arXiv:quant-ph/0304082 v1 11 Apr 2003

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 non-empty? We prove that this problem is decidable or undecidable depending on whether recognition is defined by strict or non-strict thresholds. This result is in contrast with the corresponding situation for probabilistic finite automata for which it is known that strict and non-strict thresholds both lead to undecidable problems.

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 [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. Associated to a real threshold λ, the languages recognized by the automata A with non-strict and strict threshold λ are

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

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 [Rab63]. This is true even for automata of fixed dimensions [BC03].

In this contribution, we consider the problem of determining for a quantum automata 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

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

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

prove in Theorem 2.1 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 non-strict 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 [Rab63] to be undecidable for probabilistic automata. For quantum automata our decidability results do again differ depending on whether we consider strict or non-strict inequalities. Our results are summarized in Table 1.

Before we proceed with the proofs, we first define what we mean by a quantum finite automaton. A number of different quantum automata models have been proposed in the literature and not all models are computationaly equivalent. For the "measure-many" model of quantum automata introduced by Kondacs and Watrous [KW97] the four problems of Table 1 are proved 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 a $n \times n$ 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 this 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 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.

2. Undecidability for non strict inequality

We prove in this section that the problem of determining if a quantum automata has a word of value larger than some threshold is undecidable. The proof is by reduction from Post's correspondence problem (PCP), a well-known undecidable problem. An instance of Post's correspondence problem 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 non-empty 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 [MS96] but is decidable for $k = 2$ pairs; the decidability of the cases $2 < k < 7$ is yet unresolved. 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 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 matrices in dimension 6, or by 2 matrices in dimension 42.*

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

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

These matrices are unitary, $X_a X_a^T = I = X_b X_b^T$ and they generate a free group since a result from Swierczkowski ensures us that two irrationals rotations on orthogonal axes in $\mathbb{R}^3$ generate a free group. 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 $M = M'X_1X_0$, 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 proof for all the other cases is similar.

We will now call t the row vector $(3\ 0\ 4)$. If $tX_u = tX_v$ then $tX_uX_v^{-1} = t$. 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 unitary, 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

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

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 non-negative and so the problem of determining if there exists a word w such that $\mathrm{Val}_A(w) \le 0$ is undecidable. Notice also that $\|yY_wI\|^2 = 1$ and so

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

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

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

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

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

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

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

We claim that there exists a non-empty word w over the alphabet $\{1, \dots, 7\}$ such that $\|yY_wP\| = 0$ if and only if there exists a non-empty word ν over $\{0, 1\}$ such that $\|xZ_\nu Q\| = 0$. The complete proof of this claim is given in [BT97] and is not reproduced here. □

It is possible to give a stronger form to the second part of the Theorem. We prove below that, whatever threshold $0 \leq \lambda < 1$ is used, the problem of determining if there exists a word for which $\mathrm{Val}_A(w) \geq \lambda$ is undecidable. This result follows as a corollary of the following lemma.

Lemma 2.2. *Associated to every QFA $\mathcal{A}$ and threshold $\lambda < 1$ we can effectively construct a QFA B such that the language recognized with threshold λ by B is the language recognized with threshold 0 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 unitary 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 $\sqrt{1-\lambda}$ do in general 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 square 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 have immediately $\mathrm{Val}_B(w) = a_1^2 \mathrm{Val}_A(w) + a_2^2 + a_3^2 + a_4^2$, $\|s^B\|^2 = 1$ and the lemma is proved. □

Combining Lemma 2.2 with Theorem 2.1, we immediately obtain:

Corollary 2.3. *For any rational λ, $0 \leq \lambda < 1$, there is no algorithm that decides if a given quantum automata has a word w for which* $\mathrm{Val}(w) \leq \lambda$.

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 semi-decidable. 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 $n \times n$ unitary 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 polynomials 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 zeroes 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

$$\forall X \left[(f_1(X) = 0 \wedge \cdots \wedge f_p(X) = 0) \implies f(X) \geq \lambda \right]. \tag{1}$$

This is a first-order formula over the reals, and can be decided effectively by Tarski-Seidenberg elimination methods (see [Ren92] 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 zeroes 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.

Theorem 3.1. *Let $(X_i)_{i \in \Sigma}$ be unitary 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

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

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 $\|X^k - X^l\| \le \epsilon$, that is $\|X^{-1} - X^{l-k-1}\| \le \epsilon$ (recall that $\|AB\| = \|B\|$ if A is unitary, 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 proved. 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

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

In 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

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

(a polynomial is invariant under the action of G if and only 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 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 only show 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 $Val_A(w) \leq \lambda$ for every word w. Consider the following algorithm:

- enumerate the f_i's;
- for every initial segment $f_1, \ldots, f_p$, decide whether (1) holds, and halt if it does.

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

Throughout the paper we have assumed that our unitary matrices have 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.

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 treshold 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 $\overline{\mathcal{X}}$ has also applications to quantum circuits: this algorithm can be used to decide whether a given set of quantum gates is complete (*complete* means that any unitary 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].

References

[BBC+95] Adriano Barenco, Charles H. Bennett, Richard Cleve, David P. DiVincenzo, Norman H. Margolus, Peter W. Shor, Tycho Sleator, John A. Smolin, and Harald Weinfurter. Elementary gates for quantum computation. *Physical Review A*, 52(5):3457–3467, 1995.

[BC03] Vincent D. Blondel and Vincent Canterini. Undecidable problems for probabilistic automata of fixed dimension. *Theory of Computing Systems*, 2003.

[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*, volume 26 of *Lecture Notes in Computer Science*, pages 107–112. Springer, Berlin, 1975.

[BMT77] A. Bertoni, G. Mauri, and M. Torelli. Some recursively unsolvable problems relating to isolated cutpoints in probabilistic automata. In *Proc. 4th International Colloquium on Automata, Languages and Programming*, volume 52 of *Lecture Notes in Computer Science*, pages 87–94. Springer, Berlin, 1977.

[BP02] Alex Brodsky and Nicholas Pippenger. Characterizations of 1-way quantum finite automata. *SIAM Journal on Computing*, 31(5):1456–1478, 2002.

[BT97] Vincent D. Blondel and John N. Tsitsiklis. When is a pair of matrices mortal? *Information Processing Letters*, 63(5):283–286, 1997.

[DBE95] David Deutsch, Adriano Barenco, and Artur K. Ekert. Universality in quantum computation. *Proceedings of the Royal Society of London Ser. A*, 449:669–677, 1995.

[DJK03] H. Derksen, E. Jeandel, and P. Koiran. Quantum automata and algebraic groups. in preparation, 2003.

[Jea02] E. Jeandel. Indécidabilité sur les automates quantiques, 2002. Master Thesis.

[KW97] Attila Kondacs and John Watrous. On the power of quantum finite state automata. In *IEEE Symposium on Foundations of Computer Science*, pages 66–75, 1997.

[MC00] C. Moore and J. Crutchfield. Quantum automata and quantum grammars. *Theoretical Computer Science*, 237(2):257–306, 2000.

[MS96] Yuri Matiyasevich and Gerard Senizergues. Decision problems for semi-thue systems with a few rules. In *Logic in Computer Science*, pages 523–531, 1996.

[OV90] A. Onishchik and E. Vinberg. *Lie groups and algebraic groups.* Springer Verlag, 1990.

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

[Pos46] E. L. Post. A variant of a recursively unsolvable problem. *Bulletin of the American Mathematical Society*, 52:264–268, 1946.

[Rab63] M. O. Rabin. Probabilistic automata. *Information and Control*, 6:230–245, 1963.

[Rab67] M. O. Rabin. Mathematical theory of automata. In *Proc. Sympos. Appl. Math.*, volume 19, pages 153–175, 1967.

[Ren92] J. Renegar. On the computational complexity of the first-order theory of reals; parts i-iii. *J. Symb. Comp.*, 13:255–352, 1992.

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

Vincent D. Blondel, Departement of Mathematical Engineering, Université Catholique de Louvain.

E-mail address: blondel@inma.ucl.ac.be

Emmanuel Jeandel, Laboratoire de l'Informatique du Parallélisme, Ecole Normale Supérieure de Lyon.

E-mail address: Emmanuel.Jeandel@ens-lyon.fr

Pascal Koiran, Laboratoire de l'Informatique du Parallélisme, Ecole Normale Supérieure de Lyon.

E-mail address: Pascal.Koiran@ens-lyon.fr

Natacha Portier, Laboratoire de l'Informatique du Parallélisme, Ecole Normale Supérieure de Lyon.

E-mail address: Natacha.Portier@ens-lyon.fr