|x|} M(y_i)\right) \cdot M(y_\sharp) \cdot \beta\right) \cdot \left\| \varphi \cdot \left(\prod_{i=1}^{|x|} U(x_i) \cdot P(y_i)\right) \cdot U(\sharp) \cdot P(\sharp) \right\|^2 . \qquad (14)
\end{aligned}$$

As pointed out in Section 2.1, we have that $\alpha \cdot \left(\prod_{i=1}^{|x|} M(y_i)\right) \cdot M(y_\sharp) \cdot \beta = 1$ if and only if the string $y_1 \ldots y_{|x|} y_\sharp \in C^*$ is accepted by the DFA D or, equivalently, if and only if $y_1 \ldots y_{|x|} y_\sharp \in \mathscr{L}$. Otherwise, we have $\alpha \cdot \left(\prod_{i=1}^{|x|} M(y_i)\right) \cdot M(y_\sharp) \cdot \beta = 0$. By this observation, we can rewrite (14) as

$$\begin{aligned}
&\pi \cdot \left(\prod_{i=1}^{|x|} J(x_i)\right) \cdot L(\sharp) \cdot \eta \\
&= \sum_{y_1\ldots y_{|x|}y_\sharp \in C^*} \left(\alpha \cdot \left(\prod_{i=1}^{|x|} M(y_i)\right) \cdot M(y_\sharp) \cdot \beta\right) \cdot \left\| \varphi \cdot \left(\prod_{i=1}^{|x|} U(x_i) \cdot P(y_i)\right) \cdot U(\sharp) \cdot P(\sharp) \right\|^2 \\
&= \sum_{y_1\ldots y_{|x|}y_\sharp \in \mathscr{L}} \left\| \varphi \cdot \left(\prod_{i=1}^{|x|} U(x_i) \cdot P(y_i)\right) \cdot U(\sharp) \cdot P(\sharp) \right\|^2 = \mathscr{E}_A(x),
\end{aligned}$$

where the last equality follows from Equations (8) and (9). □

Another tool which will turn out to be useful in the analysis of the decidability of PERIODICITY is the notion of a generating function. Generating functions are a well known formalism in Combinatorics and Discrete Mathematics to represent and manipulate numeric sequences (see, e.g., [69]):

Definition 2. *Given a function $p : \mathbb{N} \to \mathbb{C}$, its generating function is defined as the formal power series $G_p(z) = \sum_{k=0}^{\infty} p(k) \cdot z^k$ for any $z \in \mathbb{C}$ satisfying $|z| < 1$.*

Moreover, we will use the following property of square matrices (see, e.g., [52]):

Lemma 2. *Let the matrix $U \in \mathbb{C}^{n\times n}$ satisfy $\lim_{k\to\infty} U^k = \mathbf{0}$. Then, the matrix $(I^{(n)} - U)^{-1}$ exists, and we have $\sum_{k=0}^{\infty} U^k = (I^{(n)} - U)^{-1}$.*

Let us now focus on the unary QFC $A = (\varphi, \{U(\sigma), U(\sharp)\}, \mathscr{O} = \sum_{c\in C} c \cdot P(c), \mathscr{L})$, with control language $\mathscr{L}$ recognized by the DFA $D = (\alpha, \{M(c)\}_{c\in C}, \beta)$ and with linear representation $\mathrm{Li}(A) = (\pi, \{J(\sigma), J(\sharp)\}, \eta)$. The following limit property holds for the dynamics of $\mathrm{Li}(A)$:

Lemma 3. *For any $z \in \mathbb{C}$ such that $|z| < 1$, we have $\lim_{k\to\infty}(J(\sigma)\cdot z)^k = \mathbf{0}$.*

Proof. Let $\Gamma = \{\sigma, \sharp\}$. We first prove that for any string $x_1 \cdots x_n \in \Gamma^*$, we have

$$\left\| \pi \cdot \left(\prod_{i=1}^{n} J(x_i) \right) \right\| \le 1. \tag{15}$$

In fact, by taking into account from Definition 1 the formal statement of the components of $\mathrm{Li}(A) = (\pi, \{J(\sigma), J(\sharp)\}, \eta)$ and using the linear algebra properties from Section 2.2, we can write

$$\begin{aligned}
&\left\| \pi \cdot \left(\prod_{i=1}^{n} J(x_i) \right) \right\| = \\
&= \left\| \sum_{y_1\ldots y_n \in C^n} \left(\varphi \cdot \left(\prod_{i=1}^{n} U(x_i)\cdot P(y_i) \right) \right) \otimes \left(\varphi^* \cdot \left(\prod_{i=1}^{n} U^*(x_i)\cdot P^*(y_i) \right) \right) \otimes \left(\alpha \cdot \left(\prod_{i=1}^{n} M(y_i) \right) \right) \right\| \\
&\le \sum_{y_1\ldots y_n \in C^n} \left\| \left(\varphi \cdot \left(\prod_{i=1}^{n} U(x_i)\cdot P(y_i) \right) \right) \otimes \left(\varphi \cdot \left(\prod_{i=1}^{n} U(x_i)\cdot P(y_i) \right) \right)^* \otimes \left(\alpha \cdot \left(\prod_{i=1}^{n} M(y_i) \right) \right) \right\| \\
&= \sum_{y_1\ldots y_n \in C^n} \left\| \varphi \cdot \left(\prod_{j=1}^{n} U(x_j)\cdot P(y_j) \right) \right\|^2 \cdot \left\| \alpha \cdot \left(\prod_{i=1}^{n} M(y_i) \right) \right\|,
\end{aligned} \tag{16}$$

where the inequality follows from the triangular inequality, see Section 2.2. Let us focus on the terms in (16). As pointed out in Equation (1), we notice that the m-dimensional boolean row vector $\alpha \cdot (\prod_{i=1}^{n} M(y_i))$ is the characteristic vector of the sole state reached by the DFA D after processing the string $y_1 \cdots y_n \in C^n$. So $\|\alpha \cdot (\prod_{i=1}^{n} M(y_i))\| = 1$ clearly holds true. Moreover, by Lemma 1, we have $\sum_{y_1\ldots y_n \in C^n} \|\varphi \cdot (\prod_{i=1}^{n} U(x_i)\cdot P(y_i))\|^2 = \|\varphi\|^2$. These two observations enable us to rewrite (16) as

$$\begin{aligned}
\left\| \pi \cdot \left(\prod_{i=1}^{n} J(x_i) \right) \right\| &\le \sum_{y_1\ldots y_n \in C^n} \left\| \varphi \cdot \left(\prod_{j=1}^{n} U(x_j)\cdot P(y_j) \right) \right\|^2 \cdot \left\| \alpha \cdot \left(\prod_{i=1}^{n} M(y_i) \right) \right\| \\
&= \sum_{y_1\ldots y_n \in C^n} \left\| \varphi \cdot \left(\prod_{j=1}^{n} U(x_j)\cdot P(y_j) \right) \right\|^2 = \|\varphi\|^2 = 1,
\end{aligned}$$

where the last equality $\|\varphi\|^2 = 1$ comes from the fact that φ is the norm 1 initial superposition of the QFC A. This shows the inequality $\|\pi \cdot (\prod_{i=1}^{n} J(x_i))\| \le 1$ claimed in (15).

Let us now get back to the original problem, that is, to evaluate $\lim_{k\to\infty}(J(\sigma)\cdot z)^k$ for any $z \in \mathbb{C}$ such that $|z| < 1$. By Inequality (15), we have that $\|\pi \cdot (J(\sigma))^k\| \le 1$. In addition, since $|z| < 1$, we have that $\lim_{k\to\infty} |z|^k = 0$. These two facts yield

$$\lim_{k\to\infty} \left\| \pi \cdot (J(\sigma)\cdot z)^k \right\| = \lim_{k\to\infty} |z|^k \cdot \left\| \pi \cdot (J(\sigma))^k \right\| = 0,$$

which, in turn, implies that $\lim_{k\to\infty}(J(\sigma)\cdot z)^k = \mathbf{0}$. □

We are now ready to prove the main result of this section:

Theorem 3. PERIODICITY *is decidable.*

Proof. Given a unary QFC $A = (\varphi, \{U(\sigma), U(\sharp)\}, \mathscr{O} = \sum_{c \in C} c \cdot P(c), \mathscr{L})$ with q quantum basis states and m classical states, and an integer $d \geq 0$, we have to algorithmically decide whether or not the following property holds true:

$$\text{for every } k \in \mathbb{N}, \quad \mathscr{E}_A(k) = \mathscr{E}_A(k+d).$$

(We recall that $\mathscr{E}_A(k)$ stands for $\mathscr{E}_A(\sigma^k)$.) We can equivalently investigate the validity of this property on the linear representation $\mathrm{Li}(A) = (\pi, \{J(\sigma), J(\sharp)\}, \eta)$ of the QFC A. This is due to the fact that according to Proposition 1, the linear representation $\mathrm{Li}(A)$ provides an alternative and equivalent representation of $\mathscr{E}_A(k)$ as

$$\mathscr{E}_A(k) = \pi \cdot \left(\prod_{i=1}^{k} J(\sigma) \right) \cdot J(\sharp) \cdot \eta = \pi \cdot J(\sigma)^k \cdot \eta',$$

where for convenience of notation, we let $\eta' = J(\sharp) \cdot \eta$. Therefore, the periodicity condition we have to decide is written

$$\begin{aligned} \text{for every } k \in \mathbb{N}, \quad & \mathscr{E}_A(k) = \mathscr{E}_A(k+d) \\ & \Leftrightarrow \pi \cdot J(\sigma)^k \cdot \eta' = \pi \cdot J(\sigma)^{k+d} \cdot \eta' \\ & \Leftrightarrow \pi \cdot J(\sigma)^k \cdot \eta' - \pi \cdot J(\sigma)^{k+d} \cdot \eta' = 0 \\ & \Leftrightarrow \pi \cdot (I^{(q^2 \cdot m)} - J(\sigma)^d) \cdot J(\sigma)^k \cdot \eta' = 0. \end{aligned} \tag{17}$$

By using generating functions (see Definition 2), we can express the condition (17) as

$$\sum_{k=0}^{\infty} \left(\pi \cdot (I^{(q^2 \cdot m)} - J(\sigma)^d) \cdot J(\sigma)^k \cdot \eta' \right) \cdot z^k = \pi \cdot (I^{(q^2 \cdot m)} - J(\sigma)^d) \cdot \left(\sum_{k=0}^{\infty} (J(\sigma) \cdot z)^k \right) \cdot \eta' = 0. \tag{18}$$

As shown in Lemma 3, we have that $\lim_{k \to \infty} (J(\sigma) \cdot z)^k \to \mathbf{0}$. Thus, according to Lemma 2, we can replace $\sum_{k=0}^{\infty} (J(\sigma) \cdot z)^k$ with $(I^{(q^2 \cdot m)} - J(\sigma) \cdot z)^{-1}$ in Equation (18), obtaining

$$\pi \cdot (I^{(q^2 \cdot m)} - J(\sigma)^d) \cdot (I^{(q^2 \cdot m)} - J(\sigma) \cdot z)^{-1} \cdot \eta' = 0. \tag{19}$$

To sum up, so far, we have proved that deciding the d-periodicity of $\mathscr{E}_A$ is equivalent to deciding whether or not Equation (19) holds true. To this aim, let us elaborate on Equation (19), starting from the $q^2 \cdot m \times q^2 \cdot m$ matrix $M = (I^{(q^2 \cdot m)} - J(\sigma) \cdot z)$. It is well known (see, e.g., [52]) that

$$M^{-1} = \frac{\mathrm{adj}(M)}{\det(M)},$$

where:

- $\det(M)$ denotes the determinant of M;
- $\mathrm{adj}(M)$ denotes the *adjugate* of M, i.e., the matrix defined as follows: let $M_{[i,j]}$ be the $(q^2 \cdot m - 1) \times (q^2 \cdot m - 1)$ matrix obtained from M by deleting its ith row and jth column; then, $\mathrm{adj}(M)$ is the $q^2 \cdot m \times q^2 \cdot m$ matrix with $(\mathrm{adj}(M))_{ij} = (-1)^{i+j} \cdot \det(M_{[j,i]})$.

So by letting $P(z) = \mathrm{adj}(M) = \mathrm{adj}(I^{(q^2 \cdot m)} - J(\sigma) \cdot z)$ be the $q^2 \cdot m \times q^2 \cdot m$ matrix of polynomials in the variable z, we can equivalently rewrite Equation (19) as

$$\pi \cdot (I^{(q^2 \cdot m)} - J(\sigma)^d) \cdot P(z) \cdot \eta' = 0. \tag{20}$$

The left side of Equation (20) is easily seen to be an effectively constructible polynomial $\gamma(z)$ of degree of at most $q^2 \cdot m$. So in conclusion, we have reduced the decision problem

PERIODICITY to the problem of deciding whether or not $\gamma(z)$ is the null polynomial. This latter problem can be clearly decided by checking whether or not each coefficient of $\gamma(z)$ is zero, whence the claimed decidability result follows. □

The decidability of PERIODICITY leads to the decidability of testing the periodicity of the stochastic events induced by several models of unary QFAs, namely:

Theorem 4. *Testing the periodicity of the stochastic events induced by unary* MO-QFAs, MM-QFAs, QRAs, *or* LQFAs *is decidable.*

Proof. The proof of Theorem 2 constructively provides simulation algorithms to turn any of the listed QFA models into QFCs inducing the same stochastic events. Therefore, to test the periodicity of the stochastic event induced by a MO-QFA, MM-QFA, QRA, or LQFA A, we first construct by Theorem 2 the QFC B such that $\mathscr{E}_B = p_A$. Then, we solve PERIODICITY on B, which is decidable by Theorem 3. □

5. QFCs for Unary Regular Languages

In this section, we are going to tackle the problem of building succinct isolated cut point QFCs for *unary regular languages* by investigating their sizes with respect to that of equivalent classical (DFAs) finite state automata.

As an immediate solution, we can adopt the construction for regular languages on *general* alphabets designed in [25] and quickly addressed in Section 2.5. According to this construction, given a unary regular language $L \subseteq \sigma^*$ recognized by a (minimal) DFA with T states in the initial path and P states in the cycle (see Theorem 1), we could obtain an isolated cut point QFC A for L with 2 quantum basis states and $3 \cdot (T + P)$ classical states. Nevertheless, the size of A is far from being satisfactory given that its classical component alone, namely the incorporated DFA, is three times bigger than the original DFA for L. This size inefficiency is mainly due to the generality of the construction in [25], which was primarily devised to show that isolated cut point QFCs can recognize all regular languages without a particular emphasis on size efficiency. In addition, the resulting QFC A recognizes L by exhibiting a deterministic accepting behavior, this suggesting that the descriptional power of the quantum component is not fully exploited.

In the next section, we provide a construction scheme specifically calibrated for the unary case and aiming to obtain succinct isolated cut point QFCs. Such a scheme specifies the role of both the classical and the quantum component in the resulting QFCs as well as the way these components cooperate. Owing to this modular design framework, two versions of isolated cut point QFCs for unary regular languages are then constructed by simply adopting two different types of quantum components. This "plug-and-play" approach is clearly open to be further exploited, e.g., by hardwiring other succinct quantum components proposed in the literature [20,35,37,68].

5.1. The Construction Scheme

By Theorem 1, any unary regular language $L \subseteq \sigma^*$ can be regarded as the disjoint union of two languages: the finite language $L_T = L \cap \sigma^{<T}$ and the ultimately periodic language $L_P = L \cap \sigma^{\geq T}$. This observation inspires our modular framework for building an isolated cut point QFC A for the language L. Roughly speaking, the classical and the quantum components of A cooperate to perform the following tasks:

1. **Classical component dealing with** L_T: The classical component, i.e., the DFA D recognizing the control language of A, takes care of the finite language L_T as long as the input length is less than T. More precisely, D counts the input symbols, and if this count does not exceed $T - 1$, then the input string is accepted or rejected according to membership in L_T. If the count exceeds $T - 1$, then D "activates" the quantum component to deal with L_P.
2. **Quantum component dealing with** L_P: The quantum component M is "activated" by D on input strings of length exceeding $T - 1$ in order to recognize with an isolated

cut point the ultimately periodic language L_P. More precisely, M is going to recognize the P-periodic language that coincides with L_P on strings with lengths exceeding $T-1$.

5.2. The First Construction

We provide a first implementation out of the construction scheme outlined in Section 5.1, featuring a first type of *quantum component* for our isolated cut point QFC A recognizing the unary language $L = L_T \cup L_P$. Such a component takes care of the ultimately periodic language L_P by recognizing with an isolated cut point the P-periodic language

$$L_{P\circlearrowleft} = \{\sigma^{(T+i) \bmod P + h \cdot P} \mid 0 \leq i < P,\ h \geq 0, \text{ and } \sigma^{T+i} \in L_P\}.$$

It is not hard to see that $L_{P\circlearrowleft}$ coincides with L_P when restricted to strings with lengths exceeding $T-1$: namely, $L_P = L_{P\circlearrowleft} \cap \sigma^{\geq T}$. This quantum component is equipped with an observable to be measured at every step and with three possible outcomes: g (go), a (accept), and r (reject). Upon processing the input symbol σ, the observable outcome will always be g, while on $\sharp$, the outcome will be either a or r depending, respectively, on whether or not the input string belongs to $L_{P\circlearrowleft}$. Given that $L_P = L_{P\circlearrowleft} \cap \sigma^{\geq T}$, the control language $\mathscr{L}$ of A is designed so that acceptance by this quantum component takes place only on the strings with lengths exceeding $T-1$; on the other hand, for strings with lengths less than or equal to $T-1$, the acceptance is completely up to the classical component. Precisely, $\mathscr{L} \subseteq \{a, r, g\}^*$ is designed so that:

- Only for input strings with lengths greater than or equal to T, the outcomes a and r are considered for accepting/rejecting; correspondingly, the outcome g is ignored after T steps.
- Instead, for input strings with lengths less than T, the outcomes a and r are ignored, and the classical component counts the number of g to establish membership in L_T.

Summing up, the classical component of the QFC A is the DFA D, whose state diagram is depicted in Figure 6, which recognizes the control language language $\mathscr{L} \subseteq \{a, r, g\}^*$ defined to be the set $\mathscr{L} = \{y \in g^*\{a, r\} \mid \sigma^{|y|_g} \in L_T \text{ or } y \in g^{\geq T}a\}$.

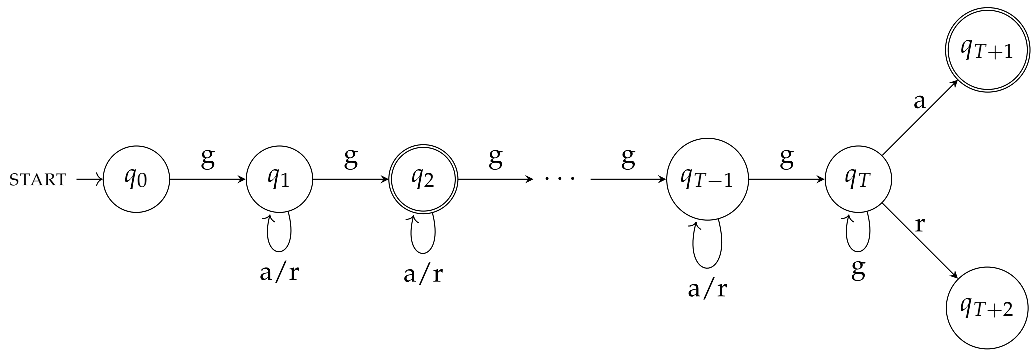

Figure 6. The state diagram of the classical component of the QFC A, i.e., the DFA D recognizing the control language $\mathscr{L} = \{y \in g^*\{a, r\} \mid \sigma^{|y|_g} \in L_T \text{ or } y \in g^{\geq T}a\} \subseteq \{a, r, g\}^*$.

Let us quickly discuss the architecture and behavior of the DFA D in Figure 6. In the initial path $\{q_0, \ldots, q_{T-1}\}$, the accepting states are set according to L_T. Namely, for every $0 \leq i < T$, we designate the state q_i as accepting if and only if $\sigma^i \in L_T$. This ensures the correct acceptance of input strings (with lengths not exceeding $T-1$ and) belonging to L_T. If the state q_T is reached, then an input string with a length of at least T is under processing, and so membership in L_P has to be checked. To this aim, once reaching q_T, the DFA D stays in q_T as long as the input symbol σ is read (i.e., as long as the outcome g is processed by D). Upon reaching the endmarker $\sharp$, the DFA D simply acknowledges the outcome a or r of the quantum component to witness, respectively, membership or not in L_P.

Formally, we define the isolated cut point QFC $A = (\varphi,\ \{U(\sigma),\ U(\sharp)\},\ \mathscr{O},\ \mathscr{L})$ for the unary regular language $L = L_T \cup L_P$ as follows:

- $\varphi = \mathrm{e}_1 \in \mathbb{C}^{2\cdot P}$ is the initial superposition;
- $U(\sigma) = Z \oplus I^{(P)}$, where $Z \in \{0,1\}^{P\times P}$ is the matrix representing the cyclic permutation, i.e., Z has 1 at the $(i, i+1)$th entries for every $1 \leq i < P$ and at the $(P,1)$th entry, while all the other entries are 0;
- $U(\sharp) = \begin{pmatrix} \mathbf{0} & I^{(P)} \\ I^{(P)} & \mathbf{0} \end{pmatrix}$;
- $\mathscr{O} = g\cdot P(g) + a\cdot P(a) + r\cdot P(r)$ is the $2\cdot P \times 2\cdot P$ observable for which the projectors are defined as $P(g) = I^{(P)} \oplus \mathbf{0}$, $P(a) = \mathbf{0} \oplus \mathrm{diag}(\chi_{L_{P\circlearrowleft}})$, and $P(r) = \mathbf{0} \oplus (I^{(P)} - P(a))$, where the vector $\chi_{L_{P\circlearrowleft}} \in \{0,1\}^P$ satisfies $(\chi_{L_{P\circlearrowleft}})_i = 1$ if and only if $\sigma^{i-1} \in L_{P\circlearrowleft}$ for every $1 \leq i \leq P$ (we notice that $\chi_{L_{P\circlearrowleft}}$ is sometimes referred to as the characteristic vector of the P-periodic language $L_{P\circlearrowleft}$);
- $\mathscr{L} = \{y \in g^*\{a,r\} \mid \sigma^{|y|_g} \in L_T \text{ or } y \in g^{\geq T}a\} \subseteq \{a,r,g\}^*$ is the control language recognized by the DFA D in Figure 6.

Each superposition of the QFC A can be regarded as being divided into two halves. Along its computation and *before* reaching the endmarker, A implements in the left half the dynamic of the quantum component taking care of $L_{P\circlearrowleft}$ by repeatedly applying $U(\sigma)$. In this phase, any measuring of $\mathscr{O}$ yields the outcome g with certainty. Upon reading the endmarker, the left half of the superposition is swapped with the right half by applying $U(\sharp)$. At this point, the final measuring of $\mathscr{O}$ takes place, yielding a or r according to membership or not, respectively, in $L_{P\circlearrowleft}$. Figure 7 quickly exemplifies the dynamic of A.

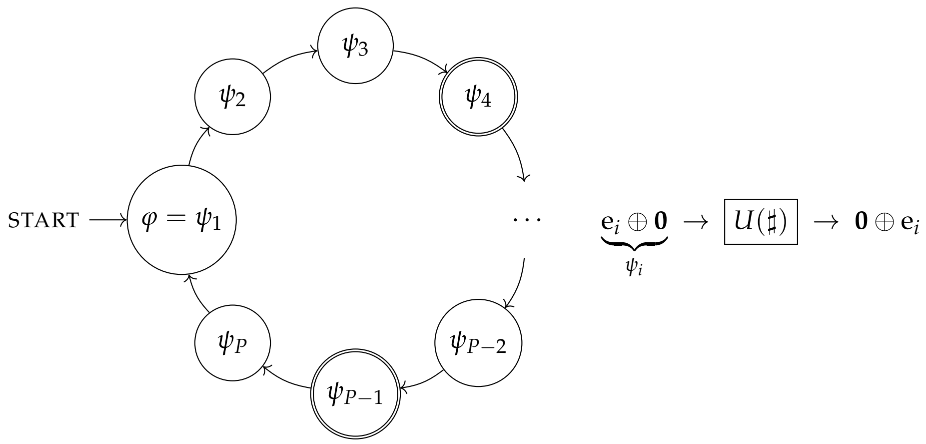

Figure 7. *Left:* The cyclic dynamic by $U(\sigma)$ of the superposition of the QFC A upon reading the input symbol σ. For a better display, we let $\psi_i = \mathrm{e}_i \oplus \mathbf{0}$, with $\mathrm{e}_i, \mathbf{0} \in \{0\}^{1\times P}$. *Right:* Superposition swapping by $U(\sharp)$ upon reading the endmarker $\sharp$.

Let $g^k y_\sharp$, with $y_\sharp \in \{a,r\}$, be the sequence of outcomes from measuring $\mathscr{O}$ along the computation of A on the input string $\sigma^k\sharp$.

- Suppose that $k < T$: It is not hard to see from the definition of the control language $\mathscr{L} = \{y \in g^*\{a,r\} \mid \sigma^{|y|_g} \in L_T \text{ or } y \in g^{\geq T}a\}$ that A exhibits with certainty an accepting computation on $\sigma^k\sharp$ if and only if $\sigma^{|g^k y_\sharp|_g} \in L_T \Leftrightarrow \sigma^k \in L_T$. Also, it is easy to verify that an accepting computation occurs with probability 0 whenever $\sigma^k \notin L_T$.
- Suppose that $k \geq T$: Again, from the definition of $\mathscr{L}$, it is not hard to see that A exhibits with certainty an accepting computation on $\sigma^k\sharp$ if and only if $g^k y_\sharp = g^k a$, and this happens if and only if $\sigma^k \in L_{P\circlearrowleft} \cap \sigma^{\geq T} = L_P$. On the other hand, no accepting computation can occur whenever $\sigma^k \notin L_P$.

In conclusion, the resulting QFC A deterministically recognizes the unary regular language L (i.e., A accepts with certainty (respectively, with probability 0) the strings in (respectively, not in) L).

Concerning the *size* of A, we have $2 \cdot P$ quantum basis states and $T + 3$ classical states, which is slightly worse than the size (i.e., $T + P$ states) of the DFA for L. However, in the next section, we are going to improve the size of A by hardwiring a smaller quantum component.

5.3. Reducing the Size

Our first type of quantum component used in Section 5.2 is actually a DFA "disguised" as a QFA. This leads to a deterministic recognition of unary regular languages but actually does not allow one to fully exploit the high descriptional power of the quantum paradigm (where an error probability in the classification of input strings is potentially permitted).

Here, we are going to build a smaller QFC for the unary regular language $L = L_T \cup L_P$ by designing a more succinct quantum component for the P-periodic language $L_{P\circlearrowleft}$: recall that $L_P = L_{P\circlearrowleft} \cap \sigma^{\geq T}$. To this aim, we need a result on the ability of MO-QFAs to induce linear approximations of periodic stochastic events by using a very limited amount of basis states (see the beginning of Section 4 for the definition of a periodic stochastic event):

Theorem 5 ([39])**.** *For any P-periodic stochastic event p, there exists a unary* MO-QFA *with at most $2\sqrt{6P} + 25$ basis states inducing the P-periodic stochastic event $\mu \cdot p + \tau$ for suitable $\mu, \tau \in \mathbb{R}$ satisfying $\mu > 0$, $\tau \geq 0$, and $\mu + \tau \leq 1$.*

Now, with any P-periodic language $\Pi \subseteq \sigma^*$, we can associate its characteristic function $\chi_\Pi : \mathbb{N} \to \{0, 1\}$, defined as

$$\chi_\Pi(k) = \begin{cases} 1 & \text{if } \sigma^k \in \Pi \\ 0 & \text{otherwise,} \end{cases}$$

which is clearly a P-periodic stochastic event. By Theorem 5, there exists a MO-QFA A_Π with $2\sqrt{6P} + 25$ basis states inducing the P-periodic stochastic event $\mu \cdot \chi_\Pi + \tau$. It is easy to see that we can use A_Π to recognize the language Π with cut point $\lambda = \tau + \frac{\mu}{2}$ and radius of isolation $\varrho = \frac{\mu}{2}$. As a matter of fact, it is always possible to have an isolated cut point $\lambda \geq \frac{1}{2}$, paying with an additional basis state (see [39] for technical details). This enables us to state

Theorem 6. *Any P-periodic language can be recognized with isolated cut point $\lambda \geq \frac{1}{2}$ by an* MO-QFA *featuring no more than $2\sqrt{6P} + 26$ basis states.*

Therefore, given the P-periodic language $L_{P\circlearrowleft}$, we let $B = (\varphi_B, \{U_B(\sigma), U_B(\sharp)\}, \mathscr{O}_B)$ with observable $\mathscr{O}_B = a \cdot P_B(a) + r \cdot P_B(r)$ be the isolated cut point MO-QFA for $L_{P\circlearrowleft}$ with $\ell = 2\sqrt{6P} + 26$ basis states resulting from Theorem 6. We can replace the quantum component in the QFC A detailed in Section 5.2 with B, thus obtaining our second version $\tilde{A} = (\varphi, \{U(\sigma), U(\sharp)\}, \mathscr{O}, \mathscr{L})$ of a QFC for the unary regular language L, where:

- $\varphi = \varphi_B \oplus \mathbf{0} \in \mathbb{C}^{2\cdot\ell}$ is the initial superposition, with $\|\varphi\| = \|\varphi_B\| = 1$;
- $U(\sigma) = U_B(\sigma) \oplus I^{(\ell)}$ and $U(\sharp) = \begin{pmatrix} \mathbf{0} & U_B(\sharp) \\ U_B(\sharp) & \mathbf{0} \end{pmatrix}$ are the unitary evolution matrices;
- $\mathscr{O} = g \cdot P(g) + a \cdot P(a) + r \cdot P(r)$ is the $2 \cdot \ell \times 2 \cdot \ell$ observable for which the projectors are defined as $P(g) = I^{(\ell)} \oplus \mathbf{0}$, $P(a) = \mathbf{0} \oplus P_B(a)$, and $P(r) = \mathbf{0} \oplus P_B(r)$;
- $\mathscr{L} = \{y \in g^*\{a, r\} \mid \sigma^{|y|_g} \in L_T \text{ or } y \in g^{\geq T}a\} \subseteq \{a, r, g\}^*$ is the control language recognized by the DFA D in Figure 6.

By reasoning as in Section 5.2, one may verify that $\tilde{A}$ is an isolated cut point QFC for the unary regular language L. Thus, we can conclude with

Theorem 7. *Let L be a unary regular language recognized by a* DFA *with T states in the initial path and P states in the cycle. Then, there exists a* QFC *recognizing L with isolated cut point $\lambda \geq \frac{1}{2}$ and featuring $4\sqrt{6P} + 52$ quantum basis states and $T + 3$ classical states.*

6. Conclusions

6.1. Summary of Novel Results

We have studied quantum finite state automata with control language (QFCs), which represent a hybrid computational model embedding a deterministic finite state automaton (DFA) as the control unit coupled with a quantum finite state automaton (QFA) as the processor. An interesting feature of QFCs is their ability to simulate several types of QFAs, thus providing a unifying framework within which to study the properties of variants of QFAs.

Owing to this simulation capability, we have shown how to reproduce measure-once, measure-many, reversible, and Latvian QFAs (MO-QFAs, MM-QFAs, QRAs, and LQFAs, respectively) by QFCs, emphasizing the simulation costs in terms of quantum and classical states. Then, we have proved the decidability of the problem PERIODICITY of testing the periodicity of the stochastic event induced by a given QFC. This, together with the listed simulation results, proves the decidability of PERIODICITY for MO-QFAs, MM-QFAs, QRAs, and LQFAs as well.

Next, we have compared the succinctness of QFCs and classical finite state automata on unary regular language recognition. Given a unary regular language recognized by a DFA consisting of an initial path of T states joined to a cycle of P states, we have designed an equivalent isolated cut point QFC with $T + 3$ classical states and $O(\sqrt{P})$ quantum basis states. This shows that a finite memory hybrid computational model can be quadratically more succinct than a classical one on the whole relevant class of unary regular languages.

6.2. Research Outlooks

Several possible lines of future research can be foreseen. Concerning QFCs as a unifying framework to study finite memory quantum paradigms, it would be worth investigating the decidability of other typical problems on QFCs (e.g., testing language emptiness, finiteness, and universality). Positive decidability results would carry on to all simulated types of QFAs. As a consequence, it would be interesting to study the possibility of simulating other types of QFAs (e.g., [20,22,70–72]) within the framework of QFCs. Another investigation could aim at characterizing the recognition power of isolated cut point QFCs as a function of the control language. For instance, one could study the class of languages recognized by isolated cut point QFCs using control languages that are, e.g., bounded, star-free, piecewise (locally) testable, commutative, etc.

Concerning QFCs as a theoretical model for the analysis of possible hardware advantages of hybrid architectures, it would be worth investigating other tasks, beyond unary regular language recognition, for which QFCs may possibly outperform classical devices. For instance, by plugging suitable types of quantum components within our modular construction scheme, one could consider building succinct isolated cut point QFCs for wider classes of regular languages, e.g., commutative or bounded regular languages, as well as for other classes of subregular languages [73,74]. Also, it would be interesting either to improve our construction for general unary regular languages, e.g., by hardwiring more size-efficient quantum components, or to show its size optimality.

Author Contributions: Conceptualization, C.M., B.P. and P.R.; methodology, C.M., B.P. and P.R.; formal analysis, C.M., B.P. and P.R.; investigation, C.M., B.P. and P.R.; writing—original draft preparation, C.M., B.P. and P.R.; writing—review and editing, C.M., B.P. and P.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Acknowledgments: The authors wish to thank the anonymous referees for valuable and helpful comments and remarks. C.M. and B.P. are members of GNCS-INdAM.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

diag(η)	The diagonal matrix having the vector η on its main diagonal
det (M)	The determinant of the matrix M
adj(M)	The adjugate matrix of the matrix M
DFA	Deterministic Finite State Automaton
QFA	Quantum Finite State Automaton
MO-QFA	Measure-Once Quantum Finite State Automaton
MM-QFA	Measure-Many Quantum Finite State Automaton
LQFA	Latvian Quantum Finite State Automaton
QRA	Quantum Reversible Finite State Automaton
QFC	Quantum Finite State Automaton with Control Language
PERIODICITY	The decision problem PERIODICITY for QFCs

References

1. Grover, L.K. A fast quantum mechanical algorithm for database search. In Proceedings of the 28th Symp. on Theory of Computing (STOC), Philadelphia, PA, USA, 22–24 May 1996; ACM: New York, NY, USA, 1996; pp. 212–219. [CrossRef]
2. Shor, P.W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.* **1997**, *26*, 1484–1509. [CrossRef]
3. Bartlett, B.; Dutt, A.; Fan, S. Deterministic photonic quantum computation in a synthetic time dimension. *Optica* **2021**, *8*, 1515–1523. [CrossRef]
4. Knill, E.; Laflamme, R.; Milburn, G. A scheme for efficient quantum computation with linear optics. *Nature* **2001**, *409*, 46–52. [CrossRef] [PubMed]
5. Kok, P.; Munro, W.J.; Nemoto, K.; Ralph, T.C.; Dowling, J.P.; Milburn, G.J. Linear optical quantum computing with photonic qubits. *Rev. Mod. Phys.* **2007**, *79*, 135–174. [CrossRef]
6. Mereghetti, C.; Palano, B.; Cialdi, S.; Vento, V.; Paris, M.G.A.; Olivares, S. Photonic realization of a quantum finite automaton. *Phys. Rev. Res.* **2020**, *2*, 013089. [CrossRef]
7. Madsen, L.S.; Laudenbach, F.; Askarani, M.F.; Rortais, F.; Vincent, T.; Bulmer, J.F.F.; Miatto, F.M.; Neuhaus, L.; Helt, L.G.; Collins, M.J.; et al. Quantum computational advantage with a programmable photonic processor. *Nature* **2022**, *606*, 75–81. [CrossRef]
8. Slussarenko, S.; Pryde, G.J. Photonic quantum information processing: A concise review. *Appl. Phys. Rev.* **2019**, *6*, 041303. [CrossRef]
9. Tian, Y.; Feng, T.; Luo, M.; Zheng, S.; Zhou, X. Experimental demonstration of quantum finite automaton. *npj Quantum Inf.* **2019**, *5*, 67. [CrossRef]
10. Burkard, G.; Ladd, T.D.; Pan, A.; Nichol, J.M.; Petta, J.R. Semiconductor spin qubits. *Rev. Mod. Phys.* **2023**, *95*, 025003. [CrossRef]
11. Flouris, K.; Jimenez, M.M.; Herrmann, H.J. Curvature-induced quantum spin-Hall effect on a Möbius strip. *Phys. Rev. B* **2022**, *105*, 235122. [CrossRef]
12. Gali, A. Recent advances in the ab initio theory of solid-state defect qubits. *Nanophotonics* **2023**, *12*, 359–397. [CrossRef]
13. Usmani, I.; Afzelius, M.; de Riedmatten, H.; Gisin, N. Mapping multiple photonic qubits into and out of one solid-state atomic ensemble. *Nat. Commun.* **2010**, *1*, 12. [CrossRef] [PubMed]
14. Wolfowicz, G.; Heremans, F.J.; Anderson, C.P.; Kanai, S.; Seo, H.; Gali, A.; Galli, G.; Awschalom, D.D. Quantum guidelines for solid-state spin defects. *Nat. Rev. Mater.* **2021**, *6*, 906–925. [CrossRef]
15. Xue, Q. Interface between condensed matter physics and quantum information science. *Quantum Front.* **2022**, *1*, 1. [CrossRef]
16. Hirvensalo, M. Quantum automata with open time evolution. *Int. J. Nat. Comput. Res.* **2010**, *1*, 70–85. [CrossRef]
17. Bianchi, M.P.; Mereghetti, C.; Palano, B. Complexity of promise problems on classical and quantum automata. In *Computing with New Resources*; LNCS; Springer: Berlin/Heidelberg, Germany, 2014; Volume 8808, pp. 161–175. [CrossRef]
18. Qiu, D.; Li, L.; Mateus, P.; Sernadas, A. Exponentially more concise quantum recognition of non-RMM regular languages. *J. Comput. Syst. Sci.* **2015**, *81*, 359–375. [CrossRef]
19. Zheng, S.; Qiu, D.; Gruska, J.; Li, L.; Mateus, P. State succinctness of two-way finite automata with quantum and classical states. *Theor. Comput. Sci.* **2013**, *499*, 98–112. [CrossRef]
20. Zheng, S.; Qiu, D.; Li, L.; Gruska, J. One-way finite automata with quantum and classical states. In *Languages Alive*; LNCS; Springer: Berlin/Heidelberg, Germany, 2012; Volume 7300, pp. 273–290. [CrossRef]
21. Li, L.; Qiu, D. Lower bounds on the size of semi-quantum finite automata. *Theor. Comput. Sci.* **2016**, *623*, 75–82. [CrossRef]

22. Zheng, S.; Gruska, J.; Qiu, D. On the state complexity of semi-quantum finite automata. *RAIRO-Theor. Inf. Appl.* **2014**, *48*, 187–207. [CrossRef]
23. Zheng, S.; Qiu, D.; Gruska, J. Power of the interactive proof systems with verifiers modeled by semi-quantum two-way finite automata. *Inf. Comput.* **2015**, *241*, 197–214. [CrossRef]
24. Bertoni, A.; Mereghetti, C.; Palano, B. Quantum computing: 1-way quantum automata. In Proceedings of the 7th Conference on Developments in Language Theory (DLT), Szeged, Hungary, 7–11 July 2003; LNCS; Springer: Berlin/Heidelberg, Germany, 2003, Volume 2710, pp. 1–20. [CrossRef]
25. Mereghetti, C.; Palano, B. Quantum finite automata with control language. *RAIRO-Theor. Inf. Appl.* **2006**, *40*, 315–332. [CrossRef]
26. Paz, A. *Introduction to Probabilistic Automata*; Academic Press: Cambridge, MA, USA, 1971.
27. Rabin, M.O. Probabilistic automata. *Inf. Control* **1963**, *6*, 230–245. [CrossRef]
28. Ambainis, A.; Freivalds, R. 1-way quantum finite automata: Strengths, weaknesses and generalizations. In Proceedings of the 39th Symp. on Foundations of Computer Science (FOCS), Palo Alto, CA, USA, 8–11 November 1998; IEEE Computer Society: Piscataway, NJ, USA, 1998; pp. 332–342. [CrossRef]
29. Brodsky, A.; Pippenger, N. Characterizations of 1-way quantum finite automata. *SIAM J. Comput.* **2002**, *31*, 1456–1478. [CrossRef]
30. Kondacs, A.; Watrous, J. On the power of quantum finite state automata. In Proceedings of the 38th Annual Symposium on Foundations of Computer Science, Miami Beach, FL, USA, 20–22 October 1997; pp. 66–75. [CrossRef]
31. Moore, C.; Crutchfield, J.P. Quantum automata and quantum grammars. *Theor. Comput. Sci.* **2000**, *237*, 275–306. [CrossRef]
32. Nayak, A. Optimal lower bounds for quantum automata and random access codes. In Proceedings of the 40th Annual Symposium on Foundations of Computer Science (FOCS), New York, NY, USA, 17–19 October 1999; IEEE Computer Society: Piscataway, NJ, USA, 1999; pp. 369–376. [CrossRef]
33. Ambainis, A.; Beaudry, M.; Golovkins, M.; Kikusts, A.; Mercer, M.; Thérien, D. Algebraic results on quantum automata. *Theory Comput. Syst.* **2006**, *39*, 165–188. [CrossRef]
34. Mercer, M. Applications of Algebraic Automata Theory to Quantum Finite Automata. Ph.D. Thesis, McGill University, Montreal, QC, Canada, 2007.
35. Ablayev, F.; Gainutdinova, A. On the lower bounds for one-way quantum automata. In Proceedings of the 25th International Symposium Mathematical Foundations of Computer Science (MFCS), Bratislava, Slovakia, 28 August–1 September 2000; LNCS; Springer: Berlin/Heidelberg, Germany, 2000; Volume 1893, pp. 132–140. [CrossRef]
36. Bianchi, M.P.; Mereghetti, C.; Palano, B. Size lower bounds for quantum automata. *Theor. Comput. Sci.* **2014**, *551*, 102–115. [CrossRef]
37. Bianchi, M.P.; Mereghetti, C.; Palano, B. Quantum finite automata: Advances on Bertoni's ideas. *Theor. Comput. Sci.* **2017**, *664*, 39–53. [CrossRef]
38. Bianchi, M.P.; Palano, B. Behaviours of unary quantum automata. *Fundam. Informaticae* **2010**, *104*, 1–15. [CrossRef]
39. Mereghetti, C.; Palano, B. On the size of one-way quantum finite automata with periodic behaviors. *RAIRO-Theor. Inf. Appl.* **2002**, *36*, 277–291. [CrossRef]
40. Li, L.; Feng, Y. On hybrid models of quantum finite automata. *J. Comput. Syst. Sci.* **2015**, *81*, 1144–1158. [CrossRef]
41. Li, L.; Qiu, D. Determining the equivalence for one-way quantum finite automata. *Theor. Comput. Sci.* **2008**, *403*, 42–51. [CrossRef]
42. Huang, F. On coverings of products of uninitialized sequential quantum machines. *Int. J. Theor. Phys.* **2019**, *58*, 1418–1440. [CrossRef]
43. Huang, F. Decompositions of average probability uninitialized sequential quantum machines. *Soft Comput.* **2022**, *26*, 5965–5974. [CrossRef]
44. Gruska, J. Algebraic methods in quantum informatics. In Proceedings of the 9th International Conference Algebraic Informatics (CAI), Thessalonkik, Greece, 21–25 May 2007; LNCS; Springer: Berlin/Heidelberg, Germany, 2007; Volume 4728, pp. 87–111. [CrossRef]
45. Li, L.; Qiu, D.; Zou, X.; Li, L.; Wu, L.; Mateus, P. Characterizations of one-way general quantum finite automata. *Theor. Comput. Sci.* **2012**, *419*, 73–91. [CrossRef]
46. Parikh, R. On context-free languages. *J. ACM* **1966**, *13*, 570–581. [CrossRef]
47. Salomaa, A. Theorems on the representation of events in Moore-automata. *Ann. Univ. Turku, Ser. A I* **1964**, *69*, 13.
48. Hopcroft, J.E.; Motwani, R.; Ullman, J.D. *Introduction to Automata Theory, Languages, and Computation*, 3rd ed.; Addison-Wesley: Boston, MA, USA, 2006.
49. Dirac, P. *The Principles of Quantum Mechanics*, 4th ed.; Oxford University Press: Oxford, UK, 1988.
50. Hughes, R. *The Structure and Interpretation of Quantum Mechanics*; Harvard University Press: Cambridge, MA, USA, 1992.
51. Marcus, M.; Minc, H. *Introduction to Linear Algebra*; The Macmillan Company: New York, NY, USA, 1965.
52. Shilov, G.E. *Linear Algebra*; Prentice-Hall: Hoboken, NJ, USA, 1971.
53. Qiu, D.; Li, L.; Mateus, P.; Gruska, J. *Quantum Finite Automata*; CRC Press: Boca Raton, FL, USA, 2016; pp. 113–144.
54. Dwork, C.; Stockmeyer, L. A time complexity gap for two–way probabilistic finite-state automata. *SIAM J. Comput.* **1990**, *19*, 1011–1023. [CrossRef]
55. Freivalds, R. Probabilistic two-way machines. In Proceedings of the 10th International Symposium Mathematical Foundations of Computer Science (MFCS), Strbske Pleso, Czechoslovakia, 31 August–4 September 1981; LNCS; Springer: Berlin/Heidelberg, Germany, 1981; Volume 118, pp. 33–45. [CrossRef]

56. Bertoni, A.; Carpentieri, M. Analogies and differences between quantum and stochastic automata. *Theor. Comput. Sci.* **2001**, *262*, 69–81. [CrossRef]
57. Bertoni, A.; Carpentieri, M. Regular languages accepted by quantum automata. *Inf. Comput.* **2001**, *15*, 174–182. [CrossRef]
58. Benioff, P. Quantum mechanical Hamiltonian models of Turing machines. *J. Stat. Phys.* **1982**, *29*, 515–546. [CrossRef]
59. Deutsch, D. Quantum theory, the Church-Turing principle and the universal quantum computer. *R. Soc. Lond. Proc. Ser. A-Math. Phys. Sci.* **1985**, *400*, 97–117. [CrossRef]
60. Gruska, J. *Quantum Computing*; McGraw-Hill: New York, NY, USA, 2000.
61. Nielsen, M.; Chuang, I. *Quantum Computation and Quantum Information*; Cambridge University Press: Cambridge, UK, 2011.
62. Pin, J.E. *Varieties of Formal Languages*; North Oxford Academic: Oxford, UK, 1986.
63. Bednárová, Z.; Geffert, V.; Mereghetti, C.; Palano, B. Boolean language operations on nondeterministic automata with a pushdown of constant height. *J. Comput. Syst. Sci.* **2017**, *90*, 99–114. [CrossRef]
64. Gruska, J. Descriptional complexity issues in quantum computing. *J. Autom. Lang. Comb.* **2000**, *5*, 191–218. [CrossRef]
65. Holzer, M.; Kutrib, M. Descriptional and computational complexity of finite automata—A survey. *Inf. Comput.* **2011**, *209*, 456–470. [CrossRef]
66. Jakobi, S.; Meckel, K.; Mereghetti, C.; Palano, B. Queue automata of constant length. In Proceedings of the 15th International Workshop on Descriptional Complexity of Formal Systems (DCFS), London, ON, Canada, 22–25 July 2013; LNCS; Springer: Berlin/Heidelberg, Germany, 2013; Volume 8031, pp. 124–135. [CrossRef]
67. Yu, S. State complexity of regular languages. *J. Autom. Lang. Comb.* **2001**, *6*, 221–234. [CrossRef]
68. Ambainis, A.; Nahimovs, N. Improved constructions of quantum automata. *Theor. Comput. Sci.* **2009**, *410*, 1916–1922. [CrossRef]
69. Wilf, H.S. *Generatingfunctionology*, 3rd ed.; CRC Press: Boca Raton, FL, USA, 2005.
70. Bhatia, A.S.; Kumar, A. Quantum ω-Automata over Infinite Words and Their Relationships. *Int. J. Theor. Phys.* **2019**, *58*, 878–889. [CrossRef]
71. Bhatia, A.S.; Kumar, A. On the power of two-way multihead quantum finite automata. *RAIRO-Theor. Inf. Appl.* **2019**, *53*, 19–35. [CrossRef]
72. Piazza, C.; Romanello, R. Mirrors and memory in quantum automata. In Proceedings of the International Conference Quantitative Evaluation of Systems (QUEST), Warsaw, Poland, 12–16 September 2022; LNCS; Springer: Berlin/Heidelberg, Germany, 2022; Volume 13479, pp. 359–380. [CrossRef]
73. Choffrut, C.; Malcher, A.; Mereghetti, C.; Palano, B. First-order logics: Some characterizations and closure properties. *Acta Inform.* **2012**, *49*, 225–248. [CrossRef]
74. Truthe, B. *Hierarchy of Subregular Language Families*; Technical report; Universitätsbibliothek Gießen, Institut für Informatik: Giessen, Germany, 2018. [CrossRef]