Unary probabilistic and quantum automata on promise problems

Aida Gainutdinova · Abuzer Yakaryılmaz

Received: 17 July 2017 / Accepted: 15 December 2017 / Published online: 26 December 2017
© Springer Science+Business Media, LLC, part of Springer Nature 2017

Abstract We continue the systematic investigation of probabilistic and quantum finite automata (PFAs and QFAs) on promise problems by focusing on unary languages. We show that bounded-error unary QFAs are more powerful than bounded-error unary PFAs, and, contrary to the binary language case, the computational power of Las-Vegas QFAs and bounded-error PFAs is equivalent to the computational power of deterministic finite automata (DFAs). Then, we present a new family of unary promise problems defined with two parameters such that when fixing one parameter QFAs can be exponentially more succinct than PFAs and when fixing the other parameter PFAs can be exponentially more succinct than DFAs.

Keywords Quantum finite automata · Unary promise problems · Bounded-error · Las-Vegas algorithms · Succinctness
1 Introduction

By a classic definition, a (computational) machine is said to recognize a language $L$, if it accepts all input strings over the given alphabet belonging to $L$. A string that is not accepted is said to be rejected. That is, by taking all members in $L$ and all non-members in $L$, we get set of all input strings over the given alphabet.

A promise problem is a generalization of the classic language recognition, similar to the generalization of a partial function over the total function: we are given two disjoint languages $L_{\text{yes}}$ and $L_{\text{no}}$, called yes- and no-instances, but their union, called the domain of a decision problem, is not necessarily equal to the set of all input strings. Thus, the machine must accept all inputs belonging to $L_{\text{yes}}$, reject all inputs belonging to $L_{\text{no}}$, but the behavior of the machine on inputs not belonging to the domain $L_{\text{yes}} \cup L_{\text{no}}$ can be quite arbitrary. (This reflects the fact that, in many applications, we do not have to distinguishing between acceptance and rejection for all input strings but, rather, only for inputs satisfying some a priori formatting requirements.)

Besides being generalization of decision problems, promise problems have also served important roles in the computational complexity. For example, it is not known whether the class $\text{BPP}$ ($\text{BQP}$), bounded-error probabilistic (quantum) polynomial time, has a complete problem, but, the class $\text{PromiseBPP}$ ($\text{PromiseBQP}$), defined on promise problems, has some complete problems (see the surveys by Goldreich [21] and Watrous [40]). We also refer the reader to [15] for a very recent and similar results on probabilistic logarithmic space.

In automata theory, the promise problems has appeared in many different forms. Similar to above results, some complete promise problems were given for non-uniform poly-size deterministic and non-deterministic automata models and so automata version of $\text{P}$ versus $\text{NP}$ problem was formulated by using such promise problems [25]. Another interesting result was given by Condon and Lipton [14]: (1) A promise emptiness problem given for probabilistic finite automata (PFAs) was shown to be undecidable by using a promise version of equality language ($\text{EQ} = \{a^n b^n | n > 0\}$). (2) Then, by using the fact that two-way bounded-error PFAs can recognize the equality language [16], a weak constant-space interactive proof system was constructed for any recursive enumerable language.

On the other hand, up to our knowledge, a systematic study (characterization and classification of the models and their related classes) of uniform automata models on promise problems have been started only recently.

An initial result was given for comparing zero-error quantum and deterministic pushdown automata [31], and it was shown that quantum model can solve a promise problem with probability 1, which cannot be solved by deterministic model. Then, Ambainis and Yakaryılmaz [9] showed that there is an infinite family of promise problems such that each promise problem from this family can be solved by a 2-state quantum finite automaton (QFA) with probability 1 but, for each natural number $n$, there is a promise problem in this family such that it cannot be solved by any $n$-state deterministic finite automaton (DFA). Thus, when solving promise problems,
the gap between the number of states in zero-error QFAs and DFAs can be arbitrarily large. Later, it was also shown that neither bounded-error probabilistic finite automata (PFAs) nor two-way alternating finite automata (2AFAs) can do better than DFAs on this family of promise problems [21,36], and, the result given in [9] was extended to more general families of promise problems [22]. We point out that, in the case of language recognition, zero-error QFAs cannot be more succinct than DFAs [27]. Moreover, the gap between bounded-error PFAs and bounded-error QFAs can be at most exponential [7,10].

After [9], a series of papers have appeared, presenting new results regarding the state complexity of classical and quantum automata on promise problems [1,2,13,18,20,23,43,45]. Parallel to the studies on state complexity, the computational power of classical and quantum automata has also been investigated.

In [20], a framework was given for probabilistic, deterministic, non-deterministic, and alternating automata on promise problems. Since the deterministic, non-deterministic, and alternating automata can simulate each other exactly, their computational power on promise problems remains the same. Similarly, the state complexity bounds given for language recognition cannot be violated when considering the promise problems. However, bounded-error one-way PFAs were shown to be not only more powerful but also more succinct (the gap cannot be bounded) than any other classical automata on promise problems. On the other hand, Las-Vegas PFAs were shown to have the same computational power as the classic finite state automata, accepting regular languages only. When considering the state complexity, Las-Vegas one-way PFAs were shown to be exponentially more succinct than DFAs on a family of promise problems, where the gap is tight.

In [35], QFAs were compared with PFAs and zero-error and Las-Vegas QFAs were shown to be more powerful than PFAs on promise problems:

- There is a promise problem solved by a zero-error two-way QFA (2QFA) but not by any sublogarithmic probabilistic Turing machine (PTM).
- There is a promise problem solved by a zero-error 2QFA in quadratic expected time, but not by any bounded-error $o(\log \log n)$-space PTMs in polynomial expected time.
- There is a promise problem solvable by a Las-Vegas one-way QFA, but not by any bounded-error one-way PFA.

We point out that it is still open whether QFAs are more powerful than PFAs in the above scenarios, when considering language recognition.

In [44], classical and quantum automata models were considered on a restricted type of promise problems where each machine defines a single promise problem.

We can summarize the relations among models in terms of their computational power as below:

\[
\text{DFA} \equiv 2\text{AFA} \equiv \text{Las-Vegas-2PFA} \subsetneq \text{bounded-error-PFA} \subsetneq \text{bounded-error-QFA} \\
\text{and} \\
\text{Las-Vegas-2PFA} \subsetneq \text{Las-Vegas-QFA} \subsetneq \text{zero-error-2QFA},
\]
where 2PFA stands for two-way PFA. Remark that the given separation results have been obtained for binary (and general) alphabets. Moreover, except for zero-error 2QFAs, all other models are equivalent to each other in the case of language recognition, since they all correspond to the class of regular languages. It is still open whether zero-error or Las-Vegas 2QFAs can recognize a non-regular language.

In this paper, we provide new results for probabilistic and quantum automata, focusing on unary promise problems. We show that bounded-error unary QFAs are more powerful than bounded-error unary PFAs. However, contrary to the binary case, the computational power of both Las-Vegas QFAs and bounded-error PFAs is the same, equal to that of DFAs. Finally, we present a new family of unary promise problems with two parameters such that, when fixing the first parameter, QFAs become exponentially more succinct than PFAs and, when fixing the second parameter, PFAs become exponentially more succinct than DFAs.

In the next section, we fix the basics for the rest of the paper. Next, in Sect. 3, we investigate the computational power of unary PFAs and QFAs. Finally, in Sect. 4, we present our gap result.

A preliminary version of this paper was presented in DLT2015 and appeared in its proceedings book [17]. This is an extended and considerable revised version. We include the omitted proofs of Theorems 7, 9, and 10. Moreover, we revised all proofs and make them shorter and more elegant. Theorem 1 is added by also considering general alphabets (the previous result was only about single letter alphabets), which also allows us to shorten the proof for the result regarding Las-Vegas unary QFAs. For the sake of completeness, we give also the definition of general QFAs.

2 Preliminaries

We start with the definitions of models and the notion of promise problems. Then, we give the basics of Markov chains, required later, in some proofs.

2.1 Definitions

An $n$-state PFA $P$ is a 5-tuple $P = (Q, \Sigma, \{A_\sigma \mid \sigma \in \Sigma\}, v_0, Q_a)$, where

- $Q = \{q_1, \ldots, q_n\}$ is a set of (classical) states,
- $\Sigma$ is an input alphabet,
- $v_0$ is a $|Q|$-dimensional stochastic initial column vector representing the initial probability distribution of the states at the beginning of the computation,
- $A_\sigma$ is a (left) stochastic transition matrix for symbol $\sigma \in \Sigma$, where $A_\sigma(j, i)$ represents the probability of going from the $i$th state to the $j$th state after reading $\sigma$, and
- $Q_a \subseteq Q$ is a set of the accepting states.

The computation of $P$ on the input $w \in \Sigma^*$ can be traced by a stochastic column vector, i.e.,

$$v_j = A_w v_{j-1},$$
where $1 \leq j \leq |w|$. After reading the whole input, the final probabilistic state is $v_f = v_{|w|}$. Based on this, we can calculate the accepting probability of $w$ by $P$, denoted $f_P(w)$, as follows:

$$f_P(w) = \sum_{q_j \in Q_a} v_f[j].$$

If all stochastic components of a PFA are restricted to have only 0s and 1s, then we obtain a DFA that starts in a certain state and switches to only one state in each step, and so the computation ends in a single state. An input is accepted by a DFA if the final state is an accepting state. Otherwise, the input is rejected.

There are different kinds of quantum finite automata (QFAs) models in the literature [10]. The general ones (e.g., [4, 24, 42]) can exactly simulate PFAs (see [38] for a pedagogical proof), but the restricted ones cannot recognize some languages recognized by PFAs (e.g., [6, 10, 12, 28]). In this paper, we present our quantum results based on the known simplest QFA model, called Moore-Crutchfield QFA (MCQFA) [30] and general QFA models (from now on we refer simply QFA).

We assume the reader knows the basics of quantum computation (see [10, 34, 38] for details).

An $n$-state MCQFA $M$ is a 5-tuple $M = (Q, \Sigma, \{U_\sigma \mid \sigma \in \Sigma\}, |v_0\rangle, Q_a)$ where, different from a PFA,

- $|v_0\rangle$ is a norm-1 complex-valued column vector that can be in a superposition of basis states (see below) and represents the initial quantum state of $M$ at the beginning of a computation, and,
- $U_\sigma$ is a unitary transition matrix for symbol $\sigma \in \Sigma$, where $U_\sigma(j, i)$ represents the amplitude of going from the $i$-th state to the $j$-th state after reading $\sigma$.

Traditionally, column vectors are represented with “ket” notation ($|\cdot\rangle$) in quantum mechanics and computations. For a given column vector $|v\rangle$, its conjugate transpose is denoted as $\langle v |$. For each $q_j \in Q$, $|q_j\rangle$ is a basis state, a zero vector except its $j$-th entry that is 1. The computation of $M$ on an input $w \in \Sigma^*$ can be traced by a norm-1 complex-valued column vector, i.e.,

$$|v_j\rangle = U_{w_j} |v_{j-1}\rangle,$$

where $1 \leq j \leq |w|$. After reading the whole input, the final quantum state is $|v_f\rangle = |v_{|w|}\rangle$. Based on this, a measurement operator is applied to see whether the automaton is in an accepting or non-accepting state. The accepting probability of $w$ by $M$ is calculated as:

$$f_M(w) = \sum_{q_j \in Q_a} |\langle q_j | v_f \rangle|^2,$$

where $\langle q_j | v_f \rangle$ returns the $j$-th entry of $|v_f\rangle$.

During the computation, a MCQFA is always in a single quantum state, which is also called as a pure state. A (general) QFA, on the other hand, can be in more than one pure state with some probabilities:
\[
\left\{ (p_j, |v_j\rangle) \mid 1 \leq j \leq k, p_j \geq 0, \text{ and } \sum_{j=1}^{k} p_j = 1 \right\},
\]
where the QFA is in pure state \(|v_j\rangle\) with probability \(p_j\). Such a mixture is called as mixed state, and it can be represented as a single mathematical object, called density matrix:

\[
\rho = \sum_{j=1}^{k} p_j |v_j\rangle \langle v_j|.
\]

Remark that \((j, j)\)-th entry of \(\rho\) gives the probability of being in the \(j\)-th state. Therefore, \(Tr(\rho) = 1\).

A QFA can apply more general quantum operators, called as superoperators. A superoperator \(E\) is composed by a finite number of operational elements:

\[
E = \left\{ E_1, \ldots, E_l \mid \sum_{j=1}^{l} E_j^\dagger E_j = I \right\},
\]

where \(E_j^\dagger\) is the conjugate transpose of \(E_j\) and I is the identity matrix. After applying a superoperator \(E\) to the mixed state \(\rho\), we obtain a new mixed state:

\[
\rho' = E(\rho) = \sum_{j=1}^{l} E_j \rho E_j^\dagger.
\]

An \(n\)-state QFA \(M\) \([24, 42]\) is 5-tuple \(M = (Q, \Sigma, \{E_\sigma \mid \sigma \in \Sigma\}, \rho_0, Q_a)\) where, different from a MCQFA,

- \(\rho_0\) is an initial mixed state of \(M\) at the beginning of a computation, and,
- \(E_\sigma\) is a superoperator for symbol \(\sigma \in \Sigma\) composed by \(l_\sigma > 0\) operational elements, \(E_\sigma = \{E_{\sigma, j} \mid 1 \leq j \leq l_\sigma\}\).

The computation of \(M\) on an input \(w \in \Sigma^*\) can be traced by mixed states (density matrices), i.e.,

\[
\rho_j = E_{w_j}(\rho_{j-1}),
\]

where \(1 \leq j \leq |w|\). After reading the whole input, the final quantum state is \(\rho_f = \rho_{|w|}\). Based on this, a measurement operator is applied to see whether the automaton is in an accepting or non-accepting state. The accepting probability of \(w\) by \(M\) is calculated as:

\[
f_M(w) = \sum_{q_f \in Q_a} \rho_f(j, j).
\]
A QFA can also be defined by reading an extra symbol, called the right end-marker ($\$), after reading the whole input in order to make some post-processing. In such a case, the final mixed state is calculated as $\rho_f = \mathcal{E}_\$ (\rho_{|w|})$ where $\mathcal{E}_\$ is the superoperator applied after reading $\$.

A non-deterministic QFA (NQFA) is a QFA that is restricted to a special acceptance mode: It separates the strings having nonzero accepting probability from the strings having zero accepting probability.

A Las-Vegas PFA or QFA never makes a wrong decision but can make the decision of “don’t know.” Formally, its set of states is divided into three disjoint sets, the set of accepting states ($Q_a$), the set of neutral states ($Q_n$), and the set of rejecting states ($Q_r = Q \setminus (Q_a \cup Q_n)$). At the end of the computation, the decision of “don’t know” is given if the automaton ends in a neutral state. The probability of giving the decision of “don’t know” (rejection) is calculated in the same way of the accepting probability by using $Q_n$ ($Q_r$) instead of $Q_a$.

A promise problem $P \subseteq \Sigma^*$ is composed by two disjoint languages $P_{yes}$ and $P_{no}$, where the former one is called a set of yes-instances and the latter one is called a set of no-instances.

A promise problem is said to be solved by a DFA if any yes-instance is accepted and any no-instance is rejected. A promise problem is said to be solved by a PFA or QFA with error bound $\epsilon < \frac{1}{2}$ if any yes-instance is accepted with probability at least $1 - \epsilon$ and any no-instance is rejected with probability at least $1 - \epsilon$. If all yes-instances are accepted exactly, then it is said the promise problem is solved with one-sided bounded error. In this case, the error bound can be greater than $\frac{1}{2}$ but it must be less than 1, i.e., $\epsilon < 1$.

A promise problem is solved by a NQFA if any yes-instance is accepted with nonzero probability and any no-instance is accepted with zero probability.

Lastly, a promise problem is said to be solved by a Las-Vegas PFA or QFA with success probability $p > 0$, if any yes-instance is accepted with probability at least $p$ and it is rejected with probability 0, and, if any no-instance is rejected with probability at least $p$ and it is accepted with probability 0.

In the case of promise problems, we do not care about the decisions on the strings from $\Sigma^* \setminus P$.

If the set of yes-instances is the complement of the set of no-instances, then we use the term of language recognition and it is said that the language composed by yes-instances is recognized by the model.

### 2.2 The theory of Markov chains

The computation of a unary PFA can be modeled as a Markov chain. Here we present some basic facts and results from the theory of Markov chains that will be used in some proofs. We refer the reader to [26] for more details and [7] and [29] for similar applications.
Let $\mathcal{P}$ be an $n$-state unary PFA defined on alphabet $\{a\}$ with transition matrix $A$. Then $\mathcal{P}$ defines a Markov chain as $\{A^i | i \geq 1\}$. In its long-term behavior, $A^i$ converges to either a single matrix $A_0$ or a set of matrices $\{A_0, \ldots, A_{D-1}\}$ for some $D$ depending on the entries of $A$ (explained later). More explicitly, in each case, given any arbitrary small $\delta > 0$, there exists an index $r$ satisfying the following properties.

- Case 1: $||A^{r+i} - A_0||_1 < \delta$ for all $i \geq 0$.
- Case 2: For $j \in \{0, \ldots, D-1\}$, $||A^{r+j+i\cdot D} - A_j||_1 < \delta$ for all $i \geq 0$.

We provide more details about the Case 2. The number $D > 1$ is called the period of the Markov chain. In the long-term behavior, the probabilities of being in some states may converge to zeros. Any such state is called transient since once the automaton leaves this state it never comes back again. The rest of states form either a single cycle ($t = 1$) or $t > 1$ disjoint cycles. In each (or the single) cycle, the computation goes from one state to another state in an order. Let $n_j$ be the number of states in the $j$-th cycle ($1 \leq j \leq t$). Then, $D = \text{lcm}(n_1, \ldots, n_t)$. Remark that $n_1 + \cdots + n_t \leq n$.

Based on the above facts, we can obtain the followings results regarding the accepting probabilities of $\mathcal{M}$ on the long inputs.

In Case 1, there is a single limiting accepting probability $\text{acc}_0$. Given any arbitrary small $\delta > 0$, there exists an index $r$ satisfying

$$|fP(a^{r+i}) - \text{acc}_0| < \delta$$

for any $i \geq 0$.

In Case 2, there are $D$ limiting accepting probabilities $\text{acc}_0, \ldots, \text{acc}_{D-1}$. Given any arbitrary small $\delta > 0$, there exists an index $r$ satisfying that for each $j \in \{0, \ldots, D-1\}$

$$|fP(a^{r+j+i\cdot D}) - \text{acc}_j| < \delta$$

for any $i \geq 0$.

### 3 The computational power of unary PFAs and QFAs

We start with Las-Vegas QFAs by also considering general alphabets.

**Theorem 1** If a promise problem $\mathcal{P} = (\mathcal{P}_{\text{yes}}, \mathcal{P}_{\text{no}})$ is solvable by a Las-Vegas QFA $\mathcal{M}_1$ with success probability $p$, then it is also solvable by an NQFA $\mathcal{M}'$.

**Proof** By converting every neutral state of $\mathcal{M}$ into a non-accepting state, we obtain $\mathcal{M}'$. Then, any member of $\mathcal{P}_{\text{yes}}$ can be accepted by $\mathcal{M}'$ with nonzero probability and any member of $\mathcal{P}_{\text{no}}$ is accepted with zero probability. \hfill $\square$

As a simple consequence of this theorem is that if $L$ is the language recognized by the NQFA $\mathcal{M}'$, then $\mathcal{P}_{\text{yes}} \subseteq L$ and $\mathcal{P}_{\text{no}} \subseteq \overline{L}$. By going one step further, we can also obtain the following result.

**Theorem 2** Any unary promise problem $\mathcal{P} = (\mathcal{P}_{\text{yes}}, \mathcal{P}_{\text{no}})$ solvable by a Las-Vegas QFA is also solvable by a DFA.

@Springer
Proof Due to Theorem 1, for the promise problem $P$, there exists a unary language $L$ recognized by a NQFA such that $P_{\text{yes}} \subseteq L$ and $P_{\text{no}} \subseteq \overline{L}$. We also know that unary NQFAs can recognize only regular languages (Page 89 of [37] and [41]) and so $L$ is regular. Thus, there exists a DFA recognizing $L$. Therefore, $P$ is also solvable by this DFA.

Since Las-Vegas QFAs and DFAs define the same class of unary promise problems, one may ask how much state-efficient QFAs can be over DFAs. Due to the result of Ambainis and Yakaryılmaz [9], we know that the gap (on unary promise problems) cannot be bounded. Remark that, in the case of language recognition, exact QFAs and DFAs have the same state complexity on every regular language [27], and bounded-error QFAs can be at most exponentially more state efficient than bounded-error PFAs (e.g., [3, 7, 10]). On the other hand, as mentioned before, over binary promise problems, Las-Vegas QFAs are known to be more powerful than bounded-error PFAs [35]. An open question here is whether exact QFAs can solve a binary promise problem that is beyond the capabilities of DFAs.

Theorem 3 If a unary promise problem $P = (P_{\text{yes}}, P_{\text{no}})$ is solved by a PFA, say $\mathcal{P}$, with error bound $\epsilon < \frac{1}{2}$, then it is also solvable by a DFA.

Proof If $P_{\text{yes}}$ or $P_{\text{no}}$ is a finite set, then $P$ is solvable by a DFA: Any finite set is regular and so it is recognizable by a DFA which separates this finite set from any other disjoint set. So we assume that both $P_{\text{yes}}$ and $P_{\text{no}}$ contain infinitely many words in the rest of the proof.

Let $\mathcal{P}$ has $n$ states and $D$ be a period of the corresponding Markov chain. So, $\mathcal{P}$ has $D$ limiting accepting probabilities, say $\text{acc}_0, \ldots, \text{acc}_{D-1}$ (if $\mathcal{P}$ has a single limiting accepting probability, then $D = 1$). We know that given any small $\delta > 0$, there is an index $r$ such that, for each $j \in \{0, \ldots, D - 1\}$, we have the inequality $|f_\mathcal{P}(a^{r+i\cdot D+j}) - \text{acc}_j| < \delta$ for all $i \geq 0$.

We pick a $\delta' > 0$ such that, for each $j \in \{0, \ldots, D - 1\}$, the interval $[\alpha_{\text{acc}(j)} - \delta', \alpha_{\text{acc}(j)} + \delta']$ contains at most one of the points in $\{1 - \epsilon, \epsilon\}$. This is always possible since the gap between these two points $(1 - 2\epsilon)$ is nonzero. For this $\delta'$, we have an index $r'$ such that, for any $j \in \{0, \ldots, D - 1\}$, $f_\mathcal{P}(a^{r'+i\cdot D+j})$ is in the interval $[\text{acc}_j - \delta', \text{acc}_j + \delta']$ for all $i \geq 0$.

Then, we can classify $\text{acc}_j$ as follows:

**accept-type:** The value of $\text{acc}_j$ is at least $\frac{1}{2}$. Then, $f_\mathcal{P}(a^{r'+i\cdot D+j})$ cannot be $\epsilon$ or less than $\epsilon$ for any $i \geq 0$.

**reject-type:** The value of $\text{acc}_j$ is less than $\frac{1}{2}$. Then, $f_\mathcal{M}(a^{r'+i\cdot D+j})$ cannot be $1 - \epsilon$ or greater than $1 - \epsilon$ for any $i \geq 0$.

It is clear that a $D$-state cyclic DFA with the following state transitions

$q_0 \rightarrow q_1 \rightarrow \cdots \rightarrow q_j \rightarrow \cdots \rightarrow q_{D-1} \rightarrow q_0 \rightarrow \cdots$
can easily follow the periodicity of \( P \). Moreover, if \( \text{acc}_j \) is accept-type (resp., reject-type), then \( q_j \) is an accepting (resp., a non-accepting) state. Thus, our cyclic DFA can make the same decisions as \( P \) on the promised strings with length at least \( r' \). The remaining (and shorter) promised strings form a finite set and a DFA with \( r' \) states can make appropriate decisions on them. Therefore, by combining these two DFAs, we can obtain a DFA with \( r' + D \) states that solves the promise problem \( P \).

\[
\square
\]

In the above proof, the values of \( r' \) and \( D \) depend on the entries of transition matrix and may not depend on the number of states \( (n) \). A simple evidence was given in \([19,20]\) that there exists an infinite family of unary promise problems solvable by 2-state PFAs, but we cannot put an upper bound on the number of states for the DFAs solving this family.

Now we show that two-sided bounded-error unary QFAs can define more promise problems than two-sided bounded-error PFAs. We start with defining a 2-state MCQFA \( M \) on the alphabet \( \{a\} \) (see \([39]\) for a similar construction): It has two states \( \{q_1, q_2\} \), \( q_1 \) is the accepting state, and \( |q_1\rangle \) is the initial quantum state. Moreover, the single unitary operator is defined on \( \mathbb{R}^2 \) that is a counter clockwise rotation with angle \( \alpha = \sqrt{2}\pi \) on the unit circle (on \( |q_1\rangle \)-\( |q_2\rangle \) plane). (See also \([8]\) for algorithmic applications of such rotations.)

Since \( \alpha \) is an irrational multiple of \( \pi \), all quantum states that are accessible from the initial one are dense on the unit circle. (In fact, the angle \( \alpha \) can be any irrational multiple of \( \pi \).) Moreover, the accepting probability of \( a^j \) is \( f_M(a^j) = \cos^2(j\alpha) \).

Let \( \theta \) be the angle satisfying \( 0 < \sin^2 \theta = \epsilon_\theta < \frac{1}{2} \). Then, \( M \) and \( \epsilon_\theta \) (or \( \alpha \) and \( \theta \)) define a promise problem \( P^\theta \) as follows:

\[
- P^\theta_{\text{yes}} = \{a^j | f_M(a^j) \geq 1 - \epsilon_\theta \} \text{ and } P^\theta_{\text{no}} = \{a^j | f_M(a^j) \leq \epsilon_\theta \}.
\]

Remark that there are infinitely many strings in \( P^\theta_{\text{yes}} \) and in \( P^\theta_{\text{no}} \).

**Theorem 4** The promise problem \( P^\theta \) is solvable by \( M \) with error bound \( \epsilon_\theta \).

On the other hand, the promise problem \( P^\theta \) is not solvable by any bounded-error PFA. Since unary bounded-error PFAs and unary DFAs have the same computational power on promise problems (Theorem 3), we give our proof for DFAs.

**Theorem 5** The promise problem \( P^\theta \) is not solvable by any DFA.

**Proof** Suppose that there is an \( n \)-state DFA \( D \) solving \( P^\theta \). Let \( a^i \) be a string in \( P^\theta_{\text{no}} \), where \( j > n \), and \( q_j \) be the state of \( D \) after reading \( a^j \). It is clear that \( q_j \) is not an accepting state. Since \( D \) is a unary DFA, it has a period \( D \leq n \) such that, after the first \( n \) steps, it always visits the same state in every \( D \) steps. Thus, none of the members of the following set is accepted by \( D \):

\[
S = \{a^{i \cdot D + j} | i \geq 0 \}.
\]

If we show that \( S \) contains a member of \( P^\theta_{\text{yes}} \), then we complete our proof since \( D \) rejects a member of \( P^\theta_{\text{yes}} \), which is a contradiction.
Remember that the members of $P^\theta$ are defined based on $M$ and $\theta$. In order to show that $S \cap P^\theta_{\text{yes}} \neq \emptyset$, we focus on the computation of $M$ on the members of $S$. Let $|v_i.D+j)$ be the state of $M$ after reading $a^{i-D+j}$ for $i \geq 0$. It is clear that $|v_{(i+1)}D+j)$ is obtained from $|v_iD+j)$ by a counter clockwise rotation of angle $\alpha' = D\alpha = D \sqrt{2}\pi$. Since $\alpha'$ is an irrational multiple of $\pi$, it is dense on the unit circle. Thus, the quantum states of some members of $S$ can be arbitrarily close to $|q_1\rangle$; for any arbitrary small $\delta > 0$, there exists an index $i'$ such that the distance between $|q_1\rangle$ and $|v_{i'\cdot D+j}\rangle$ is less than $\delta$ where the norm can be $l_1$ or $l_2$.

Therefore, for any $\theta$, we can always find a member of $S$ that is also a member of $P^\theta_{\text{yes}}$. 

4 Succinctness

For each $m$, $k \in \mathbb{Z}^+$, let $P_{m.k} = \{p_m, \ldots, p_{m+k-1}\}$ be the set of prime numbers from the $m$-th one to $(m+k-1)$-th one, and let $N_{m,k} = \prod_{i=0}^{k-1} p_{m+i}$ be the multiplication of these prime numbers.

We define a new family of unary promise problems $\text{PRIMES}_m = \{P_{m,k} \subseteq \{a\}^* | k \in \mathbb{Z}^+\}$, where

- $P_{m,k}^{\text{yes}} = \{a^i | i \equiv 0 \mod N_{m,k}\}$ and
- $P_{m,k}^{\text{no}} = \{a^i | i \mod p_j \in \left[p_j^\frac{1}{8}, \frac{3p_j}{8}\right] \cup \left[p_j^\frac{5}{8}, \frac{7p_j}{8}\right] \text{ for at least } 2k \frac{1}{3} \text{ different } p_j \text{ from the set } P_{m,k}\}$.

It is trivial that $P_{m,k}^{\text{yes}}$ contains infinitely many strings. By using Chinese Remainder Theorem, we can show that $P_{m,k}^{\text{no}}$ also contains infinitely many strings.

Lemma 1 There are infinitely many strings in $P_{m,k}^{\text{no}}$.

Proof If positive integers $i_1, i_2, \ldots, i_m$ are pairwise coprimes, then for any integers $r_1, r_2, \ldots, r_m$ satisfying $0 \leq r_j < i_j$ ($j \in \{1, 2, \ldots, m\}$), there exists a number $K$, such that $K = r_i \mod i_j$ for each $j \in \{1, 2, \ldots, m\}$. Moreover, any such $K$ is congruent modulo the product, $M = i_1 \cdot \ldots \cdot i_m$. That is all numbers of the form $K + l \cdot M$ will satisfy this condition for any $l \in \mathbb{Z}^+$.

Theorem 6 For any $m, k \in \mathbb{Z}^+$, the promise problem $P_{m,k}$ can be solvable by a $2k$-state MCQFA, say $M_{m,k}$ (shortly $M$), such that any yes-instance is accepted with probability 1 and any no-instance is rejected with probability at least $\frac{1}{3}$.

Proof We use the well-known technique given in [5,7]. The set of states of $M$ is $\{q^0_m, q^1_m, \ldots, q^0_{m+k-1}, q^1_{m+k-1}\}$ and the ones with superscript “0” are the accepting states. The initial quantum state is $|v_0\rangle = \frac{1}{\sqrt{k}} |q^0_m\rangle + \frac{1}{\sqrt{k}} |q^0_{m+1}\rangle + \cdots + \frac{1}{\sqrt{k}} |q^0_{m+k-1}\rangle$.

During reading the input, the states $|q^0_j\rangle$ and $|q^1_j\rangle$ form a small MCQFA isolated from the others, where $m \leq j \leq m+k-1$. For each letter $a$, a rotation with the angle $\frac{2\pi}{p_j}$
is applied on \( \{|q_j^0\}, |q_j^1\rangle \):
\[
U_j = \begin{pmatrix}
\cos(2\pi/p_j) & \sin(2\pi/p_j) \\
-\sin(2\pi/p_j) & \cos(2\pi/p_j)
\end{pmatrix},
\]
Then, the overall transition matrix is
\[
U = \begin{pmatrix}
U_m & 0 & \cdots & 0 \\
0 & U_{m+1} & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & U_{m+k-1}
\end{pmatrix},
\]
where 0 denotes \( 2 \times 2 \) zero matrix.
For any input \( a^i \) the final state of \( M \) is
\[
|v_i\rangle = \frac{1}{\sqrt{k}} \sum_{j=m}^{m+k-1} \left( \cos \left( i \frac{2\pi}{p_j} \right) |q_j^0\rangle + \sin \left( i \frac{2\pi}{p_j} \right) |q_j^1\rangle \right).
\]
For any yes-instance, \( i \) is multiple of \( N_{m,k} \) and so each \( \frac{2\pi}{p_j} \) will be a multiple of \( 2\pi \). Then, the final state is in a superposition of only the accepting states, i.e.,
\[
|v_i\rangle = \frac{1}{\sqrt{k}} \sum_{j=m}^{m+k-1} |q_j^0\rangle,
\]
and so the input is accepted with probability 1.
For any no-instance, on the other hand, \( \left( i \mod p_j \right) \in \left[ \frac{p_j}{8}, \frac{3p_j}{8} \right] \cup \left[ \frac{5p_j}{8}, \frac{7p_j}{8} \right] \) for at least \( \frac{2k}{3} \) different \( p_j \)'s from the set \( P_{m,k} \). If \( p_j \) is one of them, then its contribution to the overall rejecting probability is given by
\[
\frac{1}{k} \sin^2 \left( i \frac{2\pi}{p_j} \right),
\]
which takes its minimum value \( \frac{1}{2k} \) when \( \left( i \mod p_j \right) \) is equal to one of the border. Since there are at least \( \frac{2k}{3} \) of them, the overall rejecting probability is at least \( \frac{1}{3} \).

Before proving the next theorem, we present two auxiliary lemmas, based on some facts from group theory (for more details see, for example, [36]).

**Lemma 2** Let \( D \) and \( N \) be two positive integers, \( \langle ND \rangle = N \ (\text{mod} \ D) \), \( G_D = \{0, 1, \ldots, D - 1\} \) be an additive cyclic group with generator element 1, and \( G_{ND} = \langle ND \rangle \) be a cyclic subgroup of the group \( G_D \). Then one of the following statements holds:

I. \( G_{ND} = G_D \) or

\(\mathcal{S}\) Springer
2. there exists a positive integer $d$ ($1 < d \leq D$) such that both $N$ and $D$ are multiple of $d$.

**Proof** It is known that if $\gcd(D, N_D) = 1$, then the element $N_D$ generates the whole group $G_D$. So we have $G_{N_D} = G_D$ in this case.

Suppose that $\gcd(D, N_D) > 1$. Let the order of $G_{N_D}$ (the number of elements) be $D'$. Since $G_{N_D}$ is a subgroup of $G_D$, $D'$ divides $D$ and $D/D'$, say $d$, is greater than 1. Since $0 \in G_{N_D}, N \cdot D' = e \cdot D$ for some $e \in \mathbb{Z}$. Then, we can follow $D' \cdot N = e \cdot d \cdot D'$ and so $N = e \cdot d$, i.e., $N$ is a multiple of $d$. \hfill \Box

Let $N = N_{m,k}$ and $d = \gcd(N, D)$ for a given $D$ (specified later). We define two sets $S$ and $M$. The set $S$ contains all multiples of $d$:

$$S = \{d \cdot j \mid j \in \mathbb{Z}\}.$$ 

The set $M$ contains the numbers corresponding to the no-instances ($\mathbb{P}^{m,k}_{no}$):

$$M = \{i \mid i \in \mathbb{Z}, i \mod p_j \in \left[\frac{p_j}{8}, \frac{3p_j}{8}\right] \cup \left[\frac{5p_j}{8}, \frac{7p_j}{8}\right]
\text{ for at least } \frac{2k}{3} \text{ different } p_j \text{ from the set } P_{m,k}\}.$$ 

**Lemma 3** If $D < T = p_m \cdot p_{m+1} \cdots p_{m+\left\lfloor \frac{k}{2} \right\rfloor}$, then $S \cap M \neq \emptyset$.

**Proof** Let $R \subseteq P_{m,k}$ be the set containing the primes in $P_{m,k}$ that do not divide $d$:

$$R = \{p \mid p \in P_{m,k}, d \equiv 0 \pmod{p}\}.$$ 

Since $d \leq D < T = p_m \cdots p_{m+\left\lfloor \frac{k}{2} \right\rfloor}$, $d$ can be divisible by at most $\frac{k}{3}$ primes from the set $P_{m,k}$. So there are at least $\frac{2k}{3}$ primes from $P_{m,k}$ that do not divide $d$. Thus, the size of $R$ is at least $\frac{2k}{3}$: $|R| \geq \frac{2k}{3}$.

Now we define a subset of $M$, say $O \subseteq M$, as follows:

$$O = \left\{l + N \cdot i \mid i \in \mathbb{Z}, l \mod p_j \in \left[\frac{p_j}{8}, \frac{3p_j}{8}\right] \cup \left[\frac{5p_j}{8}, \frac{7p_j}{8}\right]\text{ for } p_j \in R
\text{ and } l \equiv 0 \pmod{p_j}\right\}$$

for $p_j \in P_{m,k} \setminus R$.

The existence of such $l$’s follows from the Chinese Remainder Theorem. By definition, the multiplication of all prime numbers in $(P_{m,k} \setminus R)$ divides both $l$ and $d$. Since $d = \gcd(D, N)$ and $N = N_{m,k}$, then we conclude that $d$ is a product of prime numbers from the set $(P_{m,k} \setminus R)$. Since the numbers $N$ and $l$ are both multiples of $d$, the set $O$ is also subset of $S$ and so $S \cap M \neq \emptyset$. \hfill \Box

**Theorem 7** For any $m \in \mathbb{Z}^+$, any DFA solving the promise problem $\mathbb{P}^{m,k}$ needs $\Omega(m \log m)^{\frac{k}{3}}$ states.
Proof Let \( T = p_m \cdot p_{m+1} \cdots p_{m+\left\lceil \frac{k}{3} \right\rceil} \). Assume that a DFA, say \( D \), solves the promise problem \( P_{m,k} \) with less than \( T \) states. Since \( P_{m,k}^{yes} \) and \( P_{m,k}^{no} \) contain infinitely many strings, there must exist a cycle of \( D \) states \((D < T)\) \( s_0, \ldots, s_{D-1} \) such that \( D \) visits these states in order

\[
s_0 \rightarrow s_1 \rightarrow \cdots \rightarrow s_{D-1} \rightarrow s_0 \rightarrow \cdots
\]

Without loss of generality, we suppose that \( D \) enters the state \( s_0 \) after reading the yes-instance \( a^l N_{m,k} \) for some \( l > 0 \).

We associate states \( s_0, s_1, \ldots, s_{D-1} \) with elements of additive cyclic group \( G_D = \{0, 1, \ldots, D \} \). Let \( N_D = N_{m,k} \mod D \), \( N_D \in G_D \). We consider a cyclic subgroup \( G_{N_D} = \langle N_D \rangle \) of the group \( G_D \). All elements of \( G_{N_D} \) must be accepting states.

Let \( d = \gcd(D, N_D) \). By Lemma 2, if \( d = 1 \), then \( G_D = G_{N_D} \). Since some states must be rejecting states, we conclude that \( d > 1 \). From this we have that all states \( \{s_{i,d} \mid i = 0, \ldots, \frac{D}{d} - 1\} \) are accepting states and the automaton accepts all inputs from the set \( A = \{a^{id} \mid i \in \mathbb{Z}\} \) after entering the cycle. From Lemma 3, it follows that the set \( A \) contains some no-instances. This is a contradiction. Therefore, length \( D \) of the cycle (and so the number of states required by \( D \)) cannot be less than \( T \).

The value of \( T \), which is equal to \( p_m \cdots p_{m+\left\lceil k/3 \right\rceil} \), can be calculated as follow. It is known [11] that the \( m \)-th prime number \( p_m \) satisfies \( p_m = \Theta(m \log(m)) \). Then,

\[
T = \prod_{i=m}^{m+\left\lceil k/3 \right\rceil} p_i \geq c \prod_{i=m}^{m+k/3-1} i \log(i) \geq c'(m \log(m))^{k/3} = \Omega(m \log m)^{k/3}.
\]

\( \square \)

**Theorem 8** For any \( m \in \mathbb{Z}^+ \), there is a \( O((m + \frac{k}{3}) \log(m + \frac{k}{3}))^{\frac{k}{3}} \)-state DFA \( D_{m,k} \) solving the promise problem \( P_{m,k} \).

Proof Let \( T = p_m \cdot p_{m+1} \cdots p_{m+\left\lceil \frac{k}{3} \right\rceil} \). The DFA \( D_{m,k} \), solving the promise problem \( P_{m,k} \), has \( T \) states \( q_0, \ldots, q_{T-1} \), where \( q_0 \) is an initial and the only accepting state. For each input symbol, \( D_{m,k} \) moves from the state \( q_i \) to \( q_{i+1} \mod T \). It is clear that the input \( a^i \) is accepted if and only if \( i \) is multiple of \( T \). Thus, any member of \( P_{m,k}^{yes} \) is accepted. If \( a^i \in P_{m,k}^{no} \), then there are at least \( \frac{2k}{3} \) prime numbers from \( P_{m,k} \) such that \( i \) is not multiple of any of them. Thus, such \( i \) cannot be multiple of \( T \) since \( T \) is multiple of \( \left\lceil \frac{k}{3} \right\rceil + 1 \) prime numbers from \( P_{m,k} \).

As mentioned in the previous proof, the \( m \)-th prime number \( p_m \) satisfies \( p_m = \Theta(m \log m) \) and so

\[
T = \prod_{i=m}^{m+\left\lceil k/3 \right\rceil} p_i \leq c \prod_{i=m}^{m+k/3} i \log(i) \leq c'((m + k/3) \log(m + k/3))^{k/3} = O((m + k/3) \log(m + k/3))^{k/3}.
\]

\( \square \)
Now, we give a lower and an upper bound for PFAs.

**Theorem 9** Any bounded-error PFA solving the promise problem $P_{m,k}$ needs $\Omega(k(m + k) \log m)$ states.

*Proof* Let $\mathcal{P}$ be a PFA solving $P_{m,k}$ with bounded error. Both yes and no instances are infinitely many and so for sufficiently long inputs, $\mathcal{P}$ behaves like a DFA as shown in the proof of Theorem 3. Moreover, the period of $\mathcal{P}$, say $D$, is at least $T = p_m p_{m+1} \cdots p_{m+\lfloor \frac{k}{3} \rfloor}$ due to Theorem 7. Suppose that $\mathcal{P}$ has $t$ cycles with $n_1, \ldots, n_t$ number of states. Thus, $D = \text{lcm}(n_1, \ldots, n_t)$. It is clear that for the minimal values, the number of states are simply $p_m, p_{m+1}, \ldots, p_{m+\lfloor \frac{k}{3} \rfloor}$. Therefore, $\mathcal{P}$ cannot have less than $(p_m + p_{m+1} + \cdots + p_{m+\lfloor \frac{k}{3} \rfloor})$ states. Thus, we can follow the lower bound as

$$
\sum_{i=m}^{m+\lfloor \frac{k}{3} \rfloor} p_i \geq c \int_{m-1}^{m+\frac{k}{3}-1} x \log x \, dx \geq c' \left(2m - 2 + \frac{k}{3}\right) \log \left(\frac{m-1}{\sqrt{2}}\right) = \Omega(k(m + k) \log m).
$$

$\square$

**Theorem 10** For any $m \in \mathbb{Z}^+$, there is a $O(k(m + k) \log(m + k))$-state PFA $\mathcal{P}_{m,k}$ solving the promise problem $P_{m,k}$ with one-sided error bound $\frac{1}{3}$.

*Proof* Let $\mathcal{P}_{m,k}$, shortly $\mathcal{P}$, be $(Q, \{a\}, \{A\}, v_0, Q_a)$, where

- $Q = \{q_{i,j} \mid i = 1, \ldots, k, j = 0, \ldots, p_{m+i-1} - 1\}$ and $p_m, \ldots, p_{m+k-1}$ are the primes from the set $P_{m,k}$,
- $v_0$ is the initial probabilistic state such that the automaton is in the state $q_{i,0}$ with the probability $\frac{1}{k}$ for each $i = 1, \ldots, k$, and,
- $Q_a = \{q_{i,0} \mid i = 1, \ldots, k\}$.

The transitions of $\mathcal{P}$ are deterministic: after reading each letter, it switches from state $q_{i,j}$ to $q_{i,j+1} \pmod{p_{m+i-1}}$. In fact, $\mathcal{P}$ executes $k$ copies of DFAs with equal probability. The aim of the $i$-th DFA is to determine whether the length of the input is equivalent to zero in mod $p_{m+i-1}$. By construction, it is clear that $\mathcal{P}$ accepts any yes-instance with the probability $\frac{1}{k}$ and any no-instance with probability at most $\frac{1}{3}$. The number of states is $|Q| = p_m + \cdots + p_{m+k-1}$, i.e.,

$$
|Q| = \sum_{i=m}^{m+k-1} p_i \leq O(k(m + k) \log(m + k)).
$$

$\square$

We summarize our results in this section in Table 1. The upper bound for QFAs does not depend on the value $m$. The bounds for DFAs and PFAs are almost tight and the gap between them is exponential in $k$. Currently, we do not know any better bound for QFAs. Moreover, if we pick $m = 2^k$, then we obtain an exponential gap between QFAs and PFAs.
Table 1  Summary of upper and lower bounds on the number of states for $P^{m,k}$

|        | DFA                  | PFA                  | QFA     |
|--------|----------------------|----------------------|---------|
| Lower bounds | $\Omega (m \log m \cdot \frac{k}{2})$ | $\Omega (k(m + k) \log m)$ | 1       |
| Upper bounds | $O((m + \frac{k}{2}) \log(m + \frac{k}{2}))^{\frac{k}{2}}$ | $O(k(m + k) \log(m + k))$ | $2k$    |

Acknowledgements  Yakaryılmaz was partially supported by ERC Advanced Grant MQC and CAPES with grant 88881.030338/2013-01. We thank Viliam Geffert for his corrections on the text and the anonymous reviewers for their very helpful comments.

References

1. Ablayev, F., Gainutdinova, A., Khadiev, K., Yakaryılmaz, A.: Very narrow quantum OBDDs and width hierarchies for classical OBDDs. In: DCFS, LNCS, vol. 8614, pp. 53–64. Springer (2014)
2. Ablayev, F., Gainutdinova, A., Khadiev, K., Yakaryılmaz, A.: Very narrow quantum OBDDs and width hierarchies for classical OBDDs. Lobachevskii J. Math. 37(6), 670–682 (2016)
3. Ablayev, F.M., Gainutdinova, A.: On the lower bounds for one-way quantum automata. In: MFCS, LNCS, vol. 1893, pp. 132–140. Springer (2000)
4. Ablayev, F.M., Gainutdinova, A.: Complexity of quantum uniform and nonuniform automata. In: DLT, LNCS, vol. 3572, pp. 78–87. Springer (2005)
5. Ablayev, F.M., Gainutdinova, A., Karpinski, M., Moore, C., Pollett, C.: On the computational power of probabilistic and quantum branching program. Inf. Comput. 203(2), 145–162 (2005)
6. Ambainis, A., Beaudry, M., Golovkins, M., Kikusts, A., Mercer, M., Thérien, D.: Algebraic results on quantum automata. Theory Comput. Syst. 39(1), 165–188 (2006)
7. Ambainis, A., Freivalds, R.: 1-way quantum finite automata: strengths, weaknesses and generalizations. In: FOCS'98, pp. 332–341 (1998)
8. Ambainis, A., Watrous, J.: Two-way finite automata with quantum and classical states. Theoretical Comput. Sci. 287(1), 299–311 (2002)
9. Ambainis, A., Yakaryılmaz, A.: Superiority of exact quantum automata for promise problems. Inf. Process. Lett. 112(7), 289–291 (2012)
10. Ambainis, A., Yakaryılmaz, A.: Automata from mathematics to applications, chap. Automata and quantum computing (to appear). (arXiv:1507.01988)
11. Apostol, T.M.: Introduction to Analytic Number Theory. Springer, Berlin (1976)
12. Bertoni, A., Carpentieri, M.: Analogies and differences between quantum and stochastic automata. Theor. Comput. Sci. 262(1–2), 69–81 (2001)
13. Bianchi, M.P., Mereghetti, C., Palano, B.: Complexity of promise problems on classical and quantum automata. In: Computing with New Resources, LNCS, vol. 8808, pp. 161–175. Springer (2014)
14. Condon, A., Lipton, R.J.: On the complexity of space bounded interactive proofs (extended abstract). In: FOCS’89, pp. 462–467 (1989)
15. Doron, D., Sarid, A., Ta-Shma, A.: On approximating the eigenvalues of stochastic matrices in probabilistic logspace. Comput. Complex. 26(2), 393–420 (2017)
16. Freivalds, R.: Probabilistic two-way machines. In: Proceedings of the International Symposium on Mathematical Foundations of Computer Science, pp. 33–45 (1981)
17. Gainutdinova, A., Yakaryılmaz, A.: Unary probabilistic and quantum automata on promise problems. In: Developments in Language Theory, Lecture Notes in Computer Science, vol. 9168, pp. 252–263. Springer (2015)
18. Gainutdinova, A.F.: Comparative complexity of quantum and classical OBDDs for total and partial functions. Rus. Math. 11(59), 26–35 (2015)
19. Geffert, V., Yakaryılmaz, A.: Classical automata on promise problems. In: DCFS, LNCS, vol. 8614, pp. 126–137. Springer (2014). ECCC:TR14-136
20. Geffert, V., Yakaryılmaz, A.: Classical automata on promise problems. Discret. Math. Theor. Comput. Sci. 17(2), 157–180 (2015)
21. Goldreich, O.: On promise problems: a survey. Essays Mem. Shimon Even, LNCS 3895, 254–290 (2006)
22. Gruska, J., Qiu, D., Zheng, S.: Potential of quantum finite automata with exact acceptance. Int. J. Found. Comput. Sci. 26(3), 381–398 (2015)
23. Gruska, J., Qiu, D., Zheng, S.: Generalizations of the distributed Deutsch-Jozsa promise problem. Math. Struct. Comput. Sci. 27(3), 311–331 (2017)
24. Hirvensalo, M.: Quantum automata with open time evolution. Int. J. Nat. Comput. 1(1), 70–85 (2010)
25. Kapoutsis, C.A.: Minicomplexity. J. Autom. Lang. Comb. 17(2–4), 205–224 (2012)
26. Kemeny, J.G., Snell, J.L.: Finite Markov Chains. Van Nostrand, Princeton (1960)
27. Klauck, H.: On quantum and probabilistic communication: Las Vegas and one-way protocols. In: STOC’00, pp. 644–651 (2000)
28. Kondacs, A., Watrous, J.: On the power of quantum finite state automata. In: FOCS’97, pp. 66–75 (1997)
29. Mereghetti, C., Palano, B., Pighizzini, G.: Note on the succinctness of deterministic, nondeterministic, probabilistic and quantum finite automata. Theor. Inf. Appl. 35(5), 477–490 (2001)
30. Moore, C., Crutchfield, J.P.: Quantum automata and quantum grammars. Theor. Comput. Sci. 237(1–2), 275–306 (2000)
31. Murakami, Y., Nakanishi, M., Yamashita, S., Watanabe, K.: Quantum versus classical pushdown automata in exact computation. IPSJ Digit. Cour. 1, 426–435 (2005)
32. Nakanishi, M.: Quantum pushdown automata with a garbage tape. In: SOFSEM2015 Proceedings of the forty-first International Conference on Current Trends in Theory and Practice of Computer Science, LNCS, pp. 352–363. Springer (2015). (arXiv:1402.3449)
33. Nakanishi, M., Yakaryılmaz, A.: Classical and quantum counter automata on promise problems. In: Implementation and Application of Automata, Lecture Notes in Computer Science, vol. 9223, pp. 224–237. Springer (2015)
34. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information. Cambridge University Press, Cambridge (2000)
35. Rashid, J., Yakaryılmaz, A.: Implications of quantum automata for contextuality. In: Conference on Implementation and Application of Automata, LNCS, vol. 8587, pp. 318–331. Springer (2014). ArXiv:1404.2761
36. Rotman, J.: An introduction to the theory of groups. number 148 in graduate texts in mathematics (1994)
37. Salomaa, A., Soittola, M.: Automata-Theoretic Aspects of Formal Power Series. Texts and monographs in computer science. Springer, Berlin (1978)
38. Say, A.C.C., Yakaryılmaz, A.: Quantum finite automata: A modern introduction. In: Computing with New Resources, LNCS, vol. 8808, pp. 208–222. Springer (2014)
39. Shur, A.M., Yakaryılmaz, A.: More on quantum, stochastic, and pseudo stochastic languages with few states. Nat. Comput. 15(1), 129–141 (2016)
40. Watrous, J.: Encyclopedia of Complexity and System Science, chap. Quantum computational complexity. Springer (2009). Also available at arXiv:0804.3401
41. Yakaryılmaz, A., Say, A.C.C.: Languages recognized by nondeterministic quantum finite automata. Quantum Inf. Comput. 10(9&10), 747–770 (2010)
42. Yakaryılmaz, A., Say, A.C.C.: Unbounded-error quantum computation with small space bounds. Inf. Comput. 279(6), 873–892 (2011)
43. Zheng, S., Gruska, J., Qiu, D.: On the state complexity of semi-quantum finite automata. In: Language and Automata Theory and Applications, LNCS, vol. 8370, pp. 601–612. Springer (2014)
44. Zheng, S., Li, L., Qiu, D., Gruska, J.: Promise problems solved by quantum and classical finite automata. Theor. Comput. Sci. 666, 48–64 (2017)
45. 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. 499, 98–112 (2013)