SIAM J. COMPUT.
Vol. 34, No. 6, pp. 1464–1473

DECIDABLE AND UNDECIDABLE PROBLEMS ABOUT QUANTUM AUTOMATA*

VINCENT D. BLONDEL†, EMMANUEL JEANDEL‡, PASCAL KOIRAN‡, AND NATACHA PORTIER‡

Abstract. We study the following decision problem: is the language recognized by a quantum finite automaton empty or nonempty? We prove that this problem is decidable or undecidable depending on whether recognition is defined by strict or nonstrict thresholds. This result is in contrast with the corresponding situation for probabilistic finite automata, for which it is known that strict and nonstrict thresholds both lead to undecidable problems.

Key words. quantum automata, probabilistic automata, undecidable problems, algebraic groups

AMS subject classifications. 81P68, 68Q45

DOI. 10.1137/S0097539703425861

1. Introduction. In this paper, we provide decidability and undecidability proofs for two problems associated with quantum finite automata. Quantum finite automata (QFA) were introduced by Moore and Crutchfield [MC00]; they are to quantum computers what finite automata are to Turing machines. Quantum automata are also analogous to the probabilistic finite automata introduced in the 1960s by Rabin that accept words with a certain probability (see [Rab63], [Rab67]; see also [Paz71] for a book-length treatment). A quantum automaton A assigns real values $\mathrm{Val}_A(w)$ to input words w (see below for a precise description of how these values are computed). $\mathrm{Val}_A(w)$ can be interpreted as the probability that on any given run of A on the input word w, w is accepted by A. Nonisolated cut-point recognition will be considered in this article: we do not ask for a gap between the set of $\mathrm{Val}_A(w)$ for accepted words w and the set of $\mathrm{Val}_A(w)$ for rejected words w. Associated to a real threshold λ, the languages recognized by the automaton A with nonstrict and strict threshold λ are

$$L_{\geq} = \{w : \mathrm{Val}_A(w) \geq \lambda\} \quad \text{and} \quad L_{>} = \{w : \mathrm{Val}_A(w) > \lambda\}.$$

Many properties of these languages are known in the case of probabilistic and quantum automata. For instance, it is known that the class of languages recognized by quantum automata is strictly contained in the class of languages recognized by probabilistic finite automata [BP02]. For probabilistic automata it is also known that the problem of determining if $L_{\geq}$ is empty and the problem of determining if $L_{>}$ is empty are undecidable (see [Paz71, Thm. 6.17, p. 90]). This is true even for automata of fixed dimensions [BC03]. Decidability problems on QFA were first studied in the paper by Amano and Iwama [AI99]: is the language recognized by a 1.5-way quantum automaton empty? The undecidability of this problem was proven, even in the case of isolated cut-point.

*Received by the editors April 9, 2003; accepted for publication (in revised form) December 11, 2004; published electronically August 17, 2005.
http://www.siam.org/journals/sicomp/34-6/42586.html
†Department of Mathematical Engineering, Université Catholique de Louvain (blondel@inma.ucl.ac.be).
‡Laboratoire de l'Informatique du Parallélisme, Ecole Normale Supérieure de Lyon (Emmanuel.Jeandel@ens-lyon.fr, Pascal.Koiran@ens-lyon.fr, Natacha.Portier@ens-lyon.fr).

TABLE 1
Decidable and undecidable problems for probabilistic and quantum automata.

	$L_{\geq} = \emptyset$	$L_{>} = \emptyset$	$L_{\leq} = \emptyset$	$L_{<} = \emptyset$
PFA	undecidable	undecidable	undecidable	undecidable
QFA	undecidable	decidable	undecidable	decidable

In this contribution, we consider the problem of determining for a quantum automaton A and threshold λ if there exists a word w for which $\mathrm{Val}_A(w) \geq \lambda$ and if there exists a word w for which $\mathrm{Val}_A(w) > \lambda$. We prove in Theorem 2.1 and Corollary 2.2 that the first problem is undecidable, and in Theorem 3.1 that the second problem is decidable. For quantum automata it thus makes a difference to consider strict or nonstrict thresholds. This result is in contrast with probabilistic automata, for which both problems are undecidable.

Similarly to the languages $L_{\geq}$ and $L_{>}$, one can define the languages $L_{\leq}$ and $L_{<}$ and ask whether or not they are empty (of course, emptiness of $L_{\leq}$ is equivalent to $L_{>}$ being equal to Σ^*). These two problems are known [Paz71] to be undecidable for probabilistic automata. For quantum automata our decidability results do again differ depending on whether we consider strict or nonstrict inequalities. Our results are summarized in Table 1.

Before we proceed with the proofs, we first define what we mean by a QFA. A number of different quantum automata models have been proposed in the literature and not all models are computationally equivalent. For the "measure-many" model of quantum automata introduced by Kondacs and Watrous [KW97] the four problems of Table 1 are proven undecidable in [Jea02]. The model we consider here is the so-called measure once quantum finite automaton introduced by Moore and Crutchfield [MC00]. These automata operate as follows. Let Σ be a finite set of input letters and let Σ^* denote the set of finite input words (including the empty word); typical elements of Σ^* will be denoted $w = w_1 \cdots w_{|w|}$, where $w_i \in \Sigma$ and $|w|$ denotes the length of w. The QFA A is given by a finite set of n states, $n \times n$ unitary transition matrices X_α (one for each symbol α in Σ), a (row) vector of unit norm s (the initial configuration), and an $n \times n$ orthogonal projection matrix P. Given a word $w \in \Sigma^*$, the value of w, denoted $\mathrm{Val}_A(w)$, is defined by

$$\mathrm{Val}_A(w) = \|sX_wP\|^2.$$

In this expression, $\|\cdot\|$ is the euclidean vector norm, and we use the notation X_w for the product $X_{w_1} \cdots X_{w_{|w|}}$. For a vector v, the value $\|vP\|^2$ is the probability for the quantum state v to be observed in acceptance space. The value $\mathrm{Val}_A(w)$ can thus be interpreted as the probability of observing the quantum state in acceptance space after having applied the operator sequence X_{w_1} to $X_{w_{|w|}}$ to the initial quantum state s.

The rest of the paper is organized as follows. In section 2, we reduce Post's correspondence problem to the problem of determining if a quantum automata has a word of value larger than or equal to a given threshold. Post's correspondence problem is undecidable, and this therefore proves our first result. Our reduction uses an encoding of words in three-dimensional space. In section 3, we prove decidability of the same problem for strict inequality. For the proof we use the fact that any compact matrix group is algebraic, and the group we consider can be given an effective description.

Complex versus real entries. Throughout the paper we will assume that the initial state, the unitary matrices X_α, and the projection matrix P have real rather than complex entries (i.e., these matrices are actually orthogonal). This is not a significant restriction since any quantum automaton A (with possibly complex entries) can be simulated by another quantum automaton A' with real entries by doubling the number of states. More precisely, let Q be the set of states of A. We replace each element q_j of Q by two states q_j^1 and q_j^2. Let $\phi : \mathbb{C}^n \to \mathbb{R}^{2n}$ be the $\mathbb{R}$-linear map which sends a configuration $x = \sum_{j=1}^n (\alpha_j + i\beta_j) q_j$ to $\phi(x) = \sum_{j=1}^n \alpha_j q_j^1 + \sum_{j=1}^n \beta_j q_j^2$. We replace the initial configuration s by $s' = \phi(s)$. Let X be one of the matrices of A. The rows and columns of A are indexed by elements of Q. Let $x_{jk} + iy_{jk}$ be the entry at row q_j and column q_k. Recall that a complex number $x + iy$ can be identified to the 2×2 matrix

$$\begin{pmatrix} x & -y \\ y & x \end{pmatrix}.$$

It is therefore natural to replace this entry by the 2×2 matrix

$$\begin{pmatrix} x_{jk} & -y_{jk} \\ y_{jk} & x_{jk} \end{pmatrix}.$$

The two rows and two columns of this matrix are indexed, respectively, by q_j^1, q_j^2, q_k^1, and q_k^2. By abuse of notation we also denote by ϕ the map which sends X to X'. It is easy but instructive to check that for any $v \in \mathbb{C}^n$ and for any $n \times n$ complex matrices A and B the following relations hold: $\phi(Av) = \phi(A)\phi(v)$, $\phi(AB) = \phi(A)\phi(B)$, $\phi(A^*) = \phi(A)^T$, and $v^*v = \phi(v)^T\phi(v)$. Now recall that unitary matrices, orthogonal matrices, complex matrices of orthogonal projection, and real matrices of orthogonal projection are, respectively, characterized by the following relations: $AA^* = I$, $AA^T = I$, $A = A^* = A^2$, and $A = A^T = A^2$. It follows that ϕ sends unitary matrices to orthogonal matrices, and complex matrices of orthogonal projection to real matrices of orthogonal projection. The quantum automaton A' defined by the orthogonal matrices $X'_a = \phi(X_a)$, the projection matrix $P' = \phi(P)$, and the initial configuration s' satisfies $\phi(sX_wP) = s'X'_wP'$ for any word w. Hence $\mathrm{Val}_A(w) = \mathrm{Val}_{A'}(w)$ for any word w.

2. Undecidability for nonstrict inequality. We prove in this section that the problem of determining if a quantum automata has a word of value larger than or equal to some threshold is undecidable. The proof is by reduction from Post's correspondence problem (PCP), a well-known undecidable problem. An instance of PCP is given by a finite alphabet Σ and k pairs of words $(u_i, v_i) \in \Sigma^* \times \Sigma^*$ for $i = 1, \ldots, k$. A solution to the correspondence is any nonempty word $w = w_1 \cdots w_n$ over the alphabet $\{1, \ldots, k\}$ such that $u_w = v_w$, where $u_w = u_{w_1} \ldots u_{w_n}$. This correspondence problem is known to be undecidable: there is no algorithm that decides if a given instance has a solution [Pos46]. It is easy to see that the problem remains undecidable when the alphabet Σ contains only two letters. The problem is also known to be undecidable for $k = 7$ pairs [MS05] but is decidable for $k = 2$ pairs; the decidability of the cases $2 < k < 7$ is not yet known. We are now ready to state our first result.

THEOREM 2.1. *There is no algorithm that decides for a given automaton A if there exists a nonempty word w for which $\mathrm{Val}_A(w) \leq 0$, or if there exists one for which $\mathrm{Val}_A(w) \geq 1$. These two problems remain undecidable even if the automaton is given by 7 orthogonal matrices in dimension 6.*

Proof. We proceed by reduction from PCP. For our reduction we need to encode words by orthogonal matrices. We will take matrices that represent rotations of angle $\arccos(3/5)$ on, respectively, the first and third axes:

$$X_a = \frac{1}{5}\begin{pmatrix} 3 & -4 & 0 \\ 4 & 3 & 0 \\ 0 & 0 & 5 \end{pmatrix}, \quad X_b = \frac{1}{5}\begin{pmatrix} 5 & 0 & 0 \\ 0 & 3 & -4 \\ 0 & 4 & 3 \end{pmatrix}.$$

These matrices are orthogonal, $X_a X_a^T = I = X_b X_b^T$, and they generate a free group since a result from Swierczkowski [Sw58, Sw94] ensures that if $\cos\phi \in \mathbb{Q}$, two rotations of angle ϕ on orthogonal axes in $\mathbb{R}^3$ generate a free group if and only if $\cos\phi \notin \{0, \pm\frac{1}{2}, \pm 1\}$.

In addition to that, we now prove that there exists a vector t such that $tX_u = tX_w$ implies $u = w$.

We will use here a method from [Su90]. One can show by induction that for any reduced matrix product M of k matrices[1] taken from the set $\{X_a, X_b, X_a^{-1}, X_b^{-1}\}$, we have

$$(3\ 0\ 4)M = (x_1\ x_2\ x_3)/5^k$$

with $x_1, x_2, x_3 \in \mathbb{Z}$, and 5 divides x_2 if and only if $k = 0$ (and then $M = I$).

The result is obviously true for $k = 0, 1$. Now, if $M = M' X_a X_b$, then $(3\ 0\ 4)M = (x_1\ x_2\ x_3)/5^k X_a X_b = (x_4\ x_5\ 5x_3)/5^{k+1} X_b$ for some x_4, x_5, and by induction hypothesis, 5 does not divide x_5. Now $(3\ 0\ 4)M = (x_6\ 3x_5 + 20x_3\ x_7)/5^{k+2}$ so that 5 does not divide the second term. The proofs for all the other cases are similar.

We will now call t the row vector $(3\ 0\ 4)$. If $tX_u = tX_v$, then $tX_u X_v^{-1} = t$. As the second component of t is equal to 0, the product must be trivial, and so $u = v$.

Given an instance $(u_i, v_i)_{1 \le i \le k}$ of PCP over the alphabet $\{a, b\}$ and a word $w \in \{1, \ldots, k\}^*$, we construct the matrix

$$Y_w = \frac{1}{2}\begin{pmatrix} X_{u_w} + X_{v_w} & X_{u_w} - X_{v_w} \\ X_{u_w} - X_{v_w} & X_{v_w} + X_{u_w} \end{pmatrix}.$$

These matrices are orthogonal and verify $Y_{w\nu} = Y_w Y_\nu$.

A solution of the original PCP problem is a nonempty word $w \in \{1, \ldots, k\}^*$ such that the upper-right block of the matrix Y_w is equal to zero. We may use the previously introduced vector $t = (3\ 0\ 4)$ to test this condition. We have

$$\begin{pmatrix} t & 0 \end{pmatrix} Y_w = \frac{1}{2}\begin{pmatrix} tX_{u_w} + tX_{v_w} & tX_{u_w} - tX_{v_w} \end{pmatrix},$$

and thus a solution of the PCP problem is a word w such that the last three coordinates of yY_w are equal to zero, where $y = \begin{pmatrix} t & 0 \end{pmatrix}$. This condition can be tested with a projection matrix. Defining

$$P = \begin{pmatrix} 0_3 & 0 \\ 0 & I_3 \end{pmatrix}$$

[1] A product is said to be *reduced* if no two consecutive matrices in the product are inverse from each other.

we have that the solutions of the original PCP problem are the words w for which $y\ Y_w\ P = 0$, which is equivalent to

$$\mathrm{Val}_A(w) = \|yY_wP\|^2 = 0.$$

The values taken by $\mathrm{Val}_A(w)$ are nonnegative and so the problem of determining if there exists a nonempty word w such that $\mathrm{Val}_A(w) \leq 0$ is undecidable. Notice also that $\|yY_wI\|^2 = 1$ and so

$$\|yY_w(I - P)\|^2 \leq 1$$

with equality only for $yY_wP = 0$. Thus, the problem of determining if there exists a nonempty word w such that $\mathrm{Val}_A(w) \geq 1$ is undecidable too. ∎

Theorem 2.1 deals only with nonempty words. We remove this restriction in the next result, and we reduce the number of matrices from 7 to 2.

COROLLARY 2.2. *There is no algorithm that decides for a given automaton A if there exists a word w for which $\mathrm{Val}_A(w) \leq 0$, or if there exists one for which $\mathrm{Val}_A(w) \geq 1$. These problems remain undecidable even if the automaton is given by 7 orthogonal matrices in dimension 6, or by 2 orthogonal matrices in dimension 42.*

Proof. As in the proof of Theorem 2.1, the undecidability results for the condition $\mathrm{Val}_A(w) \geq 1$ follow from those for the condition $\mathrm{Val}_A(w) \leq 0$. Hence we supply the proofs for the latter condition only. We proceed by reduction from the problem $\exists w\ \mathrm{Val}_A(w) \leq 0$ for 7 matrices in dimension 6, which is undecidable for nonempty words w as shown in Theorem 2.1. Note that the language of the nonempty w's such that $\mathrm{Val}_A(w) \leq 0$ is the union of the seven languages defined by the conditions $\mathrm{Val}_A(iw) \leq 0$ for possibly empty words w and $i \in \{1, \ldots, 7\}$. Hence the emptiness of one of these languages (say, the first one) must be undecidable. Thus, the problem of determining if there exists a word w such that $\mathrm{Val}_A(1w) \leq 0$ is undecidable.[2] For each automaton $A = ((Y_i)_{i\in\{1,\ldots,7\}}, s, P)$ we can now construct the quantum automaton $B = ((Y_i)_{i\in\{1,\ldots,7\}}, y, P)$, where $y = sY_1$. Then $\mathrm{Val}_A(1w) \leq 0$ if and only if $\mathrm{Val}_B(w) \leq 0$.

The following problem is therefore undecidable: given a quantum automaton A defined by 7 orthogonal matrices in dimension 6, is there a (possibly empty) word w such that $\mathrm{Val}_A(w) \leq 0$?

Finally, we show how to reduce the number of matrices to 2. We use a construction from Blondel and Tsitsiklis [BT97] and Blondel and Caterini [BC03]. Given the above matrices Y_i and the projection matrix P, we define

$$Z_0 = \begin{pmatrix} Y_1 & 0 & \ldots & 0 \\ 0 & Y_2 & \ddots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & Y_7 \end{pmatrix} \quad \text{and} \quad Z_1 = \begin{pmatrix} 0 & I & 0 & 0 \\ 0 & \ddots & \ddots & 0 \\ \vdots & \vdots & \ddots & I \\ I & 0 & \ldots & 0 \end{pmatrix}.$$

When taking products of these two matrices the matrix Z_1 acts as a "selecting matrix" on the blocks of Z_0. Let us define $x = \begin{pmatrix} y & 0 \end{pmatrix}$ and

$$Q = \begin{pmatrix} P & 0 & \ldots & 0 \\ 0 & P & \ddots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & P \end{pmatrix}.$$

[2] It is not difficult to show that the 6 other problems must be undecidable as well.

We claim that there exists a word w over the alphabet $\{1,\dots,7\}$ such that $\|yY_wP\| = 0$ if and only if there exists a word ν over $\{0,1\}$ such that $\|xZ_\nu Q\| = 0$. Indeed, for any word ν over $\{0,1\}$, xZ_ν is a row vector of block form $(0\cdots 0\ yY_w\ 0\cdots 0)$ for some word w over $\{1,\dots,7\}$ (the length of w is equal to the number of 0's in ν). Therefore $\|xZ_\nu Q\| = \|yY_wP\|$. Conversely, for any word w over $\{1,\dots,7\}$ there exists a word ν over $\{0,1\}$ such that xZ_ν is a row vector of block form $(yY_w\ 0\cdots 0)$, and we therefore have again the equality $\|xZ_\nu Q\| = \|yY_wP\|$. To obtain Z_ν from Y_w, one can for instance replace as in [BT97] each matrix Y_i in the product Y_w by $Z_1^{-(8-i)}Z_0Z_1^{(8-i)} = Z_1^{i-1}Z_0Z_1^{8-i}$. □

Theorem 2.1 and its corollary deal only with 0/1 thresholds. We prove below that, whichever threshold $0 < \lambda \le 1$ is used, the problem of determining if there exists a word for which $\mathrm{Val}_A(w) \ge \lambda$ is undecidable. This result follows as a corollary to the following lemma.

LEMMA 2.3. *Associated to every QFA A and threshold $0 < \lambda \le 1$ we can effectively construct a QFA B such that the language recognized with threshold λ by B is the language recognized with threshold 1 by A. Moreover, if $\lambda \in \mathbb{Q}$ and A has only rational entries, then B can be chosen with rational entries.*

Proof. The idea is to construct B by adding a state to A. Let A be given by the orthogonal matrices X_i^A, the projection matrix P^A, and the initial vector s^A. Let

$$X_i^B = \begin{pmatrix} X_i^A & 0 \\ 0 & 1 \end{pmatrix},$$

and define $s^B = \begin{pmatrix} \sqrt{\lambda}\, s^A & \sqrt{1-\lambda} \end{pmatrix}$. If we choose

$$P^B = \begin{pmatrix} P^A & 0 \\ 0 & 0 \end{pmatrix},$$

we immediately have $\mathrm{Val}_B(w) = \lambda\ \mathrm{Val}_A(w)$ and the first part of lemma is proven. The entries $\sqrt{\lambda}$ and $\sqrt{1-\lambda}$ in general do not need to be rational. It remains to show how the parameters of B can be chosen rational when those of A are. We therefore use Lagrange's theorem to write λ and $1-\lambda$ as the sum of the squares of four rational numbers, say $\lambda = a_1^2 + a_2^2 + a_3^2 + a_4^2$ and $1-\lambda = b_1^2 + b_2^2 + b_3^2 + b_4^2$.

Now, if we define

$$s^B = \begin{pmatrix} a_1 s^A & a_2\cdots a_4 & b_1\cdots b_4 \end{pmatrix}\ X_i^B = \begin{pmatrix} X_i^A & 0 \\ 0 & I_7 \end{pmatrix}\ P^B = \begin{pmatrix} P^A & 0 & 0 \\ 0 & I_3 & 0 \\ 0 & 0 & 0_4 \end{pmatrix},$$

we immediately have $\mathrm{Val}_B(w) = a_1^2\mathrm{Val}_A(w) + a_2^2 + a_3^2 + a_4^2$, $\|s^B\|^2 = 1$ and the lemma is proven. □

Combining Lemma 2.3 with Corollary 2.2, we immediately obtain the following.

COROLLARY 2.4. *For any rational λ, $0 < \lambda \le 1$, there is no algorithm that decides if a given quantum automata has a word w for which* $\mathrm{Val}(w) \ge \lambda$.

3. Decidability for strict inequality. We now prove that the problem of determining if a quantum automata has a word of value *strictly* larger than some threshold is decidable. This result points to a difference between quantum and probabilistic automata since for probabilistic automata this problem is known to be undecidable.

Once an automaton is given, one can of course always enumerate all possible words w and halt as soon as one is found for which $\mathrm{Val}_A(w) > \lambda$, and so the problem

is clearly semidecidable. In order to show that it is decidable, it remains to exhibit a procedure that halts when $\mathrm{Val}_A(w) \leq \lambda$ for all w.

Let a quantum automata A be given by a finite set of $n \times n$ orthogonal transition matrices X_i, an initial configuration s of unit norm, and a projection matrix P. The value of the word w is given by $\mathrm{Val}_A(w) = \|sX_wP\|^2$. Let $\mathcal{X}$ be the semigroup generated by the matrices X_i, $\mathcal{X} = \{X_w : w \in \Sigma^*\}$, and let $f : \mathbb{R}^{n\times n} \mapsto \mathbb{R}$ be the function defined by $f(X) = \|sXP\|^2$. We have that

$$\mathrm{Val}_A(w) = f(X_w),$$

and the problem is now that of determining if $f(X) \leq \lambda$ for all $X \in \mathcal{X}$. The function f is a (continuous) polynomial map and so this condition is equivalent to $f(X) \leq \lambda$ for all $X \in \overline{\mathcal{X}}$, where $\overline{\mathcal{X}}$ is the closure of $\mathcal{X}$ in $\mathbb{R}^{n\times n}$. The set $\overline{\mathcal{X}}$ has the interesting property that it is algebraic (see below for a proof), and so there exist polynomial mappings $f_1, \ldots, f_p : \mathbb{R}^{n\times n} \mapsto \mathbb{R}$, such that $\overline{\mathcal{X}}$ is exactly the set of common zeros of $f_1, \ldots, f_p$. If the polynomials $f_1, \ldots, f_p$ are known, the problem of determining whether $f(X) \leq \lambda$ for all $X \in \overline{\mathcal{X}}$ can be written as a quantifier elimination problem

$$\forall X \big[(f_1(X) = 0 \wedge \cdots \wedge f_p(X) = 0) \implies f(X) \leq \lambda \big]. \tag{3.1}$$

This is a first-order formula over the reals and can be decided effectively by Tarski–Seidenberg elimination methods (see [Ren92a, Ren92b, Ren92c, BPR96] for a survey of known algorithms). If we knew how to effectively compute the polynomials $f_1, \ldots, f_p$ from the matrices X_i, a decision algorithm would therefore follow immediately. In the following we solve a simpler problem: we effectively compute a sequence of polynomials whose zeros describe the same set $\overline{\mathcal{X}}$ after finitely many terms (but we may never know how many). It turns out that this is sufficient for our purposes. We will use some basic algebraic geometry. In particular, we will use the Noether (or "descending chain") property: in any field, the set of common zeros of a set of n-variate polynomials is equal to the set of common zeros of a *finite* subset of these polynomials (see any textbook on algebraic geometry, for instance, [CLO92, Prop. 1, sect. 4.6]).

THEOREM 3.1. *Let $(X_i)_{i\in\Sigma}$ be orthogonal matrices of dimension n and let $\overline{\mathcal{X}}$ be the closure of the semigroup $\{X_w : w \in \Sigma^*\}$. The set $\overline{\mathcal{X}}$ is algebraic, and if the X_i have rational entries, we can effectively compute a sequence of polynomials $f_1, \ldots, f_i, \ldots$ such that*

1. *if $X \in \overline{\mathcal{X}}$, $f_i(X) = 0$ for all i;*
2. *there exists some k such that $\overline{\mathcal{X}} = \{X : f_i(X) = 0, i = 1, \ldots, k\}$.*

Proof. We first prove that $\overline{\mathcal{X}}$ is algebraic. It is known (see, e.g., [OV90]) that every compact group of real matrices is algebraic. In fact, the proof of algebraicity in [OV90] reveals that any compact group G of real matrices of size n is the zero set of

$$\mathbb{R}[X]^G = \{f \in \mathbb{R}[X] : f(I) = 0 \text{ and } f(gX) = f(X)\, \forall\, g \text{ in } G\};$$

i.e., G is the zero set of the polynomials in $n \times n$ variables which vanish at the identity and are invariant under the action of G. We will use this property later in the proof.

To show that $\overline{\mathcal{X}}$ is algebraic, it suffices to show that $\overline{\mathcal{X}}$ is compact and is a group. The set $\overline{\mathcal{X}}$ is obviously compact (bounded and closed in a normed vector space of finite dimension). Let us show that it is a group. It is in fact known that every compact subsemigroup of a topological group is a subgroup. Here is a self-contained proof in our setting: For every matrix X, the sequence X^k admits a subsequence that is a

Cauchy sequence, by compactness. Hence for every ϵ there exists $k > 0$ and $l > k+1$ such that $\|X^k - X^l\| \leq \epsilon$, that is, $\|X^{-1} - X^{l-k-1}\| \leq \epsilon$ (recall that $\|AB\| = \|B\|$ if A is orthogonal and if $\|.\|$ is the operator norm associated to the euclidean norm). Hence, X^{-1} is in the set and the first part of the theorem is proven. For notational convenience, we will denote the group $\overline{\mathcal{X}}$ by G in the remainder of the proof.

For the second part of the theorem, we will prove that we can take

$$\{f_i\} = \{f \in \mathbb{Q}[X] : f(I) = 0 \text{ and } f(X_j X) = f(X) \,\forall\, j \text{ in } \Sigma\}.$$

In other words, this is the set $\mathbb{Q}[X]^G$ of rational polynomials which vanish at the identity and are invariant under the action of each matrix X_j. It is clear that this set is recursively enumerable. We claim that G is the zero set of the f_i's. By Noetherianity the zero set of the f_i's is equal to the zero set of a finite subset of the f_i's, so that the theorem follows immediately from this claim. To prove the claim, we will use the fact that G is the zero set of $\mathbb{R}[X]^G$. Note that

$$\mathbb{R}[X]^G = \{f \in \mathbb{R}[X] : f(I) = 0 \text{ and } f(X_j X) = f(X) \,\forall\, j \text{ in } \Sigma\}.$$

(A polynomial is invariant under the action of G if and only if it is invariant under the action of all the X_j.) This observation implies immediately that each f_i is in $\mathbb{R}[X]^G$, so that the zero set of the f_i's contains the zero set of $\mathbb{R}[X]^G$. The converse inclusion follows from the fact that any element P of $\mathbb{R}[X]^G$ can be written as a linear combination of some f_i's. Indeed, let d be the degree of P and let E_d be the set of real polynomials in $n \times n$ variables of degree at most d. The set $V_d = E_d \cap \mathbb{R}[X]^G$ is a linear subspace of E_d defined by a system of linear equations with rational coefficients (those equations are $f(I) = 0$ and $f(X_j X) = f(X)$ for all $j \in \Sigma$). Hence there exists a basis of V_d made up of polynomials with rational coefficients, that is, of elements of $\{f_i\}$. This completes the proof of the claim, and of the theorem. □

We may now apply this result to quantum automata.

THEOREM 3.2. *The two following problems are decidable:*

(i) *Given a quantum automaton A and a threshold λ, decide whether there exists a word w such that* $\mathrm{Val}_A(w) > \lambda$.

(ii) *Given a quantum automaton A and a threshold λ, decide whether there exists a word w such that* $\mathrm{Val}_A(w) < \lambda$.

Proof. We show only that problem (i) is decidable. The argument for problem (ii) is essentially the same.

As pointed out at the beginning of this section, it suffices to exhibit an algorithm which halts if and only if $\mathrm{Val}_A(w) \leq \lambda$ for every word w. Consider the following algorithm:

- enumerate the f_i's;
- for every initial segment $f_1, \ldots, f_p$, decide whether (3.1) holds, and halt if it does.

It follows from property (1) in Theorem 3.1 that $\mathrm{Val}_A(w) \leq \lambda$ for every word w if the algorithm halts. The converse follows from property (2). □

In Theorems 3.1 and 3.2 we have assumed that our orthogonal matrices have rational entries, mostly because the undecidability results of section 2 already hold for rational entries. It is not hard to relax this hypothesis. For instance, it is clear from the proofs that Theorems 3.1 and 3.2 can be generalized to matrices with real algebraic entries. Even more generally, one may allow "arbitrary" real entries by proceeding as follows. Let K be the subfield of $\mathbb{R}$ generated by the entries of our matrices. We may give a transcendence basis $\mathcal{B}$ of K and represent the entries as

algebraic numbers over $\mathcal{B}$. This purely algebraic information is sufficient to compute the sequence of polynomials (f_i) in Theorem 3.1. We also need to decide for every initial segment whether (3.1) holds. After quantifier elimination, this amounts to computing the sign of a finite number of polynomial functions of the elements of $\mathcal{B}$. In order to do this we need only assume that we have access to an oracle which for any element x of $\mathcal{B}$ and any $\epsilon > 0$ outputs a rational number q such that $|x - q| < \epsilon$ (such an oracle can be effectively implemented if the entries are computable real numbers). We use the algebraic information to determine whether a polynomial takes the value zero, and if not we use approximations to determine its sign.

In the proof of Theorem 3.2 we have bypassed the problem of explicitly computing a finite set of polynomials defining $\overline{\mathcal{X}}$. It is in fact possible to show that this problem is algorithmically solvable [DJK03]. This implies in particular that the following two problems are decidable:

(i) Decide whether a given threshold is isolated.

(ii) Decide whether a given QFA has an isolated threshold.

A threshold λ is said to be isolated if

$$\exists \epsilon > 0 \; \forall w \in \Sigma^* \; |\mathrm{Val}_A(w) - \lambda| > \epsilon.$$

It is known that these two problems are undecidable for probabilistic automata [Ber75, BMT77, BC03].

The algorithm of [DJK03] for computing $\overline{\mathcal{X}}$ also has applications to quantum circuits: this algorithm can be used to decide whether a given set of quantum gates is complete (*complete* means that any orthogonal transformation can be approximated to any desired accuracy by a quantum circuit made up of gates from the set). Much effort has been devoted to the construction of specific complete sets of gates [DBE95, BBC$^+$95], but no general algorithm for deciding whether a given set is complete was known.

Finally, we note that the proof of Theorem 3.2 does not yield any bound on the complexity of problems (i) and (ii). We hope to investigate this question in future work.

Acknowledgment. P.K. would like to thank Etienne Ghys for pointing out reference [OV90]. We are also grateful to the anonymous referee for his very careful reading of the manuscript.

REFERENCES

[AI99] M. Amano and K. Iwama, *Undecidability on quantum finite automata*, in Proceedings of the 31st ACM Symposium on Theory of Computing, ACM, New York, 1999, pp. 368–375.

[BBC$^+$95] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. H. Margolus, P. W. Shor, T. Sleator, J. A. Smolin, and H. Weinfurter, *Elementary gates for quantum computation*, Phys. Rev. A, 52 (1995), pp. 3457–3467.

[BPR96] S. Basu, R. Pollack, and M.-F. Roy, *On the combinatorial and algebraic complexity of quantifier elimination*, J. ACM, 43 (1996), pp. 1002–1045.

[BC03] V. D. Blondel and V. Canterini, *Undecidable problems for probabilistic automata of fixed dimension*, Theory Comput. Syst., 36 (2003), pp. 231–245.

[Ber75] A. Bertoni, *The solution of problems relative to probabilistic automata in the frame of the formal languages theory*, in Vierte Jahrestagung der Gesellschaft für Informatik, Lecture Notes in Comput. Sci. 26, Springer, Berlin, 1975, pp. 107–112.

[BMT77] A. Bertoni, G. Mauri, and M. Torelli, *Some recursively unsolvable problems relating to isolated cutpoints in probabilistic automata*, in Proceedings of the 4th International

Colloquium on Automata, Languages and Programming, Lecture Notes in Comput. Sci. 52, Springer, Berlin, 1977, pp. 87–94.

[BP02] A. Brodsky and N. Pippenger, *Characterizations of* 1*-way quantum finite automata*, SIAM J. Comput., 31 (2002), pp. 1456–1478.

[BT97] V. D. Blondel and J. N. Tsitsiklis, *When is a pair of matrices mortal?*, Inform. Process. Lett., 63 (1997), pp. 283–286.

[CLO92] D. Cox, J. Little, and D. O'Shea, *Ideals, Varieties, and Algorithms*, Undergrad. Texts Math., Springer, New York, 1992.

[DBE95] D. Deutsch, A. Barenco, and A. K. Ekert, *Universality in quantum computation*, Proc. Roy. Soc. London Ser. A, 449 (1995), pp. 669–677.

[DJK03] H. Derksen, E. Jeandel, and P. Koiran, *Quantum automata and algebraic groups*, J. Symbolic Comput., 39 (2005), pp. 357–371.

[Jea02] E. Jeandel, *Indécidabilité sur les automates quantiques*, Master's Thesis, 2002; available online from http://perso.ens-lyon.fr/emmanuel.jeandel/publis.html.

[KW97] 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, IEEE, Los Alamitos, 1997, pp. 66–75.

[MC00] C. Moore and J. Crutchfield, *Quantum automata and quantum grammars*, Theoret. Comput. Sci., 237 (2000), pp. 257–306.

[MS05] Y. Matiyasevich and G. Sénizergues, *Decision problems for semi-Thue systems with a few rules*, Theoret. Comput. Sci., 330 (2005), pp. 145–169.

[OV90] A. Onishchik and E. Vinberg, *Lie Groups and Algebraic Groups*, Springer, Berlin, 1990.

[Paz71] A. Paz, *Introduction to Probabilistic Automata*, Academic Press, New York, 1971.

[Pos46] E. L. Post, *A variant of a recursively unsolvable problem*, Bull. Amer. Math. Soc., 52 (1946), pp. 264–268.

[Rab63] M. O. Rabin, *Probabilistic automata*, Inform. Control, 6 (1963), pp. 230–245.

[Rab67] M. O. Rabin, *Mathematical theory of automata*, in Proc. Sympos. Appl. Math. 19, AMS, Providence, RI, 1967, pp. 153–175.

[Ren92a] J. Renegar, *On the computational complexity and geometry of the first-order theory of the reals.* I, J. Symbolic Comput., 13 (1992), pp. 255–299.

[Ren92b] J. Renegar, *On the computational complexity and geometry of the first-order theory of the reals.* II, J. Symbolic Comput., 13 (1992), pp. 301–327.

[Ren92c] J. Renegar, *On the computational complexity and geometry of the first-order theory of the reals.* III, J. Symbolic Comput., 13 (1992), pp. 329–352.

[Su90] F. E. Su, *The Banach-Tarski Paradox*, Minor Thesis, 1990.

[Sw58] S. Swierczkowski, *On a free group of rotations of the Euclidean space*, Nederl. Akad. Wetensch. Proc. Ser. A, 20 (1958), pp. 376–378.

[Sw94] S. Swierczkowski, *A class of free rotation groups*, Indag. Math. (N.S.), 5 (1994), pp. 221–226.