Quantum Finite Automata and Probabilistic Reversible Automata: R-trivial Idempotent Languages ⋆

Marats Golovkins, Maksim Kravtsev, and Vasilijs Kravcevs

Faculty of Computing, University of Latvia, Raiņa bulv. 19, Riga LV-1586, Latvia
marats AT latnet DOT lv, maksims DOT kravcevs AT lu DOT lv,
kvasilijs AT gmail DOT com

Abstract. We study the recognition of $\mathcal{R}$-trivial idempotent ($\mathcal{R}_1$) languages by various models of "decide-and-halt" quantum finite automata (QFA) and probabilistic reversible automata (DH-PRA). We introduce *bistochastic* QFA (MM-BQFA), a model which generalizes both Nayak's enhanced QFA and DH-PRA. We apply tools from algebraic automata theory and systems of linear inequalities to give a complete characterization of $\mathcal{R}_1$ languages recognized by all these models. We also find that "forbidden constructions" known so far do not include all of the languages that cannot be recognized by measure-many QFA.

1 Introduction

Measure-many quantum finite automata (MM-QFA) were defined in 1997 [11] and their language class characterization problem remains open still. The difficulties arise because the language class is not closed under Boolean operations like union and intersection [3]. The results by Brodsky and Pippenger [5] combined with the non-closure property imply that the class of languages recognized by MM-QFA is a proper subclass of the language variety corresponding to the **ER** monoid variety. The same holds for DH-PRA and for EQFA [8, 14]. In [1], it is stated that MM-QFA recognize any regular language corresponding to the monoid variety **EJ**. Since any syntactic monoid of a unary regular language belongs to **EJ**, the results in [1] imply that MM-QFA recognize any unary regular language. In [4], a new proof of this result is given by explicitly constructing MM-QFA recognizing unary languages. In the paper, we consider a sub-variety of **ER**, the variety of $\mathcal{R}$-trivial idempotent monoids $\mathbf{R_1}$ and determine which $\mathcal{R}$-trivial idempotent languages ($\mathcal{R}_1$ languages) are recognizable by MM-QFA and other "decide-and-halt" models. Since $\mathbf{R_1}$ shares a lot of the characteristic properties with **ER**, the obtained results may serve as an insight to solve the general problem. The paper is structured as follows. Section 2 describes the

⋆ Supported by the Latvian Council of Science, grant No. 09.1570 and by the European Social Fund, contract No. 2009/0216/1DP/1.1.1.2.0/09/APIA/VIAA/044. Unabridged version of this article is available at `http://arxiv.org`.

algebraic tools - monoids, morphisms and varieties. Section 3 considers completely positive maps. We apply the result by Kuperberg to obtain Theorem 3.1, which is essential to prove the limitations of QFA in terms of language recognition. Sections 4 to 7 present the main results of the paper: (1) We introduce MM-BQFA, a model which generalizes the earlier "decide-and-halt" automata models (MM-QFA, DH-PRA, EQFA) and give some characteristics of the corresponding language class. We also obtain the class of languages recognized by MO-BQFA; (2) We define how to construct a system of linear inequalities for any $\mathcal{R}_1$ language and prove that if the system is not consistent the language cannot be recognized by MM-BQFA (and MM-QFA, DH-PRA, EQFA); (3) We construct DH-PRA (this presumes also EQFA and MM-BQFA) and MM-QFA for any $\mathcal{R}_1$ language having a consistent system of inequalities. Thus, we obtain that an $\mathcal{R}_1$ language is recognizable by "decide-and-halt" models if and only if the corresponding system of linear inequalities is consistent; (4) We show that "forbidden constructions" known from [3] do not give all of the languages that cannot be recognized by MM-QFA.

2 Monoids and Varieties

Given an alphabet A, let A^* be the set of words over alphabet A. Given a word $\mathbf{x}$, let $|\mathbf{x}|$ be the length of $\mathbf{x}$. Introduce a partial order $\leqslant$ on A^*, let $\mathbf{x} \leqslant \mathbf{y}$ if and only if there exists $\mathbf{z} \in A^*$ such that $\mathbf{xz} = \mathbf{y}$. Let $\mathcal{P}(A)$ be the set of subsets of A, including the empty set $\emptyset$. Note that there is a natural partial order on $\mathcal{P}(A)$, i.e., the subset order. Given a word $\mathbf{s} \in A^*$, let $\mathbf{s}\omega$ be the set of letters of the word $\mathbf{s}$. We say that $\mathbf{u}, \mathbf{v} \in A^*$ are equivalent with respect to ω, $\mathbf{u} \sim_\omega \mathbf{v}$, if $\mathbf{u}\omega = \mathbf{v}\omega$ (that is, $\mathbf{u}$ and $\mathbf{v}$ consist of the same set of letters). Let $\mathcal{F}(A)$ be the set of all words over the alphabet A that do not contain any repeated letters. The empty word ε is an element of $\mathcal{F}(A)$. Let τ be a function such that for every $\mathbf{s} \in A^*$, any repeated letters in $\mathbf{s}$ are deleted, leaving only the first occurrence. Given $\mathbf{u}, \mathbf{v} \in A^*$, we say that $\mathbf{u} \sim_\tau \mathbf{v}$, if $\mathbf{u}\tau = \mathbf{v}\tau$. Introduce a partial order $\leqslant$ on $\mathcal{F}(A)$, let $\mathbf{v_1} \leqslant \mathbf{v_2}$ if and only if there exists $\mathbf{v} \in \mathcal{F}(A)$ such that $\mathbf{v_1}\mathbf{v} = \mathbf{v_2}$. Note that $\sim_\omega$ and $\sim_\tau$ are equivalence relations. The functions ω and τ are morphisms; $(\mathbf{uv})\omega = \mathbf{u}\omega \cup \mathbf{v}\omega$ and $(\mathbf{uv})\tau = \mathbf{u}\tau \cdot \mathbf{v}\tau$. Moreover, ω (and τ) preserves the order relation since $\mathbf{u} \leqslant \mathbf{v}$ implies $\mathbf{u}\omega \subseteq \mathbf{v}\omega$ ($\mathbf{u} \leqslant \mathbf{v}$ implies $\mathbf{u}\tau \leqslant \mathbf{v}\tau$).

A general overview on varieties of finite semigroups, monoids as well as operations on them is given in [17]. Unless specified otherwise, the monoids discussed in this section are assumed to be finite. An element e of a monoid $\mathcal{M}$ is called an *idempotent*, if $e^2 = e$. If x is an element of a monoid $\mathcal{M}$, the unique idempotent of the subsemigroup of $\mathcal{M}$ generated by x [17] is denoted by x^ω. Given a regular language $L \subseteq A^*$, words $\mathbf{u}, \mathbf{v} \in A^*$ are called *syntactically congruent*, $\mathbf{u} \sim_L \mathbf{v}$, if for all $\mathbf{x}, \mathbf{y} \in A^*$ $\mathbf{xuy} \in L$ if and only if $\mathbf{xvy} \in L$. The set of equivalence classes $A^*/\sim_L$ is a monoid, called *syntactic monoid* of L and denoted $\mathcal{M}(L)$. The morphism φ from A^* to $A^*/\sim_L$ is called *syntactic morphism.* Given a monoid variety $\mathbf{V}$, the corresponding language variety is denoted by $\boldsymbol{\mathcal{V}}$. The set of languages over A recognized by monoids in $\mathbf{V}$ is denoted by $A^*\boldsymbol{\mathcal{V}}$.

Varieties Definitions. In this paper, we refer to the following monoid varieties. The definitions for $\mathbf{G}, \mathbf{J_1} = [\![x^2 = x, xy = yx]\!], \mathbf{R_1} = [\![xyx = xy]\!], \mathbf{ER_1}, \mathbf{J}, \mathbf{R}$ may be found in [9]. The definition for $\mathbf{EJ}$ is in [18]. Also, $\mathbf{ER} = [\![(x^\omega y^\omega)^\omega x^\omega = (x^\omega y^\omega)^\omega]\!]$, the variety considered in [6]. The respective language varieties corresponding to the monoid varieties above are denoted $\boldsymbol{\mathcal{G}}$ (group languages), $\boldsymbol{\mathcal{J}_1}$ (semilattice languages), $\boldsymbol{\mathcal{R}_1}$ ($\mathcal{R}$-trivial idempotent languages, or $\mathcal{R}_1$ languages), $\boldsymbol{\mathcal{ER}_1}$, $\boldsymbol{\mathcal{J}}$, $\boldsymbol{\mathcal{R}}$, $\boldsymbol{\mathcal{EJ}}$, $\boldsymbol{\mathcal{ER}}$. It is possible to check that $\mathbf{J_1} \subset \mathbf{J} \subset \mathbf{EJ}$, $\mathbf{R_1} \subset \mathbf{R} \subset \mathbf{ER}$, $\mathbf{R_1} \subset \mathbf{ER_1} \subset \mathbf{ER}$, $\mathbf{J_1} \subset \mathbf{R_1}$, $\mathbf{J} \subset \mathbf{R}$ and $\mathbf{G} \subset \mathbf{EJ} \subset \mathbf{ER}$.

Semilattice Languages and Free Semilattices. A free semilattice over an alphabet A is defined as a monoid $(\mathcal{P}(A), \cup)$, where $\cup$ is the ordinary set union. For any alphabet A, the free semilattice $\mathcal{P}(A)$ satisfies the identities of $\mathbf{J_1}$, therefore $\mathcal{P}(A) \in \mathbf{J_1}$. Given a free semilattice $\mathcal{P}(A)$, one may represent it as a deterministic finite automaton $(\mathcal{P}(A), A, \emptyset, \cdot)$, where for every $X \in \mathcal{P}(A)$ and for every $a \in A$, $X \cdot a = X \cup \{a\}$. It is implied by results in [17] that for any semilattice language L over alphabet A, $L\omega$ is a set of final states, such that the automaton recognizes the language. Therefore, in order to specify a particular language $L \in A^*\boldsymbol{\mathcal{J}_1}$, one may identify it by indicating a particular subset of $\mathcal{P}(A)$. A free semilattice over $\{a, b, c\}$ represented as a finite automaton is depicted in Figure 1 on the left side. The states of $(\mathcal{P}(A), A, \emptyset, \cdot)$ can be separated into several levels, i.e., a state is at level k if it corresponds to an element in $\mathcal{P}(A)$ of cardinality k.

$\boldsymbol{\mathcal{R}_1}$ **languages and Free Left Regular Bands.** A free left regular band over an alphabet A is defined as a monoid $(\mathcal{F}(A), \cdot)$, where $\mathbf{x} \cdot \mathbf{y} = (\mathbf{xy})\tau$, i.e., concatenation followed by the application of τ. For any alphabet A, the free left regular band $\mathcal{F}(A)$ satisfies the identities of $\mathbf{R_1}$, therefore $\mathcal{F}(A) \in \mathbf{R_1}$. Given a free left regular band $\mathcal{F}(A)$, one may represent it as a deterministic finite automaton $(\mathcal{F}(A), A, \varepsilon, \cdot_{\mathcal{F}(A)})$. A free left regular band over $\{a, b, c\}$ represented as a finite automaton is depicted in Figure 1 on the right side. It is implied by [19] that for any $\mathcal{R}_1$ language L over alphabet A, $L\tau$ is a set of final states, such that the automaton recognizes the language. Therefore, in order to specify a particular language $L \in A^*\boldsymbol{\mathcal{R}_1}$, one may identify it by indicating a particular subset of

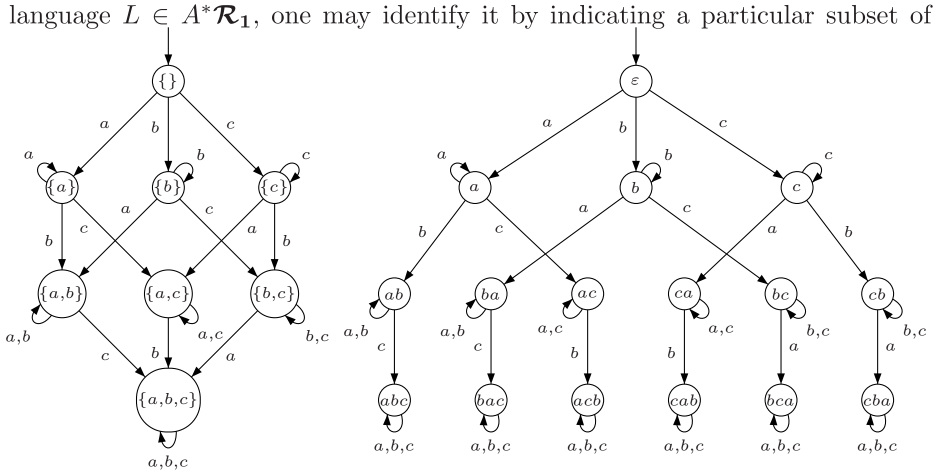

Fig. 1. Free semilattice and free left regular band over $\{a, b, c\}$.

$\mathcal{F}(A)$. For example, the semilattice language A^*aA^* may also be denoted as $\{\mathbf{a}, \mathbf{ab}, \mathbf{ba}, \mathbf{ac}, \mathbf{ca}, \mathbf{abc}, \mathbf{acb}, \mathbf{bac}, \mathbf{bca}, \mathbf{cab}, \mathbf{cba}\}$. We can also see that $\mathcal{P}(A)$ is a quotient of $\mathcal{F}(A)$. Indeed, let σ be a restriction of ω to $\mathcal{F}(A)$. The function σ is a surjective morphism from $\mathcal{F}(A)$ to $\mathcal{P}(A)$ which preserves the order relation.

Free left regular bands and free semilattices are key elements to prove that a quantum automaton may recognize a particular $\mathcal{R}_1$ language if and only if its system of linear inequalities is consistent.

3 Completely Positive Maps

In this section, we establish some facts about completely positive (CP) maps with certain properties, i.e., CP maps that describe the evolution of BQFA, defined in the next section. A comprehensive account on quantum computation and CP maps can be found in [16]. Following [16], we call a matrix $M \in \mathbb{C}^{n\times n}$ *positive*, if for any vector $X \in \mathbb{C}^n$, X^*MX is real and nonnegative. For arbitrary matrices M, N we may write $M \geqslant N$ if $M - N$ is positive. Let I_s be the identity map over $\mathbb{C}^{s\times s}$. Given Φ and Ψ, let $\Phi \bigotimes \Psi$ be the tensor product of those linear maps. A positive linear map Φ is called *completely positive*, if for any $s \geqslant 1$, $\Phi \bigotimes I_s$ is positive. Any CP map from $\mathbb{C}^{n\times n}$ to $\mathbb{C}^{m\times m}$ may be regarded as a linear operator in $\mathbb{C}^{n^2\times m^2}$. A CP map Φ is called *sub-tracial* iff for any positive M we have $\mathrm{Tr}(\Phi(M)) \leqslant \mathrm{Tr}(M)$. A CP map Φ from $\mathbb{C}^{n\times n}$ to $\mathbb{C}^{m\times m}$ is called *unital* if $\Phi(I_n) = I_m$. A CP map from $\mathbb{C}^{n\times n}$ to $\mathbb{C}^{m\times m}$ Φ is called *sub-unital* if $\Phi(I_n) \leqslant I_m$. A *composition* of CP maps $\Phi_0, ..., \Phi_m$ from $\mathbb{C}^{n\times n}$ to $\mathbb{C}^{n\times n}$ is a CP map $\Phi = \Phi_0 \circ \cdots \circ \Phi_m$ such that for any $M \in \mathbb{C}^{n\times n}$ $\Phi(M) = \Phi_0(\Phi_1(...(\Phi_m(M))...))$. A CP map Φ from $\mathbb{C}^{n\times n}$ to $\mathbb{C}^{n\times n}$ is called *bistochastic*, if it is both trace preserving and unital, i.e., for any positive M, $\mathrm{Tr}(\Phi(M)) = \mathrm{Tr}(M)$ and $\Phi(I_n) = I_n$. A CP map $\mathbb{C}^{n\times n}$ to $\mathbb{C}^{n\times n}$ is called *sub-bistochastic*, if it is both sub-unital and sub-tracial. A composition of two sub-bistochastic CP maps is a sub-bistochastic CP map. We are interested about some properties of the asymptotic dynamics resulting from iterative application of a CP sub-bistochastic map. A CP map Φ from $\mathbb{C}^{n\times n}$ to $\mathbb{C}^{n\times n}$ is called *idempotent* if $\Phi \circ \Phi = \Phi$. It is said that a CP map Φ from $\mathbb{C}^{n\times n}$ to $\mathbb{C}^{n\times n}$ *generates a unique idempotent*, denoted Φ^ω, if there exists a sequence of positive integers n_s such that 1) exists the limit $\Phi^\omega = \lim_{s\to\infty} \Phi^{n_s}$; 2) the CP map Φ^ω is idempotent; 3) for any sequence of positive integers m_s such that the limit $\lim_{s\to\infty} \Phi^{m_s}$ exists and is idempotent, $\lim_{s\to\infty} \Phi^{m_s} = \Phi^\omega$. By Kuperberg [12], for any CP sub-bistochastic map Φ from $\mathbb{C}^{n\times n}$ to $\mathbb{C}^{n\times n}$, its idempotent Φ^ω exists, is unique and is a linear projection operator in $\mathbb{C}^{n^2\times n^2}$. That implies the subsequent theorem, which ultimately is the reason why certain models of quantum finite automata cannot recognize all regular languages.

Theorem 3.1. *Let $e_1, ..., e_k$ be idempotent CP sub-bistochastic maps from $\mathbb{C}^{n\times n}$ to $\mathbb{C}^{n\times n}$. Then for any i, $1 \leqslant i \leqslant k$, (1) $\lim_{n\to\infty}(e_1 \circ ... \circ e_k)^n = (e_1 \circ ... \circ e_k)^\omega = (e_{\pi(1)} \circ ... \circ e_{\pi(k)})^\omega$, where π is a permutation in $\{1, \ldots, k\}$; (2) $(e_1 \circ ... \circ e_k)^\omega = e_i \circ (e_1 \circ ... \circ e_k)^\omega = (e_1 \circ ... \circ e_k)^\omega \circ e_i$.*

Any finite quantum system at a particular moment of time (i.e., its *mixed state*) is described by a *density matrix*. By [16, Theorem 2.5], a matrix is a density matrix if and only if it is positive and its trace is equal to 1. Informally, an $n \times n$ density matrix describes a quantum system with n states. A completely positive trace-preserving map describes an evolution of a quantum system as allowed by quantum mechanics. It maps a density matrix to a density matrix.

4 Automata Models

For the formal definitions of other indicated automata models, the reader is referred to the following references. "Classical" models: Group Automata (GA, [21]), Measure-Once Quantum Finite Automata (MO-QFA, [15, 5]), "Classical" Probabilistic Reversible Automata (C-PRA, [7, 1]), Latvian Quantum Finite Automata (LQFA, [1]). "Decide-and-halt" models: Reversible Finite Automata (RFA, [2, 9]), Measure-Many Quantum Finite Automata (MM-QFA, [11, 5, 3, 1]), "Decide-and-halt" Probabilistic Reversible Automata (DH-PRA, [7, 8]), Enhanced Quantum Finite Automata (EQFA, [14]). In case of classical acceptance, an automaton reads an input word until the last letter, and then accepts or rejects the word depending on whether the current state is final or non-final. In case of "decide-and-halt" acceptance, the automaton reads the input word until it enters a halting state. The input is accepted or rejected depending on whether the halting state is accepting or rejecting. Every word is appended with a special symbol, an end-marker, to ensure that any word is either accepted or rejected. We define MO and MM bistochastic QFA as a generalization of these models, which allows to prove the limitations of language recognition for all the models within single framework.
A bistochastic quantum finite automaton (BQFA) is a tuple $(Q, A \cup \{\#, \$\}, q_0, \{\Phi_a\})$*, where* Q *is a finite set of states,* A *- a finite input alphabet,* $\#, \$ \notin A$ *- initial and final end-markers,* q_0 *- an initial state and for each* $a \in A \cup \{\#, \$\}$ Φ_a *is a CP bistochastic transition map from* $\mathbb{C}^{|Q| \times |Q|}$ *to* $\mathbb{C}^{|Q| \times |Q|}$.
Regardless of which word acceptance model is used, each input word is enclosed into end-markers $\#, \$$. At any step, the mixed state of a BQFA may be described by a density matrix ρ. The computation starts in the state $|q_0\rangle\langle q_0|$.
Operation of a measure-once BQFA and word acceptance is the same as described for LQFA [1], only instead of sequences of unitary operations and orthogonal measurements we have arbitrary bistochastic CP maps. On input letter $a \in A$, ρ is transformed into $\Phi_a(\rho)$.
Operation of a measure-many BQFA and word acceptance is the same as described for EQFA [14], but arbitrary bistochastic CP maps are used. The set of states Q is partitioned into three disjoint subsets Q_{non}, Q_{acc} and Q_{rej} - non-halting, accepting and rejecting states, respectively. On input letter $a \in A$, ρ is transformed into $\rho' = \Phi_a(\rho)$. After that, a measurement $\{P_{non}, P_{acc}, P_{rej}\}$ is applied to ρ', where for each $i \in \{non, acc, rej\}$ $P_i = \sum_{q \in Q_i} |q\rangle\langle q|$. To describe the probability distribution $S_{\#\mathbf{u}}$ of a MM-BQFA $\mathcal{A}$ after reading some prefix $\#\mathbf{u}$, it is convenient to use density matrices ρ scaled by p, $0 \leqslant p \leqslant 1$. So the probability distribution $S_{\#\mathbf{u}}$ of $\mathcal{A}$ is a triple (ρ, p_{acc}, p_{rej}), where $\mathrm{Tr}(\rho) + p_{acc} + p_{rej} = 1$, $\rho / \mathrm{Tr}(\rho)$ is the current mixed state and p_{acc}, p_{rej} are respectively the probabil-

ities that $\mathcal{A}$ has accepted or rejected the input. So the scaled density matrix ρ may be called a *scaled mixed state*. For any $a \in A \cup \{\#, \$\}$, let $\Psi_a(\rho) = P_{non}\Phi_a(\rho)P_{non}$. After reading the next input letter a, the probability distribution is $S_{\#\mathbf{u}a} = (\Psi_a(\rho), p_{acc} + \mathrm{Tr}(P_{acc}\Phi_a(\rho)P_{acc}), p_{rej} + \mathrm{Tr}(P_{rej}\Phi_a(\rho)P_{rej}))$. For any word $\mathbf{a} = a_1 \ldots a_k$, define $\Psi_{\mathbf{a}} = \Psi_{a_k} \circ \cdots \circ \Psi_{a_1}$. Hence $\rho = \Psi_{\#\mathbf{u}}(|q_0\rangle\langle q_0|)$. Note that $\Psi_{\mathbf{a}}$ is a CP sub-bistochastic map.

Language recognition is defined in the same way as in Rabin's [20]. Suppose that an automaton $\mathcal{A}$ is one of the models from the list above. By $p_{\mathbf{x},\mathcal{A}}$ (or $p_{\mathbf{x}}$, if no ambiguity arises) we denote the probability that an input $\mathbf{x}$ is accepted by the automaton $\mathcal{A}$. We consider only bounded error language recognition.

BQFA as a generalization of other models. Since unitary operations and orthogonal measurements are bistochastic operations, MO-BQFA is a generalization of LQFA and MM-BQFA is a generalization of EQFA. Also, the Birkhoff theorem [22, Theorem 4.21] implies that MO-BQFA and MM-BQFA are generalizations of C-PRA and DH-PRA, respectively. On the other hand, BQFA are a special case of one-way general QFA, which admit any CP trace-preserving transition maps. One-way general QFA recognize with bounded error exactly the regular languages [10, 13]. So the recognition power of BQFA is also limited to regular languages only.

Comparison of the language classes. Having a certain class of automata $\boldsymbol{\mathcal{A}}$, let us denote by $\boldsymbol{\mathcal{L}}(\boldsymbol{\mathcal{A}})$ the respective class of languages. Thus $\boldsymbol{\mathcal{L}}(\text{GA}) = \boldsymbol{\mathcal{L}}(\text{MO-QFA}) = \boldsymbol{\mathcal{G}}$, $\boldsymbol{\mathcal{L}}(\text{C-PRA}) = \boldsymbol{\mathcal{L}}(\text{LQFA}) = \boldsymbol{\mathcal{L}}(\text{MO-BQFA}) = \boldsymbol{\mathcal{EJ}}$, $\boldsymbol{\mathcal{G}} \subsetneq \boldsymbol{\mathcal{L}}(\text{RFA}) \subsetneq \boldsymbol{\mathcal{ER}_1}$, $\boldsymbol{\mathcal{EJ}} \subsetneq \boldsymbol{\mathcal{L}}(\text{MM-QFA}) \stackrel{?}{=} \boldsymbol{\mathcal{L}}(\text{DH-PRA}) \stackrel{?}{=} \boldsymbol{\mathcal{L}}(\text{EQFA}) \stackrel{?}{=} \boldsymbol{\mathcal{L}}(\text{MM-BQFA}) \subsetneq \boldsymbol{\mathcal{ER}}$. Relations concerning BQFA depend on the theorem below. All the other relations are known from the references given in the list of automata models above.

Theorem 4.1. *$\boldsymbol{\mathcal{L}}(\text{MO-BQFA}) = \boldsymbol{\mathcal{EJ}}$ and $\boldsymbol{\mathcal{L}}(\text{MM-BQFA}) \subseteq \boldsymbol{\mathcal{ER}}$. $\boldsymbol{\mathcal{L}}(\text{MM-BQFA})$ is closed under complement, inverse free monoid morphisms, and word quotient.*

The proof relies on Theorem 3.1 and ideas in [1, 5] used for LQFA and MM-QFA. We find that $\boldsymbol{\mathcal{L}}(\text{MM-BQFA})$ shares a lot of properties with the language classes of other "decide-and-halt" word acceptance models. In Section 7 it is noted that MM-BQFA does not recognize any of the languages corresponding to "forbidden constructions" from [3, Theorem 4.3]. As other "decide-and-halt" models, $\boldsymbol{\mathcal{L}}(\text{MM-BQFA}) \subsetneq \boldsymbol{\mathcal{ER}}$ and $\boldsymbol{\mathcal{L}}(\text{MM-BQFA})$ is not closed under union and intersection (Corollary 6.3).

5 Linear Inequalities

In this section, we define a system of linear inequalities that an $\mathcal{R}_1$ language recognized by a MM-BQFA must satisfy. Let L be an $\mathcal{R}_1$ language and $\mathcal{S}$ - a MM-BQFA, both over alphabet A. Let $\{\mathbf{v_0}, \mathbf{v_1}, ..., \mathbf{v_R}\} = \mathcal{F}(A)$. Assume $\mathbf{v_0} = \varepsilon$. For any $\mathbf{v}$ in $\mathcal{F}(A)$, where $\mathbf{v} = a_1...a_k$ (a_i are distinct letters of A), denote by $\mathbf{v}[i]$ a prefix of $\mathbf{v}$ of length i, i.e. $\mathbf{v}[0] = \varepsilon$ and for all i, $1 \leqslant i \leqslant k$, $\mathbf{v}[i] = a_1...a_i$. Recall that $\mathcal{F}(A)$ can be viewed as an automaton that recognizes an $\mathcal{R}_1$ language L, provided $L\tau$ is its set of final states (see Section 2).

Now define a linear system of inequalities $\mathfrak{L}$ as follows: *(1) For every* $\mathbf{v}$ *in* $\mathcal{F}(A)$, *where* $\mathbf{v} = a_1...a_k$ *take the formal expression* $\mathfrak{L}(\mathbf{v}) = x_0 + x_{\mathbf{v}[0]\omega,a_1} + x_{\mathbf{v}[1]\omega,a_2} + x_{\mathbf{v}[2]\omega,a_3} + ... + x_{\mathbf{v}[k-1]\omega,a_k} + y_{\mathbf{v}\omega}$; *(2) Introduce two another variables* p_1 *and* p_2. *For any* $\mathbf{v} \in \mathcal{F}(A)$, *if* $\mathbf{v} \in L\tau$, *construct an inequality* $\mathfrak{L}(\mathbf{v}) \geqslant p_2$, *otherwise construct an inequality* $\mathfrak{L}(\mathbf{v}) \leqslant p_1$; *(3) Append the system by an inequality* $p_1 < p_2$.

Example 5.1. Consider an $\mathcal{R}_1$ language $L = \{\mathbf{ab}, \mathbf{bac}\}$ over alphabet $A = \{a, b, c\}$. Among others, the system $\mathfrak{L}(L)$ has the following inequalities:

$$\begin{array}{ll} \mathfrak{L}(\mathbf{ab}) \;= x_0 + x_{\{\},a} + x_{\{a\},b} + y_{\{a,b\}} & \geqslant p_2 \\ \mathfrak{L}(\mathbf{bac}) = x_0 + x_{\{\},b} + x_{\{b\},a} + x_{\{a,b\},c} + y_{\{a,b,c\}} & \geqslant p_2 \\ \mathfrak{L}(\mathbf{ba}) \;= x_0 + x_{\{\},b} + x_{\{b\},a} + y_{\{a,b\}} & \leqslant p_1 \\ \mathfrak{L}(\mathbf{abc}) = x_0 + x_{\{\},a} + x_{\{a\},b} + x_{\{a,b\},c} + y_{\{a,b,c\}} & \leqslant p_1 \\ & p_1 < p_2 \end{array}$$

Informally, an inequality $\mathfrak{L}(\mathbf{v})$ represents the probability of accepting a specifically defined input word $\mathbf{u}$, such that $\mathbf{u}\tau = \mathbf{v} = a_1a_2...a_k$. The variable x_0 represents the probability to accept the input after reading the initial end-marker #. The variable $y_{\mathbf{v}\omega}$ represents the cumulative probability to stay in non-halting states before reading the final end-marker \$ and accept the input after reading it. The variable $x_{\mathbf{v}[i-1]\omega,a_i}$ represents the cumulative probability to stay in non-halting states after reading a (specifically defined) prefix $\mathbf{u}[i-1]$ (of $\mathbf{u}$) such that $\mathbf{u}[i-1]\tau = \mathbf{v}[i-1]$ and to accept input after reading a (specifically defined) prefix $\mathbf{u}[i]$ such that $\mathbf{u}[i]\tau = \mathbf{v}[i]$.

Theorem 5.2. *Suppose* L *is an* $\mathcal{R}_1$ *language. If the linear system* $\mathfrak{L}$ *is not consistent, then* L *cannot be recognized by any MM-BQFA.*

Proof. Let m_l $(l = 1, 2, ...)$ be a sequence of positive integers such that for all letters $a \in A$ $\lim_{l\to\infty} \Psi_a^{m_l} = \Psi_a^{\omega}$ (existence is proved in the same way as the Kuperberg's result quoted in Section 3). Let μ be a function that assigns to any word in A^* the same word (of the same length) with letters sorted in alphabetical order. Let $\varkappa_i$, $i \in \mathbb{N}$, a morphism from A^* to A^* such that for any $a \in A$ $a\varkappa_i = a^i$. Let $\xi = \xi_l$ be an everywhere defined function from $\mathcal{F}(A)$ to A^*, such that $\varepsilon\xi = \varepsilon$ and for all $\mathbf{v} \in \mathcal{F}(A)$, if $|\mathbf{v}| = 1$ then $\mathbf{v}\xi = \mathbf{v}^{m_l}$ and otherwise, if $|\mathbf{v}| \geqslant 2$ then $\mathbf{v}\xi = (\mathbf{v}\mu\varkappa_{m_l})^l$.

Let us define the function θ (which depends on the parameter l) as follows. For any $\mathbf{v} = a_1...a_k$ let $\mathbf{v}\theta = \mathbf{v}[1]\xi...\mathbf{v}[k]\xi = a_1^{m_l}((a_1^{m_l}a_2^{m_l})\mu)^l...((a_1^{m_l}a_2^{m_l}\ldots a_k^{m_l})\mu)^l$. Let $\varepsilon\theta = \varepsilon$. Note that $\mathbf{v}[i]\theta = (\mathbf{v}[i-1]\theta)(\mathbf{v}[i]\xi)$. In the discussion preceding this theorem, the word $\mathbf{u}$ corresponding to $\mathbf{v}$ is $\mathbf{v}\theta$ and the prefix $\mathbf{u}[i]$ corresponding to $\mathbf{v}[i]$ is $\mathbf{v}[i]\theta$. Note that $\mathbf{v}\theta\tau = \mathbf{v}$.

Now take the set $\mathcal{F}(A)\theta = \{\mathbf{u_k} \mid \mathbf{u_k} = \mathbf{v_k}\theta \text{ and } 0 \leqslant k \leqslant R\}$. Let us take a positive integer i and any two words $\mathbf{u}$ and $\mathbf{u}'$ in $\mathcal{F}(A)\theta$, such that $\mathbf{u}[i-1]\omega = \mathbf{u}'[i-1]\omega$. Let $\mathbf{v} = \mathbf{u}\tau$ and $\mathbf{v}' = \mathbf{u}'\tau$. If the parameter l is sufficiently large, Theorem 3.1 implies that after reading $\mathbf{u}[i-1]$ and $\mathbf{u}'[i-1]$ the automaton $\mathcal{S}$ has essentially the same scaled density matrices (which represent the non-halting states). Suppose that $\mathbf{v}[i] = \mathbf{v}'[i] = a_1...a_i$. The automaton $\mathcal{S}$ finishes reading the prefixes $\mathbf{u}[i]$ and $\mathbf{u}'[i]$ after reading the next symbols forming the sub-word $\mathbf{v}[i]\xi$. The cumulative probabilities to stay in non-halting states while reading $\mathbf{u}[i-1]$, $\mathbf{u}'[i-1]$ and to accept input after reading $\mathbf{v}[i]\xi = \mathbf{v}'[i]\xi$ will be essentially the same. (They converge to the same value as l tends to infinity). Hence those probabilities are reflected in the system of linear inequalities by the same variable $x_{\mathbf{v}[i-1],a_i}$. Thus, if a MM-BQFA $\mathcal{S}$ recognizes an $\mathcal{R}_1$ language L, then the linear system of inequalities $\mathfrak{L}$ has to be consistent. □

If the linear system $\mathfrak{L}(L)$ is not consistent, then L cannot be recognized by any MM-QFA, DH-PRA or EQFA as well. The statement converse to Theorem 5.2 is provided in Section 6 (Theorem 6.2).

Consider the inequalities in the system $\mathfrak{L}(L)$. The only possible coefficients of variables in any linear inequality are -1, 0 and 1. Denote by $Z = \{x_0, z_1, ..., z_s, y_1, ..., y_t, p_1, p_2\}$ the set of all the variables in the system $\mathfrak{L}$, where z_i are variables of the form $x_{\mathbf{v}[i-1]\omega,a_i}$, and y_i are variables of the form $y_{\mathbf{v}\omega}$.

Proposition 5.3. *The system $\mathfrak{L}$ is consistent if and only if it has a solution where $x_0 = 0$, $y_A = 0$, $0 \leqslant p_1, p_2 \leqslant 1$ and all the other variables $z_1, ..., z_s, y_1, ..., y_{t-1}$ are assigned real values from 0 to $1/|A|$.*

6 Construction of DH-PRA and MM-QFA for $\mathcal{R}_1$ languages

Preparation of a linear programming problem. Consider an $\mathcal{R}_1$ language L over alphabet A. Construct the respective system of linear inequalities $\mathfrak{L}$. Obtain a system $\mathfrak{L}_1$ by supplementing $\mathfrak{L}$ with additional inequalities that enforce the constraints expressed in Proposition 5.3, according to which $\mathfrak{L}$ is consistent if and only if $\mathfrak{L}_1$ is consistent. Obtain a system $\mathfrak{L}_1'$ by replacing in $\mathfrak{L}_1$ the inequality $p_1 < p_2$ by $p_1 \leqslant p_2$. The linear programming problem, denoted $\mathfrak{P}$, is to maximize $p_2 - p_1$ according to the constraints expressed by $\mathfrak{L}_1'$. Since $\mathfrak{L}_1'$ is homogenous, it always has a solution where $p_1 = p_2$. Since the solution polytope of $\mathfrak{L}_1'$ is bounded, $\mathfrak{P}$ always has an optimal solution. Obviously, if the optimal solution yields $p_1 = p_2$, then $\mathfrak{L}_1$ is not consistent and therefore, by Theorem 5.2, a DH-PRA that recognizes L does not exist. Otherwise, if the optimal solution yields $p_1 < p_2$, then $\mathfrak{L}_1$ is consistent.
Automata derived from the free semilattice $\mathcal{P}(A)$. Assume $\mathfrak{L}_1$ is consistent, so we are able to obtain a solution of $\mathfrak{P}$ where $p_1 < p_2$. Given any expression Z of variables from $\mathfrak{L}_1$, let $\mathfrak{P}(Z)$ - the value which is assigned to Z by solving $\mathfrak{P}$. First, we use the obtained solution to construct probabilistic automata $\mathcal{A}_i$, $1 \leqslant i \leqslant |A|$. Those automata are not probabilistic reversible. Similarly as in the "decide-and-halt" model, the constructed automata have accepting, rejecting and non-halting states. Any input word is appended by the end-marker \$. The initial end-marker # is not used for those automata themselves. Any automaton $\mathcal{A}_i$ is a tuple $(Q_i, A \cup \{\$\}, s_i, \delta_i)$, where Q_i is a set of states, s_i - an initial state and δ_i - a transition function $Q \times A \times Q \longrightarrow [0,1]$, so $\delta_i(q, a, q')$ is a probability of transit from q to q' on reading input letter a. $\mathcal{A}_i$ is constructed

as follows: (1) Take the deterministic automaton $(\mathcal{P}(A), A, \emptyset, \cdot)$, remove all the states at level greater or equal to i. The remaining states are defined to be non-halting. The state $\emptyset$ is initial, it is the only state of $\mathcal{A}_i$ at level 0. For any a in A and state $\mathbf{s}$ at levels $\{0, \ldots, i-2\}$, $\delta_i(\mathbf{s}, a, \mathbf{s}\cdot a) = 1$. For any state $\mathbf{s}$ at level $i-1$ and any a in $\mathbf{s}$, $\delta_i(\mathbf{s}, a, \mathbf{s}) = 1$; (2) For any non-halting state $\mathbf{s}$ at levels $\{0, \ldots, i-2\}$, add a rejecting state $(\mathbf{s}\$)_{rej}$. Let $\delta_i(\mathbf{s}, \$, (\mathbf{s}\$)_{rej}) = 1$; (3) For any state $\mathbf{s}$ at level $i-1$, add $|A| - |\mathbf{s}| + 1$ accepting states $(\mathbf{s}a)_{acc}$, $a \in (A \setminus \mathbf{s}) \cup \{\$\}$. Also add $|A| - |\mathbf{s}| + 1$ rejecting states $(\mathbf{s}a)_{rej}$, $a \in (A \setminus \mathbf{s}) \cup \{\$\}$; (4) If $a \in A \setminus \mathbf{s}$, any element $\mathbf{s}'a$ in $(\mathbf{s}\sigma^{-1})a$ defines the same variable $x_{\mathbf{s},a}$ in the system of linear inequalities (σ is defined in Section 2). Let $c_{\mathbf{s},a} = \mathfrak{B}(x_{\mathbf{s},a})$. Any element $\mathbf{s}'$ in $\mathbf{s}\sigma^{-1}$ defines the same variable $y_{\mathbf{s}}$. Let $d_{\mathbf{s}} = \mathfrak{B}(y_{\mathbf{s}})$; (5) Define missing transitions for the states at level $i-1$. For any state $\mathbf{s}$ at level $i-1$ and any a in $A \setminus \mathbf{s}$, let $t_{\mathbf{s},a} = \delta_i(\mathbf{s}, a, (\mathbf{s}a)_{acc}) = c_{\mathbf{s},a}|A|$ and $\delta_i(\mathbf{s}, a, (\mathbf{s}a)_{rej}) = 1 - t_{\mathbf{s},a}$. Let $v_{\mathbf{s}} = \delta_i(\mathbf{s}, \$, (\mathbf{s}\$)_{acc}) = d_{\mathbf{s}}|A|$ and $\delta_i(\mathbf{s}, \$, (\mathbf{s}\$)_{rej}) = 1 - v_{\mathbf{s}}$; (6) The formally needed transitions outgoing the halting states for now are left undefined.

Consider an automaton $\mathcal{A}$ (Figure 2), which with the same probability $1/|A|$ executes any of the automata $\mathcal{A}_1, ..., \mathcal{A}_{|A|}$ (i.e., it uses the initial end-marker # to transit to initial states of any of those automata). By construction of $\mathcal{A}_1, ..., \mathcal{A}_{|A|}$, the automaton $\mathcal{A}$ accepts any word $\mathbf{u} \in A^*$ with probability $\mathfrak{P}(\mathfrak{L}(\mathbf{u}\tau))$. Since for any word $\mathbf{u} \in L$, $\mathfrak{P}(\mathfrak{L}(\mathbf{u}\tau)) \geqslant \mathfrak{P}(p_2)$, and for any word $\mathbf{w} \notin L$, $\mathfrak{P}(\mathfrak{L}(\mathbf{w}\tau)) \leqslant \mathfrak{P}(p_1)$, the automaton $\mathcal{A}$ recognizes the language L.

Construction of a DH-PRA. In order to construct a DH-PRA recognizing L, it remains to demonstrate that any of the automata $\mathcal{A}_1, ..., \mathcal{A}_{|A|}$ may be simulated by some DH probabilistic reversible automata, that is, for any automaton $\mathcal{A}_i$, it is possible to construct a sequence of DH-PRA $\mathcal{S}_{i,n}$, where $n \geqslant 1$, such that $p_{\mathbf{w},\mathcal{S}_{i,n}}$ converges uniformly to $p_{\mathbf{w},\mathcal{A}_i}$ on A^* as $n \to \infty$. An automaton $\mathcal{A}_i = (Q_i, A \cup \{\$\}, s_i, \delta_i)$ is used to construct a DH-PRA $\mathcal{S}_{i,n} = (Q_{i,n}, A \cup \{\$\}, s_i, \delta_{i,n})$

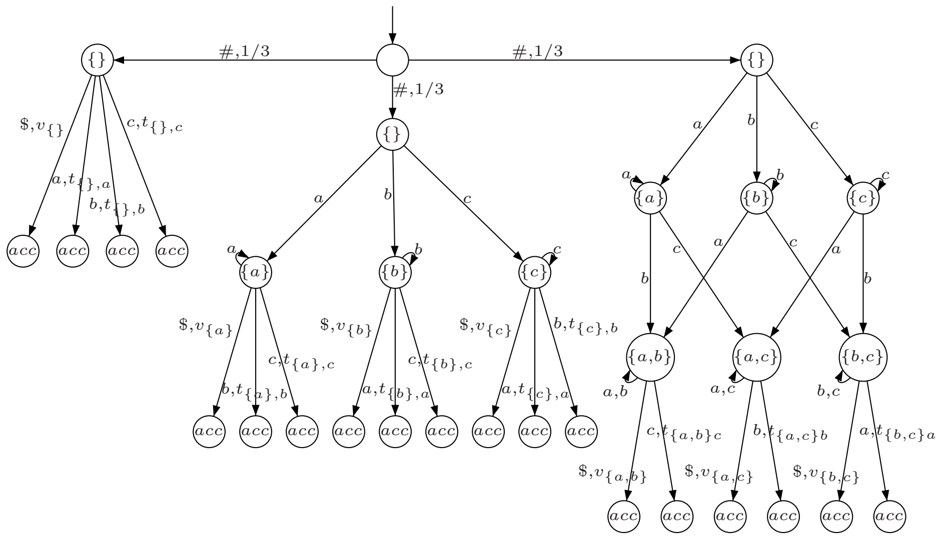

Fig. 2. An automaton $\mathcal{A}$ over alphabet $\{a, b, c\}$, the rejecting states are not shown.

as described next. Initially $Q_{i,n}$ is empty. (1) For any non-halting state $\mathbf{s}$ at level j, $0 \leqslant j \leqslant i-1$, supplement $\mathcal{S}_{i,n}$ with non-halting states denoted $\mathbf{s}_k$, where $1 \leqslant k \leqslant n^j$; (2) For any non-halting state $\mathbf{s}$ at level j, $0 \leqslant j < i-1$, supplement $\mathcal{S}_{i,n}$ with rejecting states $(\mathbf{s}\$)_{rej,k}$, where $1 \leqslant k \leqslant n^j$; (3) For any non-halting state $\mathbf{s}$ at level $i-1$, accepting state $(\mathbf{s}a)_{acc}$ and rejecting state $(\mathbf{s}a)_{rej}$, where $a \in (A \setminus \mathbf{s}) \cup \{\$\}$, supplement $\mathcal{S}_{i,n}$ with accepting states $(\mathbf{s}a)_{acc,k}$ and rejecting states $(\mathbf{s}a)_{rej,k}$, where $1 \leqslant k \leqslant n^{i-1}$.

It remains to define the transitions. For any non-halting state $\mathbf{s}$ of $\mathcal{A}_i$ at level j, $1 \leqslant j \leqslant i-1$, the states in $\{\mathbf{s}_k\}$ are grouped into n^{j-1} disjoint subsets with n states in each, so that any state in $\{\mathbf{s}_k\}$ may be denoted as $\mathbf{s}_{l,m}$, where $1 \leqslant l \leqslant n^{j-1}$ and $1 \leqslant m \leqslant n$. For any letter a in A, consider all pairs of non-halting states $\mathbf{s}, \mathbf{t}$ of $\mathcal{A}_i$ such that $\mathbf{s} \neq \mathbf{t}$ and $\delta_i(\mathbf{s}, a, \mathbf{t}) = 1$. For any fixed k and any l and m, $1 \leqslant l, m \leqslant n$, define $\delta_{i,n}(\mathbf{s}_k, a, \mathbf{s}_k) = \delta_{i,n}(\mathbf{s}_k, a, \mathbf{t}_{k,m}) = \delta_{i,n}(\mathbf{t}_{k,m}, a, \mathbf{s}_k) = \delta_{i,n}(\mathbf{t}_{k,l}, a, \mathbf{t}_{k,m}) = 1/(n+1)$. For any non-halting state $\mathbf{s}$ of $\mathcal{A}_i$ at level j, $0 \leqslant j < i-1$, $\delta_{i,n}(\mathbf{s}_k, \$, (\mathbf{s}\$)_{rej,k}) = 1$, $\delta_{i,n}((\mathbf{s}\$)_{rej,k}, \$, \mathbf{s}_k) = 1$. For the same $(\mathbf{s}\$)_{rej,k}$ and any other letter b in $A \cup \{\$\}$, define $\delta_{i,n}((\mathbf{s}\$)_{rej,k}, b, (\mathbf{s}\$)_{rej,k}) = 1$. For any non-halting state $\mathbf{s}$ of $\mathcal{A}_i$ at level $i-1$ and $a \in (A \setminus \mathbf{s}) \cup \{\$\}$, the transitions induced by a among $\mathbf{s}_k$, $(\mathbf{s}a)_{acc,k}$, $(\mathbf{s}a)_{rej,k}$ are defined by the matrix $\begin{pmatrix} 0 & 0 & 1 \\ r_1 & r_2 & 0 \\ r_2 & r_1 & 0 \end{pmatrix}$, where $r_1 = \delta_i(\mathbf{s}, a, (\mathbf{s}a)_{acc})$, $r_2 = \delta_i(\mathbf{s}, a, (\mathbf{s}a)_{rej})$. The first, second and third rows and columns are indexed by $\mathbf{s}_k$, $(\mathbf{s}a)_{acc,k}$, $(\mathbf{s}a)_{rej,k}$, respectively. Note that $r_1 + r_2 = 1$. For the same $(\mathbf{s}a)_{acc,k}, (\mathbf{s}a)_{rej,k}$ and any other letter b in $A \cup \{\$\}$, define $\delta_{i,n}((\mathbf{s}a)_{acc,k}, b, (\mathbf{s}a)_{acc,k}) = \delta_{i,n}((\mathbf{s}a)_{rej,k}, b, (\mathbf{s}a)_{rej,k}) = 1$. We have defined all the non-zero transitions for $\mathcal{S}_{i,n}$. By construction, the transition matrices induced by any letter a in $A \cup \{\$\}$ are doubly stochastic.

Lemma 6.1. *For any i, $1 \leqslant i \leqslant |A|$, $p_{\mathbf{w},\mathcal{S}_{i,n}}$ converges uniformly to $p_{\mathbf{w},\mathcal{A}_i}$ on A^* as $n \to \infty$.*

Now it is possible to construct a DH-PRA $\mathcal{S} = (Q, A \cup \{\#, \$\}, s, \delta)$, which with the same probability $1/|A|$ executes the automata $\mathcal{S}_{1,n}, \dots, \mathcal{S}_{|A|,n}$. The set of states Q is a disjoint union of $Q_1, ..., Q_{|A|}$. Take the initial state s_i of any $\mathcal{S}_{i,n}$ as the initial state s. For any $a \in A \cup \{\$\}$ and $q_1, q_2 \in Q_i$, $\delta(q_1, a, q_2) = \delta_i(q_1, a, q_2)$. For any initial states s_i and s_j of $\mathcal{S}_{i,n}$ and $\mathcal{S}_{j,n}$, $\delta(s_i, \#, s_j) = 1/|A|$. For any other state q, $\delta(q, \#, q) = 1$. So the transition matrices of $\mathcal{S}$ induced by any letter are doubly stochastic. By Lemma 6.1, $\mathcal{S}$ recognizes L if n is sufficiently large. Hence we have established one of the main results of this section:

Theorem 6.2. *Suppose L is an $\mathcal{R}_1$ language. If the linear system $\mathfrak{L}$ is consistent, then L can be recognized by a DH-PRA.*

Therefore, if the linear system $\mathfrak{L}$ is consistent, then L can be recognized by a MM-BQFA as well. Moreover, since all of the transition matrices of the constructed DH-PRA are also unitary stochastic, by [7, Theorem 5.2] L can be recognized by an EQFA.

Corollary 6.3. *The class $\boldsymbol{\mathcal{L}}$(MM-BQFA) is not closed under union and intersection. Moreover, $\boldsymbol{\mathcal{L}}$(MM-BQFA) $\subsetneq \boldsymbol{\mathcal{ER}}$.*

Proof. See the language $L = \{\mathbf{ab}, \mathbf{bac}\}$ in Example 5.1; the languages $\{\mathbf{ab}\}$ and $\{\mathbf{bac}\}$ can be recognized since their systems are consistent, while the system $\mathfrak{L}(L)$ is inconsistent. □

The construction of MM-QFA for $\mathcal{R}_1$ languages has some peculiarities which have to be addressed separately. Specifically, there exist semilattice languages that MM-QFA do not recognize with probability $1-\epsilon$ [2, Theorem 2] and therefore they can't simulate with the same accepting probabilities the automata $\mathcal{A}_1, ..., \mathcal{A}_{|A|}$. Nevertheless, since the matrices used in the construction of DH-PRA are unitary stochastic, a modified construction is still possible.

Theorem 6.4. *Suppose L is an $\mathcal{R}_1$ language. If the linear system $\mathfrak{L}$ is consistent, then L can be recognized by a MM-QFA.*

In summary, Theorems 5.2, 6.2 and 6.4 imply that an $\mathcal{R}_1$ language L can be recognized by MM-QFA if and only if the linear system $\mathfrak{L}(L)$ is consistent. MM-QFA, DH-PRA, EQFA and MM-BQFA recognize exactly the same $\mathcal{R}_1$ languages.

7 "Forbidden Constructions"

In [3, Theorem 4.3], Ķikusts has proposed "forbidden constructions" for MM-QFA; any regular language whose minimal deterministic finite automaton contains any of these constructions cannot be recognized by MM-QFA. It is actually implied by Theorem 3.1 that the same is true for MM-BQFA and other "decide-and-halt" models from Section 4. Also, by Theorem 4.1 any language that is recognized by a MM-BQFA is contained in $\boldsymbol{\mathcal{ER}}$. Therefore it is legitimate to ask whether all the $\boldsymbol{\mathcal{ER}}$ languages that do not contain any of the "forbidden constructions" can be recognized by MM-BQFA. The answer to this question is *negative*.

Theorem 7.1. *There exists an $\boldsymbol{\mathcal{ER}}$ language that does not contain any of the "forbidden constructions" and still cannot be recognized by MM-BQFA.*

Proof. It is sufficient to indicate a language in $\boldsymbol{\mathcal{R}_1}$ which satisfies the required properties and for which the system of linear inequalities is inconsistent. Such a language is $L = \{\mathbf{aedbc}, \mathbf{beca}, \mathbf{beda}, \mathbf{bedac}, \mathbf{eacb}, \mathbf{eacbd}, \mathbf{eadbc}, \mathbf{ebca}\}$ over alphabet $A = \{a, b, c, d, e\}$. □

8 Conclusion

In the paper we show which $\mathcal{R}_1$ languages can be recognized by "decide-and-halt" quantum automata. It is expected that these results can be first generalized to include any $\mathcal{R}$-trivial language, and finally, any language in $\boldsymbol{\mathcal{ER}}$, thus obtaining the solution of the language class problem for MM-QFA. To apply the same approach for $\mathcal{R}$-trivial languages, one would need to find convenient sets of finite $\mathcal{R}$-trivial and $\mathcal{J}$-trivial monoids that generate the varieties $\mathbf{R}$ and $\mathbf{J}$ and a function resembling θ (in the proof of Theorem 5.2). That is a subject for further research.

References

1. A. Ambainis, M. Beaudry, M. Golovkins, A. Ķikusts, M. Mercer, D. Thérien. Algebraic Results on Quantum Automata. *Theory of Computing Systems, Vol. 39(1), pp. 165-188*, 2006.
2. A. Ambainis, R. Freivalds. 1-Way Quantum Finite Automata: Strengths, Weaknesses and Generalizations. *Proc. 39th FOCS, pp. 332-341*, 1998.
3. A. Ambainis, A. Ķikusts, M. Valdats. On the Class of Languages Recognizable by 1-Way Quantum Finite Automata. *STACS 2001, Lecture Notes in Computer Science, Vol. 2010, pp. 75-86*, 2001.
4. M.P. Bianchi, B. Palano. Behaviours of Unary Quantum Automata. *Fundamenta Informaticae, Vol. 104, pp. 1-15*, 2010.
5. A. Brodsky, N. Pippenger. Characterizations of 1-Way Quantum Finite Automata. *SIAM Journal on Computing, Vol. 31(5), pp. 1456-1478*, 2002.
6. S. Eilenberg. Automata, Languages and Machines, Vol. B. *Academic Press, New York*, 1976.
7. M. Golovkins, M. Kravtsev. Probabilistic Reversible Automata and Quantum Automata. *COCOON 2002, Lecture Notes in Computer Science, Vol. 2387, pp. 574-583*, 2002.
8. M. Golovkins, M. Kravtsev, V. Kravcevs. On a Class of Languages Recognizable by Probabilistic Reversible Decide-and-Halt Automata. *Theoretical Computer Science, Vol. 410(20), pp. 1942-1951*, 2009.
9. M. Golovkins, J.E. Pin. Varieties Generated by Certain Models of Reversible Finite Automata. *Chicago Journal of Theoretical Computer Science, Vol. 2010, Article 2*, 2010.
10. M. Hirvensalo. Quantum Automata with Open Time Evolution. *International Journal of Natural Computing Research, Vol. 1(1), pp. 70-85*, 2010.
11. A. Kondacs, J. Watrous. On The Power of Quantum Finite State Automata. *Proc. 38th FOCS, pp. 66-75*, 1997.
12. G. Kuperberg. The Capacity of Hybrid Quantum Memory. *IEEE Transactions on Information Theory, Vol. 49-6, pp. 1465 - 1473*, 2003.
13. L. Li, D. Qiu, X. Zou, L. Li, L. Wu, P. Mateus. Characterizations of One-Way General Quantum Finite Automata. `http://arxiv.org/abs/0911.3266`, 2010.
14. M. Mercer. Lower Bounds for Generalized Quantum Finite Automata. *LATA 2008, Lecture Notes in Computer Science, Vol. 5196, pp. 373-384*, 2008.
15. C. Moore, J.P. Crutchfield. Quantum Automata and Quantum Grammars. *Theoretical Computer Science, Vol. 237(1-2), pp. 275-306*, 2000.
16. M.A. Nielsen, I.L. Chuang. Quantum Computation and Quantum Information. *Cambridge University Press*, 2000.
17. J.E. Pin. Varieties of Formal Languages. *North Oxford, London and Plenum, New-York*, 1986.
18. J.E. Pin. BG = PG, a Success Story. *NATO Advanced Study Institute. Semigroups, Formal Languages and Groups, J. Fountain (ed.), pp. 33-47, Kluwer Academic Publishers*, 1995.
19. J.E. Pin, H. Straubing, D. Thérien. Small Varieties of Finite Semigroups and Extensions. *J. Austral. Math. Soc. (Series A), Vol. 37, pp. 269-281*, 1984.
20. M.O. Rabin. Probabilistic Automata. *Information and Control, Vol. 6(3), pp. 230-245*, 1963.
21. G. Thierrin. Permutation Automata. *Mathematical Systems Theory, Vol. 2(1), pp. 83-90*, 1968.
22. F. Zhang. Matrix Theory: Basic Results and Techniques. *Springer-Verlag*, 1999.