# QMC: A Model Checker for Quantum Systems

Simon J. Gay[1], Rajagopal Nagarajan[2⋆], and Nikolaos Papanikolaou[2⋆,⋆⋆]

[1] Department of Computing Science, University of Glasgow
simon@dcs.gla.ac.uk
[2] Department of Computer Science, University of Warwick
{biju,nikos}@dcs.warwick.ac.uk

**Abstract.** We introduce a model-checking tool intended specially for the analysis of quantum information protocols. The tool incorporates an efficient representation of a certain class of quantum circuits, namely those expressible in the so-called stabiliser formalism. Models of protocols are described using a simple, imperative style simulation language which includes commands for the unitary operators in the Clifford group as well as classical integer and boolean variables. Formulas for verification are expressed using a subset of exogenous quantum propositional logic (EQPL). The model-checking procedure treats quantum measurements as the source of non-determinism, leading to multiple protocol runs, one for each outcome. Verification is performed for each run.

## 1 Introduction and Background

The novel field of quantum computation and quantum information has been growing at a rapid rate; the study of quantum information in particular has led to the emergence of communication and cryptographic protocols with no classical analogues. Quantum information protocols have interesting properties which are not exhibited by their classical counterparts, but they are most distinguished for their applications in cryptography. Notable results include the unconditional security proof [1] of quantum key distribution [2, 3] and the impossibility proof of unconditionally secure quantum bit commitment [4]. The former of these results in particular is one of the reasons for the widespread interest in this field, and it demonstrates an achievement not possible in classical cryptographic systems.

The benefits of automated verification techniques are well known for classical communication protocols, especially in the cryptographic setting. *Model-checking* has been used to uncover subtle flaws in protocols and system designs [5, 6]. Our research programme is to apply similar techniques to quantum protocols with the expectation of gaining corresponding benefits. Today, while simulation tools for quantum information

⋆ Partially supported by the EU Sixth Framework Programme (Project SecoQC: *Development of a Global Network for Secure Communication based on Quantum Cryptography*).

⋆⋆ Partially supported by the EPSRC Network EP/E006833/1 on Semantics of Quantum Computation.

systems abound (see [7] for a list), to our knowledge no other authors have developed a tool aimed at verification.
In this paper we describe just such a tool, based on our earlier work [8, 9], named QMC (Quantum Model Checker); it allows for automated verification of properties of quantum systems. Properties to be verified are expressed using a subset of EQPL [10], a state logic designed specifically for quantum information. QMC analyses systems which can be expressed within the *stabiliser formalism*, which is known to be simulable in polynomial time. This is significant, as any kind of model checking necessarily involves simulation and, in general, quantum systems cannot be simulated efficiently on classical hardware. The systems expressible in this formalism are restricted, in the sense that the set of operations which they can perform is not universal for quantum computation. Nevertheless, stabiliser circuits are sufficient to describe a number of systems of practical interest.
Existing simulation tools for quantum systems [7] are designed to help the user understand the function of a given quantum circuit; some tools have a graphical user interface, and many allow the simulation of circuits with arbitrary quantum gates, even if there is a substantial computational cost due to the limited power of the classical machine running the simulation. Simulators which allow only stabiliser circuits include CHP [11] and GraphSim [12]; the algorithms in QMC are based on those used in the former of these two, as well as on particular algorithms developed in [13]. We do not know of any previous tool which provides automated checking of a circuit specification. Another distinctive characteristic of QMC is the automatic exploration of all possibilities generated by quantum measurements, which are probabilistic by nature [14].

*Background.* We assume here a familiarity with the basic concepts of quantum information, namely, qubits, unitary operators and projective measurements. For our purposes, a quantum system is regarded as consisting of a finite number of qubits, acted upon by applying particular operators and by performing measurements, which give rise to probabilistic outcomes. We confine ourselves to the states and operations which arise in the so-called stabiliser formalism; according to the Gottesman-Knill theorem [15], quantum circuits in this formalism are simulable in polynomial time on a classical computer. The reader is referred to [14] for more details.

## 2 Tool Description

The QMC tool allows the user to model-check a property of the final quantum state produced by a particular quantum protocol. A quantum protocol is perceived as a sequence of operations on both classical variables and a single quantum state consisting of $n$ qubits. Models of protocols are expressed using a simple, imperative-style language, while properties for verification are expressed using a subset of the logic EQPL [10].

```
1  init 3; // Initialise 3-qubit system state
2  int teleportme := 0; /* 0 = |0>, 1 = |1>, 2 =|0>+|1>, 3 =|0>-|1> */
3  if ((teleportme==1) \/ (teleportme==3)) do { X q0; };
4  if (teleportme>1) do { had q0; };
5  had q1; cnot q1 q2;
6  cnot q0 q1; had q0;
7  int a,b; a:= meas q0; b := meas q1;
8  if (b==1) do { X q2; }; if (a==1) do { Z q2; };
```

**Fig. 1.** Quantum teleportation expressed in QMC's modelling language.

The tool functions by simulating the protocol step-by-step; whenever a measurement occurs in a protocol, it gives rise to different runs of the protocol, one for each possible outcome. The EQPL formula specifying the desired protocol behaviour is checked on the final quantum state for each possible run.

## 3 Modelling Language

We will demonstrate the modelling language through the use of an example, namely, the quantum teleportation protocol [14]. Teleportation is a protocol that exploits quantum entanglement in order to transmit the state of a qubit using only two classical bits. In quantum mechanics, an *entangled state* is one which cannot be decomposed into the product of the states of its components.

The model shown in Figure 1 describes the protocol as a sequence of quantum operations on a three-qubit system. The qubit to be transmitted is qubit 0 (denoted `q0`); the second and third qubits (`q1`, `q2` respectively) are placed in an entangled quantum state, to be shared between the two protocol users. The sender possesses qubits `q0` and `q1`, and the receiver possesses `q2`. The purpose of teleportation is to enable an arbitrary qubit state to be transmitted by sending only two classical values, namely the outcomes of measurements made by the sender.

To begin with, there is a parameter which may be set so as to select which of several possible input states to supply to the protocol. Depending on this parameter, the desired quantum state is constructed by applying a suitable combination of the $X$ and $H$ (Hadamard) operators (lines 2-4). The second and third qubits are placed in an entangled state (line 5). The main part of the protocol is contained in lines 6-8. After applying particular operations to `q0` (line 6), Alice measures `q0` and `q1` and stores the results $a, b$ (line 7). Then the operator $X^b Z^a$ is applied to the third qubit (line 8). The net effect of this procedure is that the state of the third qubit at the end of the protocol is the same as the original state of the first qubit at the outset. Thus teleportation makes it possible for the state of one qubit to be transferred from one user to another, using the properties of entanglement.

# 4 Property Specification

A property is always checked against a single quantum state, in particular, the final state of the whole $n$-qubit system at the end of a protocol. The logic used for specifying properties of a protocol is a subset of the state logic EQPL [10]. The meaning of formulae in EQPL is expressed in terms of valuations, which are truth-value assignments for the symbols $q_0, q_1, \cdots, q_n$ corresponding to each qubit in the system. For instance, the quantum state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ is understood as a pair of valuations $(v_1, v_2)$ for a 2-qubit system such that $v_1(q_0) = 0$, $v_1(q_1) = 0$, $v_2(q_0) = 1$, $v_2(q_1) = 1$.
The formulae accepted by QMC for verification allow the user to reason about the state of individual qubits, and involve usual logical connectives such as negation and implication. There are two levels of formulae: classical formulae, which hold only if all valuations in a state satisfy them, and quantum formulae, which are essentially logical combinations of classical formulae. For instance, the quantum conjunction in the formula $\phi_1 \curlywedge \phi_2$ is only satisfied if both the classical formulae $\phi_1$ and $\phi_2$ are satisfied in the current state. A particularly distinctive type of quantum formula is of the form $[Q]$, where $Q$ is a list of qubit symbols; this type of formula is satisfied only if the qubits listed are disentangled from all other qubits in the system. For more details the reader is referred to [10].

## 4.1 Example

The requirement for the teleportation protocol described in section 3 is that, at the end of the protocol, no matter the measurement outcomes, the third qubit will be in the same state as the first qubit was to begin with, and this qubit will be disentangled from the rest of the system. We can express this requirement, for the case where the input is the quantum state $|0\rangle$, in the input language of QMC using the statement

```
formula ([q2]) #/\ (!q2);
```

which corresponds to the EQPL formula $[q_2] \curlywedge (\neg q_2)$. The first part of the formula asserts that the last qubit (`q2`) is disentangled from the rest of the system, while the second part asserts that the current valuation assigns to this qubit a value of 0. The entire formula is true if both parts are true, indicated by the connective of quantum conjunction `#/\`.

# 5 Verification

QMC implements algorithms for evaluating EQPL formulas over stabiliser states, which are represented internally using a matrix representation (see [11]). In order to check the truth of a particular formula, its truth need to be determined for all possible valuations; the tool automatically extracts all valuations from the internal representation.
More interestingly, the tool has been designed to explore all possible executions of a particular protocol arising from different measurement outcomes. Quantum measurement is known to be probabilistic, although at

the moment QMC treats it as a source of nondeterminism. Each possible measurement outcome gives rise to a different run of the protocol model, and formulae supplied for verification are automatically checked on the final state produced by each such run. The teleportation example described in previous sections has been model-checked in this manner, and shown to perform its intended function for a given input, for all possible measurement results.

## 6 Conclusion and Future Work

We have described QMC, a model-checking tool for quantum protocols. As far as we know, it is the first dedicated verification tool (as opposed to simulation systems) for quantum protocols. QMC allows the modelling and verification of properties of protocols expressible in the quantum stabiliser formalism. The logic for expressing properties is a subset of EQPL.

We intend to extend QMC in several ways. First, to implement the whole of EQPL, including its constructs for specifying numerical probabilities and coefficients in a quantum state; eventually we will implement a temporal extension of EQPL [16] (note that despite the title, that paper does not describe a model-checking tool). Experience with case studies is likely to feed into future development of these logics. Second, to support calculation of the probability that a logical property is satisfied. Third, to use a more expressive modelling language, such as the quantum process calculus CQP [17].

## References


[1] Mayers, D.: Unconditional security in quantum cryptography. Journal of the ACM **48** (2001) 351—406

[2] Bennett, C.H., Brassard, G.: Quantum cryptography: Public key distribution and coin tossing. In: Proceedings of International Conference on Computers, Systems and Signal Processing. (1984)

[3] Ekert, A.: Quantum cryptography based on Bell's theorem. Physical Review Letters **67** (1991) 661—663

[4] Mayers, D.: Unconditionally secure quantum bit commitment is impossible. In: Fourth Workshop on Physics and Computation — PhysComp '96, Springer–Verlag (1996)

[5] Ryan, P., Schneider, S., Goldsmith, M., Lowe, G., Roscoe, B.: Modelling and Analysis of Security Protocols. Pearson Education (2001)

[6] Holzmann, G.: The SPIN Model Checker: Primer and Reference Manual. Pearson Education (2003)

[7] Glendinning, I.: Links on simulation, modelling, and error prevention for quantum computers (2006) `http://www.vcpc.univie.ac.at/~ian/hotlist/qc/simulation.shtml`.

[8] Nagarajan, R., Gay, S.J.: Formal verification of quantum protocols. Available at arXiv.org. Record: quant-ph/0203086 (2002)

[9] Gay, S.J., Nagarajan, R., Papanikolaou, N.: Probabilistic model–checking of quantum protocols. DCM 2006: Proceedings of the 2nd International Workshop on Developments in Computational Models (2005) arXiv:quant-ph/0504007.
[10] Mateus, P., Sernadas, A.: Weakly complete axiomatization of exogenous quantum propositional logic. Information and Computation **204** (2006) 771—794
[11] Aaronson, S., Gottesman, D.: Improved simulation of stabilizer circuits. Physical Review A **70** (2004)
[12] Anders, S., Briegel, H.: Fast simulation of stabilizer circuits using a graph state representation. Physical Review A **73** (2006)
[13] Audenaert, K., Plenio, M.: Entanglement on mixed stabiliser states: Normal forms and reduction procedures. New Journal of Physics **7** (2005)
[14] Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information. Cambridge University Press (2000)
[15] Gottesman, D.: The Heisenberg representation of quantum computers. In Corney, S., Delbourgo, R., Jarvis, P., eds.: Group22: Proceedings of the XXII International Colloquium on Group Theoretical Methods in Physics, International Press (1999)
[16] P. Baltazar, R. Chadha, P.M., Sernadas, A.: Towards model-checking quantum security protocols. In P. Dini *et al.*, ed.: Proceedings of the First Workshop on Quantum Security: QSec'07, IEEE Press (2007)
[17] Gay, S.J., Nagarajan, R.: Communicating quantum processes. In: POPL '05: Proceedings of the 32nd ACM Symposium on Principles of Programming Languages, Long Beach, California. (2005)

## 7 Appendix - Tool Demonstration

QMC will be demonstrated by a presentation in three parts. First, the motivation for the tool will be explained, and issues specific to quantum protocols will be described. Then, the user interface will be demonstrated, and a small example will be used to show the tool in simulation mode. The third part of the presentation will consist of the teleportation example, with model-checking over all possible measurement outcomes.