Towards model-checking quantum security protocols

P. Baltazar, R. Chadha[*], P. Mateus and A. Sernadas
SQIG-IT and IST, Portugal
{pbtz,rchadha,pmat,acs}@math.ist.utl.pt

October 17, 2006

Abstract

Logics for reasoning about quantum states have been given in the literature. In this paper, we extend one such logic with temporal constructs mimicking the standard computational tree logic used to reason about classical transition systems. We investigate the model-checking problem for this temporal quantum logic and illustrate its use by reasoning about the BB84 key distribution protocol.

1 Introduction

Reasoning about quantum programs has gained prominence due to a big potential in applications such as information processing, security, distributed systems and randomized algorithms. This has attracted research in formal reasoning about quantum states [19, 18, 13, 7] and quantum programs [12, 16, 1, 10, 2, 17, 3, 6]. Formal methods have proved to successful in design and verification of classical distributed systems and security protocols [9, 14]. Herein, we present a temporal logic for reasoning about evolution of quantum systems composed of a fixed finite set of qubits.

Our starting point is the logic dEQPL for reasoning about quantum states presented in [13, 7]. The logic dEQPL is designed around the first two postulates of quantum mechanics. The first postulate says that a quantum state is a unit vector in a complex Hilbert space and the second one says that the quantum state composed of two independent quantum states is the

[*]The author R. Chadha is now at Dept of Computer Science, Univ of Illinois, Urbana-Chmapaign, USA.

tensor product of the composing states. Herein, for efficiency reasons, we consider just a restricted sub-logic of dEQPL based on the first postulate. The models of this logic are basically the quantum states of the finite qubit system.

We give a sound and complete axiomatization of this state logic. The completeness proof, which is inspired by [7, 11], also suggests a decision procedure for the theorem-hood problem and we compute the complexity of the decision procedure assuming that all basic integer operations (addition, subtraction, multiplication and comparison) take unit time. Furthermore, assuming a floating point representation of complex numbers and assuming that basic floating point operations (addition, subtraction, multiplication and comparison) take unit time, we compute the complexity of the model-checking algorithm.

Next, we obtain quantum computational tree logic QCTL by replacing the state formulas in the standard computational tree logic (CTL) [8] by dEQPL formulas. The standard CTL is interpreted over classical states and transition relations amongst these states. QCTL is interpreted over quantum states and unitary transformations. We give a sound axiomatization of QCTL and combine the standard CTL model-checking algorithm with the dEQPL model-checking algorithm to obtain a model-checking algorithm for QCTL. The completeness of QCTL is out of scope of this paper.

Finally, we note that we do not explicitly deal with measurements in this paper, although we can reason about probabilities of outcomes of measuring all the qubits in the standard computational basis. The rest of the paper is organized as follows. Section 2 discusses the restricted dEQPL and Section 3 introduces QCTL. We discuss the BB84 protocol in Section 4 and summarize our contributions in Section 5. For lack of space, the proofs are omitted in this paper and are available at: `http://wslc.math.ist.utl.pt/ftp/pub/SernadasA/06-BCMS-quantlog13s.pdf`.

2 State logic

We discuss here briefly the restricted state logic, dEQPL. The logic is designed around the first postulate of quantum mechanics which states that each quantum system is a unit vector in a complex Hilbert space. For our purposes, we shall only deal with a finite-dimensional Hilbert space composed of a finite set of qubits. We shall thus assume a fixed finite set of qubit symbols, qB, which will represent these qubits.

A quantum state $|\psi\rangle$ therefore is a unit vector in $\mathcal{H}_{\mathsf{qB}} = \mathcal{H}(2^{\mathsf{qB}})$, the

Hilbert space generated by the set of valuations 2^{qB}. Please note that these valuations constitute what is commonly called the standard computational basis. Assuming that qB has n elements, the vector $|\psi\rangle$ is then specified by 2^n complex numbers $\langle v|\psi\rangle$ that give the projection on the basis vectors $|v\rangle$. We shall have terms in our language representing the real and complex parts of these 2^n complex numbers. Furthermore, please note that there is a natural bijection between the subsets of qB and the set of valuations over qB: a set A corresponds to a valuation v_A which valuates to true if $\mathsf{qb} \in A$ and valuates to false if $\mathsf{qb} \notin A$.

We shall also have terms in our logic that will represent the probability of outcomes if all the qubits in qB were to be measured in the standard computational basis. We are now ready to discuss the syntax and semantics of dEQPL.

2.1 Language and semantics

Syntax. The terms in dEQPL denote elements from $\mathbb{R}$, the set of real numbers. The formulas of dEQPL, henceforth called *quantum formulas*, are constructed from *comparison formulas* (formulas that compare terms) using propositional connectives. We present language of dEQPL in Table 1 using an abstract version of BNF notation [15] for a compact presentation of inductive definitions and discuss the language in detail below.

Table 1: Language of efficient EQPL

Classical formulas

$\alpha \quad := \quad \bot \,[\!]\, \mathsf{qb} \,[\!]\, (\alpha \Rightarrow \alpha)$

Term language (with the proviso $m \in \mathbb{Z}$ and $A \subseteq \mathsf{qB}$)

$t \quad := \quad x \,[\!]\, m \,[\!]\, (t + t) \,[\!]\, (t\, t) \,[\!]\, \mathrm{Re}(|\top\rangle_A) \,[\!]\, \mathrm{Im}(|\top\rangle_A) \,[\!]\, (\int \alpha)$

Quantum formulas

$\gamma \quad := \quad (t \leq t) [\!]\, \perp\!\!\!\perp \, [\!](\gamma \sqsupset \gamma)$

The first syntactic category is that of *classical formulas*. Please recall that we fixed a finite set of qubit symbols qB. Classical formulas are built from qubit symbols in qB using the classical disjunctive connectives, falsum $\bot$ and implication $\Rightarrow$. As usual, other classical connectives like $\neg, \wedge, \vee, \Leftrightarrow$ and $\top$ are introduced as abbreviations.

For the term language, we pick a denumerable sets of variables $X = \{x_k : k \in \mathbb{N}\}$ interpreted over reals. We also have a copy of integers in the set of terms. The terms $\mathrm{Re}(|\top\rangle_A)$ and $\mathrm{Im}(|\top\rangle_A)$ denote the real and complex parts of the logical amplitude $\langle v_A|\psi\rangle$, where ψ is a quantum state over qB and v_A is the (unique) valuation corresponding to the set A. The *probability term* $(\int\alpha)$ denotes the probability that classical formula α holds for an outcome of measuring the all the qubits (in qB) in the standard basis.

As usual, we may define the notion of occurrence of a term t_1 in a term t, and the notion of replacing zero or more occurrences of terms t_1 in t by t_2. If $\vec{x}$ and $\vec{t}$ are sequences of variables and terms respectively, we will write $t\{\!\{\vec{x}/\vec{t}\}\!\}$ to mean the real term obtained by substituting *all* occurrences of x_i by t_i.

The quantum formulas are built from classical formulas *comparison formulas* $(t \leq t)$ using the connectives $\perp\!\!\!\perp$ and $\sqsupset$. The set of comparison formulas shall henceforth be called qAtom. and use δ, δ' to range over this set. Please note that quantum bottom $\perp\!\!\!\perp$ and quantum implication $\sqsupset$ should not be confused with their classical (local) counterparts.

For clarity sake, we shall often drop parenthesis in formulas and terms if it does not lead to ambiguity. As expected, other quantum connectives will be introduced as abbreviations. However, before introducing a whole set of useful abbreviations, we present the semantics of the language.

Semantics. The language is interpreted over a unit vector $|\psi\rangle$ on the Hilbert space $\mathcal{H}_{\mathsf{qB}}$ spanned by all valuations over qB. For interpreting the variables, we also need the concept of an assignment. An *assignment* ρ is a map from X, the set of variables, such that $\rho(x) \in \mathbb{R}$. Given a classical state formula α and a valuation v over qB, we shall also assume the definition of satisfaction of α by v. We shall write $v \Vdash_c \alpha$ if v satisfies α.

For interpreting the probability terms $(\int\alpha)$, we shall use the *probability map* $\mu_{|\psi\rangle} : \wp(\mathsf{qB}) \to \mathbb{R}$ defined as:

$$\mu_{|\psi\rangle}(U) = \sum_{v \in U} ||\langle v|\psi\rangle||^2 .$$

For the probability terms, we shall also need the *extent* of classical formulas defined as:

$$|\alpha| = \{v \in \wp(\mathsf{qB}) : v \Vdash_c \alpha\}.$$

Given a quantum state ψ and an assignment ρ, the *denotation of terms* and *satisfaction of quantum formulas* at $|\psi\rangle$ and ρ is inductively defined in Table 2 (omitting the obvious ones).

Table 2: Semantics of dEQPL

Denotation of terms

$$
\begin{array}{lcl}
[\![x]\!]_{|\psi\rangle\rho} & = & \rho(x) \\
[\![(\int \alpha)]\!]_{|\psi\rangle\rho} & = & \mu_{|\psi\rangle}(|\alpha|_V) \\
[\![\mathrm{Re}(|\top\rangle_A)]\!]_{|\psi\rangle\rho} & = & \mathrm{Re}(\langle v_A|\psi\rangle) \\
[\![\mathrm{Im}(|\top\rangle_A)]\!]_{|\psi\rangle\rho} & = & \mathrm{Im}(\langle v_A|\psi\rangle)
\end{array}
$$

Satisfaction of quantum formulas

$$
\begin{array}{lll}
|\psi\rangle\rho \Vdash (t_1 \leq t_2) & \text{iff} & [\![t_1]\!]_{|\psi\rangle\rho} \leq [\![t_2]\!]_{|\psi\rangle\rho} \\
|\psi\rangle\rho \nVdash \perp\!\!\!\perp & & \\
|\psi\rangle\rho \Vdash (\gamma_1 \sqsupset \gamma_2) & \text{iff} & |\psi\rangle\rho \nVdash \gamma_1 \text{ or } |\psi\rangle\rho \Vdash \gamma_2
\end{array}
$$

Please note that the assignment ρ is sufficient to interpret a useful sub-language of our quantum formulas defined as:

$$
\begin{array}{lcl}
a & := & x \,[\!]\, m \,[\!]\, (a + a) \,[\!]\, (a\, a) \\
\kappa & := & (a \leq a) \,[\!]\, (\perp\!\!\!\perp) \,[\!]\, (\kappa \sqsupset \kappa)
\end{array}
$$

Henceforth, the terms of this sub-language will be called *analytical terms* and the formulas will be called *analytical formulas*.

Abbreviations. As anticipated, the proposed quantum language with the semantics above is rich enough to express interesting properties of quantum systems. To this end, it is quite useful to introduce other operations, connectives and modalities through abbreviations. We start with some additional quantum connectives:

- quantum negation: $(\boxminus \gamma)$ for $(\gamma \sqsupset \perp\!\!\!\perp)$;
- quantum disjunction: $(\gamma_1 \sqcup \gamma_2)$ for $((\boxminus \gamma_1) \sqsupset \gamma_2)$;
- quantum conjunction: $(\gamma_1 \sqcap \gamma_2)$ for $(\boxminus((\boxminus \gamma_1) \sqcup (\boxminus \gamma_2)))$;
- quantum equivalence: $(\gamma_1 \equiv \gamma_2)$ for $((\gamma_1 \sqsupset \gamma_2) \sqcap (\gamma_2 \sqsupset \gamma_1))$.

It is also useful to introduce some additional comparison formulas:

- $(t_1 < t_2)$ for $((t_1 \leq t_2) \sqcap (\boxminus(t_2 \leq t_1)))$;
- $(t_1 = t_2)$ for $((t_1 \leq t_2) \sqcap (t_2 \leq t_1))$.

Given $A \subseteq \mathsf{qB}$, the following abbreviation will also be useful:

- $(\wedge A)$ for $((\wedge_{\mathsf{qb}_k \in A} \mathsf{qb}_k) \wedge (\wedge_{\mathsf{qb}_k \notin A} (\neg\, qb_k)))$.

The above formula represents the valuation v_A in the language.

The following abbreviation denotes the square of the absolute value of $\langle v_A | \psi \rangle$:

- $||\top\rangle_A|^2$ for $((\mathrm{Re}(|\top\rangle_A))^2 + (\mathrm{Im}(|\top\rangle_A))^2)$;

The following abbreviation is also useful:

- $(\Box\alpha)$ for $(\int \alpha) = 1$.

Intuitively, the formula $(\Box\alpha)$ means that the probability α being true of the outcome of measuring all the qubits in the standard computational basis is 1.

2.2 Model-checking problem

For the model-checking procedure, we only consider closed formulas, *i.e.*, formulas without variables. We assume that a quantum state $|\psi\rangle$ over qB is modeled by a 2^n-array of pairs of real numbers, with $n = |\mathsf{qB}|$. We also assume that the basic arithmetical operations take $O(1)$ time.

We also assume the definition of the *length* of a classical formula α or a quantum formula γ as the number of symbols required to write the formula. The length of a formula ξ (classical or quantum) is given is represented by $|\xi|$.

Given a quantum state ψ and a quantum formula ψ, the first step is to evaluate all the terms occurring in γ. For the probability terms $\int \alpha$, the evaluation takes $2^n|\alpha|$ steps as we have to compute the set of valuations $\wp(\mathsf{qB})$ that satisfy α. Once, the terms are evaluated, the model checking algorithm is straightforward.

Theorem 2.1 Assuming that all basic arithmetical operations take unit time, there is an algorithm $O(|\gamma|.2^n)$ to decide if a quantum state $|\psi\rangle$ over qB satisfies γ with $|\mathsf{qB}| = n$.

Proof: First notice that the terms that consume more time to evaluate are those of the type $(\int \alpha)$ (both the terms $\mathrm{Re}(|\top\rangle_A)$ and $\mathrm{Im}(|\top\rangle_B)$ can be accessed in $O(1)$ time, since they are elements of the array). The number of terms of type $(\int \alpha)$ is bounded by $|\gamma|$. To evaluate one of these terms we require $O(2^n)$ time corresponding to traveling throughout all the valuations

satisfying α, computing the square of the real and imaginary part, and summing all these values. So, computing all $(\int \alpha)$ terms takes $O(|\gamma|.2^n)$ time.

After these values are obtained, the remaining computation (comparing terms, negating a boolean value, and making implications between boolean values) takes at most $O(|\gamma|)$ time. Hence, the total time to decide if a quantum state $|\psi\rangle$ satisfies γ is $O(|\gamma|.2^n + |\gamma|) = O(|\gamma|.2^n)$. $\diamond$

2.3 Axiomatization

We need two new concepts for the axiomatization, one of quantum tautology and a second of valid analytical formulas and ground substitutions.

Consider propositional formulas built from a countable set of propositional symbols Q using the classical connectives $\Rightarrow$ and $\bot$. A quantum formula γ is said to be a quantum tautology if there is a propositional tautology β over Q and a map σ from Q to the set of quantum formulas such that β_σ coincides with γ where β_σ is the quantum formula obtained from β by replacing all occurrences of $\bot$ by $\perp\!\!\!\perp$, $\Rightarrow$ by $\sqsupset$ and $q \in Q$ by $\sigma(q)$. For instance, the expected formula $((y_1 \sqsupset y_2) \sqsupset (y_1 \sqsupset y_2))$ is tautological (obtained, for example, from the propositional tautology $q \Rightarrow q$).

Please recall that an assignment is enough to interpret analytical formulas. We say that an analytical formula κ is *a valid analytical formula* if it holds for any assignment. It is a well-known fact from the theory of real closed fields [4] that the set of valid analytical formulas so defined is decidable. However, we shall not go into details of this result and will focus our attention on reasoning about quantum aspects only.

The axioms and inference rules of dEQPL are listed in Table 3. The only inference rule is modus ponens for quantum implication **QMP**.

The axiom **QTaut** says that a quantum tautology is an axiom. Since the set of quantum tautologies is recursive, there is no need for spelling out details of tautological reasoning. The axiom **RCF** says that if κ is a valid arithmetical formula, then any formula obtained by replacing variables with the terms of dEQPL is a tautology. Since the set of valid arithmetical formulas is recursive, we refrain from spelling out the details. The axiom **Unit** says that a quantum state is a unit vector.

The axioms **CTaut**, **Meas$\emptyset$**, **FAdd** and **Mon** reasons about probability terms $(\int \alpha)$. These axioms are basically the axioms (or minor variations of) the axioms of the probability logics in literature [11]. Hence, the probability logics in [11] can be seen as a sub-logic of dEQPL.

Finally, the axiom **Prob** relates probabilities and amplitudes. This axiom says that for any $A \subset \mathsf{qB}$, the probability of observing the valuation v_A when all qubits are measured is the square of the amplitude $|\top\rangle_A$.

Table 3: Axioms for dEQPL

Axioms		
[**QTaut**]	$\vdash$	γ for each quantum tautology γ
[**RCF**]	$\vdash$	$\kappa\{\!\{\vec{x}/\vec{t}\}\!\}$ where κ is a valid analytical formula, $\vec{x}$ and $\vec{t}$ are sequences of variables and terms
[**Unit**]	$\vdash$	$((\sum_{A\subseteq \mathsf{qB}} \|\|\top\rangle_A\|^2) = 1)$
[**CTaut**]	$\vdash$	$(\Box\alpha)$ for each classical tautology α
[**Mes$\emptyset$**]	$\vdash$	$((\int \bot) = 0)$
[**FAdd**]	$\vdash$	$(((\int(\alpha_1 \wedge \alpha_2)) = 0) \sqsupset ((\int \alpha_1 \vee \alpha_2) = (\int \alpha_1) + (\int \alpha_2)))$
[**Mon**]	$\vdash$	$((\Box(\int(\alpha_1 \Rightarrow \alpha_2))) \sqsupset ((\int \alpha_1) \leq (\int \alpha_2)))$
[**Prob**]	$\vdash$	$((\int \wedge A) = \|\|\top\rangle_A\|^2$
Inference rules		
[**QMP**]	$\gamma_1, (\gamma_1 \sqsupset \gamma_2) \vdash \gamma_2$	

The axiomatization presented above is sound and weakly complete. The proof of weak completeness follows the lines of the proof in [11, 7]. The proof of completeness also suggests an algorithm for deciding whether a formula is theorem of dEQPL or not. In order to state the complexity of this algorithm, we need a few definitions.

The central result in the proof is the Model Existence Lemma, namely, if γ is consistent then there is a quantum state ψ and an assignment ρ such that $|\psi\rangle\rho \Vdash \gamma$. A quantum formula γ is said to be consistent if $\nVdash (\boxminus\gamma)$. A quantum formula γ is a theorem if and only if $(\boxminus\gamma)$ is inconsistent.

Theorem 2.2 (Model Existence Theorem) If the quantum formula γ is consistent then there is a unit vector $|\psi\rangle$ and a ρ such that $|\psi\rangle\rho \Vdash \gamma$.

Proof: Given a classical state formula α, we can show using the axioms

CTaut, **Meas∅**, **FAdd** and **Mon** that $\vdash ((\int \alpha) = \sum_{\{A \subseteq \mathsf{qB} \mid v_A \Vdash_c \alpha\}} (\int \wedge A))$. The axiom **Prob** then gives us that $(\int \alpha) = \sum_{\{A \subseteq \mathsf{qB} \mid v_A \Vdash_c \alpha\}} ||\top\rangle_A|^2$. Hence, given a quantum formula γ, we can find an equivalent quantum formula that does not contain any probability terms.

Given a formula γ free of probability terms, consider the formula $\gamma^\dagger \stackrel{def}{=} (\gamma \sqcap (\sum_{A \subseteq \mathsf{qB}} ||\top\rangle_A|^2 = 1))$. Now γ is consistent iff $\gamma^\dagger$ is consistent. Now, for each $A \subseteq \mathsf{qB}$, pick two fresh variables x_A and y_A. Consider the formula $\gamma^{\dagger\dagger}$ obtained from $\gamma^\dagger$ by replacing each term $\mathrm{Re}(|\top\rangle_A)$ be x_A and $\mathrm{Im}(|\top\rangle_A)$ by y_A. Now, by axiom **RCF**, $\gamma^\dagger$ is consistent if and only if $\gamma^{\dagger\dagger}$ is consistent over the reals. Observe that $\gamma^{\dagger\dagger}$ is a purely analytical formula. Therefore there is an assignment, say ρ', that satisfies $\gamma^{\dagger\dagger}$ or otherwise $\vdash \boxminus \gamma^{\dagger\dagger}$ by **RCF**, and $\gamma^{\dagger\dagger}$ would not be consistent and neither would $\gamma^\dagger$, which is a contradiction. We conclude that there is such an assignment ρ', and from this assignment we can construct $|\psi\rangle$ and ρ that satisfies γ as required. $\diamond$

As there is an algorithm for deciding the consistency of analytical formulas [4], the proof of the Model Existence Lemma suggests an algorithm for deciding the consistency of quantum formulas. We shall now compute the complexity of one such algorithm. We shall need a few definitions for this.

A term t of the dEQPL is said to be a *polynomial* in variables $x_1, \ldots, x_k$ if t is of the form $(\sum m_{n_1,\ldots,n_k} x_1^{n_1} \ldots x_k^{n_k})$. The *degree* of a polynomial term is defined as expected. We will also assume for the rest of the paper that each polynomial is in a normal form: for any two summands $x_1^{n_1} \ldots x_k^{n_k}$ and $x_1^{n'_1} \ldots x_k^{n'_k}$ there is some j such that $n_j \neq n'_j$. Now, given a set of classical formulas $\mathcal{A} = \{\alpha_1, \ldots, \alpha_m\}$, a set of variables $\mathcal{V} = \{x_1, \ldots x_k, z_\alpha, \ldots, z_{\alpha_m}\}$ and a set of polynomials $\mathcal{P} = \{p_1, \ldots, p_s\}$ with variables in the set $\mathcal{V}$, we say that a comparison formula $(t \leq t')$ is an $(\mathcal{A}, \mathcal{V}, \mathcal{P})$-*atom* if t' is 0 and there is some polynomial term $p \in \mathcal{P}$ such that replacing *all* occurrences of the variables z_{α_i} by $(\int \alpha_i)$ for each $(1 \leq i \leq m)$ yields t. A dEQPL formula γ is said to be a $(\mathcal{A}, \mathcal{V}, \mathcal{P})$-*formula* if each comparison formula occurring in γ is an $(\mathcal{A}, \mathcal{V}, \mathcal{P})$-atom. We have:

Theorem 2.3 Let the set qB have n elements. Let $\mathcal{A} = \{\alpha_1, \ldots, \alpha_m\}$ be a set of classical formulas, $\mathcal{V} = \{x_1, \ldots x_k, z_\alpha, \ldots, z_{\alpha_m}\}$ be a set of variables and $\mathcal{P} = \{p_1, \ldots, p_s\}$ be a set of polynomials with variables in $\mathcal{V}$. Let the degree of each polynomial in $\mathcal{P}$ be bounded by d and let $r = 2^{n+1} + k + m$. Then, assuming that all basic integer operations take unit time, there is an $\mathsf{O}(|\gamma|(s + m + 1)^r (\max(d, 2))^{\mathsf{O}(r)})$ algorithm to decide the whether an $(\mathcal{A}, \mathcal{V}, \mathcal{P})$-formula γ is a theorem or not.

Proof: For each $\alpha_i \in \mathcal{V}$ compute the set $\mathcal{B}_i = \{A \subseteq \mathsf{qB} \mid v_A \Vdash \alpha_i\}$. Computation of each $\mathcal{B}_i$ takes at most $\mathsf{O}(2^n|\alpha_i|)$ steps, where $|\alpha_i|$ is the length of α_i. Since the sum $(\sum_{1\leq im} |\alpha_i|)$ is less than $|\gamma|$, this whole computation takes at most $\mathsf{O}(2^n|\gamma|)$ steps. Please note that $(2^n|\gamma|)$ is bounded by $|\gamma|(s+m+1)^r(\max(d,2))^{\mathsf{O}(r)}$.

Given a $(\mathcal{A},\mathcal{V},\mathcal{P})$-formula γ, let γ_1 be the formula obtained by replacing all probability terms $(\int \alpha_i)$ by z_{α_i}. Now, for each $A \subset \mathsf{qB}$, pick two fresh variables x_A and y_A and consider the formula

$$\gamma^\dagger = \gamma_1 \sqcap (\textstyle\prod_{1\leq i\leq m}(z^2_{\alpha_i} - \sum_{A\in B_i}(x_A^2+y_A^2)=0)) \sqcap ((\sum_{A\subseteq \mathsf{qB}} x_A^2+y_A^2)-1=0).$$

We make a few observations here:

- γ is consistent iff and only if $\gamma^\dagger$ is.
- $\gamma^\dagger$ is purely analytical.
- $\gamma^\dagger$ is built from comparison formulas of the form $(p \leq 0)$ or $(p=0)$ where each p is a polynomial in the set

  $$\mathcal{P}' = \mathcal{P} \cup \{(z^2_{\alpha_i} - \textstyle\sum_{A\in B_i}(x_A^2+y_A^2)) | 1 \leq i \leq m\} \cup \{(\sum_{A\subseteq \mathsf{qB}} x_A^2+y_A^2)-1\}.$$

- $\mathcal{P}'$ has $(s+m+1)$ polynomials. The degree of each polynomial is bounded by $\max(d,2)$ and is built from $r = 2^{n+1}+k+m$ variables.
- The length of $\gamma^\dagger$ is $\mathrm{O}(|\gamma| + m(\max(d,2)^{\mathrm{O}(r)}))$.

Assuming that integer operations take unit time, [4] then gives an $\mathsf{O}(|\gamma|(s+m+1)^r(\max(d,2))^{\mathsf{O}(r)})$ algorithm to decide consistency of $\gamma^\dagger$ which concludes the proof of the corollary . $\diamond$

3 Temporal logic

We now introduce a temporal version of dEQPL by adopting the temporal modalities of computational tree logic [8]. The logic is interpreted over a transition system in which the states are quantum states and the transitions are unitary operators. We also provide a sound proof system by enriching the usual CTL proof system with the axioms of the quantum state logic.

Table 4: Language of QCTL

Temporal quantum formulas

$\theta \quad := \quad \gamma \,[\!]\, (\theta \sqsupset \theta) \,[\!]\, (\mathsf{EX}\theta) \,[\!]\, (\mathsf{AF}\theta) \,[\!]\, \mathsf{E}[\theta \mathsf{U} \theta]$,
where γ is a dEQPL formula.

3.1 Languages and semantics

Syntax. The formulas of Quantum Computation Tree Logic (QCTL) are depicted in Table 4 and are obtained by enriching the quantum formulas with CTL modalities.

The intuitive semantics of the temporal modalities is similar to those in classical CTL. The modalities are composed by two symbols where the first one is chosen between E or A and the second one amongst X, F, G and the bi-modality U. The second symbol is used for temporal reasoning: X stands for ne**x**t; F for sometime in the **f**uture; G for always in the future; and U for **u**ntil. The first symbol quantifies over all computation paths: an existential (E - for there **e**xists) path or a universal (A - for **a**ll) paths. The combination of the two symbols can be easily intuited. For example, the formula $(\mathsf{EX}\theta)$ holds in a state $|\psi\rangle$ if there **e**xists a ne**x**t state of $|\psi\rangle$ (that is, a state reachable from $|\psi\rangle$ with a single transition) that satisfies θ. As usual, all CTL modalities are obtained as abbreviations from EX, AF and EU.

- $(\mathsf{AX}\theta)$ for $\boxminus\,(\mathsf{EX}(\boxminus\,\theta))$;
- $(\mathsf{EF}\theta)$ for $\boxminus(\mathsf{E}[(\boxminus\,\perp\!\!\!\perp)\mathsf{U}\theta])$;
- $(\mathsf{AG}\theta)$ for $\boxminus(\mathsf{EF}(\boxminus\,\theta))$;
- $(\mathsf{EG}\theta)$ for $\boxminus(\mathsf{AF}(\boxminus\,\theta))$;
- $\mathsf{A}[\theta_1 \mathsf{U} \theta_2]$ for $\boxminus(\mathsf{E}[(\boxminus\,\theta_2)\mathsf{U}(\boxminus\,\theta_1 \sqcap \boxminus\,\theta_2)]) \sqcap (\boxminus(\mathsf{EG}(\boxminus\,\theta_2)))$.

Semantics. In order too provide semantics to the logic, we introduce a very simple notion of quantum transition system.

Definition 3.1 (Quantum transition system) A *finite quantum transition system* over qB is a tuple

$$\mathcal{T} = (A, I, \{U_a\}_{a \in A})$$

where:

- A is the finite *set of actions*;
- I is a finite set of unit vectors in $\mathcal{H}_{\mathsf{qB}}$ called the *set of initial states*;
- $\{U_a\}_{a\in A}$ is a family of pairwise distinct unitary *transition operators* over $\mathcal{H}_{\mathsf{qB}}$;

such that the set of *reachable states* $S_{\mathcal{T}} = \{U_w|\psi\rangle : w \in A^*, |\psi\rangle \in I\}$ is finite[1]. The value $|S_{\mathcal{T}}|$ is called the *size* of $\mathcal{T}$.

The concept of quantum transition system presented above is inspired by classical transition systems. Some modifications are needed in order to cope with the quantum postulates, such as, the fact that states are unit vectors of an Hilbert space and that transitions are defined by unitary transformations.

For the sake of simplicity, we are not considering generalized measurements. However, we will be able to reason about protocols where measurements in the standard computational basis are performed at the end of the protocol, thanks to the probability terms $\int \alpha$ in the state logic. Similarly, classical states (bits) can be simulated by quantum states (qubits) that remain in the computational basis throughout the transitions.

The temporal language is interpreted over a quantum transition system $\mathcal{T}$, a state $|\psi\rangle \in S_{\mathcal{T}}$ and an assignment ρ. We also assume that, given an assignment, the values of the variables does not change with state transitions. The rigorous semantics of the logic is given in Table 5.

Table 5: Semantics of QCTL

$\mathcal{T}\|\psi\rangle\rho \Vdash \gamma$	iff	$\|\psi\rangle\rho \Vdash \gamma$;
$\mathcal{T}\|\psi\rangle\rho \Vdash (\theta_1 \sqsupset \theta_2)$	iff	$\mathcal{T}\|\psi\rangle\rho \nVdash \theta_1$ or $\mathcal{T}\|\psi\rangle\rho \Vdash \theta_2$
$\mathcal{T}\|\psi\rangle\rho \Vdash (\mathsf{EX}\theta)$	iff	$\mathcal{T}U_a\|\psi\rangle\rho \Vdash \theta$ for some $a \in A$;
$\mathcal{T}\|\psi\rangle\rho \Vdash (\mathsf{AF}\theta)$	iff	$\mathcal{T}\|\psi\rangle\rho \Vdash \theta$ or for all $a \in A$ there is a word $w \in A^*$ such that $\mathcal{T}U_aU_w\|\psi\rangle\rho \Vdash \theta$;
$\mathcal{T}\|\psi\rangle\rho \Vdash \mathsf{E}[\theta_1\mathsf{U}\theta_2]$	iff	$\mathcal{T}\|\psi\rangle\rho \Vdash \theta_2$ or for some $a \in A$ there is word $w \in A^*$ such that $\mathcal{T}U_{aw}\|\psi\rangle\rho \Vdash \theta_2$ and $\mathcal{T}U_s\|\psi\rangle\rho \Vdash \theta_1$ for every prefix $s \prec w$.

[1]We extend the transition operators to words as expected: $U_\varepsilon = I$ and $U_{wa} = U_w.U_a$ with ε the empty word.

It is easy to see that for closed formulas (*that is*, formulas without variables), we can drop the assignment in the interpretation side of the satisfaction relation. A quantum transition system $\mathcal{T}$ is said to satisfy a temporal formula θ, which we denote by $\mathcal{T} \Vdash \theta$, if $\mathcal{T}, |\psi\rangle, \rho \Vdash \theta$ for all $|\psi\rangle \in I$ and assignment ρ.

3.2 The Model-checking problem

We now address the problem of model-checking a closed temporal formula. Following the usual model-checking technique for CTL, the goal is to compute the set

$$Sat_{\mathcal{T}}(\theta) := \{|\psi\rangle \in S_{\mathcal{T}} : \mathcal{T}, |\psi\rangle \Vdash \theta\}$$

for a given quantum transition system $\mathcal{T}$ and closed formula θ. This is called the global model-checking problem. Thus, $\mathcal{T} \Vdash \theta$ iff the set of initial states I is contained in $Sat_{\mathcal{T}}(\theta)$.

For the model-checking algorithm, given a transition system $\mathcal{T} = (A, I, \{U_a\}_{a\in A})$ with $S_{\mathcal{T}}$ as the set of reachable states, we shall assume that each unitary operator U_a is input as a set of ordered pairs: $\{(\psi, U_a(\psi_2)) : \psi \in S_{\mathcal{T}}\}$ (instead of the usual matrix representation. The (global) model-checking algorithm is given in Table 6.

Table 6: Algorithm to determine $Sat_{\mathcal{T}}(\theta)$

1. $Sat_{\mathcal{T}}(\gamma) = \{|\psi\rangle \in S_{\mathcal{T}} : |\psi\rangle \Vdash \gamma\}$;
2. $Sat_{\mathcal{T}}(\theta_1 \sqsupset \theta_2) = (S_{\mathcal{T}} \setminus Sat_{\mathcal{T}}(\theta_1)) \cup Sat_{\mathcal{T}}(\theta_2)$
3. $Sat_{\mathcal{T}}(\mathsf{EX}\theta) = \bigcup_{a\in A} U_a^{-1} Sat_{\mathcal{T}}(\theta)$;
4. $Sat_{\mathcal{T}}(\mathsf{AF}\theta) = \mathbf{FixedPoint}[\lambda X.\{\cap_{a\in A} U_a^{-1} X\} \bigcup X, Sat_{\mathcal{T}}(\theta)]$;
5. $Sat_{\mathcal{T}}(\mathsf{E}[\theta_1 \mathsf{U} \theta_2]) =$
   $\mathbf{FixedPoint}[\lambda X.\{\bigcup_{a\in A}\{U_a^{-1} X \bigcap Sat_{\mathcal{T}}(\theta_1)\}, Sat_{\mathcal{T}}(\theta_2)]$.

Clearly, quantum transition systems require, in general, exponential space (over the number of qubits) to simulate with classical computers due to the exponential number of possible state superpositions (observe that we allow arbitrary quantum states as initial states). For this reason, the model checking algorithm takes exponential time on the number of qubits, but it

is polynomial on the size of the transition system and the complexity of the formula.

Theorem 3.2 Assuming that all basic arithmetical operations take unit time, the algorithm in Table 6 takes $O(|\theta|^2.|S_{\mathcal{T}}|^2.2^n)$ time.

Proof: The propositional CTL model-checking algorithm takes $O(|\theta|.|S_{\mathcal{T}}|^2)$ (see [8] for a detailed analysis). So, if we consider each quantum atom to be a propositional symbol, the time complexity of the algorithm would be $O(|\theta|.|S_{\mathcal{T}}|^2)$. Finally, since checking if a quantum atom is satisfied by a quantum states takes $O(|\theta|.2^n)$ (c.f. Theorem 2.1) we derive the desired upper bound. Recall that we consider all arithmetic computations to be $O(1)$ by using floating point representation for the real numbers. $\diamond$

3.3 Axiomatization

Given a complete axiomatization for the state logic it is easy to establish a sound proof system for the CTL extension.

We are currently investigating if the system present in Table 7 is (weakly) complete. Towards this end, we are working to establish a small model-property for the quantum temporal logic and understand the restrictions imposed by unitary transformations. For the moment, we only have the following result.

Theorem 3.3 The proof system presented in Table 7 is sound.

4 Example: BB84 protocol

In this section we reason about a simplified version of the BB84 key distribution protocol [5] to illustrate the power of QCTL. We assume the reader is conversant with this protocol since it will not be presented here.

For the sake of simplicity, we consider that the protocol distributes a key of one bit. The property we desire to model check is the soundness of the protocol, that is, if there is no interference by Eve (and no decoherence occurs) Alice and Bob will obtain the same key (provided they chose the same basis).

We start by presenting the protocol as a quantum transition system with five bits $\{b_A, b_B, k, s, e\}$ and one qubit $\{m\}$. Bit b_A encodes the basis that Alice will use to send the key k through qubit m. So, Alice sends the qubit m to Bob at the following state depending on the values of b_A and k:

Table 7: Axioms for QCTL

Axioms

[**QTeo**]		All dEQPL theorems;
[**Taut**]		All tautologies with propositional symbols substituted by QCTL formulas;
[**EX**]	$\vdash$	$(\mathsf{EX}(\theta_1 \sqcup \theta_2)) \equiv (\mathsf{EX}\theta_1) \sqcup (\mathsf{EX}\theta_2)$
[**X**]	$\vdash$	$(\mathsf{AX}(\boxminus \perp\!\!\!\perp)) \sqcap (\mathsf{EX}(\boxminus \perp\!\!\!\perp))$
[**EU**]	$\vdash$	$\mathsf{E}[\theta_1 \mathsf{U} \theta_2] \equiv \theta_2 \sqcup (\theta_1 \sqcap (\mathsf{EXE}[\theta_1 \mathsf{U} \theta_2]))$
[**AU**]	$\vdash$	$\mathsf{A}[\theta_1 \mathsf{U} \theta_2] \equiv \theta_2 \sqcup (\theta_1 \sqcap (\mathsf{AXA}[\theta_1 \mathsf{U} \theta_2]))$
[**AG1**]	$\vdash$	$(\mathsf{AG}(\theta_3 \sqsupset ((\boxminus \theta_2) \sqcap (\mathsf{EX}\theta_3)))) \sqsupset$ $(\theta_3 \sqsupset (\boxminus \mathsf{A}[\theta_1 \mathsf{U} \theta_2]))$
[**AG2**]	$\vdash$	$(\mathsf{AG}(\theta_3 \sqsupset ((\boxminus \theta_2) \sqcap (\theta_1 \sqsupset (\mathsf{AX}\theta_3))))) \sqsupset$ $(\theta_3 \sqsupset (\boxminus \mathsf{E}[\theta_1 \mathsf{U} \theta_2]))$
[**AG3**]	$\vdash$	$(\mathsf{AG}(\theta_1 \sqsupset \theta_2)) \sqsupset ((\mathsf{EX}\theta_1) \sqsupset (\mathsf{EX}\theta_2))$

Inference rules

[**QMP**]	$\theta_1, (\theta_1 \sqsupset \theta_2) \vdash \theta_2$
[**AGen**]	$\theta_1 \vdash (\mathsf{AG}\theta_1)$

- $|0\rangle$ if $b_A = k = 0$;
- $|1\rangle$ if $b_A = 0$ and $k = 1$;
- $\frac{1}{2}(|0\rangle + |1\rangle)$ if $b_A = 1$ and $k = 0$;
- $\frac{1}{2}(|0\rangle - |1\rangle)$ if $b_A = k = 1$.

Similarly, b_B encodes the basis that Bob will use to observe the qubit m he receives. Since we only allow measurements over the computational basis, if $b_B = 1$ (that is, Bob should use the diagonal measurement) Bob applies a unitary transformation to m in order to obtain the same value measuring with the computational basis that he would using the diagonal basis.

The bits s and e are used to model the status of the protocol. Bit s takes value 1 if Alice has just sent a message to Bob and value 0 otherwise. Bit e indicates if the protocol has ended or not. So, the evolution of the pair (s, e) throughout the protocol is $(0, 0) \to (1, 0) \to (0, 1)$.

Recall that bits are modeled by qubits that remain in the computational basis. Thus, the states of the bits b_A, b_B, k, s, and e will be modeled by the

elements of the computational basis $\{|b_A, b_B, k, s, e\rangle : b_a, b_B, k, s, e \in \{0, 1\}\}$. We consider the qubit m to be initialized to $|0\rangle$ and, so, the set of initial states has eight elements: $I = \{|b_a, b_B, k, 0, 0, 0\rangle : b_a, b_B, k \in \{0, 1\}\}$.

We consider that there is only action symbol a. The unitary transformation U_a can be easily described as a concatenation of two unitary operator $U_a = U_r.U_s$, where U_s deals with Alice sending the message to Bob and U_r with Bob receiving the message. The idea is that U_r behaves like the identity if the qubit was not sent by Alice while U_s will behave like the identity otherwise. Both U_s and U_r are easily described as controlled operations. The operator U_s is $U_{s4}.U_{s3}.U_{s2}.U_{s1}$ where:

- $U_{s1}|0, b_B, 1, 0, 0, m\rangle = |0, b_B, 1, 0, 0, 1 - m\rangle$ and behaves like the identity for the remaining elements of the basis;
- $U_{s2}|1, b_B, 0, 0, 0, m\rangle = |1, b_B, 0, 0, 0\rangle \otimes H|m\rangle$ and behaves like the identity for the remaining elements of the basis where H is the Hadamard transformation;
- $U_{s3}|1, b_B, 1, 0, 0, m\rangle = |1, b_B, 1, 0, 0\rangle \otimes H|1 - m\rangle$ and behaves like the identity for the remaining elements of the basis;
- $U_{s4}|b_a, b_B, k, s, 0, m\rangle = |b_a, b_B, k, 1 - s, 0, m\rangle$ and behaves like the identity for the remaining elements of the basis.

The unitary transformations U_{s1}, U_{s2} and U_{s3} deal with Alice encoding m to Bob and U_{s4} updates the state of the pair (s, e) from $(0, 0)$ to $(1, 0)$.

Similarly, the operator U_r is described by $U_{r2}.U_{r1}$ where:

- $U_{r1}|b_A, 1, k, 0, 0, m\rangle = |b_A, 1, k, 0, 0\rangle \otimes H|m\rangle$ and behaves like the identity for the remaining elements of the basis;
- $U_{r2}|b_A, b_B, k, 0, e, m\rangle = |b_A, b_B, k, 0, 1 - e, m\rangle$ and behaves like the identity for the remaining elements of the basis.

The unitary transformation U_{r1} deals with the change of basis that Bob performs when $b_B = 1$ and U_{r2} (together with U_{s4}) updates the state of the pair (s, e) from $(1, 0)$ to $(0, 1)$ (note that U_{s4} changes the state of (s, e) from $(1, 0)$ to $(0, 0)$ and that U_{r2} then changes it to $(0, 1)$).

The BB84 protocol is described by two applications of U_a over an initial state. At the end of the protocol a measurement is performed by Bob over the qubit m. Thus, the quantum transition system modeling the simple BB84 protocol is given by $\mathcal{T} = (\{a\}, I, U_a)$ with $qB = \{b_A, b_B, k, s, e, m\}$.

The soundness property states that if $b_A = b_B$ then at the end of the protocol, the key k should be the same as the value that Bob observes in m. This property can be described by the formula θ below:

$$(\Box(b_A \Leftrightarrow b_B)) \sqsupset (\mathsf{A}[(\boxminus(\Box e))\mathsf{U}((\Box e) \sqcap ((\Box k) \equiv (\textstyle\int m = 1))]).$$

It is now possible to use the algorithm in Table 6 to check that $\mathcal{T} \Vdash \theta$.

5 Conclusions

We present a temporal quantum logic combining the quantum state logic given in [7] with the computational tree logic (CTL) [8]. The model-checking algorithm of CTL was extended to deal with quantum states. The use of the quantum temporal logic was illustrated with BB84 protocol [5].

This work can be extended in several directions. First, on the state logic part, density operators could replace unit vectors thus giving a global phase independent semantics. On the temporal part, quantum transition systems should allow arbitrary measurements. For this, the state logic based on density operators is more suitable. We also plan to investigate other temporal extensions to quantum logic, like linear temporal logic and full branching time logic.

On the algorithmic side, the complexity class of the SAT and the model-checking problem for both the state and the temporal logic need to be investigated.

Acknowledgments

We thank the anonymous referees for their comments which have greatly helped the exposition. This work was partially supported by FCT and EU FEDER, namely via CLC POCTI (Research Unit 1-601), QuantLog project POCI/MAT/55796/2004 and within the recent QSec initiative of SQIG-IT. R. Chadha was also supported by the FCT and EU FEDER postdoctoral fellowship SFRH/BPD/26137/2005. Pedro Baltazar was also supported by FCT and EU FEDER PhD fellowship SFRH/BD/22698/2005.

The author R. Chadha is now at Dept of Computer Science, Univ of Illinois, Urbana-Chmapaign, USA.

References

[1] S. Abramsky and B. Coecke. A categorical semantics of quantum protocols. In *Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science (LICS 2004)*, pages 415–425. IEEE Computer Science Press, 2004.

[2] T. Altenkirch and J. Grattage. A functional quantum programming language. In *Proceedings of the 20th Annual IEEE Symposium on Logic in Computer Science (LICS)*, pages 249–258. IEEE Computer Society, 2005.

[3] A. Baltag and S. Smets. LQP: The dynamic logic of quantum information. *Mathematical Structures in Computer Science*, 2006. To appear.

[4] S. Basu, R. Pollack, and R. Marie-Françoise. *Algorithms in Real Algebraic Geometry*. Springer, 2003.

[5] C. H. Bennett and G. Brassard. Quantum cryptography: Public key distribution and coin tossing. In *Proceeding of IEEE International Conference on Computers, Systems and Signal Processing*, pages 175–179. IEEE, 1984.

[6] R. Chadha, P. Mateus, and A. Sernadas. Reasoning about quantum imperative programs. *Electronic Notes in Theoretical Computer Science*, 158:19–40, 2006. Invited talk at the Twenty-second Conference on the Mathematical Foundations of Programming Semantics.

[7] R. Chadha, P. Mateus, A. Sernadas, and C. Sernadas. Extending classical logic for reasoning about quantum systems. Preprint, CLC, Department of Mathematics, Instituto Superior Técnico, 1049-001 Lisboa, Portugal, 2005.

[8] E. M. Clarke and E. A. Emerson. Design and synthesis of synchronization skeletons using branching time temporal logics. In *Proceeding of the Workshop on Logics of Programs*, volume 131 of *LNCS*. Springer-Verlag, 1981.

[9] E. M. Clarke and J. M. Wing. Formal methods: state of the art and future directions. *ACM Comput. Surv.*, 28(4):626–643, 1996.

[10] E. D'Hondt and P. Panangaden. Quantum weakest preconditions. In Peter Selinger, editor, *Proceedings of the 2nd International Workshop*

on Quantum Programming Languages, number 33 in TUCS General Publications, pages 75–90. Turku Centre for Computer Science, 2004.

[11] R. Fagin, J. Y. Halpern, and N. Megiddo. A logic for reasoning about probabilities. *Information and Computation*, 87(1-2):78–128, 1990.

[12] E. Knill. Conventions for quantum pseudocode. Technical Report LAUR-96-2724, Los Alamos National Laboratory, 1996.

[13] P. Mateus and A. Sernadas. Weakly complete axiomatization of exogenous quantum propositional logic. *Information and Computation*, to appear.

[14] C. Meadows. Formal methods for cryptographic protocol analysis: emerging issues and trends. *IEEE Journal on Selected Areas in Communications*, 21(1):44– 54, 2003.

[15] P. Naur. Revised report on the algorithmic language Algol 60. *The Computer Journal*, 5:349–367, 1963.

[16] J. W. Sanders and P. Zuliani. Quantum programming. In *Mathematics of Program Construction*, volume 1837 of *Lecture Notes in Computer Science*, pages 80–99. Springer, 2000.

[17] P. Selinger and B. Valiron. A lambda calculus for quantum computation with classical control. In *Proceedings of the 7th International Conference on Typed Lambda Calculi and Applications (TLCA)*, volume 3461 of *Lecture Notes in Computer Science*, pages 354–368. Springer, 2005.

[18] R. van der Meyden and M. Patra. Knowledge in quantum systems. In M. Tennenholtz, editor, *Theoretical Aspects of Rationality and Knowledge*, pages 104–117. ACM, 2003.

[19] R. van der Meyden and M. Patra. A logic for probability in quantum systems. In M. Baaz and J. A. Makowsky, editors, *Computer Science Logic*, volume 2803 of *Lecture Notes in Computer Science*, pages 427–440. Springer-Verlag, 2003.