INSTITUTO DE COMPUTAÇÃO
UNIVERSIDADE ESTADUAL DE CAMPINAS

Language classes and quantum finite automata

A. B. Grilo *A. V. Moura*

Technical Report - IC-12-22 - Relatório Técnico

July - 2012 - Julho

Language classes and quantum finite automata

Alex Bredariol Grilo   Arnaldo Vieira Moura

Abstract

Richard Feynman triggered the study of Quantum Computing conjecturing that classical computers could not simulate quantum systems efficiently. Since then, quantum computational models have been proposed to study how quantum mechanics influences the power of computing devices. In this work, we study two-way finite automata with quantum and classical states (2QCFA). We systematize previous results on 2QCFAs and show that 2QCFAs can recognize ambiguous context-free languages and also non-context-free languages in polynomial time, extending previous results.

1 Introduction

Since Richard Feynman conjectured that no classical computer could simulate efficiently a quantum system, the research on Quantum Computing has developed and has achieved some important breakthroughs, like the algorithms proposed by Shor [15] and Grover [6], which can solve some important problems more efficiently than their known classical counterparts. A Quantum Computer, a device that would be able to implement these theoretical achievements, is not feasible yet and it is still a challenge for physicists and engineers.

One of the most important abstract quantum devices is the Quantum Turing Machine (QTM), proposed by Deutsch [4] and studied by Bernstein and Vazirani [3], is a quantum analog of a Turing Machine[1]. In order to study the power of quantum mechanics in computation, other simpler classical computational models have also been extended to quantum models, such as quantum finite automata [9][8][1] and quantum pushdown automata [9][5].

In this work, we study the power of 2QCFA, organizing the previous results by way of the category of languages recognized by them. We focus on the classical language hierarchy: regular languages, context-free languages (CFLs) and non-context-free languages (non-CFLs). In addition, we construct two new 2QCFA that recognize an ambiguous CFL and a non-CFL, both with one-sided error and in polynomial time. The main contribution of this work is a systematization of the state of art on languages recognized by 2QCFA and the presentation of two more 2QCFA that accept languages belonging to important language classes, extending previous results.

We begin the study in the next section describing two-way finite automata with classical and quantum states model. In Section 4, we prove that all regular languages can be recognized by 2QCFA. Then, in Section 5, we present some 2QCFA that recognize different

[1]Quantum circuits [4], another important quantum computational model, is equivalent to the QTM model [18].

categories of CFLs. Finally, in Section section 6 we present two 2QCFA that recognize two non-CFLs. In the last section, we conclude and suggest some future work.

We assume, in this paper, a basic knowledge of quantum computation and formal languages. For a revision of quantum computation, we recommend Nielsen and Chuang [10] as a more complete and broad reference, or Yanofsky and Mannucci [17] for a reference closer to Computer Scientists. Formal languages and automata are well presented in Hopcroft and Ullman [7] and Sipser [16].

2 Related Work

Initially, there were two independent definitions of Quantum Finite Automata: the Measure-Once one-way quantum finite automata (MO-1QFA), proposed by Moore and Crutchfield [9], and the Many-Measure one-way quantum finite automata (MM-1QFA), proposed by Kondacs and Waltrous [8]. In both approaches, the set of languages recognized with bounded error by 1QFA was proved to be strictly contained in the set of regular languages.

Then, Kondacs and Waltrous [8] defined the two-way quantum finite automata (2QFA) model, in which the head can move left, right or stay in the same tape cell at each move. The 2QFA model was proven to be strictly stronger than the deterministic finite automata model [8].

Later, Ambainis and Waltrous [2] proposed the two-way finite automata model with quantum and classical states (2QCFA). and devised 2QCFA for two non-regular context-free languages (CFLs) Then, Qiu [11] proved some closure properties about the class of languages recognized by 2QCFA, and used these results to derived 2QCFA for recognizing other languages. Later Zheng *et al* [13][14] proposed 2QCFA for accepting some other non-regular languages, including a non-context-language (non-CFL) that is recognized in exponential time.

3 Two-way quantum finite automata with quantum and classical states

The two-way quantum finite automata (2QFA) model proposed by Kondacs and Waltrous [8], based only in quantum states, permits the head to be at different cells of the input tape at the same moment. This characteristic brings up the problem that it could only be implemented with $\log |w|$ bits of information, for any input w. In order to solve this problem, Ambainis and Waltrous [2] have proposed a variant of the 2QFA, called two-way finite automata with quantum and classical states (2QCFA). The difference between these models is that, while a configuration of a 2QFA consists of the superposition of quantum states and head positions, in 2QCFAs a configuration is defined by a classical state, the position of the head and a superposition of quantum states.

Formally, a 2QCFA is a 9-tuple $M = (Q, S, \Sigma, \Theta, \delta, q_0, s_0, S_{acc}, S_{rej})$, where Q is the set of quantum states, S is the set of classical states, Σ is the input alphabet, Θ and δ are the evolution functions for the quantum and classical states, respectively, $q_0 \in Q$ and $s_0 \in S$

are the initial states for the quantum and classical states, respectively, and $S_{acc}, S_{rej} \subseteq S$ are the sets of accept and reject classical states. We assume $S_{acc} \cap S_{rej} = \emptyset$.

For an input w, the tape has $|w| + 2$ squares: the first one, indexed by 0, contains the left-marker ¢ , the last one contains the right-marker \$ and the i-th position of the tape, $1 \leq i \leq |w|$, contains the i-th symbol of the input w. We assume ¢, \$ $\notin \Sigma$ and we call $\Gamma = \Sigma \cup \{¢, \$\}$ the tape alphabet.

Definition 3.1 *The set of halting states is $S_{halt} = S_{acc} \cup S_{rej}$ and S_{non} the set of non-halting states is $S_{non} = S \backslash S_{halt}$.*

As dictated by quantum mechanics, an operation over quantum systems can be either a linear unitary transformation or a measurement. In the 2QCFA model, the action performed over the quantum states is determined by the classical state and the tape symbol under the head. So, the quantum evolution function Θ is defined as $\Theta : S_{non} \times \Gamma \to \mathcal{U}(\mathcal{H}(Q)) \cup \mathcal{M}(\mathcal{H}(Q))$, where $\mathcal{H}(Q)$ is the Hilbert space in which Q is a basis, $\mathcal{U}(\mathcal{S})$ is the set of unitary linear operations over the Hilbert space $\mathcal{S}$ and $\mathcal{M}(\mathcal{S})$ is the set of projective measurements[2] over the Hilbert space $\mathcal{S}$.

The transition function δ depends on whether Θ applies a measurement or a linear unitary transformation. In the former case, δ will map $S_{non} \times \Gamma \times R$ to $S \times D$, where R is the set of possible results of the measurement and D is the set of possible movements of the head, $D = \{-1, 0, 1\}$[3]. When Θ applies a linear unitary transformation, δ will use only the classical information of the current configuration, that is $\delta : S_{non} \times \Gamma \to S \times D$.

The initial configuration of a 2QCFA M is the classical initial state s_0, the head position over cell 0 and the quantum initial state is $|q_0\rangle$. Then, the computation of the 2QFCA will be to apply, consecutively, the transformation Θ and then the transformation δ, until the automaton reaches one of the classical halting states.

Since δ may depend on the result of quantum measurements, which are probabilistic in nature, the classical part of the 2QCFA also has probabilistic nature. We say that the 2QCFA accepts a string w with probability p_{acc} if it enters an accept state with probability p_{acc} when computing over w. We can define the analog rejection probability p_{rej} as the probability the 2QCFA enters in a reject state when computing over w. If we assume that the 2QCFA halts, then the acceptance and rejection probabilities sum up to 1.

Definition 3.2 *We say M recognizes a language L with zero error if M accepts w with probability 1 when $w \in L$ and M rejects w with probability 1 when $w \notin L$.*

Definition 3.3 *We say M recognizes a language L with one-sided error ϵ if M accepts w with probability 1 when $w \in L$ and M rejects w with probability at least $1 - \epsilon$ when $w \notin L$.*

4 Regular languages and 2QCFA

Regular languages are those languages that can be recognized by deterministic finite automata (DFAs), one of the simplest computational models. In this section we prove a

[2]The use of POVM-type measurements does not change the power of the computational device [2]

[3]We assume that whenever the head is in the first(last) cell, it never moves left(right).

straightforward theorem showing that all regular languages can be recognized by 2QCFA with zero error in linear time. The idea of the proof is to simulate a DFA that recognizes the language, ignoring the quantum states.

Theorem 4.1 *If L is a regular language, there is a 2QCFA M that recognizes L with zero error in linear time.*

Proof If L is a regular language, there is a DFA $D = (S_1, \Sigma, \delta_1, s_0, F)$[4] that recognizes L. Let $M = (Q, S, \Sigma, \Theta, \delta, q_0, s_0, S_{acc}, S_{rej})$ be a 2QCFA, where

$$\begin{aligned} Q &= \{q_0\}, \\ S &= S_1 \cup \{s_{acc}, s_{rej}\}, \\ q_0 &= |q_0\rangle, \\ S_{acc} &= \{s_{acc}\} \text{ and} \\ S_{rej} &= \{s_{rej}\}. \end{aligned}$$

For all classical states and symbols of the tape alphabet, the quantum evolution operator Θ will apply the identity transformation. Therefore, δ does not need to be defined after projective measurements. After the application of unitary operations, δ will be:

$$\delta(s,\sigma) = \begin{cases} (s, +1), & \text{if } \sigma = \text{¢} \\ (\delta_1(s,\sigma), +1), & \text{if } \sigma \in \Sigma \\ (s_{acc}, 0), & \text{if } \sigma = \$ \text{ and } s \in F \\ (s_{rej}, 0), & \text{if } \sigma = \$ \text{ and } s \notin F. \end{cases}$$

By construction, the 2QCFA is not affected by the quantum states and it can easily be seen that D computes the i-th position of the input string in state s iff M is over the cell indexed by i with the same classical state.

At the end, if D accepts the input, M will reach the right-end marker in a state in F. In this case, δ will move move to an accept state. Otherwise, M will reach the right-end marker in a state that does not belong to F and δ will change into a reject state. So M will accept the input string iff D accepts it.

As we can see, the transitions that do not move the head to the right are the ones that change into a halting state. So, M takes $|w| + 2$ steps to accept or reject the input.

5 Context-free languages and 2QCFA

Formally, CFLs are those languages recognized by pushdown automata (PDA) which informally are non-deterministic machines with a finite control and an infinite pushdown tape.

[4] A general introduction to regular languages and DFA in chapter 1 of Sipser [16].

Equivalently, the set of CFLs is the set of languages generated by context-free grammars (CFGs)[5].

In this section, we will start with a deterministic CFL recognizable with one-sided error by 2QCFA model. Then, in Subsection 5.2 we show two 2QCFAs that accept different non-deterministic and unambiguous CFLs. In the last subsection we show a 2QCFA that recognizes an inherently ambiguous CFL.

In this section, when we say that a 2QCFA recognizes a language, we mean it recognizes the language with one-sided error probability ϵ, $0 < \epsilon < \frac{1}{2}$, unless we explicitly state otherwise.

5.1 2QCFAs and deterministic context-free languages

A deterministic context-free language (DCFL) is a language that can be recognized by a deterministic pushdown automata (DPDA). The DCFLs are a proper subset of the context-free language. Their importance in Computer Science stems from the fact that it is possible to construct efficient parsers for them[6].

Some DCFLs have been proven to be recognized by 2QCFA in polynomial time: the language in which the number of a's is equal to the number of b's [11], language of strings in the form xcy, where $\Sigma = \{a, b, c\}$ and $x, y \in \{a, b\}^*$ [13] and $|x| = |y|$, as well as the complement of all of these languages [11].

In this subsection, we present the result obtained in [2], that shows a 2QCFA recognizing the deterministic CFL $\{a^n b^n | n \geq 0\}$.

Theorem 5.1 *The 2QFCA M described in Figure 1 recognizes the language $\{a^n b^n | n \geq 0\}$ with one-sided error ϵ, $0 < \epsilon < \frac{1}{2}$, and in time $O(|w|^4)$, where w is the input string.*

Proof See Section 4 of Ambainis and Waltrous [2].

5.2 2QCFAs and non-deterministic unambiguous context-free languages

Non-deterministic CFLs are the ones that cannot be recognized by deterministic PDA, resulting in that they are more difficult to parse classically. An unambiguous CFL is a language that can be generated by an unambiguous CFG[7].

Some results about recognition of non-deterministic unambiguous CFLs by 2QCFAs have been obtained. We will show in this subsection a 2QCFA that recognizes the language $\{a^n b^n | n \geq 0\} \cup \{a^{2n} b^n | n \geq 0\}$ in polynomial time and other 2QCFA that recognizes palindromes over the alphabet $\{a, b\}$ in exponential time.

---

[5] A general introduction for CFLs, PDAs and CFGs can be found in chapter 2 of Sipser [16].

[6] More details about DCFLs and DPDAs can be found in chapter 10 of Hopcroft and Ullman [7].

[7] An unambiguous CFG is one whose generated strings have only one derivation tree. More details about derivation trees and ambiguity can be found in chapter 4 of Hopcroft and Ullman [7].

1. Choose $0 < \epsilon < \frac{1}{2}$
2. Let $|q_0\rangle$ and $|q_1\rangle$ be the quantum states.
3. Let R_α be the unitary transformation $\begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix}$
4. Let $k = 1 - \lfloor \log \epsilon \rfloor$
5. Check classically if $w \in a^*b^*$ and if not, reject it.
6. Repeat ad infinitum
   - 6.1 Move the tape head to the first a and set the quantum state to $|q_0\rangle$
   - 6.2 While the currently scanned symbol is not \$
     - If the currently scanned symbol is a, perform $R_{\sqrt{2}}$ on the quantum state
     - If the currently scanned symbol is b, perform $R_{-\sqrt{2}}$ on the quantum state
     - Move the tape head one cell to the right
   - 6.3 Measure the quantum state and if the result is not q_0, reject
   - 6.4 Repeat two times
     - Move the tape head to the first a
     - While the currently scanned symbol is not a ¢ or \$:
       - Simulate a coin flip
       - If the coin shows "heads" move right, otherwise move left
   - 6.5 If both times the process ends at \$
     - Simulate k coin flips
     - If none was "heads", accept

Figure 1: A 2QCFA recognizing $a^n b^n$, $n \geq 0$

5.2.1 The language $\{a^n b^n | n \geq 0\} \cup \{a^{2n} b^n | n \geq 0\}$

We will describe a method for constructing a 2QCFA that recognizes the language $L = \{a^n b^n | n \geq 0\} \cup \{a^{2n} b^n | n \geq 0\}$[8]. First, we show some auxiliary lemmas. The first lemma, proves that there is a 2QCFA that recognizes the language $\{a^{kn} b^n | n \geq 0\}$ for any fixed $k \in \mathbb{N}$.

Lemma 5.2 *Let $L_k = \{a^{kn} b^n | n \geq 0\}, k \in \mathbb{N}$. For a fixed k, there is a 2QCFA that recognizes L_k with one-sided error ϵ, $0 < \epsilon < \frac{1}{2}$, in polynomial time.*

Proof See Remark 2 in Qiu [11].

The next lemma shows that the union of languages recognizable by 2QCFA in polynomial time is also recognizable by a 2QCFA in polynomial time.

Lemma 5.3 *If L_1 is recognized by 2QCFA M_1 in polynomial time with one-sided error ϵ_1 and L_2 is recognized by 2QCFA M_2 in polynomial time and with one-sided error ϵ_2, then there is a 2QCFA that recognizes $L_1 \cup L_2$ in polynomial time and with a one-sided error $\epsilon = \epsilon_1 + \epsilon_2 - \epsilon_1\epsilon_2$.*

[8]By example 4.11 of Reghizzi [12] it can be seen that L is a non-deterministic unambiguous CFL.

Proof See Theorem 2 of Qiu [11].

We now use the previous lemmas and Theorem 5.1 to show that L is recognized by a 2QCFA in polynomial time.

Theorem 5.4 *There is a 2QCFA that accepts the language* $\{a^n b^n | n \geq 1\} \cup \{a^{2n} b^n | n \geq 1\}$ *with one-sided error probability* δ*,* $0 < \delta < \frac{3}{4}$*, in polynomial time.*

Proof From Theorem 5.1 and Lemma 5.2, we can state that there are 2QCFAs M_1 and M_2 that recognize with one-sided error ϵ, and in polynomial time, the languages $\{a^n b^n | n \geq 0\}$ and $\{a^{2n} b^n | n \geq 0\}$, respectively.

Applying the Lemma 5.3 there is a 2QCFA that recognizes the language $\{a^n b^n | n \geq 0\} \cup \{a^{2n} b^n | n \geq 0\}$ in polynomial time and with one-sided error $2\epsilon - \epsilon^2$.

5.2.2 Palindromes

Let $L_{pal} = \{x | x \in \{a, b\}^*$ and $x = x^R\}$ be the languages of palindromes over the alphabet $\{a, b\}$[9]. We will now present a result of [2] that states that L_{pal} is recognizable by a 2QCFA in exponential time.

Theorem 5.5 *The 2QCFA M described by Figure 2 accepts a string* $w \in L_{pal}$ *with probability* 1*, and rejects* $w \notin L_{pal}$ *with probability at least* $1 - \epsilon$*,* $0 < \epsilon < \frac{1}{2}$*, and its expected number of steps is exponential in relation to the input size.*

Proof See section 3 of Ambainis and Waltrous [2].

5.3 2QCFAs and inherently ambiguous context-free languages

Inherently ambiguous CFLs are the ones that cannot be generated by any unambiguous CFG. Ambiguity causes problems when parsing, so inherently ambiguous CFLs are on the hardest class of CFLs for parsing.

In this subsection, we will show a 2QCFA that recognizes with one-sided error the inherently ambiguous CFL $L_{ijk} = \{a^i b^j c^k | i = j \text{ or } j = k\}$[10]. The idea behind this 2QCFA is to use the construction described in Section 5.1 to check if $i = j$, and if it is not, check if $j = k$ using the same idea.

Theorem 5.6 *The 2QCFA M described in Figure 3 recognizes with one-sided error* δ *the language* L_{ijk} *in time* $O(|w|^4)$*, where* w *is the input string and* $0 < \delta < \frac{3}{4}$*.*

Proof First, M rejects classically all inputs not of the form $a^* b^* c^*$. After this step, we know that $w = a^i b^j c^k$, for some $i, j, k \geq 0$.

We can use the Theorem 5 of [2] to show that if $i = j$, then loop 6 in Figure 3 accepts the input string with probability 1 in $O(|w|^4)$ expected number of steps. Otherwise, the

[9] By example 4.12 of Reghizzi [12] it can be seen that L_{pal} is a non-deterministic unambiguous CFL.
[10] It can be seen in Sipser [16] that L_{ijk} is an inherently ambiguous CFL.

1. Choose $0 < \epsilon < \frac{1}{2}$
2. Let $|q_0\rangle$, $|q_1\rangle$ and $|1_2\rangle$ be the quantum states.
3. Let U_a be the unitary transformation $\begin{pmatrix} \frac{4}{5} & -\frac{3}{5} & 0 \\ \frac{3}{5} & \frac{4}{5} & 0 \\ 0 & 0 & 1 \end{pmatrix}$
4. Let U_b be the unitary transformation $\begin{pmatrix} \frac{4}{5} & 0 & -\frac{3}{5} \\ 0 & 1 & 0 \\ \frac{3}{5} & 0 & \frac{4}{5} \end{pmatrix}$
5. Let $k = \max\{\log 25, -\log \epsilon\}$
6. Repeat ad infinitum
   - 6.1 While the current scanned symbol σ is not \$
     - Apply U_σ on the quantum state
     - Move the tape head to the right
   - 6.2 Move the tape head to the first symbol of the input string
   - 6.3 While the current scanned symbol σ is not \$
     - Apply U_σ^{-1} on the quantum state
     - Move the tape head to the right
   - 6.4 Measure the quantum state and reject if it is q_0
   - 6.5 Set $f = 0$
   - 6.6 While the current scanned symbol σ is not ¢
     - Simulate k coin-flips and set $f = 1$ if any flip turns up as "heads"
     - Move the tape head to the left
   - 6.7 If $f = 0$, accept

Figure 2: A 2QCFA recognizing palindromes

loop change the classical state to s (terminating that loop) with probability $1-\epsilon$, or accepts w with probability ϵ, both within $O(|w|^4)$ expected number of steps.

We can use the Theorem 5 of [2] again to prove that the loop 7 rejects the input string with probability $1-\epsilon$ when $j \neq k$, and accepts it with probability 1 if $j = k$. In both cases, the expected number of steps is $O(|w|^4)$.

So, if w is not in the form $a^i b^j c^k$ it is rejected with probability 1. If $i = j$, the input is accepted with probability 1. If $i \neq j$ and $j = k$, the input is accepted with probability $\epsilon + (1-\epsilon)1 = 1$. If $i \neq j$ and $j \neq k$, then w is accepted with probability $\delta = \epsilon + (1-\epsilon)\epsilon = 2\epsilon - \epsilon^2$. Thus, M recognizes L_{ijk} with one-sided error δ.

In all cases, the expected number of steps is $O(|w|^4)$.

6 Non-context-free languages and 2QCFA

Non-context-free languages (non-CFLs) are the languages that cannot be recognized by PDAs. In this section, we will present two non-CFLs recognized with one-sided error by 2QCFA.

1. Choose $0 < \epsilon < \frac{1}{2}$
2. Let $|q_0\rangle$ and $|q_1\rangle$ be the quantum states.
3. Let R_α be the unitary transformation $\begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix}$
4. Check classically whether $w \in a^*b^*c^*$ and if it is not, reject it.
5. Let $k = 1 - \lfloor \log \epsilon \rfloor$
6. Repeat until the classical state is not s

   6.1 Move the tape head to the first a and set the quantum state to $|q_0\rangle$

   6.2 While the currently scanned symbol is not c or $

   If the currently scanned symbol is a, perform $R_{\sqrt{2}}$ on the quantum state
   If the currently scanned symbol is b, perform $R_{-\sqrt{2}}$ on the quantum state
   Move the tape head one cell to the right

   6.3 Measure the quantum state

   6.4 If the result is not q_0

   Change the classical state to s and go to loop 7

   6.5 Repeat two times

   Move the tape head to the first a
   While the currently scanned symbol is not ¢, c or $:
   Simulate a coin flip
   Measure the coin flip and if it shows "heads" move right, otherwise move left
   If both times the process ends at c or $
   Simulate k coin flips
   If none was "heads", accept

7. Repeat ad infinitum

   7.1 Move the tape head to the first b and set the quantum state to $|q_0\rangle$

   7.2 While the currently scanned symbol is not $

   If the currently scanned symbol is b, perform $U_{\sqrt{2}}$ on the quantum state
   If the currently scanned symbol is c perform $U_{-\sqrt{2}}$ on the quantum state
   Move the tape head one cell to the right

   7.3 Measure the quantum state and if the result is not q_0, reject

   7.4 Repeat two times

   Move the tape head to the first b
   While the currently scanned symbol is not an ¢, a or $:
   Simulate a coin flip
   Measure the coin flip and if it shows "heads" move right, otherwise move left

   7.5 If both times the process ends at $, accept

   Simulate k coin flips
   If none shows "heads", accept

Figure 3: A 2QCFA recognizing $\{a^ib^jc^k | i = j \text{ or } j = k\}$

6.1 A 2QCFA recognizing $a^nb^nc^n$

In this subsection, we will show a 2QCFA that recognizes with one-sided error the non-CFL $L_n = \{a^nb^nc^n | n \geq 0\}$[11]. The idea behind this construction is similar to that one presented

[11] It can be seen in Hopcroft and Ullman [7] that L_n is not a CFL.

in 5.3.

1. Choose $0 < \epsilon < \frac{1}{2}$
2. Let $|q_0\rangle$ and $|q_1\rangle$ be the quantum states.
3. Let R_α be the unitary transformation $\begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix}$
4. Let $k = 1 - \lfloor \log \epsilon \rfloor$
5. Check classically if $w \in a^*b^*c^*$ and if not, reject it.
6. Repeat until the classical state is not s
   - 6.1 Move the tape head to the first a and set the quantum state to $|q_0\rangle$
   - 6.2 While the currently scanned symbol is not c or \$
     - If the currently scanned symbol is a, perform $R_{\sqrt{2}}$ on the quantum state
     - If the currently scanned symbol is b, perform $R_{\sqrt{2}}$ on the quantum state
     - Move the tape head one cell to the right
   - 6.3 Measure the quantum state
   - 6.4 If the result is q_1, reject
   - 6.5 Repeat two times
     - Move the tape head to the first a
     - While the currently scanned symbol is not ¢, c or \$:
       - Simulate a coin flip
       - Measure the coin flip and if it shows "heads" move right, otherwise move left
     - If both times the process ends at c or \$
       - Simulate k coin flips
       - If none shows "heads", change the classical state to s
7. Repeat ad infinitum
   - 7.1 Move the tape head to the first b and set the quantum state to $|q_0\rangle$
   - 7.2 While the currently scanned symbol is not \$
     - If the currently scanned symbol is b, perform $U_{\sqrt{2}}$ on the quantum state
     - If the currently scanned symbol is c perform $U_{-\sqrt{2}}$ on the quantum state
     - Move the tape head one cell to the right
   - 7.3 Measure the quantum state and if the result is not q_0, reject
   - 7.4 Repeat two times
     - Move the tape head to the first b
     - While the currently scanned symbol is not an ¢, a or \$:
       - Simulate a coin flip
       - Measure the coin flip and if it shows "heads" move right, otherwise move left
   - 7.5 If both times the process ends at \$, accepts
     - Simulate k coin flips
     - If none shows "heads", accept

Figure 4: A 2QCFA recognizing L_n

Theorem 6.1 *The 2QCFA M described in Figure 4 recognizes L_n with one-sided error ϵ in time $O(|w|^4)$, where w is the input string and $0 < \epsilon < \frac{1}{2}$.*

Proof First, M rejects classically all the string that are not in $a^*b^*c^*$. So, after this step, we know that $w = a^ib^jc^k$, for some $i, j, k \geq 0$.

We can use the Theorem 5 of [2] to show that if $i = j$, then loop 6 in Figure 4 change the classical state to s and terminates the loop with probability 1, within $O(|w|^5)$ expected number of steps. Otherwise, the loop changes the classical state to s and stops with probability ϵ, or rejects w with probability $1 - \epsilon$, both within $O(|w|^4)$ expected number of steps.

We can use the Theorem 5 of [2] again to prove that loop 7 rejects the input string with probability $1 - \epsilon$ when $j \neq k$ and accepts it with probability 1 if $j = k$. In both cases, the expected number of steps is $O(|w|^4)$.

So, if w is not in the form $a^ib^jc^k$ it is rejected with probability 1. If $i = j$ and $j = k$, the input is accepted with probability 1. If $i = j$ and $j \neq k$, the input is accepted with probability $1 \times \epsilon = \epsilon$. If $i \neq j$ and $j = k$, the input is accepted with probability $\epsilon \times 1 = \epsilon$. If $i \neq j$ and $j \neq k$, then w is accepted with probability $\epsilon\epsilon = \epsilon^2$. Thus, M recognizes L_n with one-sided error ϵ.

In all cases, the expected number of steps is $O(|w|^4)$.

6.2 A 2QCFA for wcw

In this subsection, we will show a 2QCFA that recognizes with one-sided error the non-CFL $L_{twin} = \{wcw | w \in \{a, b\}^*\}$ in exponential time. The idea behind this construction is use a similar automata as that described in Subsection 5.2.2.

Theorem 6.2 *The 2QCFA M of Figure 5 accepts L_{twin} with one-sided error ϵ in exponential expected time, $0 < \epsilon < \frac{1}{2}$.*

Proof See Section 4 of Zheng *et al* [14].

7 Conclusions

In this work we have presented and systematized previous results about the classes of languages recognized by 2QCFA with one-sided error. We focused on the classical families of regular languages, the class of deterministic CFL, the set of non-deterministic but unambiguous CFLs recognized in polynomial time. We also discussed a non-CFL recognized in exponential time with one-sided error. Extending these results, we showed 2QCFA that recognize in polynomial time and one-sided error an ambiguous CFL and another non-CFL.

The exact relation of the set of languages recognized by 2QCFA and the class of all CFLs is still a rich source of open problems. Is there a CFL, deterministic or not, that cannot be recognized by a 2QCFA? Is there a language that can be recognized by 2QCFA only in exponential time? If the answer to the latter is positive, then another interesting question asks for a characterization the subset of CFLs that are recognized by 2QCFA in polynomial time.

1. Choose $0 < \epsilon < \frac{1}{2}$
2. Let $|q_0\rangle$, $|q_1\rangle$ and $|1_2\rangle$ be the quantum states.
3. Let U_a be the unitary transformation $\begin{pmatrix} \frac{4}{5} & -\frac{3}{5} & 0 \\ \frac{3}{5} & \frac{4}{5} & 0 \\ 0 & 0 & 1 \end{pmatrix}$
4. Let U_b be the unitary transformation $\begin{pmatrix} \frac{4}{5} & 0 & -\frac{3}{5} \\ 0 & 1 & 0 \\ \frac{3}{5} & 0 & \frac{4}{5} \end{pmatrix}$
5. Let $k = \max\{\log 25, -\log \epsilon\}$
6. Check classically if input is of the form $\{a,b\}^*c\{a,b\}^*$ and if not, reject it
7. Repeat ad infinitum

    7.1 While the current scanned symbol σ is not c

    Apply U_σ on the quantum state

    Move the tape head to the right

    7.2 Move the tape head to the last symbol of the input string

    7.3 While the current scanned symbol σ is not c

    Apply U_σ^{-1} on the quantum state

    Move the tape head to the left

    7.4 Measure the quantum state and reject if it is q_0

    7.5 Move the head to the last symbol

    7.6 Set $f = 0$

    7.7 While the current scanned symbol σ is not ¢

    Simulate k coin-flips and set $f = 1$ if any shows "heads"

    Move the tape head to the left

    7.8 If $f = 0$, accept

Figure 5: A 2QCFA recognizing $\{wcw | w \in \{a,b\}^*\}$

References

[1] M. Amano and K. Iwama. Undecidability on quantum finite automata. In *Proceedings of the thirty-first annual ACM symposium on Theory of computing*, STOC '99, pages 368–375, New York, NY, USA, 1999. ACM.

[2] A. Ambainis and J. Watrous. Two-way finite automata with quantum and classical states. *Theor. Comput. Sci.*, 287(1):299–311, September 2002.

[3] E. Bernstein and U. V. Vazirani. Quantum complexity theory. *SIAM J. Comput.*, pages 1411–1473, 1997.

[4] D. Deutsch. Quantum theory, the Church-Turing principle and the universal quantum computer. *Proceedings of the Royal Society of London A*, 400:97–117, 1985.

[5] M. Golovkins. Quantum pushdown automata. In *Proceedings of the 27th Conference on Current Trends in Theory and Practice of Informatics*, SOFSEM '00, pages 336–346, London, UK, UK, 2000. Springer-Verlag.

[6] L.. K. Grover. A fast quantum mechanical algorithm for database search. In *28th Annual ACM Symposium on the Theory of Computing*, page 212, 1996. New York.

[7] J.E. Hopcroft and J.D. Ullman. *Introduction to automata theory, languages, and computation.* Addison-Wesley series in computer science. Addison-Wesley, 1999.

[8] A. Kondacs and J. Watrous. On the power of quantum finite state automata. In *Proceedings of the 38th Annual Symposium on Foundations of Computer Science*, pages 66–, Washington, DC, USA, 1997. IEEE Computer Society.

[9] C. Moore and J. P. Crutchfield. Quantum automata and quantum grammars. *Theor. Comput. Sci.*, 237:275–306, April 2000.

[10] M.A. Nielsen and I.L. Chuang. *Quantum computation and quantum information.* Cambridge Series on Information and the Natural Sciences. Cambridge University Press, 2000.

[11] D. Qiu. Some observations on two-way finite automata with quantum and classical states. In *Proceedings of the 4th international conference on Intelligent Computing: Advanced Intelligent Computing Theories and Applications - with Aspects of Theoretical and Methodological Issues*, ICIC '08, pages 1–8, Berlin, Heidelberg, 2008. Springer-Verlag.

[12] S.C. Reghizzi. *Formal Languages and Compilation.* Texts in Computer Science. Springer, 2009.

[13] D. Qiu S. Zheng and L. Li. Some languages recognized by two-way finite automata with quantum and classical states. 2011.

[14] D. Qiu S. Zheng and L. Li. State succinctness of two-way finite automata with quantum and classical states. *CoRR*, abs/1202.2651, 2012.

[15] P. W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. *Proceedings 35th Annual Symposium on Foundations of Computer Science*, 35:124–134, 1994.

[16] M. Sipser. *Introduction to the Theory of Computation.* Thomson Course Technology, 2006.

[17] N.S. Yanofsky and M.A. Mannucci. *Quantum computing for computer scientists.* Cambridge University Press, 2008.

[18] A. C. Yao. Quantum circuit complexity. In *34th Annual Symposium on Foundations of Computer Science, 3-5 November 1993, Palo Alto, California, USA*, pages 352–361. IEEE, 1993.