Equational reasoning about quantum protocols.

Simon J. Gay[1] and Ittoop V. Puthoor[1,2] *

[1]School of Computing Science and [2]School of Physics and Astronomy,
University of Glasgow, UK

Abstract. Communicating Quantum Processes (CQP) is a quantum process calculus that applies *formal* techniques from classical computer science to concurrent and communicating systems that combine quantum and classical computation. By employing the theory of *behavioural equivalence* between processes, it is possible to verify the correctness of a system in CQP. The *equational theory* of CQP helps us to analyse quantum systems by reducing the need to explicitly construct *bisimulation* relations. We add three new equational axioms to the existing equational theory of CQP, which helps us to analyse various quantum protocols by proving that the implementation and specification are equivalent. We summarise the necessary theory and demonstrate its application in the analysis of quantum secret sharing. Also, we illustrate the approach by verifying other interesting and important practical quantum protocols such as superdense coding, quantum error correction and remote CNOT.

Keywords: Quantum Computing, Formal methods, Quantum process calculus, Verification, Operational semantics, Equational reasoning

1 Introduction

Quantum computing is believed to be the next computing revolution as it promises to offer a very high degree of improvement over its classical counterpart by using the principles of quantum mechanics. On the other hand, quantum cryptography and quantum communication have made rapid progress already with the commercial deployment of the first secure cryptography systems [12, 13]. It has been mathematically proven that quantum cryptographic systems are unconditionally secure [14] but this doesn't provide a formal assurance to the security when these systems are implemented as a whole unit which may also include classical components. Therefore, there is still the need to develop techniques that verify the correctness of these systems. This was the prime motivation of using *process calculus* (a specialised area in *formal methods*) in modelling and analysing quantum information processing (QIP) systems that can be implemented.

Quantum process calcululi provide the techniques which help us to formally define the structure and behaviour of systems that are a combination of both

* Supported by a Lord Kelvin / Adam Smith Scholarship from the University of Glasgow.

quantum and classical subsystems. We use a particular quantum process calculus called Communicating Quantum Processes (CQP) [8], developed by Gay and Nagarajan. The other quantum process calculus which has been established is qCCS by Feng et al.[4]. The property of *behavioural equivalence* of processes in quantum process calculus helps to verify the correctness of a system. The *congruence* property of equivalence makes it more powerful by preserving the equivalence in any environment. This has been developed for CQP [2] and qCCS [5].

Equational reasoning is essential in mathematics and logics, and plays an important role in many applications of formal methods in theoretical computer science. With the use of theorem provers for equational logic, it is possible to perform automated analysis. The equational axioms reduce the need to explicitly construct bisimulation relations, which is reported in [2] for CQP with an analysis of the quantum teleportation protocol (*Teleport*).

Our Contributions. In this paper, we demonstrate the use of the equational theory of CQP [2] and introduce three new axioms that helps us to take a step further to analyse various quantum protocols that include: *quantum secret sharing* (*QSS*), *superdense coding* (*SDC*), *quantum error correction code* (*QECC*) and *Remote CNOT* (*RCNOT*). Our results show that the protocols, *QSS* and *QECC* are equivalent to the specification process *Identity*. We provide a similar reasoning for other protocols. Using the *transitivity* property of equivalence, we also prove that $\mathit{QSS} \cong^c \mathit{QECC} \cong^c \mathit{Teleport}$.

The structure of the paper is as follows. First, in Section 2 we provide in brief the fundamentals of quantum computing. We review the language of CQP in Section 3 and illustrate it with a model of quantum secret sharing. Section 4 provides a brief summary on the theory of behavioural equivalence of CQP. Section 5 summarises the equational theory of CQP, which has not previously been published other than in Davidson's thesis, and applies it to quantum secret sharing and other protocols. Section 6 concludes with an indication of directions of future work.

Related Work. Previous work on automated analysis is based on exhaustive simulation based on stabiliser formalism. Model checking tools like the QMC [10] and the equivalence checker [1] were developed for the verification of quantum protocols. Since the tool uses stabilizer formalism, it is restricted to use only the operators in the Clifford group. The equational theory of CQP is not based on the stabilizer formalism and hence is not restricted to Clifford group operations.

2 Preliminaries

We recall briefly the aspects of quantum computing that are relevant for this paper. For more detailed information we refer to [16]. A *qubit* is an information unit comprising two states ($|0\rangle$ and $|1\rangle$) which are called the *standard* basis. The *state space* $\mathbb{H}$ (or Hilbert space) of a qubit is a vector space that consists of all *superpositions* of the basis states: $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ where α and β are complex amplitudes such that $|\alpha|^2 + |\beta|^2 = 1$. The states can be represented by column vectors:

$$\begin{pmatrix}\alpha\\ \beta\end{pmatrix} = \alpha\begin{pmatrix}1\\0\end{pmatrix} + \beta\begin{pmatrix}0\\1\end{pmatrix} = \alpha|0\rangle + \beta|1\rangle$$

A system can consist of many qubits (say n qubits) and the Hilbert space is a 2^n dimensional space whose standard basis is $|00\ldots0\rangle \ldots |11\ldots1\rangle$. This is represented by *tensor product* of unit vectors which is denoted as $|0\rangle\otimes|0\rangle\cdots\otimes|0\rangle$. The evolution of the quantum state of a system can be described by quantum operations called *unitary transformations.* If the state of a qubit is represented by a column vector, then a unitary transformation is represented by a matrix. Some unitary transformations which are commonly used are the *Hadamard* (H) and the *Pauli* transformations, denoted by either $\mathsf{I}, \mathsf{X}, \mathsf{Z}, \mathsf{Y}$:

$$\mathsf{H} = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}, \mathsf{I} = \begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}, \mathsf{X} = \begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}, \mathsf{Y} = \begin{pmatrix}0 & -i\\ i & 0\end{pmatrix}, \mathsf{Z} = \begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}$$

The *measurement* operation changes the quantum state permanently and measuring the above quantum state $|\psi\rangle$ gives a result 0 with probability $|\alpha|^2$ and result 1 with probability $|\beta|^2$. We will be using the *controlled Not* transformation (or CNOT) on a pair of qubits. The action of this operation is that it flips the second qubit (target qubit) if and only if the first qubit (control qubit) is 1. We have $\mathsf{CNOT}|0x\rangle = |0x\rangle$ and $\mathsf{CNOT}|1x\rangle = |1y\rangle$ where $x, y \in \{0, 1\}$ and $y = x \oplus 1$ with $\oplus$ denoting addition modulo 2. *Entanglement* is an important phenomenon in quantum computing which is observed in a system that comprises of two or more qubits. This means that the states of the qubits are not *separable.* For example, a three qubit state $\frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$ (also called *GHZ* state) is said to be *entangled* and cannot be decomposed into single qubit states. Measurement of one of the qubits will fix the state of the others even if the entangled qubits are physically separated.

3 Communicating Quantum Processes (CQP)

CQP is based on the π-calculus [15] with primitives for quantum information. The language uses the concept that a system can be considered to be made up of independent components or *processes.* The *processes* can communicate by sending and receiving data along *channels* and these data are qubits, or classical bits. A distinctive feature of CQP is its static type system [9], the purpose of which is to classify classical and quantum data and also to enforce the no-cloning property of quantum information. In our recent work, we have extended CQP to describe and verify linear optical quantum computing (LOQC) [6, 7].

3.1 Syntax and semantics of CQP

The syntax of CQP is defined by the grammar as shown in Figure 1. We use the notation $\tilde{e} = e_1, \ldots, e_n$, and write $|\tilde{e}|$ for the length of a tuple. The syntax

$$
\begin{aligned}
T &::= \mathsf{Int} \mid \mathsf{Qbit} \mid \mathsf{Bit} \mid \widehat{\ }[\tilde{T}] \mid \mathsf{Op}(1) \mid \mathsf{Op}(2) \mid \cdots \\
v &::= \mathsf{0} \mid \mathsf{1} \mid \cdots \mid \mathsf{H} \mid \cdots \\
e &::= v \mid \mathsf{measure}\ \tilde{e} \mid \tilde{e} \mathrel{*=} e \mid e + e' \mid (e, e) \\
P &::= \mathbf{0} \mid (P|P) \mid P + P \mid e?[\tilde{x} : \tilde{T}].P \mid e![\tilde{e}].P \mid \{e\}.P \mid (\mathsf{qbit}\ x)P \mid (\mathsf{new}\ x : \widehat{\ }[T])P
\end{aligned}
$$

$$(i)$$

$$
\begin{aligned}
v &::= \ldots \mid q \mid c \\
E &::= [] \mid \mathsf{measure}\ E, \tilde{e} \mid \mathsf{measure}\ v, E, \tilde{e} \mid \ \ldots \ \mid \mathsf{measure}\ \tilde{v}, E \mid E + e \mid v + E \\
F &::= []?[\tilde{x}].P \mid []![\tilde{e}].P \mid v![[].\tilde{e}].P \mid v![v, [], \tilde{e}].P \mid \ \cdots \ \mid v![\tilde{v}, []].P \mid \{[]\}.P
\end{aligned}
$$

$$(ii)$$

Fig. 1. (i) Syntax of CQP and (ii) Internal syntax of CQP.

consists of types T, values v, expressions e (including quantum measurements and the conditional application of unitary operators $\tilde{e} \mathrel{*=} e$), and processes P. Values v consist of variables (x,y,z etc), channel names c, literal values of data types (0,1,..), unitary operators such as the Hadamard operator H. Expressions e consist of values, measurements $\mathsf{measure}\ e_1, \ldots, e_n$, applications $e_1, \ldots, e_n \mathrel{*=} e$ of unitary operators and expressions involving data operators such as $e+e'$ and a pair of values (e, e). Processes include the nil process $\mathbf{0}$, parallel composition $P|P$, inputs $e?[\widetilde{x}:\widetilde{T}].P$, outputs $e![\widetilde{e}].P$, actions $\{e\}.P$ (typically a unitary operation or measurement), typed channel restriction $(\mathsf{new}\ x : \widehat{\ }[\widetilde{T}])P$ and qubit declaration $(\mathsf{qbit}\ x)P$. In order to define the operational semantics we provide the *internal syntax* in Figure 1(ii). We assume a countably infinite set of qubit names, ranging over $q, r, \ldots$ and similarly channel names c. Values are supplemented with qubit names q which are generated at run-time and substituted for the variables used in qbit declaration. Evaluation contexts for expressions ($E[]$) and processes ($F[]$) are used to define the operational semantics [19].

The complete formal semantics are provided in [2] and we explain briefly in this paper. In CQP, the execution of a system is described by the process term (which is the case for classical process calculus) and the quantum state. Hence, the operational semantics are defined using *configurations.*

Definition 1. *A* configuration *is a tuple (σ; ω; P) where σ is a mapping from qubit names to the quantum state and ω is a list of qubit names associated with the process P*

The semantics of CQP consists of labelled transitions between configurations. For example, the configuration $([q, r \mapsto |\psi\rangle]; q; c![q] \,.\, P)$, means that the global quantum state consists of two qubits, q and r, in the specified state ($|\psi\rangle$); that the process term under consideration has access to qubit q but not to qubit r ; and that the process itself is $c![q] \,.\, P$.

Example 1. $([q, r \mapsto |\psi\rangle]; q; c![q] \,.\, P) \xrightarrow{c![q]} ([q, r \mapsto |\psi\rangle]; \emptyset; P)$.

The example illustrates an output transition where the quantum state ($|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$) is not changed by this output transition. Since qubit q is given as output, the continuation process P no longer has access to it; the final configuration has an empty list of owned qubits.

3.2 Quantum Secret Sharing

In this paper, we describe a quantum secret sharing [11] protocol that consists of three users represented by the processes: *Alice*, *Bob* and *Charlie*. *Alice* would like to send a message to *Bob* and *Charlie*. We analyse a scenario in which *Charlie* ends up with the original qubit. *Alice* encodes her message in a way such that *Bob* and *Charlie* must cooperate with each other to retrieve it. The protocol begins by applying a Hadamard and CNOT operations to qubits x, y and z in order to generate the GHZ state as described in previous section. The qubits are shared between the three users. *Alice* also possesses the qubit labelled q which is in some unknown state $|\psi\rangle$; this is the qubit she wishes to send. The CQP definitions of *Alice*, *Bob* and *Charlie* are as follows

$Alice(c, e, x) = c?[q : \mathsf{Qbit}] \,.\, \{q, x \mathrel{*}= \mathsf{CNOT}\} \,.\, \{q \mathrel{*}= \mathsf{H}\} \,.\, e![\mathsf{measure}\ q, \mathsf{measure}\ x] \,.\, \mathbf{0}$
$Bob(f, y) = \{y \mathrel{*}= \mathsf{H}\} \,.\, f![\mathsf{measure}\ y] \,.\, \mathbf{0}$
$Charlie(e, f, d, z) = e?[i : \mathsf{Bit}, j : \mathsf{Bit}] \,.\, f?[k : \mathsf{Bit}] \,.\, \{z \mathrel{*}= \mathsf{Z}^k\} \,.\, \{z \mathrel{*}= \mathsf{X}^j\} \,.\, \{z \mathrel{*}= \mathsf{Z}^i\} \,.$
$d![z] \,.\, \mathbf{0}$

Alice receives the qubit q from the environment through her channel c and performs unitary operations (CNOT and H) before measuring her qubits. She sends the outcomes which are classical bits i and j through channel e to *Charlie*. *Charlie* cannot retrieve the information without the help of *Bob*. *Bob* performs an Hadamard operation on his qubit y before measuring it. Then, he sends the outcome to *Charlie*. Using the classical bits from *Alice* and *Bob*, *Charlie* performs the necessary unitary operations on his qubit z in order to recover the original state $|\psi\rangle$. The complete system is defined as:

$$QSS(c, d) = (\mathsf{qbit}\ x, y, z)(\{x \mathrel{*}= \mathsf{H}\} \,.\, \{x, y \mathrel{*}= \mathsf{CNOT}\} \,.\, \{y, z \mathrel{*}= \mathsf{CNOT}\} \,.$$
$$(\mathsf{new}\ e, f)(Alice(c, e, x) \mid Bob(f, y) \mid Charlie(e, f, d, z)))$$

QSS process consists of *Alice*, *Bob* and *Charlie* in parallel. That is the outputs on e and f in *Alice* and *Bob* respectively synchronise with the inputs on e and f in *Charlie*. Channel e and f are designated as a private local channels. This is specified by $(\mathsf{new}\ e, f)$, which is a construct from pi-calculus to dynamically create fresh channels. The first term, $(\mathsf{qbit}\ x, y, z)$ in QSS, allocates three fresh qubits, each in state $|0\rangle$, and gives them the local names x, y and z. The next three terms create the GHZ state with qubits x, y and z. The aim is to prove that QSS is equivalent to its specification process given by the following definition:

$$Identity(c : \widehat{\ }[\mathsf{Qbit}], d : \widehat{\ }[\mathsf{Qbit}]) = c?[x : \mathsf{Qbit}] \,.\, d![x] \,.\, \mathbf{0}.$$

4 Probabilistic Branching Bisimulation of CQP

The equivalence for CQP is a form of *probabilistic branching bisimilarity* [18], adapted to the situation in which probabilistic behaviour comes from quantum measurement. A key point is that when considering matching of input or output transitions involving qubits, it is the reduced density matrices of the transmitted qubits that are required to be equal. Here, we summarise the essential definitions in [2]. Let $\xrightarrow{\tau}^{+}$ denote zero or one τ transitions; let $\Longrightarrow$ denote zero or more τ transitions; and let $\overset{\alpha}{\Longrightarrow}$ be equivalent to $\Longrightarrow\xrightarrow{\alpha}\Longrightarrow$.

Definition 2 (Probabilistic Branching Bisimulation). *An equivalence relation $\mathcal{R}$ on configurations is a* probabilistic branching bisimulation *on configurations if whenever $(s,t) \in \mathcal{R}$ the following conditions are satisfied.*

I. If $s \in \mathcal{S}_n$ and $s \xrightarrow{\tau} s'$ then $\exists t', t''$ such that $t \Longrightarrow t' \xrightarrow{\tau}^{+} t''$ with $(s,t') \in \mathcal{R}$ and $(s',t'') \in \mathcal{R}$.

II. If $s \xrightarrow{c![V,\widetilde{q}_1]} s'$ where $s' = \boxplus_{j\in\{1\ldots m\}} p_j s'_j$ and $V = \{\widetilde{v}_1, \ldots, \widetilde{v}_m\}$ then $\exists t', t''$ such that $t \Longrightarrow t' \xrightarrow{c![V,\widetilde{q}_2]} t''$ with
a) $(s,t') \in \mathcal{R}$,
b) $t'' = \boxplus_{j\in\{1\ldots m\}} p_j t''_j$,
c) for each $j \in \{1,\ldots,m\}$, $\rho_E(s'_j) = \rho_E(t''_j)$.
d) for each $j \in \{1,\ldots,m\}$, $(s'_j, t''_j) \in \mathcal{R}$.

III. If $s \xrightarrow{c?[\widetilde{v}]} s'$ then $\exists t', t''$ such that $t \Longrightarrow t' \xrightarrow{c?[\widetilde{v}]} t''$ with $(s,t') \in \mathcal{R}$ and $(s',t'') \in \mathcal{R}$.

IV. If $s \in \mathcal{S}_p$ then $\mu(s,D) = \mu(t,D)$ for all classes $D \in \mathcal{S}/\mathcal{R}$.

Here, μ is the probabilistic function that is defined in the style of [18], which is necessary when calculating the total probability of reaching a terminal state. This is needed to ensure the matching of probabilistic configurations.

Definition 3 (Probabilistic Branching Bisimilarity). *Configurations s and t are* probabilistic branching bisimilar*, denoted $s \leftrightarroweq t$, if there exists a probabilistic branching bisimulation $\mathcal{R}$ such that $(s,t) \in \mathcal{R}$.*

Definition 4 (Full probabilistic branching bisimilarity). *Processes P and Q are* full probabilistic branching bisimilar*, denoted $P \leftrightarroweq^c Q$, if for all substitutions κ and all quantum states σ, $(\sigma; \widetilde{q}; P\kappa) \leftrightarroweq (\sigma; \widetilde{q}; Q\kappa)$.*

In order to state the *congruence* theorem, we need an assumption that processes are typable. Due to space constraints, we have not presented the type system in this paper but the idea is to associate each qubit with a unique owning component of the process.

Theorem 1 (Full probabilistic branching bisimilarity is a congruence [2]). *If $P \leftrightarroweq^c Q$ then for any context $C[]$, if $C[P]$ and $C[Q]$ are typable then $C[P] \leftrightarroweq^c C[Q]$.*

$$M \mid N = \Sigma_{i=1}^{m}\alpha_i.(P_i \mid N) + \Sigma_{j=1}^{n}\beta_j.(M \mid Q_j) + \Sigma_{\alpha_i C \beta_j}\tau.(P_i \mid Q_j) \tag{E1}$$

where $M = \Sigma_{i=1}^{m}\alpha_i.P_i, N = \Sigma_{j=1}^{n}\alpha_j.Q_j$ and $\alpha_i C \beta_j$ if α_i is $c![\widetilde{e}]$ and β_j is $c?[\widetilde{x}]$

$$\{\widetilde{x} \mathrel{*}= V\}.\{\widetilde{x} \mathrel{*}= W\}.P = \{\widetilde{x} \mathrel{*}= U\}.P \qquad \text{if } U = WV \tag{QI1}$$

$$\{\widetilde{y} \mathrel{*}= \mathsf{U}^{\mathsf{measure}\ x}\}.P = \{x, \widetilde{y} \mathrel{*}= \mathsf{CU}\}.\{\mathsf{measure}\ x\}.P \tag{QI2}$$

$$\{\widetilde{y} \mathrel{*}= \mathsf{U}^{\mathsf{measure}\ x.\mathsf{measure}\ z}\}.P = \{(x,z), \widetilde{y} \mathrel{*}= \mathsf{CU}\}.\{\mathsf{measure}\ x\}\,.\,\{\mathsf{measure}\ z\}.P \tag{QI3}$$

$$\{\widetilde{x} \mathrel{*}= U\}.\{\widetilde{y} \mathrel{*}= V\}.P = \{\widetilde{y} \mathrel{*}= V\}.\{\widetilde{x} \mathrel{*}= U\}.P \quad \text{if } \widetilde{x} \cap \widetilde{y} = \emptyset \tag{QC1}$$

$$\{\widetilde{x} \mathrel{*}= U\}.\{\mathsf{measure}\ \widetilde{y}\}.P = \{\mathsf{measure}\ \widetilde{y}\}.\{\widetilde{x} \mathrel{*}= U\}.P \quad \text{if } \widetilde{x} \cap \widetilde{y} = \emptyset \tag{QC2}$$

$$\{\widetilde{x} \mathrel{*}= U\}.(\mathsf{qbit}\ \widetilde{y}).P = (\mathsf{qbit}\ \widetilde{y}).\{\widetilde{x} \mathrel{*}= U\}.P \quad \text{if } \widetilde{x} \cap \widetilde{y} = \emptyset \tag{QC3}$$

$$\{\mathsf{measure}\ \widetilde{x}\}.\{\mathsf{measure}\ \widetilde{y}\}.P = \{\mathsf{measure}\ \widetilde{y}\}.\{\mathsf{measure}\ \widetilde{x}\}.P \quad \text{if } \widetilde{x} \cap \widetilde{y} = \emptyset \tag{QC4}$$

$$\{\mathsf{measure}\ \widetilde{x}\}.(\mathsf{qbit}\ \widetilde{y}).P = (\mathsf{qbit}\ \widetilde{y}).\{\mathsf{measure}\ \widetilde{x}\}.P \quad \text{if } \widetilde{x} \cap \widetilde{y} = \emptyset \tag{QC5}$$

$$(\mathsf{qbit}\ \widetilde{x}).(\mathsf{qbit}\ \widetilde{y}).P = (\mathsf{qbit}\ \widetilde{y}).(\mathsf{qbit}\ \widetilde{x}).P \quad \text{if } \widetilde{x} \cap \widetilde{y} = \emptyset \tag{QC6}$$

$$\alpha.\{\widetilde{y} \mathrel{*}= U\}.c?[\widetilde{x}].P = \alpha.c?[\widetilde{x}].\{\widetilde{y} \mathrel{*}= U\}.P \quad \text{if } \widetilde{y} \subseteq \mathbf{n}(\alpha), \widetilde{x} \cap \widetilde{y} = \emptyset \tag{QC7}$$

$$\alpha.\{\widetilde{y} \mathrel{*}= U\}.c![\widetilde{x}].P = \alpha.c![\widetilde{x}].\{\widetilde{y} \mathrel{*}= U\}.P \quad \text{if } \widetilde{y} \subseteq \mathbf{n}(\alpha), \widetilde{x} \cap \widetilde{y} = \emptyset \tag{QC8}$$

$$\alpha.\{\mathsf{measure}\ \widetilde{y}\}.c?[\widetilde{x}].P = \alpha.c?[\widetilde{x}].\{\mathsf{measure}\ \widetilde{y}\}.P \quad \text{if } \widetilde{y} \subseteq \mathbf{n}(\alpha), \widetilde{x} \cap \widetilde{y} = \emptyset \tag{QC9}$$

$$\alpha.\{\mathsf{measure}\ \widetilde{y}\}.c![\widetilde{x}].P = \alpha.c![\widetilde{x}].\{\mathsf{measure}\ \widetilde{y}\}.P \quad \text{if } \widetilde{y} \subseteq \mathbf{n}(\alpha), \widetilde{x} \cap \widetilde{y} = \emptyset \tag{QC10}$$

$$(\mathsf{qbit}\ \widetilde{x}).c?[\widetilde{y}].P = c?[\widetilde{y}].(\mathsf{qbit}\ \widetilde{x}).P \quad \text{if } \widetilde{x} \cap \widetilde{y} = \emptyset \tag{QC11}$$

$$(\mathsf{qbit}\ \widetilde{x}).c![\widetilde{y}].P = c![\widetilde{y}].(\mathsf{qbit}\ \widetilde{x}).P \quad \text{if } \widetilde{x} \cap \widetilde{y} = \emptyset \tag{QC12}$$

$$\{\mathsf{measure}\ x\}.\mathbf{0} = \mathbf{0} \tag{QS1}$$

$$\{\widetilde{x} \mathrel{*}= U\}.\mathbf{0} = \mathbf{0} \tag{QS2}$$

$$(\mathsf{qbit}\ x).\mathbf{0} = \mathbf{0} \tag{QS3}$$

$$\alpha\,.\,\tau\,.\,P = \alpha\,.\,P \tag{TAU1}$$

$$\alpha.\{\widetilde{x} \mathrel{*}= \mathit{\Pi}\}.P\{\pi(\widetilde{q})/\widetilde{x}\} = \alpha.P \quad \text{if } \widetilde{x} \subseteq \mathbf{n}(\alpha) \tag{QP1}$$

$$(\mathsf{qbit}\ x).\{\widetilde{y}, x \mathrel{*}= U\}.P = (\mathsf{qbit}\ x).\{\widetilde{y}, x \mathrel{*}= V\}.P \qquad \text{if } U(I_{\widetilde{y}} \otimes |0\rangle) = V(I_{\widetilde{y}} \otimes |0\rangle) \tag{QD1}$$

$$c?[x : \mathsf{Bit}]\,.\,P(x) = c?[x : \mathsf{Bit}]\,.\,Q(x) \text{ if } P(x) = Q(x) \text{ for all } x \in \{0,1\} \tag{CV1}$$

$$(\mathsf{new}\ c)(P + Q) = (\mathsf{new}\ c)P + (\mathsf{new}\ c)Q \tag{R1}$$

$$(\mathsf{new}\ c)\alpha.P = \mathbf{0} \qquad \text{if } \alpha \in \{c?[\cdot], c![\cdot]\} \tag{R2}$$

$$(\mathsf{new}\ c)\alpha.P = \alpha.(\mathsf{new}\ c)P \qquad \text{if } \alpha \notin \{c?[\cdot], c![\cdot]\} \tag{R3}$$

Fig. 2. Axioms for full probabilistic branching bisimilarity.

5 Equational theory of CQP

The congruence property of behavioural equivalence guarantees that equivalent processes remain equivalent in any context, which is the foundation for equational reasoning. The axioms for full probabilistic branching bismilarity are shown in Figure 2 and have been proved sound in [2]. The axioms were used in the analysis of the quantum teleportation protocol which is reported in [2] but does not help us to verify other quantum protocols like *SDC*, *QECC* etc. In this paper, we demonstrate the usefulness of the equational theory of CQP by introducing additional three new axioms Cv1, Qi3 and Tau1, that helps us to take a step further to analyse various other important quantum protocols.

$$c?[x : \mathsf{Bit}]\,.\,P(x) = c?[x : \mathsf{Bit}]\,.\,Q(x) \text{ if } P(x) = Q(x) \text{ for all } x \in \{0,1\} \qquad (\textsc{Cv1})$$

The *classical value* rule Cv1 enables us to compare processes that are controlled by the classical bit, say x. This rule will be used when we analyse the *SDC* protocol. Rule Qi3 introduced in this paper is an extension of the identity rule Qi2. This rule, expresses the *principle of deferred measurement* [16] and helps us to analyse *QECC* protocol, where the unitary operator U is controlled by the measurement of more than one qubit

$$\{\widetilde{y} \mathrel{*}= \mathsf{U}^{\mathsf{measure}\ x.\mathsf{measure}\ z}\}.P = \{(x,z), \widetilde{y} \mathrel{*}= \mathsf{CU}\}.\{\mathsf{measure}\ x\}\,.\,\{\mathsf{measure}\ z\}.P \qquad (\textsc{Qi3})$$

The correctness of *QECC* has been proved by creating bisimulation relations [3] and in this paper, we show that we can analyse *QECC* by not creating bisimulation relations explicitly. Finally, we define the Tau1 rule that helps to remove the unnecessary τ which arise during the elimination of parallel composition.

$$\alpha\,.\,\tau\,.\,P = \alpha\,.\,P \qquad (\textsc{Tau1})$$

The new axioms introduced in the paper are proved to be sound [17].

5.1 Analysing Quantum Secret Sharing

Now, we present the use of an axiomatic approach for proving that the processes are equivalent with respect to full probabilistic branching bisimilarity that is defined earlier. We begin by applying the expansion law E1 to the definition of *QSS*, to get:

$$\begin{aligned}&(\mathsf{qbit}\ x,y,z)\,.\,\{x \mathrel{*}= \mathsf{H}\}\,.\,\{x,y \mathrel{*}= \mathsf{CNOT}\}\,.\,\{y,z \mathrel{*}= \mathsf{CNOT}\}\,.\,(\mathsf{new}\ e,f)\\&(c?[q]\,.\,(\mathit{Alice}' \mid \mathit{Bob} \mid \mathit{Charlie}) + \{y \mathrel{*}= \mathsf{H}\}\,.\,(\mathit{Alice} \mid \mathit{Bob}' \mid \mathit{Charlie}) +\\&e?[i,j]\,.\,(\mathit{Alice} \mid \mathit{Bob} \mid \mathit{Charlie}'))\end{aligned} \qquad (1)$$

where $\mathit{Alice} = c?[q]\,.\,\mathit{Alice}'$, $\mathit{Bob} = \{y{*}{=}\mathsf{H}\}\,.\,\mathit{Bob}'$ and $\mathit{Charlie} = e?[i,j]\,.\,\mathit{Charlie}'$. Using the rules R1 and R2 on Eq. 1, the third term of the sum vanishes to give:

$$\begin{aligned}&(\mathsf{qbit}\ x,y,z)\,.\,\{x \mathrel{*}= \mathsf{H}\}\,.\,\{x,y \mathrel{*}= \mathsf{CNOT}\}\,.\,\{y,z \mathrel{*}= \mathsf{CNOT}\}\,.\,(\mathsf{new}\ e,f)\\&(c?[q]\,.\,(\mathit{Alice}' \mid \mathit{Bob} \mid \mathit{Charlie}) + \{y \mathrel{*}= \mathsf{H}\}\,.\,(\mathit{Alice} \mid \mathit{Bob}' \mid \mathit{Charlie}))\end{aligned} \qquad (2)$$

Expanding Eq. 2 as before, we get:

$$\begin{aligned}&(\mathsf{qbit}\ x,y,z)\,.\,\{x \mathrel{*}= \mathsf{H}\}\,.\,\{x,y \mathrel{*}= \mathsf{CNOT}\}\,.\,\{y,z \mathrel{*}= \mathsf{CNOT}\}\,.\,(\mathsf{new}\ e,f)(c?[q]\,.\,\{y \mathrel{*}= \mathsf{H}\}\,.\\&(\mathit{Alice}' \mid \mathit{Bob}' \mid \mathit{Charlie}) + \{y \mathrel{*}= \mathsf{H}\}\,.\,c?[q]\,.\,(\mathit{Alice}' \mid \mathit{Bob}' \mid \mathit{Charlie}) + \{y \mathrel{*}= \mathsf{H}\}\,.\\&f![\mathsf{measure}\ y]\,.\,(\mathit{Alice} \mid \mathbf{0} \mid \mathit{Charlie}) + c?[q]\,.\,\{q,x \mathrel{*}= \mathsf{CNOT}\}\,.\,(\mathit{Alice}' \mid \mathit{Bob} \mid \mathit{Charlie}))\end{aligned}$$

(3)

Using restriction rules R1 − R3 and commutative identities, QC7 and QC8, we can commute between the process terms which leads to the first two terms in Eq. 3 essentially the same and the third term is eliminated to give:

$$\begin{aligned}&(\mathsf{qbit}\ x,y,z)\,.\,\{x \mathrel{*}= \mathsf{H}\}\,.\,\{x,y \mathrel{*}= \mathsf{CNOT}\}\,.\,\{y,z \mathrel{*}= \mathsf{CNOT}\}\,.\,c?[q]\,.\\ &(\{y \mathrel{*}= \mathsf{H}\}\,.\,(\mathsf{new}\ e,f)(\mathit{Alice}' \mid \mathit{Bob}' \mid \mathit{Charlie}) + \{q,x \mathrel{*}= \mathsf{CNOT}\}\,.\\ &(\mathsf{new}\ e,f)(\mathit{Alice}' \mid \mathit{Bob} \mid \mathit{Charlie}))\end{aligned}\tag{4}$$

Repeating the procedure of expansion and using the reduction rules, we get:

$$\begin{aligned}&(\mathsf{qbit}\ x,y,z)\,.\,\{x \mathrel{*}= \mathsf{H}\}\,.\,\{x,y \mathrel{*}= \mathsf{CNOT}\}\,.\,\{y,z \mathrel{*}= \mathsf{CNOT}\}\,.\,c?[q]\,.\,\{q,x \mathrel{*}= \mathsf{CNOT}\}\,.\\ &\{q \mathrel{*}= \mathsf{H}\}\,.\,\{y \mathrel{*}= \mathsf{H}\}\,.\,(\mathbf{0}+\mathbf{0}+(\mathsf{new}\ e,f)\,.\,\tau\,.\,f![\mathsf{measure}\ y]\,.\,\mathbf{0} \mid f?[k\!:\!\mathsf{Bit}]\,.\\ &\{z \mathrel{*}= \mathsf{Z}^k\}\,.\,\{z \mathrel{*}= \mathsf{X}^{\mathsf{measure}\ r}\}\,.\,\{z \mathrel{*}= \mathsf{Z}^{\mathsf{measure}\ q}\}\,.\,d![z]\,.\,\mathbf{0}\end{aligned}\tag{5}$$

where τ represents the communication between *Alice* and *Charlie*, which happens internally. Similarly, the communication between *Bob* and *Charlie* gives:

$$\begin{aligned}&(\mathsf{qbit}\ x,y,z)\,.\,\{x \mathrel{*}= \mathsf{H}\}\,.\,\{x,y \mathrel{*}= \mathsf{CNOT}\}\,.\,\{y,z \mathrel{*}= \mathsf{CNOT}\}\,.\,c?[q]\,.\\ &\{q,x \mathrel{*}= \mathsf{CNOT}\}\,.\,\{q \mathrel{*}= \mathsf{H}\}\,.\,\{y \mathrel{*}= \mathsf{H}\}\,.\,(\mathsf{new}\ e,f)\,.\,\tau\,.\,\tau\,.\\ &\{z \mathrel{*}= \mathsf{Z}^{\mathsf{measure}\ y}\}\,.\,\{z \mathrel{*}= \mathsf{X}^{\mathsf{measure}\ r}\}\,.\,\{z \mathrel{*}= \mathsf{Z}^{\mathsf{measure}\ q}\}\,.\,d![z]\,.\,\mathbf{0}\end{aligned}\tag{6}$$

After several iterations using R3 and followed by $(\mathsf{new}\ e,f)\,.\,\mathbf{0} = \mathbf{0}$, we get:

$$\begin{aligned}&(\mathsf{qbit}\ x,y,z)\,.\,\{x \mathrel{*}= \mathsf{H}\}\,.\,\{x,y \mathrel{*}= \mathsf{CNOT}\}\,.\,\{y,z \mathrel{*}= \mathsf{CNOT}\}\,.\,c?[q]\,.\\ &\{q,x \mathrel{*}= \mathsf{CNOT}\}\,.\,\{q \mathrel{*}= \mathsf{H}\}\,.\,\{y \mathrel{*}= \mathsf{H}\}\,.\,\tau\,.\,\tau\,.\,\{z \mathrel{*}= \mathsf{Z}^{\mathsf{measure}\ y}\}\,.\\ &\{z \mathrel{*}= \mathsf{X}^{\mathsf{measure}\ r}\}\,.\,\{z \mathrel{*}= \mathsf{Z}^{\mathsf{measure}\ q}\}\,.\,d![z]\,.\,\mathbf{0}\end{aligned}\tag{7}$$

Finally, we remove the two τ transitions by using the TAU1 rule and thereby arrive at the sequentialised definition of *QSS*.

Proposition 1. $\mathit{QSS}(c,d) \mathrel{\cong^c} \mathit{Identity}(c,d)$

Proof. We will now simplify Eq. 7 and transform it into the *Identity* process by using the axioms in Figure 2. Rule QI1 allows us to manipulate quantum operators by combining the unitary actions into a single operation:

$$\begin{aligned}&(\mathsf{qbit}\ x,y,z)\,.\,\{x,y,z \mathrel{*}= \mathsf{CNOT}_{yz}.\mathsf{CNOT}_{xy}.\mathsf{H}_x\}\,.\,c?[q]\,.\,\{q,x,y \mathrel{*}= \mathsf{H}_y.\mathsf{H}_q.\mathsf{CNOT}_{qx}\}\,.\\ &\{z \mathrel{*}= \mathsf{Z}^{\mathsf{measure}\ y}\}\,.\,\{z \mathrel{*}= \mathsf{X}^{\mathsf{measure}\ x}\}\,.\,\{z \mathrel{*}= \mathsf{Z}^{\mathsf{measure}\ q}\}\,.\,d![z]\,.\,\mathbf{0}\end{aligned}$$

The subscripts on the unitary operators indicates to which qubits they are applied. Applying rule QI2 to the measurement operations in the above process and noting that $\mathsf{CX} = \mathsf{CNOT}$, we get:

$$\begin{aligned}&(\mathsf{qbit}\ x,y,z)\,.\,\{x,y,z \mathrel{*}= \mathsf{CNOT}_{yz}.\mathsf{CNOT}_{xy}.\mathsf{H}_x\}\,.\,c?[q\!:\!\mathsf{Qbit}]\,.\\ &\{q,x,y \mathrel{*}= \mathsf{H}_y.\mathsf{H}_q.\mathsf{CNOT}_{qx}\}\,.\,\{y,z \mathrel{*}= \mathsf{CZ}\}\,.\,\{\mathsf{measure}\ y\}\,.\,\{x,z \mathrel{*}= \mathsf{CNOT}\}\,.\\ &\{\mathsf{measure}\ x\}\,.\,\{q,z \mathrel{*}= \mathsf{CZ}\}\,.\,\{\mathsf{measure}\ q\}\,.\,d![z]\,.\,\mathbf{0}\end{aligned}$$

We can swap the operators around due to commutativity provided that the operators are not acting on the same qubits. For example, we swap the order of the measurement on z and the controlled-Z operator on x and y because the qubits are independent; mathematically, this is due to the use of the tensor product. The commutativity of internal operators are expressed by the rules QC1-QC6. Using QC2 on the above process, we can move the measurements, and then using QI1, the unitary operators are combined to give:

$$\begin{aligned}&(\mathsf{qbit}\ x,y,z)\,.\,\{x,y,z \mathrel{*{=}} \mathsf{CNOT}_{yz}.\mathsf{CNOT}_{xy}.\mathsf{H}_x\}\,.\,c?[q\,{:}\,\mathsf{Qbit}]\,.\\ &\{q,x,y \mathrel{*{=}} \mathsf{H}_y.\mathsf{H}_q.\mathsf{CNOT}_{qx}\}\,.\,\{q,x,y,z \mathrel{*{=}} \mathsf{CZ}_{qz}.\mathsf{CNOT}_{xz}.\mathsf{CZ}_{yz}\}\,.\\ &\{\mathsf{measure}\ y\}\,.\,\{\mathsf{measure}\ x\}\,.\,\{\mathsf{measure}\ q\}\,.\,d![z]\,.\,\mathbf{0}\end{aligned}$$

The rules QC7-QC10 consider the commutativity of unitary operations with input and output actions by applying certain conditions if $\widetilde{y} \subseteq \mathbf{n}(\alpha)$ and $\widetilde{x} \cap \widetilde{y} = \emptyset$. The first condition is important as it ensures that there is no blocking behaviour. We are also able to commute qubit declarations with input and output actions since a qubit declaration is never blocking. This is expressed by the rules QC11 and QC12. We use these rules to bring the input action to the top and move the measurement operations after the output to give:

$$\begin{aligned}&c?[q]\,.\,(\mathsf{qbit}\ x,y,z)\,.\,\{x,y,z \mathrel{*{=}} \mathsf{CNOT}_{yz}.\mathsf{CNOT}_{xy}.\mathsf{H}_x\}\{q,x,y \mathrel{*{=}} \mathsf{H}_y.\mathsf{H}_q.\mathsf{CNOT}_{qx}\}\,.\\ &\{q,x,y,z \mathrel{*{=}} \mathsf{CZ}_{qz}.\mathsf{CNOT}_{xz}.\mathsf{CZ}_{yz}\}\,.\,d![z]\,.\,\{\mathsf{measure}\ y\}\,.\,\{\mathsf{measure}\ x\}\,.\,\{\mathsf{measure}\ q\}\,.\,\mathbf{0}\end{aligned}$$

With the help of the principle of deferred measurement, we were able to swap classical control for quantum control. Now we consider the *principle of implicit measurement* [16] which states that, any qubits at the end of a circuit may be assumed to be measured. This is provided by the rule QS1. Applying this rule to eliminate the measurements and combining the remaining quantum operators with QI1, we obtain:

$$\begin{aligned}&c?[q]\,.\,(\mathsf{qbit}\ x,y,z)\,.\\ &\{q,x,y,z \mathrel{*{=}} \mathsf{CZ}_{qz}.\mathsf{CNOT}_{xz}.\mathsf{CZ}_{yz}.\mathsf{H}_y.\mathsf{H}_q.\mathsf{CNOT}_{qx}.\mathsf{CNOT}_{yz}.\mathsf{CNOT}_{xy}.\mathsf{H}_x\}\,.\,d![z]\,.\,\mathbf{0}\end{aligned}$$

In a similar way, the unitary operators and qubit declarations are removed by using the rules QS2 and QS3. We see that the qubits y, q and x will each finish in the state $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. So, we apply the Hadamard operator to each using the rule QS2 which allows these operations to be added. Combining these operators to a single unitary action by using QC8 and QI1; we get

$$\begin{aligned}&c?[q]\,.\,(\mathsf{qbit}\ x,y,z)\,.\\ &\{q,x,y,z \mathrel{*{=}} \mathsf{H}_y.\mathsf{H}_q.\mathsf{H}_x.\mathsf{CZ}_{qz}.\mathsf{CNOT}_{xz}.\mathsf{CZ}_{yz}.\mathsf{H}_y.\mathsf{H}_q.\mathsf{CNOT}_{qx}.\mathsf{CNOT}_{yz}.\mathsf{CNOT}_{xy}.\mathsf{H}_x\}\,.\\ &d![z]\,.\,\mathbf{0}\end{aligned}$$

Next, we insert a permutation in order to swap the output qubit z with q. Rule QP1 defines this action where π is the permutation of qubits and the corresponding permutation on the quantum state is given by Π. Applying this rule and followed by QI1, we get

$$c?[q]\,.\,(\mathsf{qbit}\ x,y,z)\,.\,\{q,x,y,z \mathrel{*{=}} U\}\,.\,d![q]\,.\,\mathbf{0} \tag{8}$$

where $\pi(q) = z, \pi(z) = q, \pi(x) = x, \pi(y) = y$ and $U = \mathit{\Pi}.\mathsf{H}_y.\mathsf{H}_q.\mathsf{H}_x.\mathsf{CZ}_{qz}.\mathsf{CNOT}_{xz}.\mathsf{CZ}_{yz}.\mathsf{H}_y.\mathsf{H}_q.\mathsf{CNOT}_{qx}.\mathsf{CNOT}_{yz}.\mathsf{CNOT}_{xy}.\mathsf{H}_x$. Now, we have the qubit declaration $(\mathsf{qbit}\ x, y, z)$ which introduces three qubits in the combined state $|000\rangle$. We can define a linear map Q for which the action of teleportation on the single qubit q is given by UQ. Based on QI1, we use a similar rule QD1 to deal with quantum operators that appear under qubit declarations.

We have $UQ = \mathsf{I}_{qxyz}Q$ where I_{qxyz} is the identity operator on qubits q, x, y, z. Then by applying QD1 to Eq. 8, we get $c?[q] \,.\, \{q, x, y, z \mathrel{*}= \mathsf{I}\} \,.\, d![q] \,.\, \mathbf{0}$. We can now apply QI1, QC8 and QS3 to give

$$c?[q] \,.\, \{q \mathrel{*}= \mathsf{I}\} \,.\, d![q] \,.\, \mathbf{0}$$

This is a special case of QP1 where we consider identity permutation that results in the process which we are aiming for: $c?[q] \,.\, d![q] \,.\, \mathbf{0}$ □

5.2 Other quantum protocols

In this section, we will discuss the analysis of three essential quantum protocols using our axioms. The CQP definitions of all the protocols are given in Figure 3. We have omitted the types of channels in our definitions for brevity.

Superdense Coding (*SDC*): It involves two users (*Alice* and *Bob*) sharing a pair of entangled qubits. In this protocol, two classical bits are communicated by exchanging a single qubit. Alice is in possession of the first qubit, while Bob has possession of the second qubit. By sending the single qubit in her possession to Bob, it turns out Alice can communicate two classical bits to Bob. The specification process for this protocol is *CIdent*.

Proposition 2. $\mathit{SDC}(c, d) \mathrel{\underline{\leftrightarrow}}^{c} \mathit{CIdent}(c, d)$

Proof. We begin by eliminating the parallel composition in the process *SDC* as we had done earlier for *QSS*. By applying the expansion law E1 to the definition of *SDC*, to get:

$$\begin{aligned}&(\mathsf{qbit}\ x, y) \,.\, \{x \mathrel{*}= \mathsf{H}\} \,.\, \{x, y \mathrel{*}= \mathsf{CNOT}\} \,.\, (\mathsf{new}\ e)(c?[a, b] \,.\\ &(\mathit{Alice}' \mid \mathit{Bob}) + e?[x] \,.\, (\mathit{Alice} \mid \mathit{Bob}'))\end{aligned} \tag{9}$$

where $\mathit{Alice}' = c?[a, b] \,.\, \mathit{Alice}$and $\mathit{Bob}' = e?[x] \,.\, \mathit{Bob}$. Using the rules R1 – R3 on Eq. 9, the second term of the sum vanishes and rearranging the terms, we get:

$$\begin{aligned}&(\mathsf{qbit}\ x, y) \,.\, \{x \mathrel{*}= \mathsf{H}\} \,.\, \{x, y \mathrel{*}= \mathsf{CNOT}\} \,.\, c?[a, b] \,.\, (\mathsf{new}\ e)(\{x \mathrel{*}= \mathsf{X}^b\} \,.\, \{y \mathrel{*}= \mathsf{Z}^a\} \,.\\ &e![x] \,.\, \mathbf{0} \mid e?[x] \,.\, \{x, y \mathrel{*}= \mathsf{CNOT}\} \,.\, \{x \mathrel{*}= \mathsf{H}\} \,.\, d![\mathsf{measure}\ x, \mathsf{measure}\ y] \,.\, \mathbf{0})\end{aligned} \tag{10}$$

Expanding Eq. 10 as before and doing similar manipulations, we arrive at:

$$\begin{aligned}&(\mathsf{qbit}\ x, y) \,.\, \{x \mathrel{*}= \mathsf{H}\} \,.\, \{x, y \mathrel{*}= \mathsf{CNOT}\} \,.\, c?[a, b] \,.\, \{x \mathrel{*}= \mathsf{X}^b\} \,.\, \{y \mathrel{*}= \mathsf{Z}^a\} \,.\, (\mathsf{new}\ e)\\ &(e![x] \,.\, \mathbf{0} \mid e?[x] \,.\, \{x, y \mathrel{*}= \mathsf{CNOT}\} \,.\, \{x \mathrel{*}= \mathsf{H}\} \,.\, d![\mathsf{measure}\ x, \mathsf{measure}\ y] \,.\, \mathbf{0})\end{aligned} \tag{11}$$

$\mathit{Alice}(c,e,x,y) = c?[a\!:\!\mathsf{Bit}, b\!:\!\mathsf{Bit}] \,.\, \{x \mathrel{*}= \mathsf{X}^b\} \,.\, \{y \mathrel{*}= \mathsf{Z}^a\} \,.\, e![x] \,.\, \mathbf{0}$
$\mathit{Bob}(e,d,y) = e?[x\!:\!\mathsf{Qbit}] \,.\, \{x,y \mathrel{*}= \mathsf{CNOT}\} \,.\, \{x \mathrel{*}= \mathsf{H}\} \,.\, d![\mathsf{measure}\ x, \mathsf{measure}\ y] \,.\, \mathbf{0}$
$\mathit{SDC}(c,d) = (\mathsf{qbit}\ x,y)(\{x \mathrel{*}= \mathsf{H}\} \,.\, \{x,y \mathrel{*}= \mathsf{CNOT}\} \,.\, (\mathsf{new}\ e)(\mathit{Alice}(c,e,x,y) \mid \mathit{Bob}(e,d,y))$
$\mathit{CIdent}(c,d) = c?[a\!:\!\mathsf{Bit}, b\!:\!\mathsf{Bit}] \,.\, d![a,b] \,.\, \mathbf{0}$

(**i**)

$\mathit{Elsa}(a,c,d) = (\mathsf{qbit}\ x,y) a?[q\!:\!\mathsf{Qbit}, r\!:\!\mathsf{Qbit}] \,.\, \{x \mathrel{*}= \mathsf{H}\} \,.\, \{x,y \mathrel{*}= \mathsf{CNOT}\} \,.\, c![q,x] \,.\, d![r,y] \,.\, \mathbf{0}$
$\mathit{Anna}(c,e,f,g) = c?[q,x] \,.\, \{x,q \mathrel{*}= \mathsf{CNOT}\} \,.\, e?[j\!:\!\mathsf{Bit}] \,.\, \{x \mathrel{*}= \mathsf{X}^{\mathsf{measure}\ q}\} \,.\, f![\mathsf{measure}\ q] \,.$
$\{x \mathrel{*}= \mathsf{Z}^j\} \,.\, g![x] \,.\, \mathbf{0}$
$\mathit{Iven}(d,f,e,h) = d?[r,y] \,.\, \{r,y \mathrel{*}= \mathsf{CNOT}\} \,.\, \{r \mathrel{*}= \mathsf{H}\} \,.\, e![\mathsf{measure}\ r] \,.\, f?[i\!:\!\mathsf{Bit}] \,.\, \{y \mathrel{*}= \mathsf{X}^i\} \,.$
$\{y \mathrel{*}= \mathsf{Z}^{\mathsf{measure}\ r}\} \,.\, h![y] \,.\, \mathbf{0}$
$\mathit{Bob}(g,h,b) = g?[x] \,.\, h?[y] \,.\, b![x,y] \,.\, \mathbf{0}$
$\mathit{RCNOT}(a,b) = (\mathsf{new}\ c,d,e,f,g,h)(\mathit{Elsa}(a,c,d) \mid \mathit{Anna}(c,e,f,g) \mid \mathit{Iven}(d,f,e,h) \mid \mathit{Bob}(g,h,b))$
$\mathit{SCNOT}(a,b) = a?[q\!:\!\mathsf{Qbit}, r\!:\!\mathsf{Qbit}] \,.\, \{r,q \mathrel{*}= \mathsf{CNOT}\} \,.\, b![q,r] \,.\, \mathbf{0}$

(**ii**)

$\mathit{Alice}(a,b) = (\mathsf{qbit}\ y,z) a?[x\!:\!\mathsf{Qbit}] \,.\, \{x,z \mathrel{*}= \mathsf{CNot}\} \,.\, \{x,y \mathrel{*}= \mathsf{CNot}\} \,.\, b![x,y,z] \,.\, \mathbf{0}$
$\mathit{NoiseRnd}(p) = (\mathsf{qbit}\ u,v)\{u \mathrel{*}= \mathsf{H}\} \,.\, \{v \mathrel{*}= \mathsf{H}\} \,.\, p![\mathsf{measure}\ u, \mathsf{measure}\ v] \,.\, \mathbf{0}$
$\mathit{NoiseErr}(b,p,c) = b?[x\!:\!\mathsf{Qbit}, y\!:\!\mathsf{Qbit}, z\!:\!\mathsf{Qbit}] \,.\, p?[j\!:\!\mathsf{bit}, k\!:\!\mathsf{bit}] \,.\, \{x \mathrel{*}= \mathsf{X}^{jk}\} \,.\, \{y \mathrel{*}= \mathsf{X}^{j\overline{k}}\} \,.$
$\{z \mathrel{*}= \mathsf{X}^{\overline{j}k}\} \,.\, c![x,y,z] \,.\, \mathbf{0}$
$\mathit{Noise}(b,c) = (\mathsf{new}\ p)(\mathit{NoiseRnd}(p) \mid \mathit{NoiseErr}(b,p,c))$
$\mathit{BobRec}(c,p) = (\mathsf{qbit}\ s,t) c?[x,y,z] \,.\, \{x,s \mathrel{*}= \mathsf{CNot}\} \,.\, \{y,s \mathrel{*}= \mathsf{CNot}\} \,.\, \{x,t \mathrel{*}= \mathsf{CNot}\} \,.$
$\{z,t \mathrel{*}= \mathsf{CNot}\} \,.\, p![x,y,z, \mathsf{measure}\ s, \mathsf{measure}\ t] \,.\, \mathbf{0}$
$\mathit{BobCorr}(p,d) = p?[x,y,z,j\!:\!\mathsf{bit}, k\!:\!\mathsf{bit}] \,.\, \{x \mathrel{*}= \mathsf{X}^{jk}\} \,.\, \{y \mathrel{*}= \mathsf{X}^{j\overline{k}}\} \,.\, \{z \mathrel{*}= \mathsf{X}^{\overline{j}k}\} \,.$
$\{x,y \mathrel{*}= \mathsf{CNot}\} \,.\, \{x,z \mathrel{*}= \mathsf{CNot}\} \,.\, d![x] \,.\, \mathbf{0}$
$\mathit{Bob}(c,d) = (\mathsf{new}\ p)(\mathit{BobRec}(c,p) \mid \mathit{BobCorr}(p,d))$
$\mathit{QECC}(a,d) = (\mathsf{new}\ b,c)(\mathit{Alice}(a,b) \mid \mathit{Noise}(b,c) \mid \mathit{Bob}(c,d))$

(**iii**)

Fig. 3. CQP definitions of quantum protocols: (i) Superdense coding (SDC), (ii) Remote CNOT (RCNOT) and (iii) Quantum error correction (QECC).

The next is a τ transition that happens internally and then performing several iterations using R3 and followed by $(\mathsf{new}\ e) \,.\, \mathbf{0} = \mathbf{0}$, we get:

$$\begin{array}{l}(\mathsf{qbit}\ x,y) \,.\, \{x \mathrel{*}= \mathsf{H}\} \,.\, \{x,y \mathrel{*}= \mathsf{CNOT}\} \,.\, c?[a,b] \,.\, \{x \mathrel{*}= \mathsf{X}^b\} \,.\, \{y \mathrel{*}= \mathsf{Z}^a\} \,. \\ \tau \,.\, \{x,y \mathrel{*}= \mathsf{CNOT}\} \,.\, \{x \mathrel{*}= \mathsf{H}\} \,.\, d![\mathsf{measure}\ x, \mathsf{measure}\ y] \,.\, \mathbf{0}\end{array} \tag{12}$$

Then using Tau1 in Eq. 12, we arrive at the sequentialised form of definition of *SDC*:

$$\begin{array}{l}(\mathsf{qbit}\ x,y) \,.\, \{x \mathrel{*}= \mathsf{H}\} \,.\, \{x,y \mathrel{*}= \mathsf{CNOT}\} \,.\, c?[a,b] \,.\, \{x \mathrel{*}= \mathsf{X}^b\} \,.\, \{y \mathrel{*}= \mathsf{Z}^a\} \\ \{x,y \mathrel{*}= \mathsf{CNOT}\} \,.\, \{x \mathrel{*}= \mathsf{H}\} \,.\, d![\mathsf{measure}\ x, \mathsf{measure}\ y] \,.\, \mathbf{0}\end{array} \tag{13}$$

Using the rule Qi1 on Eq. 13 to combine the unitary actions to give:

$$\begin{array}{l}(\mathsf{qbit}\ x,y) \,.\, \{x,y \mathrel{*}= \mathsf{CNOT}_{xy}.\mathsf{H}_x\} \,.\, c?[a,b] \,.\, \{xy \mathrel{*}= \mathsf{H}_x.\mathsf{CNOT}_{xy}.\mathsf{Z}^a_{y.}\mathsf{X}^b_x\} \\ d![\mathsf{measure}\ x, \mathsf{measure}\ y] \,.\, \mathbf{0}\end{array} \tag{14}$$

$(\mathsf{qbit}\ x, y)a?[q\!:\!\mathsf{Qbit}, r\!:\!\mathsf{Qbit}]\,.\,\{x \mathrel{*}= \mathsf{H}\}\,.\,\{x, y \mathrel{*}= \mathsf{CNOT}\}\,.\,\{x, q \mathrel{*}= \mathsf{CNOT}\}\,.\,\{r, y \mathrel{*}= \mathsf{CNOT}\}\,.$
$\{r \mathrel{*}= \mathsf{H}\}\,.\,\{x \mathrel{*}= \mathsf{X}^{\mathsf{measure}\ q}\}\,.\,\{y \mathrel{*}= \mathsf{X}^{\mathsf{measure}\ q}\}\,.\,\{x \mathrel{*}= \mathsf{Z}^{\mathsf{measure}\ r}\}\,.\,\{y \mathrel{*}= \mathsf{Z}^{\mathsf{measure}\ r}\}\,.\,b![x, y]\,.\,\mathbf{0}$

Applying QI1 and QI2 to combine the unitary operations, we get:
$(\mathsf{qbit}\ x, y)a?[q, r]\,.\,\{q, r, x, y \mathrel{*}= \mathsf{H}_r.\mathsf{CNOT}_{ry}.\mathsf{CNOT}_{xq}.\mathsf{CNOT}_{xy}.\mathsf{H}_x\}\,.\,\{q, x, y \mathrel{*}= \mathsf{CNOT}_{qy}.\mathsf{CNOT}_{qx}\}\,.$
$\{\mathsf{measure}\ q\}\,.\,\{r, x, y \mathrel{*}= \mathsf{CZ}_{ry}.\mathsf{CZ}_{rx}\}\,.\,\{\mathsf{measure}\ r\}\,.\,b![x, y]\,.\,\mathbf{0}$
Applying QC2, QC10, and QS1 to remove the measurements:
$(\mathsf{qbit}\ x, y)a?[q, r]\,.\,\{q, r, x, y \mathrel{*}= \mathsf{H}_r.\mathsf{CNOT}_{ry}.\mathsf{CNOT}_{xq}.\mathsf{CNOT}_{xy}.\mathsf{H}_x\}\,.\,\{q, x, y \mathrel{*}= \mathsf{CNOT}_{qy}.\mathsf{CNOT}_{qx}\}\,.$
$\{r, x, y \mathrel{*}= \mathsf{CZ}_{ry}.\mathsf{CZ}_{rx}\}\,.\,b![x, y]\,.\,\mathbf{0}$

Applying QC11 and QI1,to move the input action in the front and combining the unitary operations:
$a?[q, r]\,.\,(\mathsf{qbit}\ x, y)\,.\,\{q, r, x, y \mathrel{*}= \mathsf{CZ}_{ry}.\mathsf{CZ}_{rx}.\mathsf{CNOT}_{qy}.\mathsf{CNOT}_{qx}.\mathsf{H}_r.\mathsf{CNOT}_{ry}.\mathsf{CNOT}_{xq}.\mathsf{CNOT}_{xy}.\mathsf{H}_x\}$
$.\,b![x, y]\,.\,\mathbf{0}$
Applying QS2, QC8 and QI1, to add a Hadamard operation on qubit r to give:
$a?[q, r]\,.\,(\mathsf{qbit}\ x, y)\,.\,\{q, r, x, y \mathrel{*}= \mathsf{H}_r.\mathsf{CZ}_{ry}.\mathsf{CZ}_{rx}.\mathsf{CNOT}_{qy}.\mathsf{CNOT}_{qx}.\mathsf{H}_r.\mathsf{CNOT}_{ry}.\mathsf{CNOT}_{xq}.\mathsf{CNOT}_{xy}.$
$\mathsf{H}_x\}\,.\,b![x, y]\,.\,\mathbf{0}$
Applying QP1 a permutation operator to perform $\pi(q) = x$ and $\pi(x) = q$, we get:
$a?[q, r]\,.\,(\mathsf{qbit}\ x, y)\,.\,\{q, r, x, y \mathrel{*}= \mathit{\Pi}.\mathsf{H}_r.\mathsf{CZ}_{ry}.\mathsf{CZ}_{rx}.\mathsf{CNOT}_{qy}.\mathsf{CNOT}_{qx}.\mathsf{H}_r.\mathsf{CNOT}_{ry}.\mathsf{CNOT}_{xq}.\mathsf{CNOT}_{xy}.$
$\mathsf{H}_x\}\,.\,b![q, y]\,.\,\mathbf{0}$

Applying QS2, QC8 and QI1 to add a Hadamard operation on qubit x to give:
$a?[q, r]\,.\,(\mathsf{qbit}\ x, y)\,.\,\{q, r, x, y \mathrel{*}= \mathsf{H}_x.\mathit{\Pi}.\mathsf{H}_r.\mathsf{CZ}_{ry}.\mathsf{CZ}_{rx}.\mathsf{CNOT}_{qy}.\mathsf{CNOT}_{qx}.\mathsf{H}_r.\mathsf{CNOT}_{ry}.\mathsf{CNOT}_{xq}.$
$\mathsf{CNOT}_{xy}.\mathsf{H}_x\}\,.\,b![q, y]\,.\,\mathbf{0}$
Applying QP1 a permutation operator as before to perform $\pi(r) = y$ and $\pi(y) = r$, we get:
$a?[q, r]\,.\,(\mathsf{qbit}\ x, y)\,.\,\{q, r, x, y \mathrel{*}= \mathit{\Pi}.\mathsf{H}_x.\mathit{\Pi}.\mathsf{H}_r.\mathsf{CZ}_{ry}.\mathsf{CZ}_{rx}.\mathsf{CNOT}_{qy}.\mathsf{CNOT}_{qx}.\mathsf{H}_r.\mathsf{CNOT}_{ry}.\mathsf{CNOT}_{xq}.$
$\mathsf{CNOT}_{xy}.\mathsf{H}_x\}\,.\,b![q, r]\,.\,\mathbf{0}$
Applying QD1, we get: $a?[q, r]\,.\,(\mathsf{qbit}\ x, y)\,.\,\{r, q \mathrel{*}= \mathsf{CNOT}\}\,.\,\{x, y \mathrel{*}= \mathsf{I}\}\,.\,b![q, r]\,.\,\mathbf{0}$
Applying QC8, QS2, QC3 and QS3, we get: $a?[q, r]\,.\,\{r, q \mathrel{*}= \mathsf{CNOT}\}\,.\,b![q, r]\,.\,\mathbf{0}$

Fig. 4. Reasoning about Remote CNOT

To move the input actions to the top, we apply QC7 and QC11 on Eq. 14 to give:

$$\begin{array}{l} c?[a, b]\,.\,(\mathsf{qbit}\ x, y)\,.\,\{x, y \mathrel{*}= \mathsf{CNOT}_{xy}.\mathsf{H}_x\}\{xy \mathrel{*}= \mathsf{H}_x.\mathsf{CNOT}_{xy}.\mathsf{Z}^a_{y.}\mathsf{X}^b_x\} \\ d![\mathsf{measure}\ x, \mathsf{measure}\ y]\,.\,\mathbf{0} \end{array} \tag{15}$$

Applying QI1 on Eq. 15, we arrive at the sequential definition of SDC.

$$c?[a, b]\,.\,(\mathsf{qbit}\ x, y)\,.\,\{x, y \mathrel{*}= \mathsf{H}_x.\mathsf{CNOT}_{xy}.\mathsf{Z}^a_{y.}\mathsf{X}^b_x.\mathsf{CNOT}_{xy}.\mathsf{H}_x\}\,.\,d![\mathsf{measure}\ x, \mathsf{measure}\ y]\,.\,\mathbf{0}$$

Then by applying the rules QI1 to combine the unitary operations into a single action and using QC7 and QC11 to move the input action to the beginning of the process, we get:

$$c?[a\!:\!\mathsf{Bit}, b\!:\!\mathsf{Bit}]\,.\,(\mathsf{Qbit} : x, y)\,.\,\{xy \mathrel{*}= U^{ab}\}\,.\,d![\mathsf{measure}\ x, \mathsf{measure}\ y]\,.\,\mathbf{0} \tag{16}$$

Here, $U^{ab} = \mathsf{H}_x.\mathsf{CNOT}_{xy}.\mathsf{Z}^a_y.\mathsf{X}^b_x.\mathsf{CNOT}_{xy}.\mathsf{H}_x$, is a unitary operator which depends on the classical bits a and b. Now, let $P(a, b) = (\mathsf{Qbit} : x, y)\,.\,\{xy \mathrel{*}=$

$U^{ab}\}\,.\,d![\mathsf{measure}\ x, \mathsf{measure}\ y]\,.\,\mathbf{0}$ and $Q(a,b) = d![a,b]\,.\,\mathbf{0}$ be two processes that are parameterised by the classical bits a and b. It can be proven easily that $P(a,b) \underline{\leftrightarrow}^c Q(a,b)$ for all possible values of a and b. Hence using Cv1, Eq. 16 $\underline{\leftrightarrow}^c$ $c?[a,b]\,.\,d![a,b]\,.\,\mathbf{0}$, which is the specification process *CIdent*. □

Remote CNOT (*RCNOT*): The protocol [20] demonstrates the concept of teleporting a quantum logic gate. Our definitions for the protocol are shown in Figure 3(ii) consisting of four users. *Anna* and *Iven* have in their possession qubits q and r respectively, which they have received from *Elsa*. Also, *Elsa* has prepared an EPR pair with qubits x and y before sharing it with *Anna* and *Iven*. The objective of the protocol is that *Anna* and *Iven* would like to perform a CNOT operation with their qubits q and r, without communicating any quantum information between them. *Anna* entangles her qubits q and x by performing a CNOT and *Iven* performs the same with his qubits in addition to a H operation on r before measuring it. He then sends the result to *Anna*. She measures her qubit q and performs certain unitary operations on x based on the outcome of her's and *Iven*'s measurements. Also, she sends her measurement outcome to *Iven*. Hence, *Anna* and *Iven* communicate only their classical results between them, which are used to perform unitary operation on their EPR pair. Essentially *Iven*'s qubit y is a CNOT operation of q and r and they communicate their EPR pair qubits (x and y) to *Bob*. The specification of *RCNOT* is *SCNOT*.

Proposition 3. $RCNOT(a,b) \underline{\leftrightarrow}^c SCNOT(a,b)$

Proof. The proof is provided in Figure 4. □

Quantum Error Correction (*QECC*): *QECC* consists of three processes: *Alice*, *Bob* and *Noise*. *Alice* wishes to send a qubit to *Bob* over a noisy channel, represented by *Noise*. She uses a error correcting code based on threefold repetition [16]. The code is able to correct single bit-flip error in each block of three transmitted qubits, so for the purpose of this example, in each block of three qubits, *Noise* either applies X to one of them or does nothing. *Bob* uses the appropriate decoding procedure to recover *Alice*'s original qubit. The CQP definitions are provided in Figure 3 (iii) and this system is equivalent to *Identity*.

Proposition 4. $QECC(a,d) \underline{\leftrightarrow}^c Identity(a,d)$

Proof. The proof is provided in Figure 5 and alternatively given in [3] by constructing a bisimulation. In Figure 5, we begin with the sequentialised definition of *QECC* which is obtained in the same way as we had done for the previous protocols. □

Proposition 5. $Teleport \underline{\leftrightarrow}^c QSS \underline{\leftrightarrow}^c QECC$

Proof. Quantum teleportation (*Teleport*) is a protocol which allows two users who share an entangled pair of qubits to exchange an unknown quantum state by communicating only two classical bits. The CQP definition of *Teleport* protocol and the proof that $Teleport \underline{\leftrightarrow}^c Identity$ are provided in [2]. We prove the proposition easily using the *transitivity* of $\underline{\leftrightarrow}^c$ as we have seen that *QECC* and *QSS* are equivalent to *Identity* through Propositions 1 and 4. □

$(\mathsf{qbit}\ y, z) a?[x] \,.\, \{x, y, z \mathrel{*}= \mathsf{CNOT}_{xy}.\mathsf{CNOT}_{xz}\} \,.\, (\mathsf{qbit}\ u, v)\{u, v \mathrel{*}= \mathsf{H}_v.\mathsf{H}_u\} \,.$
$\{x \mathrel{*}= \mathsf{X}^{\mathsf{measure}\ u.\mathsf{measure}\ v}\} \,.\, \{y \mathrel{*}= \mathsf{X}^{\mathsf{measure}\ u.\mathsf{measure}\ v}\} \,.\, \{z \mathrel{*}= \mathsf{X}^{\mathsf{measure}\ u.\mathsf{measure}\ v}\} \,.\, (\mathsf{qbit}\ s, t) \,.$
$\{x, y, z, s, t \mathrel{*}= \mathsf{CNOT}_{zt}.\mathsf{CNOT}_{xt}.\mathsf{CNOT}_{ys}.\mathsf{CNOT}_{xs}\} \,.\, \{x \mathrel{*}= \mathsf{X}^{\mathsf{measure}\ s.\mathsf{measure}\ t}\} \,.$
$\{y \mathrel{*}= \mathsf{X}^{\mathsf{measure}\ s.\mathsf{measure}\ t}\} \,.\, \{z \mathrel{*}= \mathsf{X}^{\mathsf{measure}\ s.\mathsf{measure}\ t}\} \,.\, \{x, y, z \mathrel{*}= \mathsf{CNOT}_{xz}.\mathsf{CNOT}_{xy}\} \,.\, d![x] \,.\, \mathbf{0}$

Applying Qi3, Qc2, Qs1 and Qc3, we get:$(\mathsf{qbit}\ y, z) \,.\, a?[x] \,.\, (\mathsf{qbit}\ u, v) \,.\, (\mathsf{qbit}\ s, t) \,.$
$\{x, y, z, u, v \mathrel{*}= \mathsf{CNOT}_{(uv)z}.\mathsf{CNOT}_{(uv)y}.\mathsf{CNOT}_{(uv)x}.\mathsf{H}_v.\mathsf{H}_u.\mathsf{CNOT}_{xy}.\mathsf{CNOT}_{xz}\} \,.$
$\{x, y, z, s, t \mathrel{*}= \mathsf{CNOT}_{xz}.\mathsf{CNOT}_{xy}.\mathsf{CNOT}_{(st)z}.\mathsf{CNOT}_{(st)y}.\mathsf{CNOT}_{(st)x}.\mathsf{CNOT}_{zt}.\mathsf{CNOT}_{xt}.\mathsf{CNOT}_{ys}.\mathsf{CNOT}_{xs}\}$
$.\, d![x] \,.\, \mathbf{0}$

Applying Qc11 and Qi1, we get:
$a?[x] \,.\, (\mathsf{qbit}\ y, z, u, v, s, t) \,.\, \{x, y, z, u, v, s, t* = \mathsf{CNOT}_{xz}.\mathsf{CNOT}_{xy}.\mathsf{CNOT}_{(st)z}.\mathsf{CNOT}_{(st)y}.\mathsf{CNOT}_{(st)x}.$
$\mathsf{CNOT}_{zt}.\mathsf{CNOT}_{xt}.\mathsf{CNOT}_{ys}.\mathsf{CNOT}_{xs}.\mathsf{CNOT}_{(uv)z}.\mathsf{CNOT}_{(uv)y}.\mathsf{CNOT}_{(uv)x}.\mathsf{H}_v.\mathsf{H}_u.\mathsf{CNOT}_{xy}.$
$\mathsf{CNOT}_{xz}\} \,.\, d![x] \,.\, \mathbf{0}$

Applying Qs2, Qi1 and Qd1, we get: $a?[x] \,.\, (\mathsf{qbit}\ y, z, u, v, s, t) \,.\, \{x, y, z, u, v, s, t \mathrel{*}= \mathsf{I}\} \,.\, d![x] \,.\, \mathbf{0}$
Applying Qi1, we get: $a?[x] \,.\, (\mathsf{qbit}\ y, z, u, v, s, t) \,.\, \{x \mathrel{*}= \mathsf{I}\} \,.\, \{y, z, u, v, s, t \mathrel{*}= \mathsf{I}\} \,.\, d![x] \,.\, \mathbf{0}$
Applying Qc10, Qs1, Qc12 andQs3, we get: $a?[x] \,.\, d![x] \,.\, \mathbf{0}$

Fig. 5. Reasoning about quantum error correction

The congruence property helps to analyse a combination of systems. For example, if we consider a process defined as *System* $(c, d) = (\mathsf{new}\ a)\,\mathit{Teleport}(c, a) \mid \mathit{QECC}(a, d)$. We can consider this equivalent to a process $(\mathsf{new}\ a)\,\mathit{Teleport}(c, a) \mid \mathit{Identity}(a, d)$ by using Proposition 4. This is also equivalent to $(\mathsf{new}\ a)\mathit{Identity}(c, a) \mid \mathit{Identity}(a, d)$ which is equivalent to $\mathit{Identity}(c, d)$.

6 Conclusion and Future Work

We have explained the use of the quantum process calculus CQP in analysing various quantum protocols. We have summarised the theory of equational axioms based on the concept of behavioural equivalence which is presented in full detail in [2]. We present the analysis of *QSS* by using the equational axioms and have verified the correctness of *QSS* and other quantum protocols.

Verification of the quantum protocols using the bisimulation relations requires hard work. First, we need to perform the computation of the *System* (that models the system of interest) and the *Specification*, which expresses the desired behaviour of *System*, and then we need to establish a bisimulation relation. Because of equational reasoning, we show that we can reduce the need to explicitly construct bisimulation relations. The next step for this line of research is to prove the completeness of these laws. The axioms provide the additional advantage for automated reasoning which is our long-term goal following the recent work on automated equivalence checking [1].

References

1. E. Ardeshir-Larijani, S. J. Gay, and R. Nagarajan. Verification of concurrent quantum protocols by equivalence checking. In *Proceedings of the 20th International Conference on TACAS*, volume 8413, pages 500–514. LNCS, 2014.
2. T. A. S. Davidson. *Formal Verification Techniques using Quantum Process Calculus.* PhD thesis, University of Warwick, 2011.
3. T. A. S. Davidson, S. J. Gay, R. Nagarajan, and I. V. Puthoor. Analysis of a quantum error correcting code using quantum process calculus. In *Proceedings of the International Workshop on QPL*, volume 95, pages 67–80. EPTCS, 2011.
4. Y. Feng, R. Duan, Z. Ji, and M. Ying. Probabilistic bisimilarities between quantum processes. arXiv:cs.LO/0601014, 2006.
5. Y. Feng, R. Duan, and M. Ying. Bisimulation for quantum processes. In *ACM Symposium on Principles of Programming Languages*, pages 523–534. ACM, 2011.
6. S. Franke-Arnold, S. J. Gay, and I. V. Puthoor. Quantum process calculus for linear optical quantum computing. In *Proceedings of the 5th Conference on Reversible Computation (RC)*, volume 7948, pages 234–246. LNCS, 2013.
7. S. Franke-Arnold, S. J. Gay, and I. V. Puthoor. Verification of linear optical quantum computing using quantum process calculus. In *Proceedings of the Combined International Workshop on Expressiveness in Concurrency and Structural Operational Semantics (EXPRESS/SOS)*, volume 160, pages 111–129. EPTCS, 2014.
8. S. Gay and R. Nagarajan. Communicating Quantum Processes. In *ACM Symposium on Principles of Programming Languages*, pages 145–157. ACM, 2005.
9. S. J. Gay and R. Nagarajan. Types and Typechecking for Communicating Quantum Processes. *Mathematical Structures in Computer Science*, 16(3):375–406, 2006.
10. S. J. Gay, N. Papanikolaou, and R. Nagarajan. QMC: a model checker for quantum systems. In *Proceedings of the 20th International Conference on Computer Aided Verification*, number 5123, pages 543–547. LNCS, 2008.
11. M. Hillery, V. Buzek, and A. Berthiaume. Quantum secret sharing. *Phys. Rev. A*, 59:1829–1834, 1999.
12. IDQ. http://www.idquantique.com/company/presentation.html.
13. MagiQ. http://www.magiqtech.com/magiq/home.html.
14. D. Mayers. Unconditional security in quantum cryptography. *Journal of the ACM*, 48(3):351–406, 2001.
15. R. Milner. *Communicating and Mobile Systems: the Pi-Calculus.* Cambridge University Press, 1999.
16. M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information.* Cambridge University Press, 2000.
17. I. V. Puthoor. *Theory and applications of quantum process calculus.* PhD thesis, University of Glasgow, 2015.
18. N. Trčka and S. Georgievska. Branching bisimulation congruence for probabilistic systems. *Electronic Notes in Theoretical Computer Science*, 220(3):129 – 143, 2008.
19. A. K. Wright and M. Felleisen. A syntactic approach to type soundness. *Information and Computation*, 115(1):38–94, 1994.
20. X. Zhou, D. W. Leung, and I. L. Chuang. Methodology for quantum logic gate construction. *Phys. Rev. A*, 62, 2000.