A Framework for Automated Verification of Quantum Protocols*

NICK PAPANIKOLAOU

nikos@dcs.warwick.ac.uk
http://www.warwick.ac.uk/go/nikos

*Joint work with RAJAGOPAL NAGARAJAN (Warwick) and SIMON GAY (Glasgow).

Quantum Computation and Information

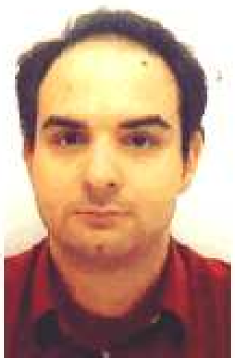
- Intensive research over past 10-20 years on quantum computation and quantum information
  - Chance of solving problems hitherto considered impossible
- Recent upsurge of interest
  - Implementation of practical quantum communication systems esp. quantum cryptography
- Increasing need for **design**, **simulation**, **analysis** tools
- Two levels of analysis:
  - **High-level:** properties of systems with both quantum & classical components
  - **Low-level:** properties of quantum subsystems, esp. quantum protocols and quantum algorithms

Our Research Programme

- To develop a verification tool enabling analysis of quantum protocols at both levels.
- We wish to facilitate automated reasoning about:
  - *Quantum state*
  - *Time*
  - *Knowledge* of agents
- Approach: **model-checking**

Raja

Simon



Nick

Background: Qubits

- Quantum bits (**qubits**): **superpositions** of basis vectors, e.g.:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

- Quantum state: a vector belonging to complex vector space (Hilbert space)
- Continuous state space; countably infinite
- n–qubit systems:
  - state space grows exponentially: 2^n basis vectors.
  - states are either:
    - either **decomposable** (products of individual states)
    - or **entangled** (cannot be decomposed)

Background: Quantum Gates

- Transformations or **operations on quantum states** are *linear* and *reversible*.

$$A|\psi\rangle = |\psi'\rangle \qquad \text{where } A^{-1}A = I \text{ and } A = A^{\dagger}$$

- **Quantum operators** or **quantum gates** are described by matrices.
- Common gates:
  - Controlled NOT (on 2 qubits) $\text{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$
  - Hadamard $H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
  - Pauli gates $\sigma_0, \ldots, \sigma_3$ (identity, bit flip, phase flip, bit and phase flip)
  - Phase gate Φ_θ (rotation by θ)

Background: Quantum Measurement

- The current state of any n-qubit system is unknown until it is measured.
- Measurement is **destructive** and **probabilistic**.
  - It collapses the current state to one of the n basis vectors at random.
- Measurement is the only way to extract a classical result from a quantum computation.
- **Example:** Measuring a qubit

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

  - with respect to basis $\{|0\rangle, |1\rangle\}$ gives $|0\rangle$ or $|1\rangle$ at random.
  - with respect to other basis $\{|a\rangle, |b\rangle\}$ gives $|a\rangle$ or $|b\rangle$ at random.

A Simple Protocol: Dense Coding

- Initial state of entangled pair shared by Alice and Bob:

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

1. To transmit integer n ($0 \leq n \leq 3$), Alice applies Pauli transformation σ_n to her qubit x.
2. She physically transfers qubit x to Bob.
3. Bob applies CNOT to qubits x and y.
4. Bob applies Hadamard to x.
5. Bob measures x and y. The result uniquely determines n.

A Modelling Language: CQP

- Simon Gay (Glasgow) and Rajagopal Nagarajan (Warwick) have developed a quantum process algebra, CQP, for modelling such protocols.
- CQP has a formal semantics and a type system.
- Example: modelling the dense coding protocol in CQP:

$$\mathit{Alice}(x : \mathsf{Qbit}, q : \widehat{\ }[\mathsf{Qbit}], n : 0..3) = \{x \mathrel{*}= \sigma_n\} \,.\, q![x] \,.\, \mathbf{0}$$

$$\mathit{Bob}(y : \mathsf{Qbit}, q : \widehat{\ }[\mathsf{Qbit}]) = q?[x : \mathsf{Qbit}] \,.\, \{x, y \mathrel{*}= \mathsf{CNot}\} \,.\, \{x \mathrel{*}= \mathsf{H}\} \,.\, \mathit{Use}(\mathsf{measure}\ x, y)$$

$$\mathit{System}(x : \mathsf{Qbit}, y : \mathsf{Qbit}, n : 0..3) = (\mathsf{new}\ q : \widehat{\ }[\mathsf{Qbit}])(\mathit{Alice}(x, q, n) \mid \mathit{Bob}(y, q))$$

Verification using PRISM

- PRISM: Probabilistic Model Checker (Kwiatkowska, Norman, Parker)
  - `http://www.cs.bham.ac.uk/~dxp/prism`
  - Suitable for verifying properties of concurrent systems exhibiting probabilism
- Quantum behaviour is inherently probabilistic
- We have used PRISM to analyse some simple properties of a quantum cryptographic protocol, as well as dense coding, teleportation, and more
  - not nearly as powerful as a general security proof
  - can only model a handful of qubits and steps

Finite State Spaces for Quantum Protocols

- Need to develop a general approach to:
  - identify finite set of quantum states in a Hilbert space of dimension n, which is closed under the operations that arise in a protocol
  - we did this manually for 2-3 qubits
- It turns out that we can represent states of interest by Pauli operators; a closed group of operations (the **Clifford group**) transforms any one Pauli operator into another Pauli operator (viz. *stabilizer formalism*)
  - The Clifford group operations are the ones which mostly arise in quantum protocols
  - So we only need to represent a handful of operators and their effects on one another in order to simulate a whole class of quantum protocols.

Towards a Verification Tool

- Research on the foundations of quantum theory led to the development of **quantum logic,** which differs from classical propositional logic.
- Some authors have developed quantum logics for reasoning about finite sets of qubits.
- A tool for checking whether a protocol model satisfies a given formula would be highly desirable.
- As is the case for classical security protocols, a tool which allows us to reason about:
  - knowledge of agents in a quantum protocol
  - quantum state at various times during the computation

  is extremely valuable, and is likely to assist protocol designers and implementors.

Review and Conclusion

- Overall we have reported on work-in-progress, namely the design and implementation of a verification tool for quantum protocols.
- We covered the basics of QCQI.
- We looked at a simple quantum protocol.
- We reviewed CQP and the use of PRISM.
- We discussed some of the considerations entering into the design of a practical verification tool.