Complexity of Promise Problems on Classical and Quantum Automata

Maria Paola Bianchi, Carlo Mereghetti(✉), and Beatrice Palano

Dipartimento di Informatica, Università degli Studi di Milano via Comelico 39, 20135 Milano, Italy
{bianchi,mereghetti,palano}@di.unimi.it

Abstract. We consider the promise problem A^{N,r_1,r_2} on a *unary* alphabet $\{\sigma\}$ studied by Gruska et al. in [21]. This problem is formally defined as the pair $A^{N,r_1,r_2} = (A^{N,r_1}_{yes}, A^{N,r_2}_{no})$, with $0 \leq r_1 \neq r_2 < N$, $A^{N,r_1}_{yes} = \{\sigma^n \mid n \equiv r_1 \bmod N\}$ and $A^{N,r_2}_{no} = \{\sigma^n \mid n \equiv r_2 \bmod N\}$. There, it is shown that a measure-once one-way quantum automaton can solve exactly A^{N,r_1,r_2} with only 3 basis states, while any one-way deterministic finite automaton requires d states, d being the smallest integer such that $d \mid N$ and $d \nmid (r_2 - r_1) \bmod N$. Here, we introduce the promise problem $\textsc{Diof}^{a,N}_{r_1,r_2}$ as an extension of A^{N,r_1,r_2} to *general* alphabets. Even for this problem, we show the same descriptional superiority of the quantum paradigm over one-way deterministic automata. Moreover, we prove that even by adding features to classical automata, namely nondeterminism, probabilism, two-way motion, we cannot obtain automata for A^{N,r_1,r_2} and $\textsc{Diof}^{a,N}_{r_1,r_2}$ smaller than one-way deterministic.

Keywords: Classical and quantum automata · Promise problem · Descriptional complexity

1 Introduction

Several features have been added to the original model of *one–way deterministic* finite automaton (1DFA) [38]. Thus, we saw one-way nondeterminism (1NFA) [38], one-way probabilism (1PFA) [37], and the ability of scanning input strings back and forth, yielding the definition of *two-way* devices (e.g., 2DFA) [39]. However, simulation results show that the computational power of 1NFAs, 1PFAs with isolated cut point, and 2DFAs does not exceed that of 1DFA, i.e., the class of *regular languages*.

Beside these classical models, other types of finite automata based on the quantum paradigm [18] are introduced and investigated in the literature [2,4,9,16,22,25,28,29,35]. The first and simplest variant of *one-way quantum finite automaton* (1QFA) is the measure-once model, where the probability of accepting words is evaluated by "observing" just once, at the end of input processing.

Partially supported by MIUR under the project "PRIN: Automi e Linguaggi Formali: Aspetti Matematici e Applicativi."

C.S. Calude et al. (Eds.): Gruska Festschrift, LNCS 8808, pp. 161–175, 2014.
DOI: 10.1007/978-3-319-13350-8_12

Surprisingly enough, measure-once 1QFAs working with isolated cut point are proved to single out a proper subclass of regular languages, namely group (or reversible) languages [5,14].

In addition to computational power, several works in the literature investigate the descriptional power of these models, i.e., their ability to provide succinct language representations. To this regard, a fundamental tool is to study how the number of states changes when turning one automaton into another. The first widely known result in this realm compares nondeterminism with determinism for one-way finite automata: each n-state 1NFA can be simulated by a 1DFA with 2^n states [38]. Moreover, this bound is tight [33]. Another tool to get deeper insights into the descriptional power of different models of finite automata is to test them on very specific tasks, such as recognizing *unary languages*, i.e., languages over single-letter alphabets [15,32]. Some results along this line of research for probabilistic and quantum automata can be found in [8,11,12,31]. Further results on the descriptional power of quantum automata are contained in [7,10,27,30].

The same questions on the computational and descriptional power of different models of finite automata have been extended from language recognition to more general tasks known as *promise problem* solving. A promise problem on an alphabet Σ is specified by two nonempty disjoint subsets of Σ^* called *yes*-instances and *no*-instances. Unlike language recognition, the union of the *yes*-instances and *no*-instances may be a proper subset of Σ^*. A device which solves the promise problem accepts *yes*-instances, rejects *no*-instances and is allowed arbitrary behavior on the remaining strings. Intuitively, this device is "promised" that the input is either a *yes*-instance or a *no*-instance, and is only required to distinguish between these two cases.

Recently, the study of promise problems has focused on quantum devices. The first result in this realm is given by Murakami et al. [34], who showed the existence of a promise problem solvable exactly by a quantum pushdown automaton, but not by any deterministic pushdown automaton. Concerning finite automata, Ambainis and Yakaryilmaz [3] showed the existence of a family of promise problems which can be solved exactly by a 2-state 1QFA, whereas the size of corresponding 1DFAs and exact 1PFAs grows without bound. Gruska et al. showed further results on the succinctness of 1QFAs for promise problems in [20,21,42–44].

In this paper, we consider the *unary* promise problem introduced in [21] as $A^{N,r_1,r_2} = (A^{N,r_1}_{yes}, A^{N,r_2}_{no})$, with $0 \leq r_1 \neq r_2 < N$, $A^{N,r_1}_{yes} = \{\sigma^n \mid n \equiv r_1 \bmod N\}$ and $A^{N,r_2}_{no} = \{\sigma^n \mid n \equiv r_2 \bmod N\}$. Gruska et al. show that a measure-once 1QFA can solve exactly A^{N,r_1,r_2} with only 3 basis states, while a 1DFA requires d states, d being the smallest integer such that $d \mid N$ and $d \nmid (r_2 - r_1) \bmod N$. Here, we introduce the promise problem $\mathrm{DIOF}^{a,N}_{r_1,r_2}$ as an extension of A^{N,r_1,r_2} to *general* alphabets. Even for this problem, we show the same descriptional superiority of 1QFAs over 1DFAs. Moreover, we prove that adding features to classical automata, namely nondeterminism, probabilism, two-way motion, does not lead to finite automata smaller than 1DFAs for solving the promise problems A^{N,r_1,r_2} and $\mathrm{DIOF}^{a,N}_{r_1,r_2}$. To analyze these latter devices, we use the tool of normal forms for unary automata, namely: the Chrobak normal form for one-way nondeterministic finite automata [15], the cyclic normal form for one-way probabilistic finite automata [13], and a simplified form for two-way deterministic finite automata called sweeping [26,41]. Putting automata in such forms, enables us to point out their ultimate periodic behavior, from which we determine optimal lower limits for their descriptional power.

2 Preliminaries

2.1 Arithmetics and Linear Algebra

The set of natural (integer) numbers is denoted by $\mathbb{N}$ ($\mathbb{Z}$). The *greatest common divisor* of $a_1, \ldots, a_s \in \mathbb{Z}$ is denoted by $\gcd(a_1, \ldots, a_s)$. Their *least common multiple* is denoted by $\mathrm{lcm}(a_1, \ldots, a_s)$. For $a, b \in \mathbb{N}$, the notation $a \mid b$ ($a \nmid b$) stands for a divides (does not divide) b. For $N > 0$, the notation $a \equiv b \bmod N$ means that $a \bmod N = b \bmod N$. Clearly, $a \mid b$ if and only if $b \equiv 0 \bmod a$. By the Fundamental Theorem of Arithmetic, any integer $z > 1$ can be univocally expressed as a product $z = \prod_{i=1}^{s} z_i^{k_i}$, where $z_1 < \cdots < z_s$ are primes and $k_1, \ldots, k_s$ are positive integers. This product is the *prime factorization* of z. Given $a_1, \ldots, a_s, z \in \mathbb{Z}$, a linear Diophantine equation with variables $x_1, \ldots, x_s$ ranging over $\mathbb{Z}$ writes as $a_1x_1 + \cdots + a_sx_s = z$. It is a very well known fact that this equation has solutions in $\mathbb{N}$ if and only if $\gcd(a_1, \ldots, a_s) \mid z$.

We quickly recall some notions of linear algebra, useful to describe the quantum world. For more details, we refer the reader to, e.g., [40]. The field of real (complex) numbers is denoted by $\mathbb{R}$ ($\mathbb{C}$). Given a complex number $z = a + ib$, we denote its *conjugate* by $z^* = a - ib$ and its *modulus* by $|z| = \sqrt{zz^*}$. We let $\mathbb{C}^{n\times m}$ and $\mathbb{C}^n$ (shorthand for $\mathbb{C}^{1\times n}$) denote, respectively, the set of $n \times m$ matrices and n-dimensional row vectors with entries in $\mathbb{C}$. The identity matrix is denoted by I. We let $e_j = (0, \ldots, 0, 1, 0, \ldots, 0)$ be the characteristic vector having 1 in its jth component and 0 elsewhere.

Given a matrix $M \in \mathbb{C}^{n\times m}$, we let M_{ij} denote its (i, j)th entry. The *transpose* of M is the matrix $M^T \in \mathbb{C}^{m\times n}$ satisfying ${M^T}_{ij} = M_{ji}$, while we let M^* be the matrix satisfying $M^*_{ij} = (M_{ij})^*$. The *adjoint* of M is the matrix $M^\dagger = (M^T)^*$. 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 $(CD)_{ij} = \sum_{k=1}^{m} C_{ik}D_{kj}$.

A *Hilbert space* of dimension n is the linear space $\mathbb{C}^n$ of n-dimensional complex row vectors equipped with sum and product by elements in $\mathbb{C}$, in which the *inner product* $\langle \varphi, \psi \rangle = \varphi\psi^\dagger$ is defined, for $\varphi, \psi \in \mathbb{C}^n$. The *norm* of a vector $\varphi \in \mathbb{C}^n$ is given by $\|\varphi\| = \sqrt{\langle \varphi, \varphi \rangle}$. If $\langle \varphi, \psi \rangle = 0$ (and $\|\varphi\| = 1 = \|\psi\|$), then φ and ψ are *orthogonal* (*orthonormal*). The set of orthonormal vectors $\{e_1, \ldots, e_n\}$ is called the *canonical basis* of $\mathbb{C}^n$. 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 generated by $X \cup Y$.

A matrix $M \in \mathbb{C}^{n \times n}$ is said to be:

- *Boolean:* whenever its entries are either 0 or 1.
- *Stochastic:* whenever its entries are reals from the interval $[0, 1]$ and each row sums to 1.
- *Unitary:* whenever $MM^\dagger = I = M^\dagger M$; equivalently, M is unitary if and only if it preserves the norm, i.e., $\|\varphi M\| = \|\varphi\|$ for any $\varphi \in \mathbb{C}^n$. It is easy to see that the product of unitary matrices is unitary as well.

A matrix $H \in \mathbb{C}^{n \times n}$ is said to be *Hermitian (or self-adjoint)* whenever $H = H^\dagger$. A matrix $P \in \mathbb{C}^{n \times n}$ is a *projector* if and only if P is Hermitian and idempotent, i.e., $P^2 = P$. The eigenvalues of a projector are either 0 or 1. More generally, given the Hermitian matrix H, let $c_1, \ldots, c_s$ be its eigenvalues and $E_1, \ldots, E_s$ the corresponding eigenspaces. It is well known that each eigenvalue c_k is real, that E_i is orthogonal to E_j for $i \neq j$, and that $E_1 \dot{+} \cdots \dot{+} E_s = \mathbb{C}^n$. Thus, every vector $\varphi \in \mathbb{C}^n$ can be uniquely decomposed as $\varphi = \varphi_1 + \cdots + \varphi_s$ for unique $\varphi_j \in E_j$. The linear transformation $\varphi \mapsto \varphi_j$ is the projector P_j onto the subspace E_j. Actually, the Hermitian matrix H is biunivocally determined by its eigenvalues and projectors as $H = \sum_{i=1}^{s} c_i P_i$, where $\sum_{i=1}^{s} P_i = I$.

2.2 Languages and Classical Finite Automata

We assume familiarity with basics in formal language theory (see, e.g., [23]). The set of all words (including the empty word ε) over a finite alphabet Σ is denoted by Σ^*. For a word $\omega \in \Sigma^*$, we let $|\omega|$ denote its length and ω_i its ith symbol. For $\sigma \in \Sigma$, we let $|\omega|_\sigma$ denote the number of occurrences of σ in ω. A language on Σ is any set $L \subseteq \Sigma^*$.

In what follows, we quickly outline the types of classical finite automata we shall be dealing with. For extensive presentations, the reader is referred to [23] for deterministic and nondeterministic automata, and to [36] for probabilistic automata.

A *one-way deterministic finite automaton* (1DFA) is defined by the 5-tuple $A = \langle S, \Sigma, \tau, s_1, F \rangle$, where $S = \{s_1, \ldots, s_{|S|}\}$ is the finite set of states, Σ the input alphabet, $s_1 \in S$ the initial state, $F \subseteq S$ the set of accepting states, and $\tau : S \times \Sigma \to S$ is the transition function. An input word is *accepted* by A if the induced computation starting from the initial state ends in some accepting state after consuming the whole input. A linear representation for the 1DFA A is the 3-tuple $\langle \alpha, \{M(\sigma)\}_{\sigma \in \Sigma}, \beta \rangle$, where $\alpha \in \{0,1\}^{|S|}$ is the characteristic row vector of the initial state, $M(\sigma) \in \{0,1\}^{|S| \times |S|}$ is the boolean stochastic matrix satisfying $M(\sigma)_{ij} = 1$ if and only if $\tau(s_i, \sigma) = s_j$, and $\beta \in \{0,1\}^{|S| \times 1}$ is the characteristic column vector of the final states. The behavior of A on an input $\omega \in \Sigma^*$ is given by $p_A(\omega) = \alpha M(\omega) \beta$, where we let $M(\omega) = \prod_{i=1}^{|\omega|} M(\omega_i)$. The language accepted by A is the set $L = \{\omega \in \Sigma^* \mid p_A(\omega) = 1\}$.

A *one-way nondeterministic finite automaton* (1NFA) is defined similarly to a 1DFA, but the transition function now maps to possibly empty subsets of S, i.e., $\tau : S \times \Sigma \to 2^S$. This dynamic describes the possibility to have zero or more than one next state at each move. A word is accepted if there exists a computation starting from the initial state and ending in some accepting state after consuming the whole input. More formally, the linear representation for a 1NFA A is the 3-tuple $\langle \alpha, \{M(\sigma)\}_{\sigma\in\Sigma}, \beta\rangle$, where α, β are as before, while $M(\sigma) \in \{0,1\}^{|S|\times|S|}$ is the boolean (not necessarily stochastic) matrix satisfying $M(\sigma)_{ij} = 1$ if and only if $s_j \in \tau(s_i, \sigma)$. The accepted language is now defined as the set $L = \{\omega \in \Sigma^* \mid p_A(\omega) \geq 1\}$.

A *one-way probabilistic finite automaton* (1PFA) is defined similarly to above devices but now, for any given state and input symbol, the transition function returns a probability distribution over the possible next states. As a consequence, an accepting probability is associated with each input word. More formally, the linear representation for a 1PFA A is the 3-tuple $\langle \alpha, \{M(\sigma)\}_{\sigma\in\Sigma}, \beta\rangle$, where β is defined as above, $\alpha \in [0,1]^{|S|}$ is a stochastic row vector representing the initial probability distribution on S, and $M(\sigma) \in [0,1]^{|S|\times|S|}$ is the stochastic matrix where $M(\sigma)_{ij}$ is the *probability* that A moves from the ith to the jth state upon reading σ. Thus, the behavior $p_A(\omega)$ now returns the probability that A accepts the input word $\omega \in \Sigma^*$. The function $p_A : \Sigma^* \to [0,1]$ is also called the stochastic event induced by A. Given $\lambda \in [0,1]$, the language accepted by A with cut point λ is the set $L = \{\omega \in \Sigma^* \mid p_A(\omega) > \lambda\}$. Moreover, λ is said to be *isolated* if there exists a positive δ such that $|p_A(\omega) - \lambda| \geq \delta$, for any $\omega \in \Sigma^*$.

In a *two-way deterministic finite automaton* (2DFA), moves are dictated by a partial transition function[1] $\tau : S \times (\Sigma \cup \{\vdash, \dashv\}) \to S \times \{-1, +1\}$, where $\vdash, \dashv \notin \Sigma$ are two special symbols called left and right endmarker, respectively. In a move, the 2DFA reads an input symbol, changes its state, and moves the input head one cell to the right or to the left depending on whether τ returns $+1$ or -1, respectively. An input word $\omega \in \Sigma^*$ for A is stored on an input tape surrounded by the two endmarkers, so that the tape content is $\vdash \omega \dashv$. The machine accepts ω if the induced computation starting from the initial state with the head on the left endmarker reaches an accepting state with the head on either of the endmarkers. Although 2DFAs do not have a finite linear representation, we let $p_A(\omega) = 1$ ($p_A(\omega) = 0$) to denote that ω is accepted (not accepted) by A. Clearly, the accepted language is the set $L = \{\omega \in \Sigma^* \mid p_A(\omega) = 1\}$.

It is well known that 1DFAs, 1NFAs, isolated cut point 1PFAs and 2DFAs share the same computational power, i.e., they characterize the class of regular languages. Nevertheless, they have different descriptional power: representation of regular languages may be much more ?economical? — in terms of number of states — in one system than another. For instance, the following are the state costs of simulating n-state automata models by 1DFAs:

- 1NFAs: 2^n [33,38],
- 1PFAs with δ-isolated cut point: $(1 + 1/(2\delta))^{n-1}$ [1,36,37],
- 2DFAs: $n(n^n - (n-1)^n)$ [24,39].

These costs are optimal, except the one for isolated cut point 1PFAs which is "quasi optimal".

[1] In the deterministic case, we do not consider stationary moves since they can be easily removed without augmenting the number of states.

A language $L \subseteq \Sigma^*$ is called *unary* whenever $|\Sigma| = 1$. Unary regular languages are accepted by unary finite automata, i.e., automata having single-symbol input alphabets. It is folklore that any unary 1DFA consists of an initial path followed by a cycle (see Figure 1).

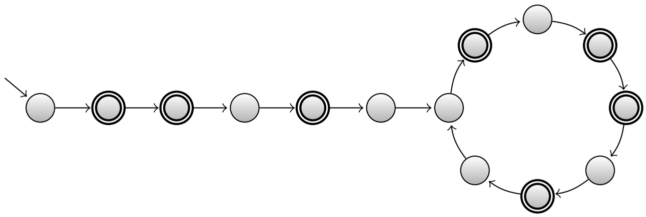

Fig. 1. A unary 1DFA. When depicting unary automata we will always omit the symbol label, since it would be redundant.

It is well known that unary automata show relevant differences from automata on general alphabets. For instance, it is proved in [15] that the optimal state cost of simulating n-state unary 1NFAs and 2DFAs by 1DFAs is "only" $\mathrm{e}^{(1+o(1))\sqrt{n \cdot \ln n}}$. For unary 1PFAs, several recent simulation results may be found, e.g., in [11,13,17,31]. All these simulations crucially rely on the fact that unary automata can be put in some "normal forms". In Section 4, we will recall such forms, and use them to get our results on promise problems.

2.3 Quantum Mechanics and Quantum Automata

Before outlining the model of quantum automaton we shall consider, we quickly present the main ingredients of the mathematical description of a quantum system possessing $Q = \{q_1, \ldots, q_m\}$ *basis states* and reacting to a set of impulses represented by the alphabet $\Sigma = \{\sigma_1, \ldots, \sigma_H\}$. (For more details, we refer the reader to, e.g., [18].) Every basis state $q_i \in Q$ can be represented by its characteristic vector $e_i \in \{0,1\}^m$. At any given time, the *quantum state* of the system is represented by a superposition $\pi = \sum_{k=1}^{m} \alpha_k e_k$, where the coefficients α_k are complex *amplitudes* and $\|\pi\| = 1$. With every symbol $\sigma_i \in \Sigma$, we associate a unitary transformation $U(\sigma_i) : \mathbb{C}^m \to \mathbb{C}^m$. An *observable* is described by an Hermitian matrix $\mathcal{O} = c_1 P_1 + \cdots + c_s P_s$. With the system being in the quantum state π, we can operate:

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

Let us now see how quantum finite automata fit in this picture. *One-way quantum finite automata* (1QFAs) are computational devices particularly interesting because of their simplicity. Moreover, their analysis provides a good insight into the nature of quantum computation, since 1QFAs are a theoretical model for a quantum computer with finite memory. From the point of view of computational capabilities, 1QFAs present both advantages and disadvantages with respect to their classical (deterministic, nondeterministic or probabilistic) counterpart. Essentially, quantum superposition offers some computational advantages on probabilistic superposition seen for 1PFAs. On the other hand, quantum dynamics are reversible: because of limitation of memory, it is generally impossible to simulate classical automata by quantum automata. Limitations due to reversibility can be partially attenuated by systematically introducing measurements of suitable observables as computational steps.

Several models of quantum automata are proposed in the literature [2,6**?**]. Basically, they differ in measurement policy. In this paper, we only focus on the *measure-once model* [5,14,28], where the transformation on an input symbol is realized by a unitary operator and a *unique* measurement is performed at the end of computation. More formally, a measure-once 1QFA with m basis states and input alphabet Σ is a system $A = \langle Q, \Sigma, \{U(\sigma)\}_{\sigma \in \Sigma \cup \{\vdash, \dashv\}}, e_1, Q_a \rangle$, where:

- $Q = \{e_1, \ldots, e_m\}$ is the canonical basis of the Hilbert space $\mathbb{C}^m$; its elements are the basis states,
- Σ is a finite alphabet of input symbols, and $\vdash, \dashv \notin \Sigma$ are the left and right endmarkers,
- with any $\sigma \in \Sigma \cup \{\vdash, \dashv\}$, a unitary matrix $U(\sigma) \in \mathbb{C}^{m \times m}$ is associated,
- $e_1 = (1, 0, \ldots, 0) \in \mathbb{C}^m$ is the initial basis state,
- $Q_a \subseteq Q$ is the set of accepting basis states, identifying the projection matrix $P_a = \sum_{\{i \mid e_i \in Q_a\}} e_i^T e_i \in \mathbb{C}^{m \times m}$ which biunivocally determines the observable $\mathcal{O} = 1 \cdot P_a + 0 \cdot (I - P_a)$.

The behavior of A is the stochastic event $p_A : \Sigma^* \to [0,1]$ defined, for any $x = x_1 x_2 \cdots x_n \in \Sigma^*$, by

$$p_A(x) = \|e_1 U(\vdash) U(x_1) U(x_2) \cdots U(x_n) U(\dashv) P_a\|^2.$$

The language accepted by A with (isolated) cut-point λ is defined as in Section 2.2 for 1PFAs.

From a computational power point of view, in [5,28] it is proved that measure-once 1QFAs are strictly less powerful than classical automata. In fact, with isolated cut point, they characterize the class of group (or reversible) languages, a proper subclass of regular languages. However, from a descriptional power point of view, they are shown to greatly outperform classical models. E.g., in [7,8,30], several families of regular languages are provided, on which measure-once 1QFAs are exponentially smaller than classical paradigms.

We remark that measure-once 1QFAs are originally introduced in [5,28] without the endmarkers, with an arbitrary initial unitary vector, and with an arbitrary accepting subspace. More precisely, the automaton is represented as $B =$

$\langle Q, \Sigma, \{U(\sigma)\}_{\sigma\in\Sigma}, \pi, P\rangle$, where Q, Σ and $U(\sigma)$ are defined as above, $\pi \in \mathbb{C}^m$ is a unitary vector and P is a projector. In this case, the event induced by B on x is $p_B(x) = \|\pi U(x_1)U(x_2)\cdots U(x_n)P\|^2$. Actually, the two models are equivalent:

– The automaton A is equivalent to $A' = \langle Q, \Sigma, \{U'(\sigma)\}_{\sigma\in\Sigma}, \pi', P_a\rangle$, where $U'(\sigma) = U(\dashv)^\dagger U(\sigma)U(\dashv)$ and $\pi' = e_1 U(\vdash)U(\dashv)$. In fact

$$p_{A'}(x) = \|e_1 U(\vdash)U(\dashv)U^\dagger(\dashv)U(x_1)U(\dashv)\cdots U^\dagger(\dashv)U(x_n)U(\dashv)P_a\|^2 = p_A(x).$$

– The automaton B is equivalent to $B' = \langle Q, \Sigma, \{U(\sigma)\}_{\sigma\in\Sigma\cup\{\vdash,\dashv\}}, e_1, Q_a\rangle$, where $U(\vdash)$ is a unitary matrix with π as the first row, so that $e_1 U(\vdash) = \pi$. To set Q_a, we notice that P is similar to the diagonal matrix P_a built on the eigenvalues of P, i.e., $P = VP_aV^\dagger$ with V being a unitary matrix [40]. Moreover, as recalled in Section 2.1, such eigenvalues are either 0 or 1. So, we let Q_a be the unique subset of Q such that $P_a = \sum_{\{i \mid e_i\in Q_a\}} e_i^T e_i$ holds. In addition, V being unitary, we let $U(\dashv) = V$, and notice that multiplying a vector by $V^\dagger$ does not change the vector norm. So, we can write

$$\begin{aligned} p_{B'}(x) &= \|e_1 U(\vdash)U(x_1)U(x_2)\cdots U(x_n)U(\dashv)P_a\|^2 \\ &= \|\pi U(x_1)U(x_2)\cdots U(x_n)VP_aV^\dagger\|^2 = p_B(x). \end{aligned}$$

Throughout the rest of the paper, we will simply write 1QFA, understanding the designation "measure-once".

3 Quantum Automata for Promise Problems

We recall that a *promise problem* over an alphabet Σ is a pair $A = (A_{yes}, A_{no})$, where $A_{yes}, A_{no} \subseteq \Sigma^*$ are nonempty disjoint sets. An automaton M solves A *with isolated cut point* λ if there exists a $\delta \in \left(0, \frac{1}{2}\right]$ such that

– for any $\omega \in A_{yes},\ p_M(\omega) \geq \lambda + \delta$, and
– for any $\omega \in A_{no},\ p_M(\omega) \leq \lambda - \delta$.

If $\lambda = \delta = \frac{1}{2}$, then A is solved by M *exactly*.

It is easy to see that the classical membership problem for a nonempty language $L \subseteq \Sigma^*$ may be regarded as the promise problem $(L, \Sigma^* \setminus L)$.

In [21], Gruska et al. propose the promise problem $A^{N,r_1,r_2} = (A^{N,r_1}_{yes}, A^{N,r_2}_{no})$ on the unary alphabet $\{\sigma\}$, with $0 \leq r_1 \neq r_2 < N$,

$$A^{N,r_1}_{yes} = \{\sigma^n \mid n \equiv r_1 \bmod N\} \text{ and } A^{N,r_2}_{no} = \{\sigma^n \mid n \equiv r_2 \bmod N\}.$$

For the sake of readability, when referring to this problem throughout the rest of the paper, we let

$$l = (r_2 - r_1) \bmod N.$$

The following result is proved in [21]:

Theorem 1. *The promise problem* A^{N,r_1,r_2} *can be solved exactly by a* 3 *basis states* 1QFA*, while the minimal* 1DFA *has* d *states, where* d *is the smallest positive integer such that* $d \mid N$ *and* $d \nmid l$*.*

We recall that the minimal 1DFA for A^{N,r_1,r_2} addressed in Theorem 1 consists of a cycle of length d with a unique final state at distance r_1 from the initial state. We also notice that, by fixing $N = 2^{k+1}$, $r_1 = 0$ and $r_2 = 2^k$, we obtain the promise problem studied in [3], for which an unbounded size gap between quantum and deterministic finite automata solution is established. So, Theorem 1 extends this gap to other values of N (e.g., for prime N).

Let us now introduce a generalization of the promise problem A^{N,r_1,r_2} on the multi-letter alphabet $\Sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_H\}$. For $0 \leq r_1 \neq r_2 < N$ and $a = (a_1, a_2, \ldots, a_H) \in \mathbb{N}^H$ satisfying $\gcd(a_1, a_2, \ldots, a_H, N) = 1$, we define the promise problem $\text{DIOF}^{a,N}_{r_1,r_2} = (\text{DIOF}^{a,N,r_1}_{yes}, \text{DIOF}^{a,N,r_2}_{no})$ as

$$\begin{aligned}\text{DIOF}^{a,N,r_1}_{yes} &= \{\omega \in \Sigma^* \mid (a_1|\omega|_{\sigma_1} + a_2|\omega|_{\sigma_2} + \cdots + a_H|\omega|_{\sigma_H}) \equiv r_1 \bmod N\},\\ \text{DIOF}^{a,N,r_2}_{no} &= \{\omega \in \Sigma^* \mid (a_1|\omega|_{\sigma_1} + a_2|\omega|_{\sigma_2} + \cdots + a_H|\omega|_{\sigma_H}) \equiv r_2 \bmod N\}.\end{aligned}$$

As above, when referring to this problem, we let $l = (r_2 - r_1) \bmod N$. Notice that the condition $\gcd(a_1, a_2, \ldots, a_H, N) = 1$ ensures that $\text{DIOF}^{a,N,r_1}_{yes}$ and $\text{DIOF}^{a,N,r_2}_{no}$ are nonempty sets for any r_1, r_2. In addition, the condition $r_1 \neq r_2$ ensures disjointness. By suitably adapting the technique in [21], we exhibit succinct 1QFAs for the family $\text{DIOF}^{a,N}_{r_1,r_2}$:

Theorem 2. *The promise problem* $\text{DIOF}^{a,N}_{r_1,r_2}$ *can be solved exactly by a* 3 *basis states* 1QFA*.*

Proof. To get our 1QFA, we apply the same construction exhibited in Theorem 1 in [21]. The only difference is that, instead of having the unique matrix U_a performing a rotation of an angle θ, we here have matrices $U(\sigma_j)$ performing rotations of angles θa_j, for $1 \leq j \leq H$. As a consequence, the product $U(\omega) = \prod_{i=1}^{|\omega|} U(\omega_i)$ describing the computation of the 1QFA on any given input word ω now yields the matrix

$$U(\omega) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\left(\theta \sum_{j=1}^{H} a_j|\omega|_{\sigma_j}\right) & \sin\left(\theta \sum_{j=1}^{H} a_j|\omega|_{\sigma_j}\right) \\ 0 & -\sin\left(\theta \sum_{j=1}^{H} a_j|\omega|_{\sigma_j}\right) & \cos\left(\theta \sum_{j=1}^{H} a_j|\omega|_{\sigma_j}\right) \end{pmatrix}.$$

The rest of the proof proceeds as in [21]. □

4 Classical Automata for Promise Problems

Let us now analyze the size required by classical automata for solving the promise problems A^{N,r_1,r_2} and $\text{DIOF}^{a,N}_{r_1,r_2}$. First we consider one-way models: both in the nondeterministic and probabilistic case, we obtain the same size lower bound as for 1DFAs (Theorem 1). Then, we extend this result to 2DFAs.

To study the solution of the promise problem A^{N,r_1,r_2} on 1NFAs, we recall the Chrobak normal form for unary automata [15]. This form extends the structure of unary 1DFAs displayed in Figure 1 and, roughly speaking, consists of an initial path at the end of which a nondeterministic move leads to more than one cycle (see Figure 2).

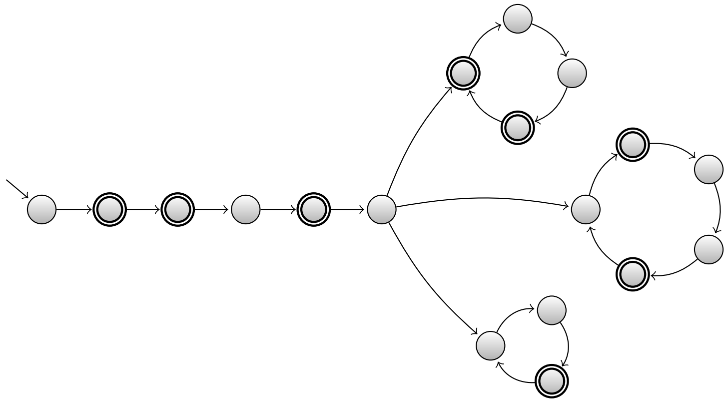

Fig. 2. A unary 1NFA in Chrobak normal form with 3 cycles

More formally, a unary 1NFA $A = \langle S, \{\sigma\}, \tau, s_0, F\rangle$ is in *Chrobak normal form* if S can be partitioned into $m+1$ disjoint sets $S_0, C_1, \ldots, C_m$ such that:

- $S_0 = \{s_0, s_1, \ldots, s_t\}$,
- for $1 \leq i \leq m$, $C_i = \{p_{i,0}, p_{i,1}, \ldots, p_{i,y_i-1}\}$,
- for $1 \leq i \leq m$ and $0 \leq j < y_i$, $\tau(p_{i,j}, \sigma) = \left\{p_{i,(j+1) \bmod y_i}\right\}$, i.e., C_i is a cycle of length y_i,
- for $0 \leq i < t$, $\tau(s_i, \sigma) = \{s_{i+1}\}$, i.e., S_0 is a path of length t,
- $\tau(s_t, \sigma) = \{p_{1,0}, p_{2,0}, \ldots, p_{m,0}\}$, i.e., s_t is the only state where a nondeterministic move takes place, leading to a single state in each cycle.

In [15], it is proved the following:

Lemma 1. *Each unary n-state* 1NFA *can be simulated by a* 1NFA *in Chrobak normal form having $O(n^2)$ states in the initial path and at most n states in the cycles.*

The Chrobak normal form is crucial to obtain the following result:

Theorem 3. *The minimal* 1NFA *solving the promise problem A^{N,r_1,r_2} has d states, where d is the smallest positive integer such that $d \mid N$ and $d \nmid l$.*

Proof. The minimal 1DFA addressed in Theorem 1, is an example of d-state 1NFA for A^{N,r_1,r_2}. So, we only need to prove minimality.

Suppose there exists a 1NFA which solves A^{N,r_1,r_2} with $p < d$ states. By Lemma 1, we can convert this 1NFA into an equivalent 1NFA M in Chrobak normal form having t (which is $O(p^2)$) states in the initial path and at most p states in the cycles.

Let $\alpha \in \mathbb{N}$ satisfy $\alpha N > t$. Since $\sigma^{\alpha N + r_1} \in A_{yes}^{N,r_1}$, there exists an accepting state s reachable by M on input $\sigma^{\alpha N + r_1}$. Moreover, since $\alpha N + r_1$ exceeds the length of the initial path, s belongs to a cycle of length $\ell \leq p$. This implies that the same state s is reachable by M on input $\sigma^{\alpha N + r_1 + \beta\ell}$, and therefore this word is accepted. Let $g = \gcd(\ell, N)$. If $g \mid l$, then there exist $\beta, \gamma \in \mathbb{N}$ such that the Diophantine equation $\beta\ell = \gamma N + l$ holds. However, for a suitable $\alpha' \in \mathbb{N}$, we get

$$\sigma^{\alpha N + r_1 + \beta\ell} = \sigma^{(\alpha+\gamma)N + r_1 + l} = \sigma^{(\alpha+\gamma)N + r_1 + (r_2 - r_1) \bmod N} = \sigma^{\alpha' N + r_2} \in A_{no}^{N,r_2},$$

and we have a contradiction. Therefore it must be $g \nmid l$. But since by definition

$$g \leq \ell \leq p < d$$

and $g \mid N$, we get a contradiction with the minimality of d. Hence, any 1NFA for A^{N,r_1,r_2} must have at least d states. □

Theorem 3 shows that for the promise problem A^{N,r_1,r_2} nondeterminism does not help in saving states. We are going to show that even the use of probabilism does not lead to smaller automata.

To this aim, we recall the *cyclic normal form* [13] for unary 1PFAs. This form is similar to Chrobak normal form, the main difference being in accepting states and the move from s_t, i.e., the last state of the initial path. Each cycle must contain exactly one accepting state, however from s_t many different states, even belonging to the same cycle, can be reached by the only allowed probabilistic move (see Figure 3).

In [13], it is proved the following:

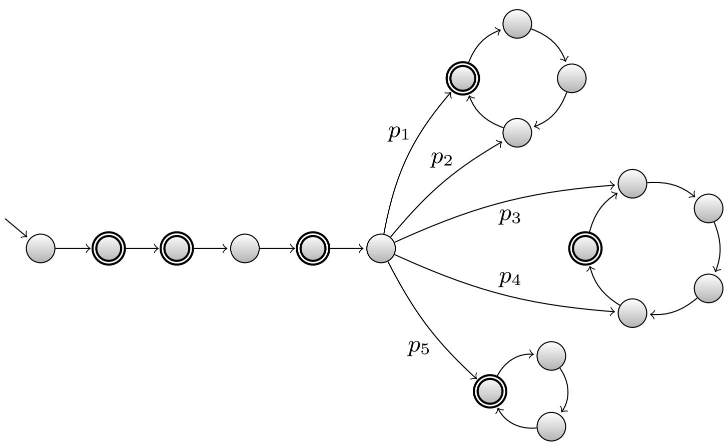

Fig. 3. A unary 1PFA in cyclic normal form with the constraint $\sum_i p_i = 1$

Lemma 2. *Each unary n-state* 1PFA *with isolated cut point can be converted into an equivalent* 1PFA *in cyclic normal form with isolated cut point (not necessarily keeping the same cut point and isolation) and with at most n states in the cycles.*

The cyclic normal form allows us to obtain the following result:

Theorem 4. *The minimal* 1PFA *solving with isolated cut point the promise problem A^{N,r_1,r_2} has d states, where d is the smallest positive integer such that $d \mid N$ and $d \nmid l$.*

Proof. The minimal 1DFA addressed in Theorem 1 can obviously be regarded as a d-state 1PFA solving exactly A^{N,r_1,r_2}.

To show minimality, suppose there exists a 1PFA which solves A^{N,r_1,r_2} with isolated cut point and $p < d$ states. By Lemma 2, we can convert this 1PFA into an isolated cut point 1PFA M in cyclic normal form having t states in the initial path and a set of cycles of lengths $\ell_1, \ell_2, \dots \ell_z$, such that $\sum_{i=1}^{z} \ell_i \leq p$.

We choose $\alpha \in \mathbb{N}$ satisfying $\alpha N > t$. The word $\sigma^{\alpha N + r_1}$ brings M from the initial state to a state s in a cycle with a given probability. By letting $L = \mathrm{lcm}(\ell_1, \ell_2, \dots \ell_z)$, for any $\beta \in \mathbb{N}$ the word $\sigma^{\beta L}$ brings M from s back to s with certainty. Since this holds for any state s reachable by M on input $\sigma^{\alpha N + r_1}$, we have that $p_M(\sigma^{\alpha N + r_1}) = p_M(\sigma^{\alpha N + r_1 + \beta L})$.

Let $g = \gcd(L, N)$. If $g \mid l$, then there exist $\beta, \gamma \in \mathbb{N}$ such that the Diophantine equation $\beta L = \gamma N + l$ holds. So, for a suitable $\alpha' \in \mathbb{N}$, we have $\sigma^{\alpha N + r_1 + \beta L} = \sigma^{\alpha' N + r_2} \in A_{no}^{N,r_2}$, and we get

$$p_M(\sigma^{\alpha N + r_1}) = p_M(\sigma^{\alpha N + r_1 + \beta L}) = p_M(\sigma^{\alpha' N + r_2}).$$

However, $\sigma^{\alpha N + r_1} \in A_{yes}^{N,r_1}$ and we have a contradiction on M having an isolated cut point. Therefore, it must be $g \nmid l$. Let $g = \prod_{i=1}^{h} g_i^{b_i}$ be the prime factorization of g. Then, there exists $g_\kappa^{b_\kappa} \nmid l$. In addition, notice that $g_\kappa^{b_\kappa} \mid L$ and since $g_\kappa^{b_\kappa}$ is a prime power there exists $1 \leq j \leq z$ such that $g_\kappa^{b_\kappa} \mid \ell_j$. This implies that

$$g_\kappa^{b_\kappa} \leq \ell_j \leq p < d.$$

This, together with $g_\kappa^{b_\kappa} \mid N$ and $g_\kappa^{b_\kappa} \nmid l$, contradicts the minimality of d. Hence, any isolated cut point 1PFA for A^{N,r_1,r_2} must have at least d states. □

Even by using two-way motion, we do not manage to design automata smaller than 1DFAs, 1NFAs, and 1PFAs for solving the promise problem A^{N,r_1,r_2}. To this aim, we consider a simplified form for unary 2DFAs: a 2DFA is called *sweeping* if its input head changes its direction at the endmarkers only [41]. The following simulation result is proved in [26]:

Lemma 3. *For each unary n-state* 2DFA*, there exists an equivalent sweeping* 2DFA *with $n + 1$ states.*

This lemma allows us to show the following result:

Theorem 5. *The minimal sweeping* 2DFA *solving the promise problem* A^{N,r_1,r_2} *has* d *states, where* d *is the smallest positive integer such that* $d \mid N$ *and* $d \nmid l$. *Moreover, any* 2DFA *for* A^{N,r_1,r_2} *must have at least* $d-1$ *states.*

Proof. Again, the minimal 1DFA addressed in Theorem 1 can obviously be regarded as a d-state sweeping 2DFA solving A^{N,r_1,r_2}. We now prove that this automaton is a minimal sweeping 2DFA for A^{N,r_1,r_2}.

Suppose there exists, a sweeping 2DFA M with $p < d$ states. Consider the computation of M accepting $\sigma^{N+r_1} \in A^{N,r_1}_{yes}$ in z traversals. Since $p < N$, in every traversal M must enter a cycle. Let C_i be the cycle entered along the ith traversal and ℓ_i the length of C_i. By letting $L = \mathrm{lcm}(\ell_1, \ldots, \ell_z)$, it is not hard to see that the computation of M on input $\sigma^{N+r_1+\beta L}$ leads to the same accepting state as for σ^{N+r_1}. Indeed, this computation has again z traversals and in the ith traversal the cycle C_i is repeated $\frac{\beta L}{\ell_i}$ more times. Now, consider $g = \gcd(L, N)$ and $\prod_{i=1}^{s} g_i^{b_i}$ its prime factorization. If $g \mid l$, then there exist $\beta, \gamma \in \mathbb{N}$ such that the Diophantine equation $\beta L = \gamma N + l$ holds. So, for a suitable $\alpha \in \mathbb{N}$, we have that

$$\sigma^{N+r_1+\beta L} = \sigma^{\alpha N + r_2} \in A^{N,r_2}_{no}$$

is accepted, leading to a contradiction. Therefore, $g \nmid l$ must hold. In this case, by proceeding analogously to the proof of Theorem 4, we get a contradiction with the minimality of d. Hence, any sweeping 2DFA for A^{N,r_1,r_2} must have at least d states.

Finally, since converting a 2DFA into sweeping costs at most one additional state (Lemma 3), we get that any 2DFA solving A^{N,r_1,r_2} must have at least $d-1$ states. □

We conclude this section by addressing the size cost of solving the promise problem $\mathrm{DIOF}^{a,N}_{r_1,r_2}$ with classical automata.

Theorem 6. *To solve the promise problem* $\mathrm{DIOF}^{a,N}_{r_1,r_2}$, *the minimal* 1DFA, 1NFA, *isolated cut point* 1PFA, *and sweeping* 2DFA *have* d *states, where* d *is the smallest positive integer such that* $d \mid N$ *and* $d \nmid l$. *Moreover, any* 2DFA *for* $\mathrm{DIOF}^{a,N}_{r_1,r_2}$ *must have at least* $d-1$ *states.*

Proof. For the upper bound, we can design 1DFAs, 1NFAs, 1PFAs, and 2DFAs solving $\mathrm{DIOF}^{a,N}_{r_1,r_2}$ with the same cyclic structure of the 1DFA for A^{N,r_1,r_2}. The only difference is that, while the automaton for A^{N,r_1,r_2} moves one state forward in the cycle upon reading σ, the automaton for $\mathrm{DIOF}^{a,N}_{r_1,r_2}$ moves a_i states forward on input σ_i, for $1 \le i \le H$.

For the lower bound, it suffices to notice that A^{N,r_1,r_2} is a particular case of $\mathrm{DIOF}^{a,N}_{r_1,r_2}$ with $|\Sigma| = 1$ and $a = 1$. □

References

1. Ambainis, A.: The complexity of probabilistic versus deterministic finite automata. In: Asano, T., Igarashi, Y., Nagamochi, H., Miyano, S., Suri, S. (eds.) ISAAC 1996. LNCS, vol. 1178, pp. 233–238. Springer, Heidelberg (1996)

2. Ambainis, A., Beaudry, M., Golovkins, M., Kikusts, A., Mercer, M., Thérien, D.: Algebraic results on quantum automata. Theory of Comp. Sys. **39**, 165–188 (2006)
3. Ambainis, A., Yakaryilmaz, A.: Superiority of exact quantum automata for promise problems. Information Processing Letters **112**, 289–291 (2012)
4. Ambainis, A., Watrous, J.: Two-way finite automata with quantum and classical states. Theoretical Computer Science **287**, 299–311 (2002)
5. Bertoni, A., Carpentieri, M.: Regular languages accepted by quantum automata. Information and Computation **165**, 174–182 (2001)
6. Bertoni, A., Mereghetti, C., Palano, B.: Quantum computing: 1-way quantum automata. In: Ésik, Z., Fülöp, Z. (eds.) DLT 2003. LNCS, vol. 2710, pp. 1–20. Springer, Heidelberg (2003)
7. Bertoni, A., Mereghetti, C., Palano, B.: Small size quantum automata recognizing some regular languages. Theoretical Computer Science **340**, 394–407 (2005)
8. Bertoni, A., Mereghetti, C., Palano, B.: Some formal tools for analyzing quantum automata. Theoretical Computer Science **356**, 14–25 (2006)
9. Bertoni, A., Mereghetti, C., Palano, B.: Trace monoids with idempotent generators and measure-only quantum automata. Natural Computing **9**, 383–395 (2010)
10. Bianchi, M.P., Mereghetti, C., Palano, B.: Size Lower Bounds for Quantum Automata. In: Mauri, G., Dennunzio, A., Manzoni, L., Porreca, A.E. (eds.) UCNC 2013. LNCS, vol. 7956, pp. 19–30. Springer, Heidelberg (2013)
11. Bianchi, M.P., Mereghetti, C., Palano, B., Pighizzini, G.: On the size of unary probabilistic and nondeterministic automata. Fund. Informaticae **112**, 119–135 (2011)
12. Bianchi, M.P., Palano, B.: Behaviours of unary quantum automata. Fund. Informaticae **104**, 1–15 (2010)
13. Bianchi, M.P., Pighizzini, G.: Normal forms for unary probabilistic automata. Theoretical Informatics and Applications **46**, 495–510 (2012)
14. Brodsky, A., Pippenger, N.: Characterizations of 1-way quantum finite automata. SIAM J. Computing **5**, 1456–1478 (2002)
15. Chrobak, M.: Finite automata and unary languages. Theoretical Computer Science **47**, 149–158 (1986)
16. Golovkins, M., Kravtsev, M.: Probabilistic Reversible Automata and Quantum Automata. In: Ibarra, O.H., Zhang, L. (eds.) COCOON 2002. LNCS, vol. 2387, pp. 574–583. Springer, Heidelberg (2002)
17. Gramlich, G.: Probabilistic and Nondeterministic Unary Automata. In: Rovan, B., Vojtáš, P. (eds.) MFCS 2003. LNCS, vol. 2747, pp. 460–469. Springer, Heidelberg (2003)
18. Gruska, J.: Quantum Computing. McGraw-Hill, London, New York (1999)
19. Gruska, J.: Descriptional complexity issues in quantum computing. J. Automata, Languages and Combinatorics **5**, 191–218 (2000)
20. Gruska J., Qiu, D., Zheng, S.: Generalizations of the distributed Deutsch-Jozsa promise problem (2014), http://arxiv.org/abs/1402.7254arXiv:1402.7254
21. Gruska J., Qiu, D., Zheng, S.: Potential of quantum finite automata with exact acceptance (2014). http://arxiv.org/abs/1404.1689arXiv:1404.1689
22. Hirvensalo, M.: Quantum automata with open time evolution. Int. J. Natural Computing Research **1**, 70–85 (2010)
23. Hopcroft, J.E., Motwani, R., Ullman, J.D.: Introduction to Automata Theory, Languages, and Computation. Addison-Wesley, Reading (2001)
24. Kapoutsis, C.A.: Removing bidirectionality from nondeterministic finite automata. In: Jedrzejowicz, J., Szepietowski, A. (eds.) MFCS 2005. LNCS, vol. 3618, pp. 544–555. Springer, Heidelberg (2005)

25. Kondacs A., Watrous, J.: On the power of quantum finite state automata. In: Proc. 38th Symposium on Foundations of Computer Science (FOCS 1997), pp. 66–75 (1997)
26. Kunc, M., Okhotin, A.: Describing Periodicity in Two-Way Deterministic Finite Automata Using Transformation Semigroups. In: Mauri, G., Leporati, A. (eds.) DLT 2011. LNCS, vol. 6795, pp. 324–336. Springer, Heidelberg (2011)
27. Mercer, M.: Lower bounds for generalized quantum finite automata. In: Martín-Vide, C., Otto, F., Fernau, H. (eds.) LATA 2008. LNCS, vol. 5196, pp. 373–384. Springer, Heidelberg (2008)
28. Moore, C., Crutchfield, J.: Quantum automata and quantum grammars. Theoretical Computer Science **237**, 275–306 (2000)
29. Mereghetti, C., Palano, B.: Quantum finite automata with control language. Theoretical Informatics and Applications **40**, 315–332 (2006)
30. Mereghetti, C., Palano, B.: Quantum automata for some multiperiodic languages. Theoretical Computer Science **387**, 177–186 (2007)
31. Mereghetti, C., Palano, B., Pighizzini, G.: Note on the succinctness of deterministic, nondeterministic, probabilistic and quantum finite automata. Theoretical Informatics and Applications **5**, 477–490 (2001)
32. Mereghetti, C., Pighizzini, G.: Optimal simulations between unary automata. SIAM J. Comput. **30**, 1976–1992 (2001)
33. Meyer, A., Fischer, M.: Economy of description by automata, grammars, and formal systems. In: Proc. 12th Annual IEEE Symposium on Switching and Automata Theory, pp. 188–91 (1971)
34. Murakami, Y., Nakanishi, M., Yamashita, S., Watanabe, K.: Quantum versus classical pushdown automata in exact computation. IPSJ Digital Courier **1**, 426–435 (2005)
35. Nayak, A.: Optimal lower bounds for quantum automata and random access codes. In: Proc. 40th Symposium on Foundations of Computer Science (FOCS 1999), pp. 369–376 (1999)
36. Paz, A.: Introduction to Probabilistic Automata. Academic Press, New York, London (1971)
37. Rabin, M.O.: Probabilistic automata. Information and Control **6**, 230–245 (1963)
38. Rabin, M., Scott, D.: Finite automata and their decision problems. IBM J. Res. Develop. **3**, 114–125 (1959)
39. Shepherdson, J.C.: The reduction of two-way automata to one-way automata. IBM J. Res. Develop. **3**, 198–200 (1959)
40. Shilov,G: Linear Algebra. Prentice Hall (1971) (Reprinted by Dover, 1977)
41. Sipser, M.: Lower bounds on the size of sweeping automata. J. Comput. System Sci. **21**, 195–202 (1980)
42. Zheng, S., Gruska, J., Qiu, D.: On the State Complexity of Semi-quantum Finite Automata. In: Dediu, A.-H., Martín-Vide, C., Sierra-Rodríguez, J.-L., Truthe, B. (eds.) LATA 2014. LNCS, vol. 8370, pp. 601–612. Springer, Heidelberg (2014)
43. Zheng, S., Qiu, D., Gruska, J., Li, L., Mateus, P.: State succinctness of two-way finite automata with quantum and classical states. Theoretical Computer Science **499**, 98–112 (2013)
44. Zheng, S., Qiu, D., Li, L., Gruska, J.: One-Way finite automata with quantum and classical states. In: Bordihn, H., Kutrib, M., Truthe, B. (eds.) Languages Alive. LNCS, vol. 7300, pp. 273–290. Springer, Heidelberg (2012)