Mika Hirvensalo

Improved Undecidability Results on the Emptiness Problem of Probabilistic and Quantum Cut-Point Languages

Turku Centre *for* Computer Science

TUCS Technical Report
No 769, May 2006

Improved Undecidability Results on the Emptiness Problem of Probabilistic and Quantum Cut-Point Languages

Mika Hirvensalo
TUCS–Turku Centre for Computer Science, and
Department of Mathematics, University of Turku
FIN-20014 Turku, Finland
mikhirve@utu.fi

Supported by the Academy of Finland under grant 208797.

TUCS Technical Report
No 769, May 2006

Abstract

We give constructions of probabilistic and MO-type quantum automata that have undecidable emptiness problem for the cut-point languages. The sizes of the automata over a binary alphabet are 25 and 21, rather than 47 and 43 given in [1] and [2], respectively.

Keywords: Undecidability, Stochastic Automata, Probabilistic Automata, Quantum Automata, Post Correspondence Problem, Cut-Point Languages

TUCS Laboratory
Discrete Mathematics for Information Technology

1 Introduction

A finite (deterministic) automaton consists of a finite set of *states* and a *transition function*, which describes the dynamics of the computation (see [15] for formal definitions). The states of the automaton are divided into *final* (accepting) and *non-accepting* states, and the type of the state after reading the last symbol determines if the word belongs to the language accepted by the automaton or not. Languages accepted by finite automata are called *regular*. The Pumping Lemma [15] makes it is clear that the *emptiness problem* of finite deterministic automata is algorithmically solvable.

In this article, we study two variants of finite automata: Probabilistic automata [11] and quantum automata of measure-once (MO) type [10]. It is known that the emptiness problems of *cut-point languages* and *strict cut-point languages* defined by probabilistic automata are undecidable [11],[1], and that so is the emptiness problem cut-point languages defined by quantum automata [2]. Quite surprisingly, the emptiness problem of strict cut-point languages determined by quantum automata turns out to be decidable [2]. The decision procedure for the said problem originates from the fact that the unitary matrices defining a quantum automata of MO-type generate a set whose closure is an *algebraic group*. For any such group (as well as for any algebraic set) there exists a finite set of defining polynomials, and a decision procedure follows from Tarski's general result for the real closed fields (see [13] for a detailed representation).

In this article, we improve the undecidability results of [1] and [2] by constructing automata with undecidable emptiness problems of smaller size than found previously. In [1] and [2] it has been shown that the emptiness problem for probabilistic cut-point languages and quantum cut-point languages is undecidable for automata sizes 47 and 43, respectively. Here we prove the undecidability results for automata of sizes 25, and 21, respectively.

2 Preliminaries

A vector $\boldsymbol{y} \in \mathbb{R}^n$ (seen as a column vector) is a *probability distribution*, if its coordinates are all nonnegative and sum up to 1. A matrix $M \in \mathbb{R}^{n\times n}$ is called a *Markov matrix* or *stochastic matrix*, if all its columns are probability distributions. We also say that a matrix M is *doubly stochastic*, if M and M^T both are stochastic matrices. Markov matrices M have the following property: if $\boldsymbol{y}$ is a probability distribution, so is $M\boldsymbol{y}$. More generally, matrices whose column entries sum up to 1 preserve vector's coordinate sum:

$$\sum_{i=1}^{n}(M\boldsymbol{y})_i = \sum_{i=1}^{n}\sum_{j=1}^{n} M_{ij}\boldsymbol{y}_j = \sum_{j=1}^{n}\boldsymbol{y}_j \sum_{i=1}^{n} M_{ij} = \sum_{j=1}^{n}\boldsymbol{y}_j.$$

Hence the Markov matrices preserve the L_1*-norm*

$$||\boldsymbol{x}||_1 = |\boldsymbol{x}_1| + \ldots + |\boldsymbol{x}_n|$$

of all vectors with nonnegative coordinates. Clearly a product of two Markov matrices is again a Markov matrix.

A *unitary matrix* $U \in \mathbb{C}^{n\times n}$ is a matrix whose columns form an orthonormal set with respect to *Hermitean inner product*

$$\langle \boldsymbol{x}, \boldsymbol{y}\rangle = \boldsymbol{x}_1^*\boldsymbol{y}_1 + \ldots + \boldsymbol{x}_n^*\boldsymbol{y}_n,$$

where c^* stands for the complex conjugate of c. The orthonormality of the columns is equivalent to $U^*U = I$, where U^* is the *adjoint matrix* of U defined as $(U^*)_{ij} = (U_{ji})^*$. Hence for a unitary matrix U we have $U^* = U^{-1}$, and therefore also $UU^* = I$, which is to say that also the rows of a unitary matrix form an orthonormal set.

Another equivalent characterization of the unitarity can be given in terms of L_2*-norm*

$$||\boldsymbol{x}||_2 = \sqrt{\langle \boldsymbol{x}, \boldsymbol{x}\rangle} = \sqrt{|\boldsymbol{x}_1|^2 + \ldots + |\boldsymbol{x}_n|^2}.$$

A matrix U is unitary if and only if $||U\boldsymbol{x}||_2 = ||\boldsymbol{x}||_2$ for each $\boldsymbol{x} \in \mathbb{C}^n$ [7]. In the sequel we denote $||\boldsymbol{x}||_2 = ||\boldsymbol{x}||$, unless otherwise stated. It is plain that a product of two unitary matrices is unitary.

Any subspace $V \subseteq \mathbb{C}^n$ defines an (orthogonal) *projection* by $\boldsymbol{x} \mapsto \boldsymbol{x}_V$, where $\boldsymbol{x} = \boldsymbol{x}_V + \boldsymbol{x}_{V^\perp}$ is the (unique) decomposition of $\boldsymbol{x}$ with $\boldsymbol{x}_V \in V$ and $\boldsymbol{x}_{V^\perp} \in V^\perp$ (the orthogonal complement of V). Each projection is a linear mapping, and it can be shown that $P \in \mathbb{C}^{n\times n}$ is a matrix of a projection if and only if $P^2 = P$ and $P^* = P$.

A *probabilistic automaton* (PFA, see [11] for further details) over an alphabet Σ is a triplet $(\boldsymbol{x}, \{M_a \mid a \in \Sigma\}, \boldsymbol{y})$, where $\boldsymbol{y} \in \mathbb{R}^n$ ($n = |\Sigma|$) is an *initial probability distribution*, each $M_a \in \mathbb{R}^{n\times n}$ is a Markov matrix, and $\boldsymbol{x} \in \mathbb{R}^n$ is the *final state vector* whose ith coordinate is 1, if the ith state is final, and 0 otherwise.

An equivalent definition of a probabilistic automaton can be given by using a transition function $\delta : Q \times \Sigma \times Q \mapsto [0, 1]$, where $Q = \{q_1, \ldots, q_n\}$ is the state set and $\delta(q_i, a, q_j) = (M_a)_{ji}$.

For any probabilistic automaton P we define a function $f_P : \Sigma^* \to [0, 1]$ as follows: If $w = a_1 \ldots a_r$, where $a_i \in \Sigma$, then

$$f_P(w) = \boldsymbol{x}^T M_{a_r} \cdot \ldots \cdot M_{a_1}\boldsymbol{y}. \tag{1}$$

The interpretation of (1) is as follows: the ith coordinate the initial distribution $\boldsymbol{y}$ stands for the probability of the automaton being initially in the ith state. Then, after reading the first letter a_1 of the input word, the ith coordinate of vector $M_{a_1}\boldsymbol{y}$ represents the probability that the automaton has entered ith state. Similarly, $M_{a_2}M_{a_1}\boldsymbol{y}$ represents the distribution of states after reading input letters a_1 and a_2. Finally, the ith coordinate of $M_{a_r} \cdot \ldots \cdot M_{a_1}\boldsymbol{y}$ gives the probability that the automaton is in the ith state after reading the whole input word, and $\boldsymbol{x}^T M_{a_r} \cdot \ldots \cdot M_{a_1}\boldsymbol{y}$ is the probability that starting from the initial distribution of states and reading word w, the automaton enters into one of the final states (corresponding to those coordinates where $\boldsymbol{x}$ has 1). If $w = a_1 \ldots a_r$, we use notation $M_w = M_{a_1} \cdot \ldots \cdot M_{a_r}$, so we can rewrite (1) as

$$f_P(w) = \boldsymbol{x}^T M_{w^R}\boldsymbol{y},$$

where $w^R = a_r \dots a_1$ is the *mirror image* of word $w = a_1 \dots a_r$. Also, instead of the final state vector we could use the *final state projection*, which is a diagonal matrix defined as $P_{ii} = 1$, if the ith state is final (in this case $\boldsymbol{x}_i = 1$), and $P_{ii} = 0$ otherwise (in this case $\boldsymbol{x}_i = 0$). It is then clear that

$$P(\boldsymbol{y}_1, \dots, \boldsymbol{y}_n)^T = (P_{11}\boldsymbol{y}_1, \dots, P_{nn}\boldsymbol{y}_n)^T,$$

and hence the sum of coordinates of $P\boldsymbol{y}$ is equal to $\boldsymbol{x}^T\boldsymbol{y}$, and function $f_P(w)$ can be expressed as

$$f_P(w) = ||PM_{w^R}\boldsymbol{y}||_1 , \tag{2}$$

which is analogous to the form appearing in the definition of quantum automata.

A *measure-once quantum automaton* (MO-QFA) (see also [10]) over an alphabet Σ ($n = |\Sigma|$) is a triplet $(P, \{U_a \mid a \in \Sigma\}, \boldsymbol{y})$, where $\boldsymbol{y} \in \mathbb{C}^n$ is an *initial amplitude vector* of unit L_2-norm, each $U_a \in \mathbb{C}^{n\times n}$ is a unitary matrix, and $P \in \mathbb{C}^{n\times n}$ is the *measurement projection*. A quantum automaton Q defines a function $f_Q : \Sigma \to [0, 1]$ by

$$f_Q(w) = ||PU_{w^R}\boldsymbol{y}||^2 . \tag{3}$$

We also define *integer-weighted automata* ($\mathbb{Z}$FA) (see [5] for details) exactly as we defined PFA, but instead of initial distribution and Markov matrices, we have an initial vector in $\mathbb{Z}^n$ and matrices with integer entries. As PFAs, $\mathbb{Z}$FAs could also be defined by the means of transition function $\delta : Q \times \Sigma \times Q \to \mathbb{Z}$. A $\mathbb{Z}$FA $Z = (\boldsymbol{x}, \{M_a \mid a \in \Sigma\}, \boldsymbol{y})$ defines a function $f_Z : \Sigma^* \to \mathbb{Z}$ by

$$f_Z(w) = \boldsymbol{x}^T M_{w^R}\boldsymbol{y}.$$

For PFA and MO-QFA and a fixed $\lambda \in [0, 1]$ we define *cut-point languages* and *strict cut-point languages*: For any $\lambda \in [0, 1]$ and automaton A,

$$L_{\geq\lambda}(A) = \{w \in \Sigma^* \mid f_A(w) \geq \lambda\},$$

and

$$L_{>\lambda}(A) = \{w \in \Sigma^* \mid f_A(w) > \lambda\}.$$

It is known that there are cut-point languages that are not regular [11].

In this article we study both problems $L_{\geq\lambda}(A) = \emptyset$? and $L_{>\lambda}(A) = \emptyset$?, and construct PFAs and MO-QFAs having an undecidable emptiness problem of smaller size than found previously.

As in [1] and [2], we prove the undecidability results by showing that for a given instance $\mathcal{I}$ of *Post Correspondence Problem* (PCP) (see [8]), one can construct an automaton that accepts words if and only $\mathcal{I}$ has a solution. The following theorem [9] is the basis of our constructions:

Theorem 1. *For $k \geq 7$, it is undecidable whether an instance $\mathcal{I} = \{(u_1, v_1), \dots, (u_k, v_k)\}$ of PCP has a solution $u_{i_1}u_{i_2}\dots u_{i_n} = v_{i_1}v_{i_2}\dots v_{i_n}$.*

We will also use the following variant of PCP [4], [6]:

Theorem 2. *There are instances $\mathcal{I} = \{(u_1, v_1), \ldots, (u_k, v_k)\}$ of PCP such that all minimal solutions*[1] $u_{i_1}u_{i_2}\ldots u_{i_n} = v_{i_1}v_{i_2}\ldots v_{i_n}$ *are of form $i_1 = 1$, $i_n = k$, and $i_2 \ldots i_{n-1} \in \{2, \ldots, k-1\}^+$. For $k \geq 7$, it is undecidable whether such a solution exists.*

The instances of the above theorem are called *Claus* instances. In fact, all undecidability proofs of PCP known to author are for Claus instances.

3 Probabilistic automata

Let $\mathcal{I} = \{(u_1, v_1), \ldots, (u_k, v_k)\}$ be an instance of the PCP. We can assume that u_i and v_i are over a binary alphabet $\Sigma = \{1, 2\}$, and construct a PFA P such that for some $\lambda \in [0, 1]$ $L_{>\lambda}(P) \neq \emptyset$ if and only if $\mathcal{I}$ has a solution. We also explain how to modify the construction to get a PFA P' such that $L_{\geq\lambda}(P') \neq \emptyset$ if and only if $\mathcal{I}$ has a solution.

Step 1. (Embedding $\mathcal{I}$ in integer matrices) Let $\sigma : \Sigma^* \to \mathbb{N} = \{1, 2, 3 \ldots\}$ be the bijection defined as $\sigma(i_1 i_2 \ldots i_n) = \sum_{j=1}^{n} i_j 2^{n-j}$.

The first target is to find, for some d, an embedding $\gamma : \Sigma^* \times \Sigma^* \mapsto \mathbb{Z}^{d\times d}$ and (column) vectors $\boldsymbol{x}$, $\boldsymbol{y} \in \mathbb{Z}^d$ such that $\boldsymbol{x}^T\gamma(u, v)\boldsymbol{y}$ includes expression $(\sigma(u) - \sigma(v))^2$.

Obviously with

$$\gamma_0(u, v) = \begin{pmatrix} 2^{|u|} & 0 & 0 \\ 0 & 2^{|v|} & 0 \\ \sigma(u) & \sigma(v) & 1 \end{pmatrix},$$

$\boldsymbol{x} = (0, 0, 1)^T$ and, $\boldsymbol{y} = (1, -1, 0)^T$ we have $\boldsymbol{x}^T\gamma_0(u, v)\boldsymbol{y} = \sigma(u) - \sigma(v)$, hence the tensor products satisfy

$$(\boldsymbol{x}^T \otimes \boldsymbol{x}^T)(\gamma_0(u, v) \otimes \gamma_0(u, v))(\boldsymbol{y} \otimes \boldsymbol{y}) = (\boldsymbol{x}^T\gamma_0(u, v)\boldsymbol{y})^2 = (\sigma(u) - \sigma(v))^2.$$

However, the dimension of the matrix $\gamma_0(u, v) \otimes \gamma_0(u, v)$ is 9. A construction with a smaller dimension was given in [1]:

$$\gamma(u, v) = \begin{pmatrix} 2^{2|u|} & 0 & 0 & 0 & 0 & 0 \\ 0 & 2^{|uv|} & 0 & 0 & 0 & 0 \\ 0 & 0 & 2^{2|v|} & 0 & 0 & 0 \\ \sigma(u)2^{|u|} & \sigma(v)2^{|u|} & 0 & 2^{|u|} & 0 & 0 \\ 0 & \sigma(u)2^{|v|} & \sigma(v)2^{|v|} & 0 & 2^{|v|} & 0 \\ \sigma(u)^2 & 2\sigma(u)\sigma(v) & \sigma(v)^2 & 2\sigma(u) & 2\sigma(v) & 1 \end{pmatrix}. \quad (4)$$

It is straightforward to see that $\gamma(u_1, v_1)\gamma(u_2, v_2) = \gamma(u_1u_2, v_1v_2)$, and by choosing $\boldsymbol{x}_1 = (0, 0, 0, 0, 0, 1)^T$, and $\boldsymbol{y}_1 = (-1, 1, -1, 0, 0, 1)^T$ we get $\boldsymbol{x}_1^T\gamma(u, v)\boldsymbol{y}_1 = 1 - (\sigma(u) - \sigma(v))^2$. Hence $\boldsymbol{x}_1^T\gamma(u, v)\boldsymbol{y}_1 \leq 1$ always, and $\boldsymbol{x}_1^T\gamma(u, v)\boldsymbol{y}_1 = 1$ if and only if $u = v$.

[1] A solution to PCP is *minimal* if it is not a concatenation of two solutions.

We define $A_i = \gamma(u_i, v_i)$ for each $i \in \{1, \ldots, k\}$. Clearly $\mathcal{I}$ has a solution if and only if $\boldsymbol{x}_1^T A_{j_1} A_{j_2} \ldots A_{j_n} \boldsymbol{y}_1 = 1$ for some $j_1 j_2 \ldots j_n \in \{1, \ldots, k\}^+$, and $\boldsymbol{x}_1^T A_{j_1} A_{j_2} \ldots A_{j_n} \boldsymbol{y}_1 \leq 1$ anyway. As before, we denote $A_{j_1} A_{j_2} \ldots A_{j_n} = A_w$, where $w = j_1 j_2 \ldots j_n$, and A_ϵ is defined to be the identity matrix. Thus $\mathcal{I}$ has a solution if and only if $\boldsymbol{x}_1^T A_w \boldsymbol{y}_1 > 0$ for some $w \in \{1, \ldots, k\}^+$ ($\boldsymbol{x}_1^T \boldsymbol{y}_1 = 1$).

Remark 1. Notice that $(\boldsymbol{x}_1, \{A_1, \ldots, A_k\}, \boldsymbol{y}_1)$ is a $\mathbb{Z}$FA with 6 states, over an alphabet of k symbols. Hence the problem "is $f_Z(w) > 0$ for some nonempty word w"? is undecidable for integer-weighted automata.

Step 2. (Reducing the number of matrices) We can assume that $\mathcal{I}$ is a Claus instance. Since all solutions $u_{i_1} \ldots u_{i_n} = v_{i_1} \ldots v_{i_n}$ of Claus instances have $i_1 = 1$, $i_n = k$, and $i_2 \ldots i_{n-k} \in \{2, \ldots, k-1\}^+$ we can define $\boldsymbol{x}_2 = (\boldsymbol{x}_1^T A_1)^T$ and $\boldsymbol{y}_2 = A_k \boldsymbol{y}_1$, $B_1 = A_2$, …, $B_{k-2} = A_{k-1}$ to get another $\mathbb{Z}$FA $Z = (\boldsymbol{x}_2, \{B_1, \ldots, B_{k-2}\}, \boldsymbol{y}_2)$. Notice that Z has 6 states and is over an alphabet of $k-2$ symbols. Moreover, $f_Z(w) = \boldsymbol{x}_2^T B_w \boldsymbol{y}_2 = \boldsymbol{x}_1^T A_1 B_w A_k \boldsymbol{y}_1$, so $f_Z(w) > 0$ for some nonempty word w if and only is $\mathcal{I}$ has a solution.

Step 3. (Reducing the number of matrices to 2) Let us denote the transition function of the $\mathbb{Z}$FA Z introduced in Step 2 by $\delta(q_i, c, q_j) = (B_c)_{ji}$ for each $i, j \in \{1, \ldots, 6\}$ and $c \in \{1, \ldots, k-2\}$. To find two matrices C_1 and C_2 that will encode the essential properties of $B_1, \ldots, B_{k-2}$, we encode the $k-2$ input symbols of automaton Z into binary strings and add some extra states adjacent to each state of Z that will decode the binary strings back to symbols in set $\{1, \ldots, k-2\}$. However, the state q_6 needs no decoder, since according to (4) we have, for each c, $\delta(q_6, c, q_i) = 1$, if $i = 6$ and 0 otherwise.

We will use an injective morphism $\psi : \{1, \ldots, k-2\}^* \to \{1, 2\}^*$ defined as $\psi(i) = 1^{i-1}2$ for $i < k-2$, and $\psi(k-2) = 1^{k-3}$. Now if $\{q_1, \ldots, q_6\}$ is the state set of automaton Z, we define a new automaton Z' with states $q_{i,j}$, where $i \in \{1, \ldots, 5\}$ and $j \in \{1, \ldots, k-3\}$, plus state $q_{6,1}$, so we have $5(k-3)+1 = 5k-14$ states altogether for Z'.

The transition function δ' of the automaton Z' will be defined as (for $(i, r) \neq (6, 6)$)

$$\delta'(q_{i,j}, 1, q_{r,s}) = \begin{cases} \delta(q_i, k, q_r), & \text{if } j = k-3, \text{ and } s = 1, \\ 1, & \text{if } i = r < 5 \text{ and } j+1 = s < k-2, \\ 0 & \text{otherwise.} \end{cases}$$

$$\delta'(q_{i,j}, 2, q_{r,s}) = \begin{cases} \delta(q_i, j, q_r) & \text{if } s = 1, \\ 0 & \text{otherwise,} \end{cases}$$

$\delta'(q_{6,1}, c, q_{6,1}) = 1$ for $c \in \{1, 2\}$, and $\delta'(q_{i,j}, c, q_{r,s}) = 0$ for the cases not defined before. See Figure 3 for a graphical representation of automaton Z'.

Finally we enumerate all $5k-14$ states $q_{i,j}$ in some way, and define vector $\boldsymbol{x}_3 \in \mathbb{Z}^{5k-14}$ such that all its coordinates are zero, except each corresponding to state $q_{i,1}$ ($i \in \{1, \ldots, 6\}$), whose value is chosen to be $(\boldsymbol{x}_2)_i$. Vector $\boldsymbol{y}_3 \in \mathbb{Z}^{5k-14}$ is defined analogously. We denote the transition matrices of this new automaton by C_1 and C_2. The dimension of the matrices is $5k-14$.

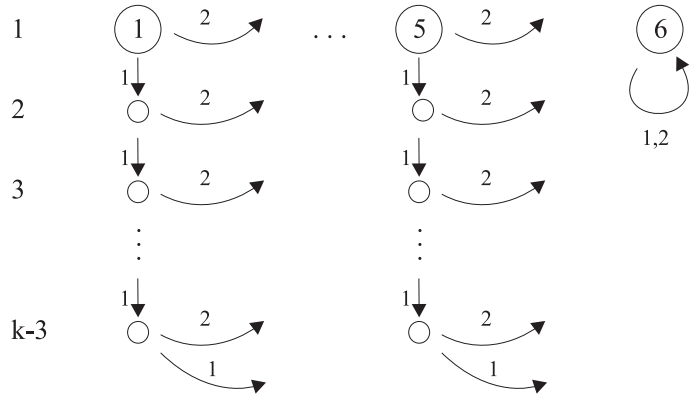

Figure 1: Automaton Z'. The weights of the arrows are not shown in the picture.

With these definitions, $\boldsymbol{x}_3^T C_{\psi(w)^R} \boldsymbol{y}_3 = \boldsymbol{x}_2^T B_{w^R} \boldsymbol{y}_2$ for each $w \in \{1, \ldots, k-2\}^*$, and $\boldsymbol{x}_3^T C_{w^R} \boldsymbol{y}_3 = 0$, if $w \in \Sigma^+$ is not in the image of ψ. Hence $\mathcal{I}$ has a solution if and only if there is a $w \in \Sigma^+$ such that $\boldsymbol{x}_3^T C_w \boldsymbol{y}_3 = 1$. Notice again that $\boldsymbol{x}_3^T C_w \boldsymbol{y}_3 \leq 1$ for each $w \in \Sigma^*$, so $\mathcal{I}$ has a solution if and only if $\boldsymbol{x}_3^T B_w \boldsymbol{y}_3 > 0$ for some $w \in \Sigma^+$.

Step 4. (Changing the initial and final vectors into probability distributions) For $i \in \{1, 2\}$ let

$$D_i = \begin{pmatrix} 0 & 0 & 0 \\ C_i \boldsymbol{y}_3 & C_i & 0 \\ \boldsymbol{x}_3^T C_i \boldsymbol{y}_3 & \boldsymbol{x}_3^T C_i & 0 \end{pmatrix},$$

and notice that

$$\begin{aligned} D_u D_v &= \begin{pmatrix} 0 & 0 & 0 \\ C_u \boldsymbol{y}_3 & C_u & 0 \\ \boldsymbol{x}_3^T C_u \boldsymbol{y}_3 & \boldsymbol{x}_3^T C_u & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 \\ C_v \boldsymbol{y}_3 & C_v & 0 \\ \boldsymbol{x}_3^T C_v \boldsymbol{y}_3 & \boldsymbol{x}_3^T C_v & 0 \end{pmatrix} \\ &= \begin{pmatrix} 0 & 0 & 0 \\ C_{uv} \boldsymbol{y}_3 & C_{uv} & 0 \\ \boldsymbol{x}_3^T C_{uv} \boldsymbol{y}_3 & \boldsymbol{x}_3^T C_{uv} & 0 \end{pmatrix} = D_{uv}. \end{aligned}$$

Hence with $\boldsymbol{x}_4 = (0, \ldots, 0, 1)^T$, $\boldsymbol{y}_4 = (1, 0, \ldots, 0)^T$, we have clearly $\boldsymbol{x}_4^T D_w \boldsymbol{y}_4 = \boldsymbol{x}_3^T C_w \boldsymbol{y}_3$ if $w \neq \epsilon$, and $\boldsymbol{x}_4^T \boldsymbol{y}_4 = 0$. Now each D_i is a $(5k-12) \times (5k-12)$-matrix and $\boldsymbol{x}_4$ and $\boldsymbol{y}_4$ are probability distributions. Furthermore, $\mathcal{I}$ has a solution if and only if $\boldsymbol{x}_4^T D_w \boldsymbol{y}_4 > 0$ for some $w \in \Sigma^*$.

Step 5. (Embedding the matrices in stochastic ones, Part 1) This and the following part of the construction is due to P. Turakainen [14]. Define $(5k-10) \times (5k-10)$-matrices E_1 and E_2 by

$$E_i = \begin{pmatrix} 0 & 0 & 0 \\ \boldsymbol{t}_i & D_i & 0 \\ s_i & \boldsymbol{r}_i^T & 0 \end{pmatrix},$$

where $\boldsymbol{t}_i$, $\boldsymbol{r}_i$, and s_i are chosen such that the row and the column sums of E_i are zero. Notice that the sum of coordinates of $\boldsymbol{t}_i$ and $\boldsymbol{r}_i$ are equal (both equal to $-\sum_r \sum_s (D_i)_{rs}$), hence s_i is definable.

It is easy to verify that

$$E_u E_v = \begin{pmatrix} 0 & 0 & 0 \\ \boldsymbol{t}_u & D_u & 0 \\ s_u & \boldsymbol{r}_u^T & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 \\ \boldsymbol{t}_v & D_v & 0 \\ s_v & \boldsymbol{r}_v^T & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ \boldsymbol{t}_{uv} & D_{uv} & 0 \\ s_{uv} & \boldsymbol{r}_{uv}^T & 0 \end{pmatrix} = E_{uv},$$

and that also the row and column sums of E_{uv} are zero.

Let $\boldsymbol{x}_5 = (0, \boldsymbol{x}_4^T, 0)^T$ and $\boldsymbol{y}_5 = (0, \boldsymbol{y}_4^T, 0)^T$. Then $\boldsymbol{x}_5^T E_w \boldsymbol{y}_5 = \boldsymbol{x}_4^T D_w \boldsymbol{y}_4$ and hence $\boldsymbol{x}_5^T E_w \boldsymbol{y}_5 > 0$ for some word $w \in \Sigma^*$ if and only if $\mathcal{I}$ has a solution.

Step 6. (Embedding the matrices in stochastic ones, Part 2) Let $\mathbf{1}$ be an $n \times n$-matrix with all entries 1. Clearly $\mathbf{1}^2 = n\mathbf{1}$, which implies that $\mathbf{1}^i = n^{i-1}\mathbf{1}$ for $i \geq 1$. In the continuation, n will be chosen as $n = 5k - 10$. Since the row and columns sums of each E_w ($w \neq \epsilon$) are zero, we have $E_w\mathbf{1} = \mathbf{1}E_w = 0$, whenever $w \neq \epsilon$.

Define F_1 and F_2 by $F_i = E_i + c\mathbf{1}$, where $c \in \mathbb{N}$ is chosen so large that each entry of F_1 and F_2 is positive. Then the sum of the entries of F_i in each column (and row) is equal to $c(5k-10)$, and consequently matrices

$$G_i = \frac{1}{c(5k-10)} F_i$$

are (doubly) stochastic. Since $E_i\mathbf{1} = \mathbf{1}E_i = 0$, we have

$$F_w = E_w + (c\mathbf{1})^{|w|} = E_w + c^{|w|}(5k-10)^{|w|-1}\mathbf{1}$$

whenever $w \neq 1$, which implies that

$$G_w = \frac{1}{(c(5k-10))^{|w|}} E_w + \frac{1}{5k-10}\mathbf{1}.$$

Now letting $\boldsymbol{x}_6 = \boldsymbol{x}_5$, $\boldsymbol{y}_6 = \boldsymbol{y}_5$ we get (to compute $\boldsymbol{x}_6^T\mathbf{1}\boldsymbol{y}_6$, recall that $\boldsymbol{x}_6$ and $\boldsymbol{y}_6$ have exactly one coordinate equal to 1, and all other coordinates 0).

$$\boldsymbol{x}_6^T G_w \boldsymbol{y}_6 = \frac{1}{(c(5k-10))^{|w|}} \boldsymbol{x}_5^T E_w \boldsymbol{y}_5 + \frac{1}{5k-10}.$$

Hence $\mathcal{I}$ has a solution if and only if there is $w \in \Sigma^*$ such that

$$\boldsymbol{x}_6^T G_w \boldsymbol{y}_6 > \frac{1}{5k-10},$$

and $(\boldsymbol{x}_6, \{G_1, G_2\}, \boldsymbol{y}_6)$ is a ($5k-10$ -state) PFA P such that the $L_{>\frac{1}{5k-10}}(P) \neq \emptyset$ if and only if $\mathcal{I}$ has a solution.

Remark 2. According to Theorem 2, we conclude that the problem $L_{>\lambda}(P)$ is undecidable for a $5 \cdot 7 - 10 = 25$-state PFA over a binary alphabet.

Modification: Step 3.5. We can define matrices

$$C_i' = \begin{pmatrix} C_i & 0 \\ 0 & 1 \end{pmatrix}$$

and $\boldsymbol{x}_3' = (\boldsymbol{x}_3^T, 1)^T$, $\boldsymbol{y}_3' = (\boldsymbol{y}_3^T, -1)^T$ to notice that $\boldsymbol{x}_3'^T C_w' \boldsymbol{y}_3' = \boldsymbol{x}_3^T C_w \boldsymbol{y}_3 - 1$. Hence $\boldsymbol{x}_3'^T C_w' \boldsymbol{y}_3' \geq 0$ if and only if $\mathcal{I}$ has a solution $w \in \Sigma^+$. Then the construction above gives an automaton P' with $5k-9$ states such that $L_{\geq\lambda}(P') \neq \emptyset$ if and only if $\mathcal{I}$ has a solution.

4 Quantum Automata

Lemma 1. *Let* $U_1 = \frac{1}{5}\begin{pmatrix} 3 & -4 \\ 4 & 3 \end{pmatrix}$, $U_2 = \frac{1}{5}\begin{pmatrix} 3 & 4i \\ 4i & 3 \end{pmatrix}$, *and* $\boldsymbol{y} = (1,0)^T$. *If*

$$U_{c_1} \cdot \ldots \cdot U_{c_r}\boldsymbol{y} = U_{d_1} \cdot \ldots \cdot U_{d_s}\boldsymbol{y}, \tag{5}$$

where $c_1 \ldots c_r$, $d_1 \ldots d_s \in \{1,2\}^*$, *then* $r = s$ *and* $c_i = d_i$ *for each* i.

Proof. We say that a product

$$T_1 \cdot \ldots \cdot T_r, \tag{6}$$

where each $T_i \in \{U_1, U_1^{-1}, U_2, U_2^{-1}\}$ is *reduced*, if $T_i \in \{U_1, U_1^{-1}\} \implies T_{i+1} \in \{U_2, U_2^{-1}\}$ and $T_i \in \{U_2, U_2^{-1}\} \implies T_{i+1} \in \{U_1, U_1^{-1}\}$. Following the idea of [12] we will show by induction on r that each reduced product (6), where $r > 0$, is of form

$$\frac{1}{5^r}\begin{pmatrix} a_r & * \\ b_r & * \end{pmatrix}, \tag{7}$$

where a_r, $b_r \in \mathbb{Z}[i]$ and a_r is not divisible by 5. To start with, the case $r = 1$ is trivial and $r = 2$ can be treated by straightforward computation.

Now we assume that the claim holds for reduced products shorter than r and divide the induction step for reduced product T of length $r+1$ into several cases:

1. $T = U_2^{\epsilon_2}U_1^{\epsilon_1}T'$, where $\epsilon_1, \epsilon_2 \in \{-1, 1\}$,
2. $T = U_1^{\epsilon_1}U_2^{\epsilon_2}T'$, where $\epsilon_1, \epsilon_2 \in \{-1, 1\}$,
3. $T = U_1^{\epsilon}U_1^{\epsilon}T'$, where $\epsilon \in \{-1, 1\}$,
4. $T = U_2^{\epsilon}U_2^{\epsilon}T'$, where $\epsilon \in \{-1, 1\}$.

Multiplying (7) from left by $U_1^{\epsilon_1}$ and $U_2^{\epsilon_2}$ give recurrences

$$\begin{pmatrix} a_{r+1} \\ b_{r+1} \end{pmatrix} = \begin{pmatrix} 3a_r - \epsilon_1 4b_r \\ \epsilon_1 4a_r + 3b_r \end{pmatrix} \text{ and } \begin{pmatrix} a_{r+1} \\ b_{r+1} \end{pmatrix} = \begin{pmatrix} 3a_r + \epsilon_2 4ib_r \\ \epsilon_2 4ia_r + 3b_r \end{pmatrix},$$

respectively, and hence we can find out that in case 1

$$\begin{aligned} a_{r+1} &= 3a_r - \epsilon_1 4b_r = 3a_r - \epsilon_1 4(\epsilon_2 4ia_{r-1} + 3b_{r-1}) \\ &= 3a_r - \epsilon_1\epsilon_2 16ia_{r-1} - 12\epsilon_1 b_{r-1} \\ &= 3a_r - \epsilon_1\epsilon_2 25ia_{r-1} + \epsilon_1\epsilon_2 9ia_{r-1} - 12\epsilon_1 b_{r-1} \\ &= 3a_r - \epsilon_1\epsilon_2 25ia_{r-1} + \epsilon_1\epsilon_2 3i(3a_{r-1} + 4i\epsilon_2 b_{r-1}) \\ &= (3 + \epsilon_1\epsilon_2 3i)a_r - \epsilon_1\epsilon_2 25ia_{r-1}. \end{aligned}$$

In the rest of the cases we have $a_{r+1} = (3 - \epsilon_1\epsilon_2 3i)a_r + \epsilon_1\epsilon_2 25ia_{r-1}$, $a_{r+1} = 6a_r - 25a_{r-1}$, and $a_{r+1} = 6a_r - 25a_{r-1}$, respectively. In all the cases we can use the induction assumption $5 \nmid a_r$ to get $5 \nmid a_{r+1}$.

Denoting $u = c_1 \ldots c_r$, $v = d_1 \ldots d_s \in \Sigma^*$ We can write equation (5) in a more compact way as

$$U_u\boldsymbol{y} = U_v\boldsymbol{y}, \tag{8}$$

where $|u| = r$ and $|v| = s$. If (8) holds for some $u \neq v$, we can assume without loss of generality that $U_u = U_1 U_{u'}$ and $U_v = U_2 U_{v'}$. Thus we get

$$U_{v'}^{-1} U_2^{-1} U_1 U_{u'} \boldsymbol{y} = \boldsymbol{y}, \tag{9}$$

where $U_{v'}^{-1} U_2^{-1} U_1 U_{u'}$ is a reduced product of length $r + s \geq 1$, and we can write (9) as

$$\begin{pmatrix} a_{r+s} \\ b_{r+s} \end{pmatrix} = \begin{pmatrix} 5^{r+s} \\ 0 \end{pmatrix}.$$

This contradicts the previously observed fact that $5 \nmid a_{r+s}$. Notice that the same contradiction can be obtained also if one of the words u or v is empty. □

Corollary 1. *$U_u \boldsymbol{y} \neq -U_v \boldsymbol{y}$ for all words u and v.*

Proof. If $U_u \boldsymbol{y} = -U_v \boldsymbol{y}$, then clearly $u \neq v$, for otherwise we would have $\boldsymbol{y} = -\boldsymbol{y}$. But if $U_u \boldsymbol{y} = -U_v \boldsymbol{y}$ for $u \neq v$, we would have, as in the previous proof, a non-trivial reduced product with left upper corner divisible with 5, again a contradiction. □

Corollary 2. *The semigroup generated by unitary matrices U_1 and U_2 is free.*

Proof. If $U_u = U_v$, then also $U_u \boldsymbol{y} = U_v \boldsymbol{y}$, and the previous lemma implies that $u = v$. □

For $u, v \in \Sigma^*$ we define

$$\gamma(u, v) = \frac{1}{2} \begin{pmatrix} U_u + U_v & U_u - U_v \\ U_u - U_v & U_u + U_v \end{pmatrix} \tag{10}$$

If is a straightforward task to verify that $\gamma(u, v)$ is a unitary matrix, and that $\gamma(u_1, v_1)\gamma(u_2, v_2) = \gamma(u_1 u_2, v_1 v_2)$. Moreover,

$$\gamma(u, v)(1, 0, 0, 0)^T = \frac{1}{2} \begin{pmatrix} (U_u + U_v)\boldsymbol{y} \\ (U_u - U_v)\boldsymbol{y} \end{pmatrix}. \tag{11}$$

By Lemma 1, $u = v$ if and only if the two last coordinates of (11) are zero. Hence if we denote $\boldsymbol{y}_1 = (1, 0, 0, 0)^T$ and

$$P_1 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

then P_1 is the projection onto the last two coordinates and we have

$$||P_1 \gamma(u, v) \boldsymbol{y}_1||^2 = 0$$

if and only if $u = v$.

Step 1. (Embedding an instance of PCP in unitary matrices) Let again $\mathcal{I} = \{(u_1, v_1), \ldots, (u_k, v_k)\}$ be an instance of PCP, u_i and v_i over a binary alphabet

$\Sigma = \{1, 2\}$. We define $A_i = \gamma(u_i, v_i)$ for each $i \in \{1, \ldots, k\}$. Hence $\mathcal{I}$ has a solution if and only if there exists $w \in \{1, \ldots, k\}^+$ such that

$$||P_1 A_w \boldsymbol{y}_1||^2 = 0.$$

Step 2. (Getting rid of $P_1\boldsymbol{y}_1 = \mathbf{0}$ and reducing the number of matrices) We assume that $\mathcal{I} = \{(u_1, v_1), \ldots, (u_k, v_k)\}$ is a Claus instance, i.e., an instance of PCP such that all solutions $u_{i_1} \ldots u_{i_n} = v_{i_1} \ldots v_{i_n}$ are of form $i_1 = 1$, $i_n = k$, and $i_2 \ldots i_{n-1} \in \{2, \ldots, k-1\}^+$.

We define $B_1 = A_2$, …, $B_{k-2} = A_{k-1}$. A new initial vector is defined as $\boldsymbol{y}_2 = A_k\boldsymbol{y}_1$, and a new final projection is defined as $P_2 = A_1^{-1}P_1A_1$. Since A_1 and A_k are unitary, it is easy to see that $||\boldsymbol{y}_2|| = 1$, and that P_2 is a projection. Since also A_1^{-1} is unitary, we have

$$||P_2 B_w \boldsymbol{y}_2||^2 = \left|\left|A_1^{-1}P_1A_1B_wA_k\boldsymbol{y}_1\right|\right|^2 = ||P_1A_1B_wA_k\boldsymbol{y}_1||^2,$$

so $||P_2B_w\boldsymbol{y}_2||^2 = 0$ if and only if $\mathcal{I}$ has a solution. Moreover, $||P_2\boldsymbol{y}_2||^2 = ||P_1A_1A_k\boldsymbol{y}|| \neq 0$, since by the assumption, $u_1u_k \neq v_1v_k$.

Step 3. (Reducing the number of matrices to 2) Define

$$C_1 = \begin{pmatrix} B_1 & 0 & \cdots & 0 \\ 0 & B_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & B_{k-2} \end{pmatrix} \text{ and } C_2 = \begin{pmatrix} 0 & I & 0 & \cdots & 0 \\ 0 & 0 & I & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & I \\ I & 0 & 0 & \cdots & 0 \end{pmatrix}.$$

C_1 and C_2 are clearly unitary $4(k-2) \times 4(k-2)$-matrices. Let also

$$P_3 = \begin{pmatrix} P_2 & 0 & \cdots & 0 \\ 0 & P_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & P_2 \end{pmatrix}$$

and $\boldsymbol{y}_3 = (\boldsymbol{y}_2^T, \mathbf{0}^T, \ldots, \mathbf{0}^T)^T$.

It is easy to verify that $C_2C_1C_2^{-1} = \mathrm{diag}(B_2, \ldots, B_{k-2}, B_1)$, which implies that $C_2^iC_1C_2^{-i} = \mathrm{diag}(B_{i+1}, \ldots, B_{k-2}, B_1, \ldots, B_i)$. Now $C_2^{-1} = C_2^{k-3}$, so the inverse can be always replaced with a positive power, and hence for any word $w \in \{1, \ldots, k-2\}^*$ there is a word $w' \in \Sigma^*$ such that $C_{w'} = \mathrm{diag}(B_w, \ldots)$.

On the other hand, both C_1 and C_2 are block matrices with exactly one nonzero block in each row and column. The said property always inherits to the products formed of C_1 and C_2, and hence C_w for any $w \in \Sigma^*$ is a block matrix that has at most one nonzero block in each row and column, but any nonzero block in C_w is of form $B_{w'}$, where $w' \in \{1, \ldots, k-2\}^*$.

Hence $\mathcal{I}$ has a solution if and only if $||P_3C_w\boldsymbol{y}_3||^2 = 0$ for some $w \in \Sigma^*$. Notice carefully that $||P_2\boldsymbol{y}_2||^2 \neq 0$ implies that also $||P_3\boldsymbol{y}_3||^2 \neq 0$. This is a very important feature in this step, since if we would have $||P_2\boldsymbol{y}_2||^2 = 0$ (as would be

the case without Step 2), the new automaton would always allow words of form $2^{r(k-2)}$, since $\left|\left|P_3 C_2^{r(k-2)} \boldsymbol{y}_3\right|\right|^2 = ||P_3 \boldsymbol{y}_3||^2$ for each $r \in \mathbb{Z}$.

Step 4. (Setting the threshold) Notice that since $I - P_3$ is a projection orthogonal to P_3, we have

$$||C_w \boldsymbol{y}_3||^2 = ||P_3 C_w \boldsymbol{y}_3||^2 + ||(I - P_3) C_w \boldsymbol{y}_3||^2 ,$$

and since $||C_w \boldsymbol{y}_3|| = 1$ always (each C_w is unitary), we have

$$||(I - P_3) C_w \boldsymbol{y}_3||^2 \leq 1$$

with equality if and only if $\mathcal{I}$ has a solution. Therefore,

$$||(I - P_3) C_w \boldsymbol{y}_3||^2 \geq 1$$

for some $w \in \Sigma^*$ if and only if $\mathcal{I}$ has a solution.

Let $0 < \lambda < 1$ and define, for each $i \in \Sigma$,

$$D_i = \begin{pmatrix} C_i & 0 \\ 0 & 1 \end{pmatrix}.$$

Let also $\boldsymbol{y}_4 = (\sqrt{\lambda}\boldsymbol{y}_3^T, \sqrt{1-\lambda})^T \in \mathbb{R}^{4k-7}$, and

$$P_4 = \begin{pmatrix} I - P_3 & 0 \\ 0 & 0 \end{pmatrix}.$$

Now D_1 and D_2 are $(4k-7) \times (4k-7)$-matrices, and

$$||P_4 D_w \boldsymbol{y}_4||^2 = \left|\left|\sqrt{\lambda}(I - P_3) C_w \boldsymbol{y}_4\right|\right|^2 = \lambda(1 - ||P_3 C_w \boldsymbol{y}_3||^2).$$

Thus $||P_4 D_w \boldsymbol{y}_4||^2 \geq \lambda$ for some word $w \in \Sigma^*$ if and only if $\mathcal{I}$ has a solution.

If an automata with defining constants in $\mathbb{Q}[i]$ is required, one can choose $\lambda = \frac{9}{25}$, for example.

From the construction it follows that $Q = (P_4, \{D_1, D_2\}, \boldsymbol{y}_4)$ is MO-QFA such that $L_{\geq\lambda}(Q) \neq \emptyset$ if and only if $\mathcal{I}$ has a solution.

Remark 3. Letting $k = 7$ we see that the problem studied is undecidable for a 21-state MO-QFA over a binary alphabet. Skipping Step 3 we could as well obtain the undecidability result for a 4-state MO-QFA over a 5-symbol alphabet.

Acknowledgement

Thanks to Dr. Vesa Halava for pointing out Theorem 2 and to Dr. Matti Soittola for reviewing earlier versions of this article and pointing out the usefulness of [12] when proving Lemma 1.

References

[1] Vincent D. Blondel and Vincent Canterini: *Undecidable Problems for Probabilistic Automata of Fixed Dimension*. Theory of Computing Systems 36, 231–245 (2003).

[2] Vincent D. Blondel, Emmanuel Jeandel, Pascal Koiran, and Natacha Portier: *Decidable and Undecidable Problems About Quantum Automata*. SIAM Journal of Computing 34:6, 1464–1473 (2005).

[3] Harm Derksen, Emmanuel Jeandel, and Pascal Koiran: *Quantum automata and algebraic groups*. Journal of Symbolic Computation 39, pp. 357–371 (2005).

[4] V. Claus: *Some remarks on $PCP(k)$ and related problems*. Bulletin of EATCS 12, 54–61 (1980).

[5] Samuel Eilenberg: *Automata, languages, and machines Vol. A*. Academic Press (1974).

[6] Vesa Halava, Tero Harju, and Mika Hirvensalo: *Lowering the Undecidability Bounds for Integer Matrices Using Claus Instances of the PCP*. A manuscript (2006).

[7] Mika Hirvensalo: *Quantum Computing, 2nd edition*. Springer (2003).

[8] Tero Harju and Juhani Karhumäki: *Morphishms*. In G. Rozenberg and A. Salomaa (eds): *Handbook of Formal Languages*, Springer (1997).

[9] Y. Matiyasevich and G. Sénizergues: *Decision problems for semi-Thue systems with a few rules*. Theoretical Computer Science 330:1, pp. 145–169 (2005).

[10] Cristopher Moore and James P. Crutchfield: *Quantum Automata and Quantum Grammars*. Theoretical Computer Science 237, 275–306 (2000).

[11] Azaria Paz: *Introduction to probabilistic automata*. Academic Press (1971).

[12] S. Świerczkowski: On a free group of rotations of the euclidean space. Indagationes Mathematicae 20, 376–378 (1958).

[13] James Renegar: *On the Complexity and Geometry of the First-order Theory of the Reals. Parts I, II, and III*. Journal of Symbolic Computation 13(3), 255–352 (1992).

[14] Paavo Turakainen: *Generalized automata and stochastic languages*. Proceedings of American Mathematical Society 21, 303–209 (1969).

[15] Yu Sheng: *Regular Languages*. In G. Rozenberg and A. Salomaa (eds): *Handbook of Formal Languages*, Springer (1997).

Lemminkäisenkatu 14 A, 20520 Turku, Finland | www.tucs.fi

University of Turku
- Department of Information Technology
- Department of Mathematics

Åbo Akademi University
- Department of Computer Science
- Institute for Advanced Management Systems Research

Turku School of Economics and Business Administration
- Institute of Information Systems Sciences

ISBN 952-12-1726-X
ISSN 1239-1891