Techniques for Formal Modelling and Analysis *of* Quantum Systems

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

[1] School of Computing Science, University of Glasgow, UK
Simon.Gay@glasgow.ac.uk

[2] Department of Computer Science, School of Science and Technology,
Middlesex University, London, UK
R.Nagarajan@mdx.ac.uk

Abstract. Quantum communication and cryptographic protocols are well on the way to becoming an important practical technology. Although a large amount of successful research has been done on proving their correctness, most of this work does not make use of familiar techniques from formal methods such as formal logics for specification, formal modelling languages, separation of levels of abstraction, and compositional analysis. We argue that these techniques will be necessary for the analysis of large-scale systems that combine quantum and classical components. We summarize the results of our investigation using different approaches: behavioural equivalence in process calculus, model-checking and equivalence checking. Quantum teleportation is used as an example to illustrate our techniques.

Prologue

We were both PhD students of Samson Abramsky, in the Theory and Formal Methods group, which he led, in the Department of Computing at Imperial College London. During our time at Imperial in the early 1990s, the group provided a superbly stimulating and well-resourced environment and established many lasting friendships. Samson's involvement in the "CONFER" project enabled us to go to a number of workshops around Europe, meeting other researchers and PhD students who remain colleagues to this day.

Samson's big ideas during that time were game semantics and interaction categories, both of which made use of structures drawn from the development of linear logic. Game semantics led to the definition of fully abstract models of functional programming languages, while interaction categories [2,3] aimed to provide a Curry-Howard-style logical basis for typed concurrent programming. Both of us worked on interaction categories; indeed, we worked closely together

[⋆] Partially supported by "Process Algebra Approach to Distributed Quantum Computation and Secure Quantum Communication", Australian Research Council Discovery Project DP110103473.

B. Coecke, L. Ong, and P. Panangaden (Eds.): Abramsky Festschrift, LNCS 7860, pp. 264–276, 2013.

during the overlapping period of our PhDs. After he had provided the basic theory of interaction categories (and there was a lot of it!), Samson gave us the freedom to explore its consequences and develop applications and examples.

In 1995 Samson moved to Edinburgh, and Simon moved to Royal Holloway to take up a lectureship. Game semantics became a large, active and successful research area, which occupied Samson for several more years. Simon's attention switched to π-calculus, which had been a sideline during his PhD, and especially to the topic of session types. Raja remained at Imperial for a few more years, working as a researcher on a joint project led by Chris Hankin and Samson.

In 2000, all of us relocated: Samson moved to Oxford, Simon to Glasgow and Raja took up a lectureship at Warwick. Around this time, both Raja and Samson independently became interested in quantum computing. Raja's interest was in the use of formal methods, successfully developed in classical computing, to analyse and verify quantum protocols. He recruited Simon to collaborate on an early application [27] of process calculus and model-checking to quantum systems. Simon's attention had also been caught by Peter Selinger's paper [29] on the denotational semantics of a quantum programming language, and there seemed to be exciting opportunities for new applications of familiar techniques from theoretical computer science. Samson, with Bob Coecke, developed the programme of categorical quantum mechanics [4], based on various elaborations of compact closed categories—a connection, at least formally, with interaction categories.

Thus we again found ourselves working in the same area as Samson, and enjoyed being involved in the development of these new angles on quantum information processing, drawing on techniques and tools from semantics. Ian Mackie and Simon organised the EPSRC-funded QNET network, with strong involvement and support from Samson's group in Oxford, which led to several successful workshops and the book *Semantic Techniques in Quantum Computation* [16], and has been followed by the Computer Science and Physics network run by Samson, Bob Coecke, Andreas Döring and Jamie Vicary.

The present paper, which we are delighted to be able to contribute to Samson's Festschrift volume, combines an introduction to the field of quantum information, and an overview of our own work in this area.

1 Introduction

Quantum computing and quantum communication (more generally, *quantum information processing*) appear in the media from time to time, usually with misleading statements about the principles of quantum mechanics, the nature of quantum information processing, and the power of quantum algorithms. In this article, we begin by clarifying the fundamental concepts of quantum information and discussing what quantum computing systems are and are not capable of. We then outline several reasons for being interested in quantum information processing. Moving on to the main theme, we first motivate the application of formal methods to quantum information processing. We then describe the different techniques we have used in specification and verification of quantum protocols, illustrating them with an example.

There are several reasons to be interested in quantum information processing. First, the subject is really about understanding the information-processing power permitted by the laws of physics, and this is a fundamental scientific question. Second, quantum algorithms might help to solve certain classes of problem more efficiently; if, however, NP-complete problems cannot be solved efficiently even by a quantum computer, then understanding why not is also a question of fundamental interest. Third, quantum cryptography provides a neat answer, in advance, to any threat that quantum computing might pose to classical cryptography. Fourth, as integrated circuit components become smaller, quantum effects become more difficult to avoid. Quantum computing might be necessary in order to continue the historical trend of miniaturization, even if it offers no complexity-theoretic improvement. Finally, Feynman [15] suggested that quantum computers could be used to simulate complex (quantum) physical systems whose behaviour is hard to analyze classically.

Will QIP become practically significant? Some aspects are already practical: there are companies selling Quantum Key Distribution systems today. Whether or not there is a real demand for quantum cryptography remains to be seen, but it seems likely that the promise of absolute security will attract organizations that feel they cannot take any chances. Quantum computing seems to be feasible in principle, although there are still formidable scientific and engineering challenges. But many experimental groups are working hard, and physicists and engineers are very clever. Remember that in 1949 the statement "In the future, computers may weigh no more than 1.5 tonnes" was a speculative prediction.

The remainder of this paper is organised as follows. In Section 2 we give a brief introduction to the main ideas of quantum information processing. In Section 3 we motivate the development of formal methods for quantum systems, and introduce the three strands that we have been working on. In Sections 4, 5 and 6 we explain, in turn, the use of process calculus, model-checking, and equivalence-checking, using quantum teleportation as an example in each case. Finally, Section 7 concludes.

2 Quantum Information Processing

The idea of quantum information processing (QIP) is to represent information by means of physical systems whose behaviour *must* be described by the laws of quantum physics. Typically this means very small systems, such as a single atom (in which the spin state, up or down, gives the basic binary distinction necessary for digital information representation) or a single photon (in which polarization directions are used). Information is then processed by means of operations that arise from quantum physics. Quantum mechanics leads to several fundamental properties of quantum information, which between them lead to various counter-intuitive effects and the possiblity of behaviour that cannot occur in classical systems.

2.1 Superposition

The state of a classical bit is either 0 or 1. The state of a quantum bit (qubit) is $\alpha|0\rangle + \beta|1\rangle$, where the states $|0\rangle$ and $|1\rangle$ are the *basis states* (in the *standard* or *computational* basis). In general, α and β are complex numbers and $|\alpha|^2 + |\beta|^2 = 1$. If both α and β are non-zero, then the state is a *superposition* of the basis states, for example $\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$. It is not correct to say, as often stated in the media, that a qubit can be in two states at once. It is in one state, but that state may be a superposition of the basis states. Note that any two orthogonal states may form a basis. For example, the pair $\{\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle, \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle\}$, sometimes written $\{|+\rangle, |-\rangle\}$, forms the *Hadamard* basis. Although we often work with the standard basis, it does not have a privileged status; indeed, whether or not a particular quantum state is regarded as a superposition depends on the choice of basis. The state $|+\rangle$ is in a superposition with respect to the standard basis, but not with respect to the Hadamard basis.

2.2 Measurement

It is not possible to inspect the contents of a quantum state. The most we can do is a measurement. Measuring a qubit that is in state $\alpha|0\rangle + \beta|1\rangle$, in the standard basis, has a random result: with probability $|\alpha|^2$ the result is $|0\rangle$, and with probability $|\beta|^2$ the result is $|1\rangle$. After the measurement, the qubit is in the basis state corresponding to the result.

2.3 Operations on a Superposition

An operation acts on every basis state in a superposition. For example, starting with the three-qubit state $\frac{1}{2}|000\rangle + \frac{1}{2}|010\rangle - \frac{1}{2}|110\rangle - \frac{1}{2}|111\rangle$ and applying the operation "invert the second bit" produces the state $\frac{1}{2}|010\rangle + \frac{1}{2}|000\rangle - \frac{1}{2}|100\rangle - \frac{1}{2}|101\rangle$. This is sometimes known as *quantum parallelism* and in the media it is often described as carrying out an operation simultaneously on a large number of values. However, it is not possible to discover the results of these simultaneous operations. A measurement would produce just one of the basis states. This is absolutely not a straightforward route to "parallelism for free".

2.4 No Cloning

It is not possible to define an operation that reliably makes a perfect copy of an unknown quantum state. This is known as the *no cloning theorem*. It contrasts sharply with the classical situation, where the existence of uniform copying procedures is one of the main advantages of digital information. Every word in the statement of the no cloning theorem is significant. For example, with the knowledge that a given qubit is either $|0\rangle$ or $|1\rangle$, it is possible to discover its state (by means of a simple measurement) and then set another qubit to the same state, thus creating a copy. It is also possible in general to create approximate copies, or to copy with a certain probability of perfect success but a certain probability of complete failure. It is possible to transfer an unknown quantum state from one physical carrier to another, but the process destroys the original state. This is known as *quantum teleportation*, and we will return to it later.

2.5 Entanglement

The states of two or more qubits can be correlated in a way that is stronger than any possible classical correlation. An example is the two-qubit state $\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$. Measuring either qubit produces, with equal probability, the state $|00\rangle$ or $|11\rangle$. Measuring the other qubit is then guaranteed to produce the same result as the first measurement. This correlation is preserved by quantum operations on the state, in a way that cannot be reproduced classically. This phenomenon is called *entanglement* and it is a key resource for quantum algorithms and communication protocols.

3 Formal Methods for QIP

The correctness of quantum algorithms and protocols can be analyzed mathematically. Simple protocols such as teleportation can be checked with a few lines of algebra, Shor's [30] and Grover's [23] algorithms have been extensively studied, and Mayers [26] and others have proved the security of quantum key distribution. But what about *systems*, which are constructed from separate components and combine quantum and classical computation and communication? Experience in classical computing science has shown that correctness of a complete implemented system is a very different question from correctness of the idealized mathematical protocol that it claims to implement. This is the *raison d'être* of the field of formal methods.

Our 2002 paper [27] suggested applying formal methods to quantum systems, with the same motivation as for classical systems:

- *formal modelling languages*, for unambiguous definitions;
- analysis of *systems*, rather than idealized situations;
- *systematic verification methodologies*, rather than *ad hoc* reasoning;
- the possibility of *tool support.*

We have been working on three strands: (1) the quantum process calculus CQP [17,19], partly in collaboration with Davidson [11]; (2) quantum model-checking based on temporal logic, in collaboration with Papanikolaou [20,21,28]; (3) quantum equivalence-checking, in collaboration with Ardeshir-Larijani [5]. Our work on process calculus has focussed on the development of basic theory, leading up to the definition of behavioural equivalence. This approach has also been studied by Ying *et al.*, who have developed qCCS [32]. Our work on model-checking uses a different style of specification language, more closely related to Promela. Some further work [10] makes connections between these two themes. Related work on model-checking include [7,14]. Our most recent work addresses the question of equivalence of sequential quantum programs, expressed in a language based on Selinger's QPL [29].

4 Quantum Teleportation in CQP

Teleportation [8] is a protocol for transferring an unknown qubit state from one participant, Alice, to another, Bob. The protocol uses classical communication — in fact, communication of just two classical bits — to achieve the transfer of a quantum state which is specified by two complex numbers. The trick is that there must be some pre-existing entanglement, shared by Alice and Bob.

Let x and y refer to two qubits that, together, are in the entangled state $\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$. Let u be a qubit in an unknown state, that is given to Alice. The protocol consists of the following steps.

1. Alice applies the *controlled not* (CNot) operator to u and x. This is a two-qubit operator whose effect on each basis state is to invert the second bit if and only if the first bit is 1.
2. Alice applies the *Hadamard* (H) operator to x. This operator is a change of basis from $\{|0\rangle, |1\rangle\}$ to $\{\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\}$.
3. Alice measures u and x, obtaining a two-bit classical result.
4. Alice sends this two-bit classical value to Bob.
5. Bob uses this classical value to determine which of four so-called *Pauli* operators I, X, Y or Z should be applied to y. In the definition below, we use the notation $\sigma_0 = I$, $\sigma_1 = \mathsf{X}$, $\sigma_2 = \mathsf{Z}$, $\sigma_3 = \mathsf{Y}$. (This is non-standard but convenient for this example; usually σ_2 and σ_3 are exchanged). The operators are defined as follows:

$$\begin{array}{lll} I: & \text{identity} & \\ \mathsf{X}: & |0\rangle \mapsto |1\rangle & |1\rangle \mapsto |0\rangle \\ \mathsf{Y}: & |0\rangle \mapsto i|1\rangle & |1\rangle \mapsto -i|0\rangle \\ \mathsf{Z}: & |0\rangle \mapsto |0\rangle & |1\rangle \mapsto -|1\rangle \end{array}$$

6. The state of y is now the original state of u (and u has lost its original state and is in a basis state).

Although the measurement in step 3 has a probabilistic result, the use of the classical value to determine a compensating operation in step 5 means that the complete protocol is deterministic in its effect on the state of Bob's qubit.

The teleportation protocol is often described by the circuit diagram in Figure 1.

The following definitions in the process calculus CQP (Communicating Quantum Processes) [17,19] model the teleportation protocol. *Alice*, *Bob* and *Teleport* are processes; q is a formal parameter representing a qubit; *in*, *out*, a and b are formal parameters representing channels; c is a private channel; x, y are local names for freshly allocated qubits, which will be instantiated with the names of actual qubits during execution. The language is based on pi-calculus and most of the syntax should be familiar.

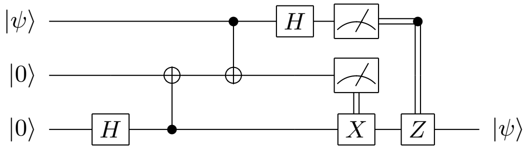

Fig. 1. Quantum teleportation as a circuit diagram

$$\begin{array}{ll}
\mathit{Alice}(q, in, out) & = in?[u] \,.\, \{u, q \mathrel{*}= \mathsf{CNot}\} \,.\, \{u \mathrel{*}= \mathsf{H}\} \,.\, out![\mathsf{measure}\ u, q] \,.\, \mathbf{0} \\
\mathit{Bob}(q, in, out) & = in?[r] \,.\, \{y \mathrel{*}= \sigma_r\} \,.\, out![y] \,.\, \mathbf{0} \\
\mathit{Teleport}(a, b) & = (\mathsf{qbit}\ x, y)(\{x \mathrel{*}= \mathsf{H}\} \,.\, \{x, y \mathrel{*}= \mathsf{CNot}\} \,. \\
& \qquad (\mathsf{new}\ c)(\mathit{Alice}(x, a, c) \mid \mathit{Bob}(y, c, b))
\end{array}$$

In *Teleport*, the actions before (new c) put the qubits x and y into the necessary entangled state. In order to help with writing a specification, *Alice* is given the qubit to be teleported as a message on channel *in*, and at the end of the protocol, *Bob* outputs the final qubit on *out*.

CQP has an operational semantics defined by labelled transition rules; it also has a type system in which the no cloning theorem is represented by linear typing. The example above, for simplicity, does not include type declarations.

The desired behaviour of teleportation is that a qubit (quantum state) is received on a and the same quantum state is sent on b; the protocol should behave like an identity operation:

$$\mathit{Identity}(a, b) = a?[x] \,.\, b![x] \,.\, \mathbf{0}$$

We can now write a specification of teleportation:

$$\mathit{Teleport}(a, b) \cong \mathit{Identity}(a, b)$$

where $\cong$ is a behavioural equivalence. Equivalent processes cannot be distinguished by any observer: they output the same values in the same circumstances, they produce the same probability distributions of measurement results, and in general interact in the same way with their environment.

As usual, we would like behavioural equivalence to be a congruence:

$$\forall P, Q, C. \quad P \cong Q \Rightarrow C[P] \cong C[Q]$$

where C is a process context. Congruence supports equational reasoning, and the universal composability properties defined by Canetti [9] in a different setting. Developing a congruence for a quantum process calculus was an open problem for several years [24], but very recently we have defined a congruence for CQP [11] and Feng *et al.* have independently defined one for qCCS [13]. Our equivalence is a form of probabilistic branching bisimulation [31], with appropriate extensions to deal with the quantum state. We have proved that the specification of teleportation is satisfied. The work on bisimulation and congruence for CQP is joint with Tim Davidson.

5 Model-Checking for Quantum Protocols

In this section we introduce the Quantum Model Checker (QMC) and its application to the verification of quantum protocols. QMC, which we have implemented together with Nick Papanikolaou, is a software tool. It automatically explores all possible behaviours arising from a protocol model, and enables logic properties expressed with Quantum Computation Tree Logic (QCTL) [6] to be checked over the resulting structure.

In QMC, the quantum state $|\psi\rangle$ is represented internally in an implicit way: rather than storing the so-called *state vector representation* of $|\psi\rangle$ (which grows exponentially in length as a function of the total number of qubits in $|\psi\rangle$), we use the *stabilizer array representation* [1], which is a binary representation of the set of Pauli operators that fix (or stabilize) $|\psi\rangle$. Using the stabilizer array representation, we gain significant computational benefits in terms of both space and time when simulating a given protocol, given that simulation of stabilizer circuits is performed using a polynomial time algorithm and the representation of the state grows polynomially with the total number of qubits.

5.1 Quantum Teleportation in QMC

We have designed an imperative-style concurrent specification language for the needs of the quantum model-checking tool QMC. For the purpose of this paper, we will demonstrate the syntax of this language by example. In this language the teleportation protocol (assuming we are trying to teleport the state $|\psi\rangle = |0\rangle$) may be expressed by the program in Figure 2. Working within the stabilizer formalism, we can teleport any of the one-qubit stabilizer states: $|0\rangle$, $|1\rangle$, $\frac{1}{\sqrt{2}}(|0\rangle \pm |1\rangle)$, $\frac{1}{\sqrt{2}}(|0\rangle \pm i|1\rangle)$.

In our setting, we allow for global variables (such as `e1`, `e2`), typed communication channels (such as `ch`) which are always global, and local (private) variables for each process (such as `a`,`b`,`c`,`d`,`q`). Communication is asynchronous, with executability rules restricting the way in which process interleaving is performed. For instance, the process `Bob` cannot start unless channel `ch` is filled with a value.

A protocol model will always consist of definitions of one or more processes; the commands performed by each of these processes must be interleaved (so as to emulate concurrent execution), and non–determinism (which occurs explicitly in selection structures (`if :: a -> ... :: b -> ...fi`) and implicitly when measurements are performed) must be resolved, producing an execution tree for the modelled system.

5.2 Specifying Properties

The properties of quantum protocols which we are interested in reasoning about are properties of the quantum state (e.g. which qubits are ‘active’ in a given state, which qubits are entangled with the rest of the system) over time. We are also interested in the outcomes of measurements, and the way in which the values

```
program Teleport;
var e1,e2:qubit; ch:channel of integer;
process Alice;
var q:qubit; a,b:integer;
begin
 q := newqubit;
 e1 := newqubit; e2 := newqubit;
 had e1; cnot e1 e2;
 cnot q e1; had q;
 a := meas q;
 b := meas e1;
 ch!a; ch!b;
end;
process Bob;
var c,d: integer;
begin
 ch?c; ch?d;
 if
 :: ((c=1) and (d=0)) -> X q; break;
 :: ((c=0) and (d=1)) -> Z q; break;
 :: ((c=1) and (d=1)) -> X q; Z q; break;
 :: ((c=0) and (d=0)) -> break;
 fi
end;
endprogram.
```

Fig. 2. QMC source program for quantum teleportation

of classical variables evolve. We use quantum computation tree logic (QCTL) [6] for this purpose.

QCTL adds the usual temporal connectives (AX, EF, EU) of computational tree logic [12] to the propositional logic EQPL [25]. The meaning of formulae in EQPL is expressed in terms of valuations, which are truth-value assignments for the symbols $\mathsf{qb}_0, \mathsf{qb}_1, \ldots, \mathsf{qb}_n$ 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 $v_1(\mathsf{qb}_0) = 0$, $v_1(\mathsf{qb}_1) = 0$, $v_2(\mathsf{qb}_0) = 1$, $v_2(\mathsf{qb}_1) = 1$.

The formulae accepted by the QMC tool 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 ϕ_1 and ϕ_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 variables $\mathsf{qb}_i, \mathsf{qb}_j, \ldots$; this type of formula is satisfied only if the qubits listed are disentangled from all other qubits in the system.

Example of Property for Verification. The requirement for the teleportation protocol is that, at the end of the protocol, no matter what 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 specification

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

which corresponds to the EQPL formula $[q_2]$ ⅄ $(\neg q_2)$. The first part of the formula asserts that the last qubit (q2) is disentangled from the rest of the quantum state, 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 (we represent ⅄ in ASCII form by `#/\`).

Alternatively, it is also possible in QMC to specify that the final state of a chosen qubit is the same as the initial state of a chosen qubit, again with the requirement (which is checked) that the chosen qubits are not entangled with the rest of the state. With this approach, we can define a model which non-deterministically chooses a state to teleport, and specify that the state is teleported, independently of its particular value; exhaustive model-checking then verifies that all stabilizer states are correctly teleported.

6 Beyond Stabilizer States: Checking Equivalence

Our work on quantum model-checking is based on the stabilizer formalism, because according to the Gottesman-Knill Theorem [22], that is what we can efficiently simulate with classical algorithms. This has two effects on the results we can obtain: (1) we can only analyse quantum systems whose operations are restricted to the Clifford group (which consists of all the operators we have seen so far along with a *phase* operator); (2) when exhaustively analysing the behaviour of a system on all possible quantum inputs, we can only consider inputs that are stabilizer states. Note, however, that the stabilizer formalism, although efficiently classically simulatable, contains many entangled states and supports a range of interesting quantum protocols.

We can avoid the second of the above limitations by taking advantage of linearity, if we focus our attention on systems that compute functions, mapping a quantum input to a quantum output. Protocols such as teleportation and error-correction can be formulated in this way. Furthermore, correctness of such a protocol can be expressed as equivalence with a particular specification protocol which is taken to be obviously correct. For example, the specification of teleportation is that a quantum state is transferred from input to output. If the teleportation protocol is formulated as a function, then its specification is that it should be equivalent to the identity function.

```
program Teleportation_Specification
input  q0:qbit
output q0:qbit
```

```
program Teleportation_Implementation
input q0:qbit
// Preparing an entangled pair.
newqbit q1;
newqbit q2;
q1 *= H;
q1,q2 *= CNot;
//Entangling the input qubit.
q0,q1 *= CNot;
q0 *= H;
// Measurement and corrections.
measure q0 then q2*=Z else q2*=I end;
measure q1 then q2*=X else q2*=I end
// The quantum state is now on q2
output q2:qbit
```

Fig. 3. Teleportation: Specification and Implementation

The appropriate way to view a quantum protocol as a function is to consider its action as a superoperator, i.e. a linear operator on the space of density matrices. In this way, both measurements and unitary operators are taken into account. By linearity, to check that superoperators f and g are equivalent, it is sufficient to choose a basis for the space of density matrices and check that for each basis element v, $f(v) = g(v)$. It turns out that it is possible to choose a basis consisting only of stabilizer states [18], and this brings equivalence-checking into the realm of automated analysis in the stabilizer formalism.

With Ebrahim Ardeshir-Larijani, we have implemented a tool [5] which takes as input two programs and checks whether or not they are equivalent, by evaluating them on all elements of a stabilizer basis. The language is based on Selinger's QPL [29]. For example, verification of a teleportation protocol consists of checking equivalence of the two programs in Figure 3. In comparison with the discussion of teleportation in Section 4, this model does not define Alice and Bob separately, and removes the communication; the protocol has been converted into a sequential program.

Compared with the verification of teleportation by the QMC system, we have a stronger conclusion: that *all* quantum states are successfully teleported, not just stablizer states. Retrospectively, we can now intepret the QMC verification as a guarantee that all states are correctly teleported, assuming that it is reasonable to view a QMC program as a superoperator, because QMC checked all stabilizer states and therefore included a basis. Equivalence checking requires less computation than QMC, because a stabilizer basis is smaller than the set of all stabilizer states (for n qubits there are approximately $2^{(n^2)/2}$ stabilizer states but a basis for the space of density matrices has only 2^{2n} elements).

7 Conclusion

We have outlined the principles of quantum information processing, and argued that formal methods will be necessary in order to guarantee the correctness of practical quantum systems. We have illustrated three particular approaches: behavioural equivalence in process calculus, model-checking and equivalence checking. We used quantum teleportation as a running example.

Future work will include the development of equational axiomatizations of behavioural equivalence in CQP, improving the efficiency of QMC and extending equivalence checking to include concurrent programs. On the more practical side, we intend to work on more substantial examples including cryptographic systems.

References

1. Aaronson, S., Gottesman, D.: Improved simulation of stabilizer circuits. Physical Review A 70, 052328 (2004)
2. Abramsky, S., Gay, S.J., Nagarajan, R.: Interaction categories and the foundations of typed concurrent programming. In: Broy, M. (ed.) Deductive Program Design: Proceedings of the 1994 Marktoberdorf International Summer School. NATO ASI Series F: Computer and Systems Sciences. Springer (1995)
3. Abramsky, S.: Interaction Categories (Extended Abstract). In: Burn, G.L., Gay, S.J., Ryan, M.D. (eds.) Theory and Formal Methods 1993: Proceedings of the First Imperial College Department of Computing Workshop on Theory and Formal Methods. Workshops in Computer Science, pp. 57–70. Springer (1993)
4. Abramsky, S., Coecke, B.: A categorical semantics of quantum protocols. In: Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science (LICS 2004), pp. 415–425. IEEE Computer Society (2004); Also arXiv:quant-ph/0402130
5. Ardeshir-Larijani, E., Gay, S.J., Nagarajan, R.: Equivalence checking of quantum protocols. In: Piterman, N., Smolka, S.A. (eds.) TACAS 2013 (ETAPS 2013). LNCS, vol. 7795, pp. 478–492. Springer, Heidelberg (2013)
6. Baltazar, P., Chadha, R., Mateus, P.: Quantum computation tree logic – model checking and complete calculus. International Journal of Quantum Information 6(2), 219–236 (2008)
7. Baltazar, P., Chadha, R., Mateus, P., Sernadas, A.: Towards model-checking quantum security protocols. In: First International Conference on Quantum, Nano, and Micro Technologies, ICQNM. IEEE Computer Society (2007)
8. Bennett, C.H., Brassard, G., Crépeau, C., Jozsa, R., Peres, A., Wootters, W.K.: Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels. Physical Review Letters 70, 1895–1899 (1993)
9. Canetti, R.: Universally composable security: A new paradigm for cryptographic protocols. In: 42nd IEEE Symposium on Foundations of Computer Science, FOCS, pp. 136–145. IEEE Computer Society (2001)

10. Davidson, T., Gay, S.J., Mlnařík, H., Nagarajan, R., Papanikolaou, N.: Model checking for Communicating Quantum Processes. International Journal of Unconventional Computing 8(1), 73–98 (2012)
11. Davidson, T.A.S.: Formal Verification Techniques using Quantum Process Calculus. PhD thesis, University of Warwick (2011)
12. Emerson, E.A.: Temporal and modal logic, vol. B: Formal Models and Semantics, pp. 995–1072. MIT Press (1990)
13. Feng, Y., Duan, R., Ying, M.: Bisimulation for quantum processes. In: 38th ACM Symposium on Principles of Programming Languages, POPL. ACM (2011)
14. Feng, Y., Yu, N., Ying, M.: Model checking quantum Markov chains. arXiv:1205.2187 [quant-ph] (2012)
15. Feynman, R.P.: Simulating physics with computers. International Journal of Theoretical Physics 21(6-7), 467–488 (1982)
16. Gay, S.J., Mackie, I.C. (eds.): Semantic Techniques in Quantum Computation. Cambridge University Press (2010)
17. Gay, S.J., Nagarajan, R.: Communicating quantum processes. In: 32nd ACM Symposium on Principles of Programming Languages, POPL, pp. 145–157 (2005); Also arXiv:quant-ph/0409052
18. Gay, S.J.: Stabilizer states as a basis for density matrices. arXiv:1112.2156 (2011)
19. Gay, S.J., Nagarajan, R.: Types and typechecking for Communicating Quantum Processes. Mathematical Structures in Computer Science 16(3), 375–406 (2006)
20. Gay, S.J., Nagarajan, R., Papanikolaou, N.: QMC: A model checker for quantum systems. In: Gupta, A., Malik, S. (eds.) CAV 2008. LNCS, vol. 5123, pp. 543–547. Springer, Heidelberg (2008)
21. Gay, S.J., Papanikolaou, N., Nagarajan, R.: Specification and verification of quantum protocols. In: Semantic Techniques in Quantum Computation. Cambridge University Press (2010)
22. Gottesman, D.: Class of quantum error-correcting codes saturating the quantum Hamming bound. Physical Review A 54, 1862 (1996)
23. Grover, L.: A fast quantum mechanical algorithm for database search. In: 28th ACM Symposium on the Theory of Computation, STOC, pp. 212–219. ACM Press (1996)
24. Lalire, M.: Relations among quantum processes: bisimilarity and congruence. Mathematical Structures in Computer Science 16(3), 407–428 (2006)
25. Mateus, P., Sernadas, A.: Weakly complete axiomatization of exogenous quantum propositional logic. Information and Computation 204(5), 771–794 (2006)
26. Mayers, D.: Unconditional Security in Quantum Cryptography. Journal of the ACM 48(3), 351–406 (2001)
27. Nagarajan, R., Gay, S.J.: Formal verification of quantum protocols. arXiv:quant-ph/0203086 (March 2002)
28. Papanikolaou, N.K.: Model Checking Quantum Protocols. PhD thesis, University of Warwick (2009)
29. Selinger, P.: Towards a quantum programming language. Mathematical Structures in Computer Science 14(4), 527–586 (2004)
30. Shor, P.W.: Algorithms for quantum computation: discrete logarithms and factoring. In: 35th IEEE Symposium on Foundations of Computer Science, FOCS (1994)
31. Trčka, N., Georgievska, S.: Branching bisimulation congruence for probabilistic systems. Electronic Notes in Theoretical Computer Science 220(3), 129–143 (2008)
32. Ying, M., Feng, Y., Duan, R., Ji, Z.: An algebra of quantum processes. ACM Transactions on Computational Logic 10(3), 19 (2009)