An Axiomatization for Quantum Processes to Unifying Quantum and Classical Computing

Yong Wang[1] (ORCID)

Abstract

We establish an axiomatization for quantum processes, which is a quantum generalization of process algebra ACP (Algebra of Communicating Processes). We use the framework of a quantum process configuration $\langle p, \varrho \rangle$, but we treat it as two relative independent part: the structural part p and the quantum part ϱ, because the establishment of a sound and complete theory is dependent on the structural properties of the structural part p. We let the quantum part ϱ be the outcomes of execution of p to examine and observe the function of the basic theory of quantum mechanics. We establish not only a strong bisimilarity for quantum processes, but also a weak bisimilarity to model the silent step and abstract internal computations in quantum processes. The relationship between quantum bisimilarity and classical bisimilarity is established, which makes an axiomatization of quantum processes possible. An axiomatization for quantum processes called qACP is designed, which involves not only quantum information, but also classical information and unifies quantum computing and classical computing. qACP can be used easily and widely for verification of most quantum communication protocols.

Keywords Quantum processes · Process algebra · Algebra of communicating processes · Axiomatization

1 Introduction

The basic principles of quantum mechanics are widely adopted in computation and communication. As a relative novel computation pattern, quantum computing [21] brings the dawn of solving the so-called NP problem because of the strong parallel computation power of quantum computing. And also, many basic principles of quantum mechanics, such as Heisenberg uncertainty principle and quantum no-cloning theorem, provide

✉ Yong Wang
  wangy@bjut.edu.cn

[1] College of Computer Science and Technology, Faculty of Information Technology,
  Beijing University of Technology, Beijing, China

quantum communication protocols the so-called provable security. Now, some quantum communication protocols, especially quantum key distribution protocols, have already been commercially available.

Process algebra [1] is well known in capturing traditional computation, especially parallelism and concurrence, in an interleaving pattern, such as CCS (Calculus of Concurrent Process) [2, 3], CSP (Communicating Sequential Processes) [4] and ACP (Algebra of Communicating Processes) [5]. To unify quantum computing and classical computing under the same process algebra framework, is attractive and has an important significance, because most quantum communication protocols involve quantum information and classical information, quantum computing and classical computing.

In this paper, we design an axiomatization called qACP for quantum processes with a quantum generalization of process algebra ACP, which unifies quantum computing and classical computing. qACP consists of not only an operational semantics based on classical structural operational semantics, but also an equational logic, by use of which, most quantum communication protocols can be verified easily.

This paper is organized as follows. In Section 2, we introduce the related works. We do not repeat some preliminaries, including basic concepts and conclusions about basic linear algebra, basic quantum mechanics, equational logic, structural operational semantics and process algebra ACP, please refer to [5, 6] for details. In Section 3, we extend classical structural operational semantics to support quantum processes. The basic quantum process algebra (BQPA) is introduced in Section 4, Quantum Process Algebra with Parallelism (QPAP) and Algebra of Quantum Communicating Processes (AQCP) are designed in Section 5. To capture infinite computing in quantum processes, we discuss recursion in Section 6. To model silent step and abstract internal computation, silent step and abstraction operator are introduced in Section 7. We unify qACP and classical ACP in Section 8. An example of the famous BB84 protocol [18] is verified by use of qACP in Section 9. qACP can be extended easily in an elegant way, which is shown in Section 10. Finally, we conclude this paper in Section 11.

2 Related Works

Quantum process algebra provides formal tools for modeling, analysis and verification of quantum communication protocols, which combines quantum communications and quantum computing together. Gay and Nagarajan [9, 10] defined a language called CQP (Communicating Quantum Processes) by adding primitives for quantum measurements and transformation of quantum states to π-calculus. An operational semantics and a type system for CQP were also presented to prove that the semantics preserves typing and typing guarantees that each qubit is owned by a unique process within a system.

Jorrand and Lalire [11–14] defined a language called QPAlg (Quantum Process Algebra), in which, based on CCS [2, 3], primitives of unitary transformations and quantum measurements were added to CCS. An operational semantics based on probabilistic branching bisimulation was given in QPAlg.

Ying et al. [15] was introduced as a kind of algebra of pure quantum processes (no classical data involved) based on CCS. qCCS aimed at providing a suitable framework, in which the mechanism of quantum concurrent computation can be understood, and interactions and conjugation of computation and communication in quantum systems can be observed. In

qCCS, quantum operations (super operators) were chosen to describe transformations of quantum states, and quantum variables and their substitutions were carefully treated. An operational semantics for qCCS based on exact (strong) bisimulation and an approximation version of bisimulation were presented for qCCS.

Based on [15], several kind of bisimulations were presented for qCCS, such as probabilistic bisimulation [8], a kind of weak probabilistic bisimulation [17], open bisimulation [19] and symbolic bisimulation [16, 20]. These bisimulations provided qCSS with more semantic models. In some bisimulations, not only pure quantum data, but also classical data could be involved in qCCS.

In this paper, we propose an axiomatization of quantum processes called qACP, which is a quantum generalization of process algebra ACP. This work uses some results of the previous works, especially qCCS, in the following ways. (1) qACP still uses the concept of a quantum process configuration $\langle p, \varrho \rangle$ [8, 8–12, 15, 17, 20], which is usually consisted of a process term p and state information ϱ of all (public) quantum information variables. (2) Like qCCS, quantum operations are chosen to describe transformations of quantum states, and behave as the atomic actions of a pure quantum process. Quantum measurements are treated as quantum operations, so probabilistic bisimulations are avoided.

There are several innovations in this paper, we enumerate them as follows. (1) A weak bisimilarity (quantum branching bisimulation equivalence) is established for quantum processes. This weak bisimilarity is in a non-probabilistic way that follows [15] and can be used to model silent step and abstract internal actions. (2) We still use the framework of a quantum process configuration $\langle p, \varrho \rangle$, but we treat it as two relative independent part: the structural part p and the quantum part ϱ, because the establishment of a sound and complete theory is dependent on the structural properties of the structural part p. We let the quantum part ϱ be the outcomes of execution of p to examine and observe the function of the basic theory of quantum mechanics. We establish the relationship between quantum bisimilarity and classical bisimilarity, including strong bisimilarity and weak bisimilarity, which makes an axiomatization of quantum processes possible. (3) We establish a series of axiomatizations of quantum process algebra, including BQPA (Basic Quantum Process Algebra), QPAP (Quantum Process Algebra with Parallelism), AQCP (Algebra of Quantum Communicating Processes), AQCP with guarded linear recursion, and $AQCP_\tau$ with guarded linear recursion. Though these axiomatizations are based on classical axiomatizations of ACP which is based on the structural analysis the process p, they are not trivial and ordinary, because it is also necessary to examine if the outcomes ϱ of execution of p obey the basic quantum mechanics theory. For example, the associativity law of sequential composition $\cdot, (x \cdot y) \cdot z = x \cdot (y \cdot z)$ (x, y, z range over the collection of process terms), is based on the associativity of quantum operations. And the behaviors of the silent step τ in quantum processes and that in classical processes are different under the framework of quantum process configuration $\langle p, \varrho \rangle$. (4) In this paper, qACP and classical ACP are unified under the framework of quantum process configuration $\langle p, \varrho \rangle$. This unifying means that quantum information and classical information can be mixed in qACP and quantum computing and classical computing are unified in qACP. Thus, qACP can be used widely for verification of quantum communication protocols, which involve not only quantum information, but also classical information. (5) As a result of axiomatization, qACP has not only an operational semantics, but also an equation logic which makes the qACP can be used easily. qACP also inherits the modularity of ACP, and can be extended in an elegant way.

3 Structural Operational Semantics Extended to Support Quantum Processes

In the above section, operational semantics are described by labelled transitions among process configurations, and a process term is enough to represent a process configuration. But in quantum processes, to avoid the abuse of quantum information which may violate the no-cloning theorem, a quantum process configuration $\langle p, \varrho \rangle$ [8–12, 15, 17, 20] is usually consisted of a process term p and state information ϱ of all (public) quantum information variables. Though quantum information variables are not explicitly defined in qACP and are hidden behind quantum operations, more importantly, the state information ϱ is the effects of execution of a series of quantum operations on involved quantum systems, the execution of a series of quantum operations should not only obey the restrictions of the structure of the process terms, but also those of quantum mechanics principles. Through the state information ϱ, wecan check and observe the functions of quantum mechanics principles, such as quantum entanglement, quantum measurement, etc.

So, the operational semantics of quantum processes should be defined based on quantum process configuration $\langle p, \varrho \rangle$, in which $\varrho = \sigma$ of two state information ϱ and σ means equality under the framework of quantum information and quantum computing, that is, these two quantum processes are in the same quantum state. Several important concepts used in this paper are following. Here, we use α, β to denote quantum operations in contrast to classical actions a, b.

Definition 3.1 (Quantum process configuration) A quantum process configuration is defined to be a pair $\langle p, \varrho \rangle$, where p is a process (graph) called structural part of the configuration, and $\varrho \in \mathcal{D}(\mathcal{H})$ specifies the current state of the environment, which is called its quantum part.

Definition 3.2 (Quantum process graph) A quantum process (graph) $\langle p, \varrho \rangle$ is an LTS in which one state s is elected to be the root. If the LTS contains a transition $s \xrightarrow{\alpha} s'$, then $\langle p, \varrho \rangle \xrightarrow{\alpha} \langle p', \varrho' \rangle$ where $\langle p', \varrho' \rangle$ has root state s'. Moreover, if the LTS contains a transition sP, then $\langle p, \varrho \rangle P$. (1) A quantum process $\langle p_0, \varrho_0 \rangle$ is finite if and only if the process p_0 is finite. (2) A quantum process $\langle p_0, \varrho_0 \rangle$ is regular if and only if the process p_0 is regular.

Definition 3.3 (Quantum transition system specification) A quantum process transition rule ϱ is an expression of the form $\frac{H}{\pi}$, with H a set of expressions $\langle t, \varrho \rangle \xrightarrow{\alpha} \langle t', \varrho' \rangle$ and $\langle t, \varrho \rangle P$ with $t, t' \in \mathbb{T}(\Sigma)$ and $\varrho, \varrho' \in \mathcal{D}(\mathcal{H})$, called the (positive) premises of ϱ, and $t \xrightarrow{\alpha} t'$, called structural part of H and denoted as H_s. And π an expression $\langle t, \varrho \rangle \xrightarrow{\alpha} \langle t', \varrho' \rangle$ or $\langle t, \varrho \rangle P$ with $t, t' \in \mathbb{T}(\Sigma)$ and $\varrho, \varrho' \in \mathcal{D}(\mathcal{H})$, called the conclusion of ϱ, and $t \xrightarrow{\alpha} t'$, called structural part of π and denoted as π_s. The left-hand side of π is called the source of ϱ. $\frac{H_s}{\pi_s}$ is called the structural part of ϱ and denoted as ϱ_s. A quantum process transition rule ϱ is closed if and only its structural part ϱ_s is closed. A quantum transition system specification (QTSS) is a (possible infinite) set of transition rules.

Definition 3.4 (Quantum bisimulation) A bisimulation relation $\mathcal{B}$ is a binary relation on quantum processes such that: (1) if $\langle p, \varrho \rangle \mathcal{B} \langle q, \sigma \rangle$ and $\langle p, \varrho \rangle \xrightarrow{\alpha} \langle p', \varrho' \rangle$ then

$\langle q, \sigma \rangle \xrightarrow{\alpha} \langle q', \sigma' \rangle$ with $\langle p', \varrho' \rangle \mathcal{B} \langle q', \sigma' \rangle$; (2) if $\langle p, \varrho \rangle \mathcal{B} \langle q, \sigma \rangle$ and $\langle q, \sigma \rangle \xrightarrow{\alpha} \langle q', \sigma' \rangle$ then $\langle p, \varrho \rangle \xrightarrow{\alpha} \langle p', \varrho' \rangle$ with $\langle p', \varrho' \rangle \mathcal{B} \langle q', \sigma' \rangle$; (3) if $\langle p, \varrho \rangle \mathcal{B} \langle q, \sigma \rangle$ and $\langle p, \varrho \rangle P$, then $\langle q, \sigma \rangle P$;(4) if $\langle p, \varrho \rangle \mathcal{B} \langle q, \sigma \rangle$ and $\langle q, \sigma \rangle P$,then $\langle p, \varrho \rangle P$. Two quantum process $\langle p, \varrho \rangle$ and $\langle q, \sigma \rangle$ are bisimilar, denoted by $\langle p, \varrho \rangle \underline{\leftrightarrow} \langle q, \sigma \rangle$, if there is a bisimulation relation $\mathcal{B}$ such that $\langle p, \varrho \rangle \mathcal{B} \langle q, \sigma \rangle$.

Definition 3.5 (Relation between quantum bisimulation and classical bisimulation) For two quantum processes, $\langle p, \varrho \rangle \underline{\leftrightarrow} \langle q, \sigma \rangle$, with $\varrho = \sigma$, if and only if $p \underline{\leftrightarrow} q$ and $\varrho' = \sigma'$,where ϱ evolves into ϱ' after execution of p and σ evolves into σ' after execution of q.

Definition 3.6 (Quantum branching bisimulation) A branching bisimulation relation $\mathcal{B}$ is a binary relation on the collection of quantum processes such that: (1) if $\langle p, \varrho \rangle \mathcal{B} \langle q, \sigma \rangle$ and $\langle p, \varrho \rangle \xrightarrow{\alpha} \langle p', \varrho' \rangle$ then either $\alpha \equiv \tau$ and $\langle p', \varrho' \rangle \mathcal{B} \langle q, \sigma \rangle$ or there is a sequence of (zero or more) τ-transitions $\langle q, \sigma \rangle \xrightarrow{\tau} \cdots \xrightarrow{\tau} \langle q_0, \sigma_0 \rangle$ such that $\langle p, \varrho \rangle \mathcal{B} \langle q_0, \sigma_0 \rangle$ and $\langle q_0, \sigma_0 \rangle \xrightarrow{\alpha} \langle q', \sigma' \rangle$ with $\langle p', \varrho' \rangle \mathcal{B} \langle q', \sigma' \rangle$; (2) if $\langle p, \varrho \rangle \mathcal{B} \langle q, \sigma \rangle$ and $\langle q, \sigma \rangle \xrightarrow{\alpha} \langle q', \sigma' \rangle$ then either $\alpha \equiv \tau$ and $\langle p, \varrho \rangle \mathcal{B} \langle q', \sigma' \rangle$ or there is a sequence of (zero or more) τ-transitions $\langle p, \varrho \rangle \xrightarrow{\tau} \cdots \xrightarrow{\tau} \langle p_0, \varrho_0 \rangle$ such that $\langle p_0, \varrho_0 \rangle \mathcal{B} \langle q, \sigma \rangle$ and $\langle p_0, \varrho_0 \rangle \xrightarrow{\alpha} \langle p', \varrho' \rangle$ with $\langle p', \varrho' \rangle \mathcal{B} \langle q', \sigma' \rangle$; (3) if $\langle p, \varrho \rangle \mathcal{B} \langle q, \sigma \rangle$ and $\langle p, \varrho \rangle P$, then there is a sequence of (zero or more) τ-transitions $\langle q, \sigma \rangle \xrightarrow{\tau} \cdots \xrightarrow{\tau} \langle q_0, \sigma_0 \rangle$ such that $\langle p, \varrho \rangle \mathcal{B} \langle q_0, \sigma_0 \rangle$ and $\langle q_0, \sigma_0 \rangle P$;(4) if $\langle p, \varrho \rangle \mathcal{B} \langle q, \sigma \rangle$ and $\langle q, \sigma \rangle P$, then there is a sequence of (zero or more) τ-transitions $\langle p, \varrho \rangle \xrightarrow{\tau} \cdots \xrightarrow{\tau} \langle p_0, \varrho_0 \rangle$ such that $\langle p_0, \varrho_0 \rangle \mathcal{B} \langle q, \sigma \rangle$ and $\langle p_0, \varrho_0 \rangle P$. Two quantum processes $\langle p, \varrho \rangle$ and $\langle q, \sigma \rangle$ are branching bisimilar, denoted by $\langle p, \varrho \rangle \underline{\leftrightarrow}_b \langle q, \sigma \rangle$, if there is a branching bisimulation relation $\mathcal{B}$ such that $\langle p, \varrho \rangle \mathcal{B} \langle q, \sigma \rangle$.

Definition 3.7 (Relation between quantum branching bisimulation and classical branching bisimulation) For two quantum processes, $\langle p, \varrho \rangle \underline{\leftrightarrow}_b \langle q, \sigma \rangle$, with $\varrho = \sigma$, if and only if $p \underline{\leftrightarrow}_b q$ and $\varrho' = \sigma'$,where ϱ evolves into ϱ' after execution of p and σ evolves into σ' after execution of q.

Definition 3.8 (Quantum rooted branching bisimulation) A rooted branching bisimulation relation $\mathcal{B}$ is a binary relation on quantum processes such that: (1) if $\langle p, \varrho \rangle \mathcal{B} \langle q, \sigma \rangle$ and $\langle p, \varrho \rangle \xrightarrow{\alpha} \langle p', \varrho' \rangle$ then $\langle q, \sigma \rangle \xrightarrow{\alpha} \langle q', \sigma' \rangle$ with $\langle p', \varrho' \rangle \underline{\leftrightarrow}_b \langle q', \sigma' \rangle$; (2) if $\langle p, \varrho \rangle \mathcal{B} \langle q, \sigma \rangle$ and $\langle q, \sigma \rangle \xrightarrow{\alpha} \langle q', \sigma' \rangle$ then $\langle p, \varrho \rangle \xrightarrow{\alpha} \langle p', \varrho' \rangle$ with $\langle p', \varrho' \rangle \underline{\leftrightarrow}_b \langle q', \sigma' \rangle$; (3) if $\langle p, \varrho \rangle \mathcal{B} \langle q, \sigma \rangle$ and $\langle p, \varrho \rangle P$,then $\langle q, \sigma \rangle P$;(4) if $\langle p, \varrho \rangle \mathcal{B} \langle q, \sigma \rangle$ and $\langle q, \sigma \rangle P$, then $\langle p, \varrho \rangle P$. Two quantum processes $\langle p, \varrho \rangle$ and $\langle q, \sigma \rangle$ are rooted branching bisimilar, denoted by $\langle p, \varrho \rangle \underline{\leftrightarrow}_{rb} \langle q, \sigma \rangle$, if there is a rooted branching bisimulation relation $\mathcal{B}$ such that $\langle p, \varrho \rangle \mathcal{B} \langle q, \sigma \rangle$.

Definition 3.9 (Relation between quantum rooted branching bisimulation and classical rooted branching bisimulation) For two quantum processes, $\langle p, \varrho \rangle \underline{\leftrightarrow}_{rb} \langle q, \sigma \rangle$, with $\varrho = \sigma$, if and only if $p \underline{\leftrightarrow}_{rb} q$ and $\varrho' = \sigma'$, where ϱ evolves into ϱ' after execution of p and σ evolves into σ' after execution of q.

Definition 3.10 (Congruence) Let Σ be a signature and $\mathcal{D}(\mathcal{H})$ be the state space of the environment. An equivalence relation $\mathcal{B}$ on $\langle t \in \mathcal{T}(\Sigma), \varrho \in \mathcal{D}(\mathcal{H}) \rangle$

is a congruence, i.e., for each $f \in \Sigma$, if $\langle s_i, \varrho_i \rangle \mathcal{B} \langle t_i, \sigma_i \rangle$ for $i \in \{1, \cdots, ar(f)\}$, then $f(\langle s_1, \varrho_1 \rangle, \cdots, \langle s_{ar(f)}, \varrho_{ar(f)} \rangle) \mathcal{B} \{(\langle t_1, \sigma_1 \rangle, \cdots, \langle t_{ar(f)}, \sigma_{ar(f)} \rangle)$. An equivalence relation $\mathcal{B}$ on $\langle t \in \mathcal{T}(\Sigma), \varrho \in \mathcal{D}(\mathcal{H}) \rangle$ is a congruence, if for each $f \in \Sigma$, $s_i \mathcal{B} t_i$ for $i \in \{1, \cdots, ar(f)\}$, and $f(s_1, \cdots, s_{ar(f)}) \mathcal{B} \{(t_1, \cdots, t_{ar(f)})$.

Definition 3.11 (Quantum conservative extension) Let T_0 and T_1 be QTSSs over signature Σ_0 and $\mathcal{D}(\mathcal{H}_0)$, and Σ_1 and $\mathcal{D}(\mathcal{H}_1)$, respectively. The QTSS $T_0 \oplus T_1$ is a conservative extension of T_0 if the LTSs generated by T_0 and $T_0 \oplus T_1$ contain exactly the same transitions $\langle t, \varrho \rangle \xrightarrow{\alpha} \langle t', \varrho' \rangle$ and $\langle t, \varrho \rangle P$ with $t \in \mathcal{T}(\Sigma_0)$ and $\varrho \in \mathcal{D}(\mathcal{H}_0)$, and $T_0 \oplus T_1 = \langle \Sigma_0 \cup \Sigma_1, \mathcal{D}(\mathcal{H}_0 \otimes \mathcal{H}_1) \rangle$.

Definition 3.12 (Relation between quantum conservative extension and classical conservative extension) The QTSS $T_0 \oplus T_1$ is a quantum conservative extension of T_0 with transitions $\langle t, \varrho \rangle \xrightarrow{\alpha} \langle t', \varrho' \rangle$ and $\langle t, \varrho \rangle P$, if its corresponding TSS $T_0' \oplus T_1'$ is a conservative extension of T_0' with transitions $t \xrightarrow{\alpha} t'$ and tP.

4 BQPA – Basic Quantum Process Algebra

In the following, the variables x, x', y, y', z, z' range over the collection of process terms, the variables υ, ω range over the set A of atomic quantum operations, $\alpha, \beta \in A, s, s', t, t'$ are closed items, τ is the special constant silent step, δ is the special constant deadlock, and the predicate $\xrightarrow{\alpha} \sqrt{}$ represents successful termination after execution of the quantum operation α.

BQPA includes three kind of operators: the execution of atomic quantum operation α, the alternative composition operator $+$ and the sequential composition operator $\cdot$. Each finite process can be represented by a closed term that is built from the set A of atomic quantum operations, the alternative composition operator $+$, and the sequential composition operator $\cdot$. The collection of all basic process terms is called Basic Quantum Process Algebra (BQPA), which is abbreviated to BQPA.

4.1 Transition Rules of BQPA

We give the transition rules under quantum transition system specification (QTSS) for BQPA as follows.

$$\frac{}{\langle \upsilon, \varrho \rangle \xrightarrow{\upsilon} \langle \sqrt{}, \upsilon(\varrho) \rangle}$$

$$\frac{\langle x, \varrho \rangle \xrightarrow{\upsilon} \langle \sqrt{}, \varrho' \rangle}{\langle x + y, \varrho \rangle \xrightarrow{\upsilon} \langle \sqrt{}, \varrho' \rangle}$$

$$\frac{\langle x, \varrho \rangle \xrightarrow{\upsilon} \langle x', \varrho' \rangle}{\langle x + y, \varrho \rangle \xrightarrow{\upsilon} \langle x', \varrho' \rangle}$$

$$\frac{\langle y,\varrho\rangle \xrightarrow{\upsilon} \langle \surd,\varrho'\rangle}{\langle x+y,\varrho\rangle \xrightarrow{\upsilon} \langle \surd,\varrho'\rangle}$$

$$\frac{\langle y,\varrho\rangle \xrightarrow{\upsilon} \langle y',\varrho'\rangle}{\langle x+y,\varrho\rangle \xrightarrow{\upsilon} \langle y',\varrho'\rangle}$$

$$\frac{\langle x,\varrho\rangle \xrightarrow{\upsilon} \langle \surd,\varrho'\rangle}{\langle x\cdot y,\varrho\rangle \xrightarrow{\upsilon} \langle y,\varrho'\rangle}$$

$$\frac{\langle x,\varrho\rangle \xrightarrow{\upsilon} \langle x',\varrho'\rangle}{\langle x\cdot y,\varrho\rangle \xrightarrow{\upsilon} \langle x'\cdot y,\varrho'\rangle}$$

where $\upsilon(\varrho)$ represents the new state of a quantum system, whose origin state is ϱ, after the execution of the atomic quantum operation υ.

– The first transition rule says that each atomic quantum operation υ can terminate successfully, and the state of the environment would be changed from ϱ to $\upsilon(\varrho)$.
– The next four transition rules say that $s+t$ can execute alternatively, that is, it can execute either s or t.
– The last two transition rules say that $s\cdot t$ can execute sequentially, that is, it executes s in the first, after successful termination of s, then execution of t follows.

4.2 Axiomatization for BQPA

We design an axiomatization $\mathcal{E}_{\text{BQPA}}$ for BQPA modulo quantum bisimulation equivalence as Table 1 shows.

Several important conclusions are following.

Theorem 4.1 *Quantum bisimulation equivalence is a congruence with respect to BQPA.*

Proof The structural part of QTSSs for BQPA are all in panth format, so bisimulation equivalence that they induce is a congruence. According to the definition of quantum bisimulation, quantum bisimulation equivalence that QTSSs for BQPA induce is also a congruence. □

Theorem 4.2 $\mathcal{E}_{\text{BQPA}}$ *is sound for BQPA modulo quantum bisimulation equivalence.*

Proof Since quantum bisimulation is both an equivalence and a congruence for BQPA, only the soundness of the first clause in the definition of the relation = is needed to be checked.

Table 1 Axioms for BQPA

No.	Axiom
QA1	$x+y=y+x$
QA2	$(x+y)+z=x+(y+z)$
QA3	$x+x=x$
QA4	$(x+y)\cdot z=x\cdot z+y\cdot z$
QA5	$(x\cdot y)\cdot z=x\cdot(y\cdot z)$

That is, if $s = t$ is an axiom in $\mathcal{E}_{\mathrm{BQPA}}$ and σ a closed substitution that maps the variable in s and t to basic quantum process terms, then we need to check that $\langle \sigma(s), \varrho \rangle \leftrightarrow \langle \sigma(t), \sigma \rangle$.

Since axioms in $\mathcal{E}_{\mathrm{BQPA}}$ (same as $\mathcal{E}_{\mathrm{BPA}}$) are sound for BPA modulo bisimulation equivalence, according to the definition of quantum bisimulation, we only need to check if $\varrho' = \sigma'$, where ϱ evolves into ϱ' after execution of $\sigma(s)$ and σ evolves into σ' after execution of $\sigma(t)$. For example, the axiom QA5 is sound for BQPA modulo quantum bisimulation equivalence, based on the associativity of quantum operations, that is, $(\sigma(s) \cdot \sigma(t)) \cdot \sigma(u)(\varrho) = \sigma(s) \cdot (\sigma(t) \cdot \sigma(u))(\sigma)$ for any $\varrho = \sigma$. □

Theorem 4.3 $\mathcal{E}_{\mathrm{BQPA}}$ *is complete for BQPA modulo quantum bisimulation equivalence.*

Proof To prove that $\mathcal{E}_{\mathrm{BQPA}}$ is complete for BQPA modulo quantum bisilumation equivalence, it means that $\langle s, \varrho \rangle \leftrightarrow \langle t, \sigma \rangle$ implies $s = t$.

It was already proved that $\mathcal{E}_{\mathrm{BQPA}}$ (same as $\mathcal{E}_{\mathrm{BPA}}$) is complete for BPA modulo bisimulation equivalence, that is, $s \leftrightarrow t$ implies $s = t$. $\langle s, \varrho \rangle \leftrightarrow \langle t, \sigma \rangle$ with $\varrho = \sigma$ means that $s \leftrightarrow t$ with $\varrho = \sigma$ and $\varrho' = \sigma'$, where ϱ evolves into ϱ' after execution of s and σ evolves into σ' after execution of t, according to the definition of quantum bisimulation equivalence. The completeness of $\mathcal{E}_{\mathrm{BQPA}}$ for BPA modulo bisimulation equivalence determines that $\mathcal{E}_{\mathrm{BQPA}}$ is complete for BQPA modulo quantum bisimulation equivalence. □

5 AQCP – Algebra of Quantum Communicating Processes

It is well known that process algebra captures parallelism and concurrency by means of the so-called interleaving pattern in contrast to the so-called true concurrency. Quantum processes can execute in parallel and communicate with each other, since the actions used to communicate are not quantum operations, which means that after the execution of communicating actions, the quantum state maintains unchanged. We introduce a new set C of atomic communicating actions. A merge operator $\parallel$ and a communication function $\gamma : C \times C \rightarrow C$ can be used to capture the parallelism and the communication.

In the following, the variables υ, ω range over the set A of atomic quantum operations, and the variable ν, μ range over the set C of atomic communicating actions.

The merge $\langle s \parallel t, \varrho \rangle$ can choose to execute an initial transition of process term s or an initial transition of process term t, and change the quantum state, which is captured by the following four transition rules.

$$\frac{\langle x, \varrho \rangle \xrightarrow{\upsilon} \langle \sqrt{}, \varrho' \rangle}{\langle x \parallel y, \varrho \rangle \xrightarrow{\upsilon} \langle y, \varrho' \rangle}$$

$$\frac{\langle x, \varrho \rangle \xrightarrow{\upsilon} \langle x', \varrho' \rangle}{\langle x \parallel y, \varrho \rangle \xrightarrow{\upsilon} \langle x' \parallel y, \varrho' \rangle}$$

$$\frac{\langle y, \varrho \rangle \xrightarrow{\upsilon} \langle \sqrt{}, \varrho' \rangle}{\langle x \parallel y, \varrho \rangle \xrightarrow{\upsilon} \langle x, \varrho' \rangle}$$

$$\frac{\langle y, \varrho \rangle \xrightarrow{\upsilon} \langle y', \varrho' \rangle}{\langle x \parallel y, \varrho \rangle \xrightarrow{\upsilon} \langle x \parallel y', \varrho' \rangle}$$

And also the merge $\langle s \parallel t, \varrho \rangle$ can choose to execute a communication of initial transitions of the process term s and t, and does not change the quantum state, which is expressed by the following four transition rules.

$$\frac{\langle x, \varrho \rangle \xrightarrow{\upsilon} \langle \checkmark, \varrho \rangle \quad \langle y, \varrho \rangle \xrightarrow{\mu} \langle \checkmark, \varrho \rangle}{\langle x \parallel y, \varrho \rangle \xrightarrow{\gamma(\upsilon, \mu)} \langle \checkmark, \varrho \rangle}$$

$$\frac{\langle x, \varrho \rangle \xrightarrow{\upsilon} \langle \checkmark, \varrho \rangle \quad \langle y, \varrho \rangle \xrightarrow{\mu} \langle y', \varrho \rangle}{\langle x \parallel y, \varrho \rangle \xrightarrow{\gamma(\upsilon, \mu)} \langle y', \varrho \rangle}$$

$$\frac{\langle x, \varrho \rangle \xrightarrow{\upsilon} \langle x', \varrho \rangle \quad \langle y, \varrho \rangle \xrightarrow{\mu} \langle \checkmark, \varrho \rangle}{\langle x \parallel y, \varrho \rangle \xrightarrow{\gamma(\upsilon, \mu)} \langle x', \varrho \rangle}$$

$$\frac{\langle x, \varrho \rangle \xrightarrow{\upsilon} \langle x', \varrho \rangle \quad \langle y, \varrho \rangle \xrightarrow{\mu} \langle y', \varrho \rangle}{\langle x \parallel y, \varrho \rangle \xrightarrow{\gamma(\upsilon, \mu)} \langle x' \parallel y', \varrho \rangle}$$

5.1 Left Merge and Communication Merge

Since there does not exist a sound and complete finite axiomatization for BPA extended with the merge, modulo bisimulation equivalence, it is can be proved that there does not exist a sound and complete axiomatization for BQPA extended with the merge modulo quantum bisimulation equivalence either. This can be overcome by defining two extra operator that are called left merge $\lfloor\!\lfloor$ and communication merge $\mid$. We call BQPA extended with the merge operator $\parallel$, the left merge operator $\lfloor\!\lfloor$ and the communication merge operator $\mid$ as Quantum Process Algebra with Parallelism, which is abbreviated to QPAP.

5.1.1 Transition Rules of QPAP

The left merge $\langle s \lfloor\!\lfloor t, \varrho \rangle$ takes its initial transition from the process term s and changes the quantum state, and then behaves as the merge $\parallel$, which is expressed by the following two transition rules.

$$\frac{\langle x, \varrho \rangle \xrightarrow{\upsilon} \langle \checkmark, \varrho' \rangle}{\langle x \lfloor\!\lfloor y, \varrho \rangle \xrightarrow{\upsilon} \langle y, \varrho' \rangle}$$

$$\frac{\langle x, \varrho \rangle \xrightarrow{\upsilon} \langle x', \varrho' \rangle}{\langle x \lfloor\!\lfloor y, \varrho \rangle \xrightarrow{\upsilon} \langle x' \parallel y, \varrho' \rangle}$$

The communication merge $\langle s \mid t, \varrho \rangle$ executes as initial transition a communication between initial transition of the process term s and t, and does not change the quantum state, and then behaves as the merge operator $\parallel$, which is captured by the following four transition rules.

$$\frac{\langle x, \varrho \rangle \xrightarrow{\upsilon} \langle \checkmark, \varrho \rangle \quad \langle y, \varrho \rangle \xrightarrow{\mu} \langle \checkmark, \varrho \rangle}{\langle x \mid y, \varrho \rangle \xrightarrow{\gamma(\upsilon, \mu)} \langle \checkmark, \varrho \rangle}$$

$$\frac{\langle x, \varrho\rangle \xrightarrow{\upsilon} \langle \checkmark, \varrho\rangle \quad \langle y, \varrho\rangle \xrightarrow{\mu} \langle y', \varrho\rangle}{\langle x \mid y, \varrho\rangle \xrightarrow{\gamma(\upsilon,\mu)} \langle y', \varrho\rangle}$$

$$\frac{\langle x, \varrho\rangle \xrightarrow{\upsilon} \langle x', \varrho\rangle \quad \langle y, \varrho\rangle \xrightarrow{\mu} \langle \checkmark, \varrho\rangle}{\langle x \mid y, \varrho\rangle \xrightarrow{\gamma(\upsilon,\mu)} \langle x', \varrho\rangle}$$

$$\frac{\langle x, \varrho\rangle \xrightarrow{\upsilon} \langle x', \varrho\rangle \quad \langle y, \varrho\rangle \xrightarrow{\mu} \langle y', \varrho\rangle}{\langle x \mid y, \varrho\rangle \xrightarrow{\gamma(\upsilon,\mu)} \langle x' \parallel y', \varrho\rangle}$$

It must be pointed out that the communication function $\gamma(\upsilon,\mu)$ of two communicating actions υ and μ is used to exchange data between two interleaving quantum processes. Due to the quantum no-cloning theorem, the data must be exchanged by references (the names of the quantum variables), but not by values.

Theorem 5.1 *QPAP is a conservative extension of BQPA.*

Proof Since the corresponding TSS of BQPA is source-dependent, and the transition rules for merge operator $\parallel$, left merge operator $\parallel\!\!\parallel$ and communication merge |contain only a fresh operator in their source, so the corresponding TSS of QPAP is a conservative extension of that of BQPA. That means that QPAP is a conservative extension of BQPA. □

Theorem 5.2 *Quantum bisimulation equivalence is a congruence with respect to QPAP.*

Proof The structural part of QTSSs for QPAP and BQPA are all in panth format, so bisimulation equivalence that they induce is a congruence. According to the definition of quantum bisimulation, quantum bisimulation equivalence that QTSSs for QPAP induce is also a congruence. □

5.1.2 Axiomatization for QPAP

We design an axiomatization for QPAP illustrated in Table 2.
    Then, we can get the soundness and completeness theorems as follows.

Theorem 5.3 $\mathcal{E}_{\mathrm{QPAP}}$ *is sound for QPAP modulo quantum bisimulation equivalence.*

Table 2 Axioms for QPAP

No.	Axiom
QM1	$x \parallel y = (x \parallel\!\!\parallel y + y \parallel\!\!\parallel x) + x \mid y$
QLM2	$\upsilon \parallel\!\!\parallel y = \upsilon \cdot y$
QLM3	$(\upsilon \cdot x) \parallel\!\!\parallel y = \upsilon \cdot (x \parallel y)$
QLM4	$(x + y) \parallel\!\!\parallel z = x \parallel\!\!\parallel z + y \parallel\!\!\parallel z$
QCM5	$\upsilon \mid \mu = \gamma(\upsilon,\mu)$
QCM6	$\upsilon \mid (\mu \cdot y) = \gamma(\upsilon,\mu) \cdot y$
QCM7	$(\upsilon \cdot x) \mid \mu = \gamma(\upsilon,\mu) \cdot x$
QCM8	$(\upsilon \cdot x) \mid (\mu \cdot y) = \gamma(\upsilon,\mu) \cdot (x \parallel y)$
QCM9	$(x + y) \mid z = x \mid z + y \mid z$
QCM10	$x \mid (y + z) = x \mid y + x \mid z$

Proof Since quantum bisimulation is both an equivalence and a congruence for QPAP, only the soundness of the first clause in the definition of the relation = is needed to be checked. That is, if $s = t$ is an axiom in $\mathcal{E}_{\mathrm{QPAP}}$ and σ a closed substitution that maps the variable in s and t to basic quantum process terms, then we need to check that $\langle \sigma(s), \varrho \rangle \leftrightarrow \langle \sigma(t), \sigma \rangle$.

Since axioms in $\mathcal{E}_{\mathrm{QPAP}}$ (same as $\mathcal{E}_{\mathrm{PAP}}$) are sound for PAP modulo bisimulation equivalence, according to the definition of quantum bisimulation, we only need to check if $\varrho' = \sigma'$ when $\varrho = \sigma$, where ϱ evolves into ϱ' after execution of $\sigma(s)$ and σ evolves into σ' after execution of $\sigma(t)$. We can find that every axiom in Table 2 meets the above condition. □

Theorem 5.4 *$\mathcal{E}_{\mathrm{QPAP}}$ is complete for QPAP modulo quantum bisimulation equivalence.*

Proof To prove that $\mathcal{E}_{\mathrm{QPAP}}$ is complete for QPAP modulo quantum bisilumation equivalence, it means that $\langle s, \varrho \rangle \leftrightarrow \langle t, \sigma \rangle$ implies $s = t$.

It was already proved that $\mathcal{E}_{\mathrm{QPAP}}$ (same as $\mathcal{E}_{\mathrm{PAP}}$) is complete for PAP modulo bisimulation equivalence, that is, $s \leftrightarrow t$ implies $s = t$. $\langle s, \varrho \rangle \leftrightarrow \langle t, \sigma \rangle$ with $\varrho = \sigma$ means that $s \leftrightarrow t$ with $\varrho = \sigma$ and $\varrho' = \sigma'$, where ϱ evolves into ϱ' after execution of s and σ evolves into σ' after execution of t, according to the definition of quantum bisimulation equivalence. The completeness of $\mathcal{E}_{\mathrm{QPAP}}$ for PAP modulo bisimulation equivalence determines that $\mathcal{E}_{\mathrm{QPAP}}$ is complete for QPAP modulo quantum bisimulation equivalence. □

5.2 Deadlock and Encapsulation

The mismatch of two communicating action pair v and μ can cause a deadlock (nothing to do), we introduce the deadlock constant δ and extend the communication function γ to $\gamma : C \times C \rightarrow C \cup \{\delta\}$. So, the introduction about communication merge | in the above section should be with $\gamma(v, \mu) \neq \delta$. We also introduce a unary encapsulation operator ∂_H for sets H of atomic communicating actions, which rename all actions in H into δ. QPAP extended with deadlock constant δ and encapsulation operator ∂_H is called the Algebra of Quantum Communicating Processes, which is abbreviated to AQCP.

5.2.1 Transition Rules of AQCP

The encapsulation operator $\partial_H(t)$ can execute all transitions of process term t of which the labels are not in H, and does not change the quantum state, which is expressed by the following two transition rules.

$$\frac{\langle x, \varrho \rangle \xrightarrow{v} \langle \surd, \varrho \rangle}{\langle \partial_H(x), \varrho \rangle \xrightarrow{v} \langle \surd, \varrho \rangle} \quad v \notin H$$

$$\frac{\langle x, \varrho \rangle \xrightarrow{v} \langle x', \varrho \rangle}{\langle \partial_H(x), \varrho \rangle \xrightarrow{v} \langle \partial_H(x'), \varrho \rangle} \quad v \notin H$$

Theorem 5.5 *AQCP is a conservative extension of QPAP.*

Proof Since the corresponding TSS of QPAP is source-dependent, and the transition rules for encapsulation operator ∂_H contain only a fresh operator in their source, so the

corresponding TSS of AQCP is a conservative extension of that of QPAP. That means that AQCP is a conservative extension of QPAP.   □

Theorem 5.6 *Quantum bisimulation equivalence is a congruence with respect to AQCP.*

Proof The structural part of QTSSs for AQCP and QPAP are all in panth format, so bisimulation equivalence that they induce is a congruence. According to the definition of quantum bisimulation, quantum bisimulation equivalence that QTSSs for AQCP induce is also a congruence.   □

5.2.2 Axiomatization for AQCP

The axioms for AQCP are shown in Table 3.

The soundness and completeness theorems are following.

Theorem 5.7 $\mathcal{E}_{\text{AQCP}}$ *is sound for AQCP modulo quantum bisimulation equivalence.*

Proof Since quantum bisimulation is both an equivalence and a congruence for AQCP, only the soundness of the first clause in the definition of the relation = is needed to be checked. That is, if $s = t$ is an axiom in $\mathcal{E}_{\text{AQCP}}$ and σ a closed substitution that maps the variable in s and t to basic quantum process terms, then we need to check that $\langle \sigma(s), \varrho \rangle \underline{\leftrightarrow} \langle \sigma(t), \sigma \rangle$.

Since axioms in $\mathcal{E}_{\text{AQCP}}$ (same as $\mathcal{E}_{\text{ACP}}$) are sound for ACP modulo bisimulation equivalence, according to the definition of quantum bisimulation, we only need to check if $\varrho' = \sigma'$ when $\varrho = \sigma$, where ϱ evolves into ϱ' after execution of $\sigma(s)$ and σ evolves into σ' after execution of $\sigma(t)$. We can find that every axiom in Table 3 meets the above condition.   □

Theorem 5.8 $\mathcal{E}_{\text{AQCP}}$ *is complete for AQCP modulo quantum bisimulation equivalence.*

Proof To prove that $\mathcal{E}_{\text{AQCP}}$ is complete for AQCP modulo quantum bisilumation equivalence, it means that $\langle s, \varrho \rangle \underline{\leftrightarrow} \langle t, \sigma \rangle$ implies $s = t$.

It was already proved that $\mathcal{E}_{\text{AQCP}}$ (same as $\mathcal{E}_{\text{ACP}}$) is complete for ACP modulo bisimulation equivalence, that is, $s \underline{\leftrightarrow} t$ implies $s = t$. $\langle s, \varrho \rangle \underline{\leftrightarrow} \langle t, \sigma \rangle$ with $\varrho = \sigma$ means that $s \underline{\leftrightarrow} t$ with $\varrho = \sigma$ and $\varrho' = \sigma'$, where ϱ evolves into ϱ' after execution of s and σ evolves into σ' after execution of t, according to the definition of quantum bisimulation equivalence. The

Table 3 Axioms for AQCP

No.	Axiom
QA6	$x + \delta = x$
QA7	$\delta \cdot x = \delta$
QD1	$v \notin H \ \partial_H(v) = v$
QD2	$v \in H \ \partial_H(v) = \delta$
QD3	$\partial_H(\delta) = \delta$
QD4	$\partial_H(x + y) = \partial_H(x) + \partial_H(y)$
QD5	$\partial_H(x \cdot y) = \partial_H(x) \cdot \partial_H(y)$
QLM11	$\delta \| x = \delta$
QCM12	$\delta | x = \delta$
QCM13	$x | \delta = \delta$

completeness of $\mathcal{E}_{AQCP}$ for ACP modulo bisimulation equivalence determines that $\mathcal{E}_{AQCP}$ is complete for AQCP modulo quantum bisimulation equivalence. □

6 Recursion

To capture infinite computing, recursion is introduced in this section. In the following, E,F,G are guarded linear recursion specifications, X,Y,Z are recursive variables. We first introduce several important concepts, which come from [5].

Definition 6.1 (Recursive specification) A recursive specification is a finite set of recursive equations

$$X_1 = t_1(X_1, \cdots , X_n)$$

$$...$$

$$X_n = t_n(X_1, \cdots , X_n)$$

where the left-hand sides of X_i are called recursion variables, and the right-hand sides $t_i(X_1, \cdots , X_n)$ are process terms in AQCP with possible occurrences of the recursion variables $X_1, \cdots , X_n$.

Definition 6.2 (Solution) Processes $p_1, \cdots , p_n$ are a solution for a recursive specification $\{X_i = t_i(X_1, \cdots , X_n) | i \in \{1, \cdots , n\}\}$ (with respect to bisimulation equivalence) if $p_i \underline{\leftrightarrow} t_i(p_1, \cdots , p_n)$ for $i \in \{1, \cdots , n\}$.

Definition 6.3 (Guarded recursive specification) A recursive specification

$$X_1 = t_1(X_1, \cdots , X_n)$$

$$...$$

$$X_n = t_n(X_1, \cdots , X_n)$$

is guarded if the right-hand sides of its recursive equations can be adapted to the form by applications of the axioms in $\mathcal{E}_{AQCP}$ and replacing recursion variables by the right-hand sides of their recursive equations,

$$\alpha_1 \cdot s_1(X_1, \cdots , X_n) + \cdots + \alpha_k \cdot s_k(X_1, \cdots , X_n) + \beta_1 + \cdots + \beta_l$$

where $\alpha_1, \cdots , \alpha_k, \beta_1, \cdots , \beta_l \in A \cup C$, and the sum above is allowed to be empty, in which case it represents the deadlock δ.

Definition 6.4 (Linear recursive specification) A recursive specification is linear if its recursive equations are of the form

$$\alpha_1 X_1 + \cdots + \alpha_k X_k + \beta_1 + \cdots + \beta_l$$

where $\alpha_1, \cdots , \alpha_k, \beta_1, \cdots , \beta_l \in A \cup C$, and the sum above is allowed to be empty, in which case it represents the deadlock δ.

6.1 Transition Rules of Guarded Recursion

For a guarded recursive specifications E with the form

$$X_1 = t_1(X_1, \cdots, X_n)$$

$$\cdots$$

$$X_n = t_n(X_1, \cdots, X_n)$$

the behavior of the solution $\langle X_i | E \rangle$ forthe recursion variable X_i in E, where $i \in \{1, \cdots, n\}$, isexactly the behavior of their right-hand sides $t_i(X_1, \cdots, X_n)$, which is captured by the following two transition rules.

$$\frac{\langle t_i(\langle X_1 | E \rangle, \cdots, \langle X_n | E \rangle), \varrho \rangle \xrightarrow{\upsilon} \langle \sqrt{}, \varrho' \rangle}{\langle \langle X_i | E \rangle, \varrho \rangle \xrightarrow{\upsilon} \langle \sqrt{}, \varrho' \rangle}$$

$$\frac{\langle t_i(\langle X_1 | E \rangle, \cdots, \langle X_n | E \rangle), \varrho \rangle \xrightarrow{\upsilon} \langle y, \varrho' \rangle}{\langle \langle X_i | E \rangle, \varrho \rangle \xrightarrow{\upsilon} \langle y, \varrho' \rangle}$$

Theorem 6.1 *AQCP with guarded recursion is a conservative extension of AQCP.*

Proof Since the corresponding TSS of AQCP is source-dependent, and the transition rules for guarded recursion contain only a fresh constant in their source, so the corresponding TSS of AQCP with guarded recursion is a conservative extension of that of AQCP. That means that AQCP with guarded recursion is a conservative extension of AQCP. □

Theorem 6.2 *Quantum bisimulation equivalence is a congruence with respect to AQCP with guarded recursion.*

Proof The structural part of QTSSs for guarded recursion and AQCP are all in panth format, so bisimulation equivalence that they induce is a congruence. According to the definition of quantum bisimulation, quantum bisimulation equivalence that QTSSs for AQCP with guarded recursion induce is also a congruence. □

6.2 Axiomatization for Guarded Recursion

The RDP (Recursive Definition Principle) and the RSP (Recursive Specification Principle) are shown in Table 4.

Theorem 6.3 $\mathcal{E}_{\text{AQCP}}$ + *RDP* + *RSP is sound for AQCP with guarded recursion modulo quantum bisimulation equivalence.*

Table 4 Recursive definition principle and recursive specification principle

No.	Axiom
RDP	$\langle X_i | E \rangle = t_i(\langle X_1 | E, \cdots, X_n | E \rangle)$ $(i \in \{1, \cdots, n\})$
RSP	if $y_i = t_i(y_1, \cdots, y_n)$ for $i \in \{1, \cdots, n\}$, then $y_i = \langle X_i | E \rangle$ $(i \in \{1, \cdots, n\})$

Proof　Since quantum bisimulation is both an equivalence and a congruence for AQCP with guarded recursion, only the soundness of the first clause in the definition of the relation = is needed to be checked. That is, if $s = t$ is an axiom in $\mathcal{E}_{AQCP}$ + RDP + RSP and σ a closed substitution that maps the variable in s and t to basic quantum process terms, then we need to check that $\langle \sigma(s), \varrho \rangle \underleftrightarrow{} \langle \sigma(t), \sigma \rangle$.

Since axioms in $\mathcal{E}_{AQCP}$ + RDP + RSP (same as $\mathcal{E}_{ACP}$ + RDP + RSP) are sound for ACP with guarded recursion modulo bisimulation equivalence, according to the definition of quantum bisimulation, we only need to check if $\varrho' = \sigma'$ when $\varrho = \sigma$, where ϱ evolves into ϱ' after execution of $\sigma(s)$ and σ evolves into σ' after execution of $\sigma(t)$. We can find that every axiom in Table 4 meets the above condition.　□

Theorem 6.4　*$\mathcal{E}_{AQCP}$ + RDP + RSP is complete for AQCP with linear recursion modulo quantum bisimulation equivalence.*

Proof　To prove that $\mathcal{E}_{AQCP}$ + RDP + RSP is complete for AQCP with linear recursion modulo quantum bisilumation equivalence, it means that $\langle s, \varrho \rangle \underleftrightarrow{} \langle t, \sigma \rangle$ implies $s = t$.

It was already proved that $\mathcal{E}_{AQCP}$ + RDP + RSP (same as $\mathcal{E}_{ACP}$ + RDP +RSP) is complete for ACP with linear recursion modulo bisimulation equivalence, that is, $s \underleftrightarrow{} t$ implies $s = t$. $\langle s, \varrho \rangle \underleftrightarrow{} \langle t, \sigma \rangle$ with $\varrho = \sigma$ means that $s \underleftrightarrow{} t$ with $\varrho = \sigma$ and $\varrho' = \sigma'$, where ϱ evolves into ϱ' after execution of s and σ evolves into σ' after execution of t, according to the definition of quantum bisimulation equivalence. The completeness of $\mathcal{E}_{AQCP}$ + RDP + RSP for ACP with linear recursion modulo bisimulation equivalence determines that $\mathcal{E}_{AQCP}$ + RDP + RSP is complete for AQCP with linear recursion modulo quantum bisimulation equivalence.　□

7 Abstraction

A quantum program has internal implementations and external behaviors. Abstraction technology abstracts away from the internal steps to check if the internal implementations really display the desired external behaviors. This makes the introduction of special silent step constant τ and the abstraction operator τ_I.

Firstly, we introduce the concept of guarded linear recursive specification, which comes from [5].

Definition 7.1 (Guarded linear recursive specification)　A recursive specification is linear if its recursive equations are of the form

$$\alpha_1 X_1 + \cdots + \alpha_k X_k + \beta_1 + \cdots + \beta_l$$

where $\alpha_1, \cdots, \alpha_k, \beta_1, \cdots, \beta_l \in A \cup C \cup \{\tau\}$.

A linear recursive specification E is guarded if there does not exist an infinite sequence of τ-transitions $\langle X|E \rangle \xrightarrow{\tau} \langle X'|E \rangle \xrightarrow{\tau} \langle X''|E \rangle \xrightarrow{\tau} \cdots$.

7.1 Silent Step

A τ-transition is silent, which is means that it can be eliminated from a quantum process graph. τ is an internal step and keep silent from an external observer, but please remember,

τ is a quantum operation in nature. This fact makes that τ must influence the state of all quantum variables ϱ, that is, τ is not really silent for a quantum process configuration $\langle p, \varrho \rangle$. To make τ keep silent, the definition of ϱ must be changed, that is, ϱ does not record the state of all quantum variables, some variables must be moved away. But, what variables should be moved away? The quantum variables that τ may influence are called private variables. These private variables include not only the variables τ directly manipulates, but also those variables which are entangled with the variables that τ directly manipulates. The quantum variables that τ can not influence are called public variables. In the following, ϱ records the state of all public variables. We use the symbol $\tau(\varrho)$ to denote the state of all public quantum variables after execution of τ. From an external view, we can see that $\varrho = \tau(\varrho)$.

The processing of τ in quantum processes is some what farfetched. But, it is the only choice under the framework of quantum process configuration $\langle p, \varrho \rangle$. Otherwise, the concept of branching bisimulation (weak bisimilarity) and the theory of abstraction can not be established.

Now, the set A of all quantum operations is extended to $A \cup \{\tau\}$, C to $C \cup \{\tau\}$, and γ to γ : $C \cup \{\tau\} \times C \cup \{\tau\} \rightarrow C \cup \{\delta\}$.

7.1.1 Transition Rules of Silent Step

τ keeps silent from an external observer, which is expressed by the following transition rules.

$$\frac{}{\langle \tau, \varrho \rangle \xrightarrow{\tau} \langle \sqrt{}, \tau(\varrho) \rangle}$$

Transition rules for alternative composition, sequential composition and guarded linear recursion that involves τ-transitions are omitted.

Theorem 7.1 *AQCP with silent step and guarded linear recursion is a conservative extension of AQCP with guarded linear recursion.*

Proof The corresponding TSS of AQCP with silent step and guarded linear recursion is a conservative extension of that of AQCP with guarded linear recursion. That means that AQCP with silent step and guarded linear recursion is a conservative extension of AQCP with guarded linear recursion. □

Theorem 7.2 *Quantum rooted branching bisimulation equivalence is a congruence with respect to AQCP with silent step and guarded linear recursion.*

Proof The structural part of QTSSs for AQCP with silent step and guarded linear recursion are all in RBB cool format by incorporating the successful termination predicate $\downarrow$ in the transition rules, so rooted branching bisimulation equivalence that they induce is a congruence. According to the definition of quantum rooted branching bisimulation, quantum rooted branching bisimulation equivalence that QTSSs for AQCP with silent step and guarded linear recursion induce is also a congruence. □

7.1.2 Axioms for Silent Step

The axioms for silent step is shown in Table 5.

Theorem 7.3　*$\mathcal{E}_{AQCP}$ + QB1,QB2 + RDP + RSP is sound for AQCP with silent step and guarded linear recursion, modulo quantum rooted branching bisimulation equivalence.*

Proof　Since quantum rooted branching bisimulation is both an equivalence and a congruence for AQCP with silent step and guarded linear recursion, only the soundness of the first clause in the definition of the relation = is needed to be checked. That is, if $s = t$ is an axiom in $\mathcal{E}_{AQCP}$+ QB1,QB2 + RDP + RSP and σ a closed substitution that maps the variable in s and t to basic quantum process terms, then we need to check that $\langle \sigma(s), \varrho \rangle \underline{\leftrightarrow}_{rb} \langle \sigma(t), \sigma \rangle$.

Since axioms in $\mathcal{E}_{AQCP}$ + QB1,QB2 + RDP + RSP (same as $\mathcal{E}_{ACP}$ + QB1,QB2 + RDP + RSP) are sound for ACP with silent step and guarded linear recursion modulo rooted branching bisimulation equivalence, according to the definition of quantum rooted branching bisimulation, we only need to check if $\varrho' = \sigma'$ when $\varrho = \sigma$, where ϱevolves into ϱ' after execution of $\sigma(s)$ and σ evolves into σ' after execution of $\sigma(t)$. We can find that every axiom in Table 5 meets the above condition.　□

Theorem 7.4　*$\mathcal{E}_{AQCP}$ + QB1,QB2 + RDP + RSP is complete for AQCP with silent step and guarded linear recursion, modulo quantum rooted branching bisimulation equivalence.*

Proof　To prove that $\mathcal{E}_{AQCP}$ + QB1,QB2 + RDP + RSP is complete for AQCP with silent step and guarded linear recursion modulo quantum rooted branching bisilumation equivalence, it means that $\langle s, \varrho \rangle \underline{\leftrightarrow}_{rb} \langle t, \sigma \rangle$ implies $s = t$.

It was already proved that $\mathcal{E}_{AQCP}$ + QB1,QB2 + RDP + RSP (same as $\mathcal{E}_{ACP}$ + QB1,QB2 + RDP + RSP) is complete for ACP with silent step and guarded linear recursion modulo rooted branching bisimulation equivalence, that is, $s \underline{\leftrightarrow}_{rb} t$ implies $s = t$. $\langle s, \varrho \rangle \underline{\leftrightarrow}_{rb} \langle t, \sigma \rangle$ with $\varrho = \sigma$ means that $s \underline{\leftrightarrow}_{rb} t$ with $\varrho = \sigma$ and $\varrho' = \sigma'$, where ϱ evolves into ϱ' after execution of s and σ evolves into σ' after execution of t, according to the definition of quantum rooted branching bisimulation equivalence. The completeness of $\mathcal{E}_{AQCP}$ + QB1,QB2 + RDP + RSP for ACP with silent step and guarded linear recursion modulo rooted branching bisimulation equivalence determines that $\mathcal{E}_{AQCP}$ + QB1,QB2 + RDP + RSP is complete for AQCP with silent step and guarded linear recursion modulo quantum rooted branching bisimulation equivalence.　□

7.2 Abstraction

Abstraction operator τ_I is used to abstract away the internal implementations. AQCP extended with silent step τ and abstraction operator τ_I is denoted by AQCP$_\tau$.

Table 5　Axioms for silent step

No.	Axiom
QB1	$\upsilon \cdot \tau = \upsilon$
QB2	$\upsilon \cdot (\tau \cdot (x + y) + x) = \upsilon \cdot (x + y)$

7.2.1 Transition Rules of Abstraction Operator

Abstraction operator $\tau_I(t)$ renames all labels of transitions of t that are in the set I into τ, and does not change the state of all public quantum variables, which is captured by the following four transition rules.

$$\frac{\langle x, \varrho \rangle \xrightarrow{\upsilon} \langle \sqrt{}, \varrho' \rangle}{\langle \tau_I(x), \varrho \rangle \xrightarrow{\upsilon} \langle \sqrt{}, \varrho' \rangle} \quad \upsilon \notin I$$

$$\frac{\langle x, \varrho \rangle \xrightarrow{\upsilon} \langle x', \varrho' \rangle}{\langle \tau_I(x), \varrho \rangle \xrightarrow{\upsilon} \langle \tau_I(x'), \varrho' \rangle} \quad \upsilon \notin I$$

$$\frac{\langle x, \varrho \rangle \xrightarrow{\upsilon} \langle \sqrt{}, \varrho' \rangle}{\langle \tau_I(x), \varrho \rangle \xrightarrow{\tau} \langle \sqrt{}, \tau(\varrho) \rangle} \quad \upsilon \in I$$

$$\frac{\langle x, \varrho \rangle \xrightarrow{\upsilon} \langle x', \varrho' \rangle}{\langle \tau_I(x), \varrho \rangle \xrightarrow{\tau} \langle \tau_I(x'), \tau(\varrho) \rangle} \quad \upsilon \in I$$

Note that $\varrho = \tau(\varrho) = \tau(\varrho')$ in the sense of public variables.

Theorem 7.5 $AQCP_\tau$ *with guarded linear recursion is a conservative extension of AQCP with silent step and guarded linear recursion.*

Proof The corresponding TSS of $AQCP_\tau$ with guarded linear recursion is a conservative extension of that of AQCP with silent step and guarded linear recursion. That means that $AQCP_\tau$ with guarded linear recursion is a conservative extension of AQCP with silent step and guarded linear recursion. □

Theorem 7.6 *Quantum rooted branching bisimulation equivalence is a congruence with respect to $AQCP_\tau$ with guarded linear recursion.*

Proof The structural part of QTSSs for $AQCP_\tau$ with guarded linear recursion are all in RBB cool format by incorporating the successful termination predicate ↓ in the transition rules, so rooted branching bisimulation equivalence that they induce is a congruence. According to the definition of quantum rooted branching bisimulation, quantum rooted branching bisimulation equivalence that QTSSs for $AQCP_\tau$ with guarded linear recursion induce is also a congruence. □

7.2.2 Axiomatization for Abstraction Operator

The axioms for abstraction operator are shown in Table 6.

Before we introduce the cluster fair abstraction rule, the concept of cluster is given firstly, which comes from [5].

Definition 7.2 (Cluster) Let E be a guarded linear recursive specification, and $I \subseteq A$. Two recursion variable X and Y in E are in the same cluster for I if and only if there exist

Table 6 Axioms for abstraction operator

No.	Axiom
QTI1	$\upsilon \notin I \ \tau_I(\upsilon) = \upsilon$
QTI2	$\upsilon \in I \ \tau_I(\upsilon) = \tau$
QTI3	$\tau_I(\delta) = \delta$
QTI4	$\tau_I(x + y) = \tau_I(x) + \tau_I(y)$
QTI5	$\tau_I(x \cdot y) = \tau_I(x) \cdot \tau_I(y)$

sequences of transitions $\langle X|E \rangle \xrightarrow{\beta_1} \cdots \xrightarrow{\beta_m} \langle Y|E \rangle$ and $\langle Y|E \rangle \xrightarrow{\eta_1} \cdots \xrightarrow{\eta_n} \langle X|E \rangle$, where $\beta_1, \cdots$ $,\beta_m, \eta_1, \cdots, \eta_n \in I \cup \{\tau\}$.

α or αX is an exit for the cluster C if and only if: (1) α or αX is a summand at the right-hand side of the recursive equation for a recursion variable in C, and (2) in the case of αX, either $\alpha \notin I \cup \{\tau\}$ or $X \notin C$.

Theorem 7.7 $\mathcal{E}_{AQCP_\tau} + RSP + RDP + CFAR$ *is sound for* $AQCP_\tau$ *with guarded linear recursion, modulo quantum rooted branching bisimulation equivalence.*

Proof Since quantum rooted branching bisimulation is both an equivalence and a congruence for $AQCP_\tau$ with guarded linear recursion, only the soundness of the first clause in the definition of the relation = is needed to be checked. That is, if $s = t$ is an axiom in $\mathcal{E}_{AQCP_\tau}$ + RSP + RDP + CFAR and σ a closed substitution that maps the variable in s and t to basic quantum process terms, then we need to check that $\langle \sigma(s), \varrho \rangle \underset{rb}{\leftrightarrow} \langle \sigma(t), \sigma \rangle$.

Since axioms in $\mathcal{E}_{AQCP_\tau}$ + RSP + RDP + CFAR (same as $\mathcal{E}_{ACP_\tau}$ + RSP + RDP + CFAR) are sound for ACP_τ with guarded linear recursion modulo rooted branching bisimulation equivalence, according to the definition of quantum rooted branching bisimulation, we only need to check if $\varrho' = \sigma'$ when $\varrho = \sigma$, where ϱ evolves into ϱ' after execution of $\sigma(s)$ and σ evolves into σ' after execution of $\sigma(t)$. We can find that every axiom in Tables 6 and 7 meets the above condition. $\square$

Theorem 7.8 $\mathcal{E}_{AQCP_\tau} + RSP + RDP + CFAR$ *is complete for* $AQCP_\tau$ *with guarded linear recursion, modulo quantum rooted branching bisimulation equivalence.*

Proof To prove that $\mathcal{E}_{AQCP_\tau}$ + RSP + RDP + CFAR is complete for $AQCP_\tau$ with guarded linear recursion modulo quantum rooted branching bisilumation equivalence, it means that $\langle s, \varrho \rangle \underset{rb}{\leftrightarrow} \langle t, \sigma \rangle$ implies $s = t$.

It was already proved that $\mathcal{E}_{AQCP_\tau}$ + RSP + RDP + CFAR (same as $\mathcal{E}_{ACP_\tau}$ + RSP + RDP + CFAR) is complete for ACP_τ with guarded linear recursion modulo rooted branching bisimulation equivalence, that is, $s \underset{rb}{\leftrightarrow} t$ implies $s = t$. $\langle s, \varrho \rangle \underset{rb}{\leftrightarrow} \langle t, \sigma \rangle$ with $\varrho = \sigma$ means that $s \underset{rb}{\leftrightarrow} t$ with $\varrho = \sigma$ and $\varrho' = \sigma'$, where ϱ evolves into ϱ' after execution of s and σ evolves into σ' after execution of t, according to the definition of quantum rooted branching bisimulation equivalence. The completeness of $\mathcal{E}_{AQCP_\tau}$ + RSP + RDP + CFAR for ACP_τ with guarded linear recursion modulo rooted branching bisimulation equivalence determines that

Table 7 Cluster fair abstraction rule

No.	Axiom
CFAR	If X is in a cluster for I with exits $\{\upsilon_1 Y_1, \cdots, \upsilon_m Y_m, \omega_1, \cdots, \omega_n\}$, then $\tau \cdot \tau_I(\langle X|E \rangle) = \tau \cdot \tau_I(\upsilon_1 \langle Y_1|E \rangle, \cdots, \upsilon_m \langle Y_m|E \rangle, \omega_1, \cdots, \omega_n)$

$\mathcal{E}_{\text{AQCP}_\tau} + \text{RSP} + \text{RDP} + \text{CFAR}$ is complete for AQCP_τ with guarded linear recursion modulo quantum rooted branching bisimulation equivalence.   □

8 Unifying Quantum and Classical Computing

We use a quantum process configurations $\langle p, \varrho \rangle$ to represent information related to the execution of a quantum process, in which p represents the structural properties of a quantum process and ϱ expresses the quantum properties of a quantum process. We have established a whole theory about quantum processes based on ACP, which is called qACP.

In qACP, the set A of actions is consisted of atomic quantum operations, and also the deadlock δ and the silent step τ. The execution of an atomic quantum operation α not only influences of the structural part p, but also changes the state of quantum variables ϱ. We still use the framework of a quantum process configuration p, ϱ under the situation of classical computing. In classical computing, the execution of a (classical) atomic action a only influence the structural part p, and maintain the quantum state ϱ unchanged. Note that, this kind of actions are already introduced in AQCP in Section 5, which are called quantum communicating actions, and range over the set C of quantum communicating actions. In nature, quantum communicating actions are some kind of classical actions in contrast to quantum operations, because they are unrelated to the quantum state ϱ. The difference of a quantum communicating action and a classical communicating action is that they exchange different contents, a classical communicating action exchange the classical data by value or by reference, while a quantum communicating action exchange the quantum variables only by reference. We extend the set C of quantum communicating actions to classical atomic actions (including classical communicating actions), and variables ν, μ range over C, and $a, b \in C$.

Base on the fact that a classical action a does not affect the quantum state ϱ, we can generalize classical ACP under the framework of quantum process configuration $\langle p, \varrho \rangle$. We only take an example of BPA, while PAP, ACP, ACP with guarded linear recursion, ACP_τ with guarded linear recursion are omitted.

We give the transition rules under quantum transition system specification (QTSS) for BPA as follows.

$$\frac{}{\langle \upsilon, \varrho \rangle \xrightarrow{\upsilon} \langle \surd, \varrho \rangle}$$

$$\frac{\langle x, \varrho \rangle \xrightarrow{\upsilon} \langle \surd, \varrho \rangle}{\langle x + y, \varrho \rangle \xrightarrow{\upsilon} \langle \surd, \varrho \rangle}$$

$$\frac{\langle x, \varrho \rangle \xrightarrow{\upsilon} \langle x', \varrho \rangle}{\langle x + y, \varrho \rangle \xrightarrow{\upsilon} \langle x', \varrho \rangle}$$

$$\frac{\langle y, \varrho \rangle \xrightarrow{\upsilon} \langle \surd, \varrho \rangle}{\langle x + y, \varrho \rangle \xrightarrow{\upsilon} \langle \surd, \varrho \rangle}$$

$$\frac{\langle y, \varrho \rangle \overset{v}{\to} \langle y', \varrho \rangle}{\langle x + y, \varrho \rangle \overset{v}{\to} \langle y', \varrho \rangle}$$

$$\frac{\langle x, \varrho \rangle \overset{v}{\to} \langle \surd, \varrho \rangle}{\langle x \cdot y, \varrho \rangle \overset{v}{\to} \langle y, \varrho \rangle}$$

$$\frac{\langle x, \varrho \rangle \overset{v}{\to} \langle x', \varrho \rangle}{\langle x \cdot y, \varrho \rangle \overset{v}{\to} \langle x' \cdot y, \varrho \rangle}$$

We design an axiomatization $\mathcal{E}_{BPA}$ for BPA modulo quantum bisimulation equivalence as Table 8 shows.

We can get the following conclusions naturally.

Theorem 8.1 *Quantum bisimulation equivalence is a congruence with respect to BPA.*

Theorem 8.2 $\mathcal{E}_{BPA}$ *is sound for BPA modulo quantum bisimulation equivalence.*

Theorem 8.3 $\mathcal{E}_{BPA}$ *is complete for BPA modulo quantum bisimulation equivalence.*

Note that, the behavior of deadlock constant δ quantum computing is the same as that of classical computing. But, the behavior of silent step τ is different under the framework of quantum process configuration $\langle p, \varrho \rangle$ for quantum computing and classical computing, just because τ in quantum computing can affect the state of all quantum variables, while τ in classical computing really keeps silent.

Making classical ACP (including BPA, PAP, ACP, ACP with guarded linear recursion, and ACP$_\tau$ with guarded linear recursion) being under the framework of quantum process configuration $\langle p, \varrho \rangle$ for classical computing is trivial, because ϱ is meaningless only for classical computing. But, in the view of unifying quantum computing and classical computing, this work would be very important. Fortunately, qACP and classical ACP are unified under the framework of quantum process configuration $\langle p, \varrho \rangle$, that is, qACP and classical ACP have the same equational logic (axiomatization $\mathcal{E}_{qACP}$ and $\mathcal{E}_{ACP}$) and the same semantic model (strong quantum bisimilarity and weak quantum bisimilarity).

The unifying of qACP and classical ACP has an important significance, because most quantum protocols, like the famous BB84 protocol [18], are mixtures of quantum information and classical information, and those of quantum computing and classical computing. This unifying can be used widely in verification for all quantum protocols.

Table 8 Axioms for BPA

No.	Axiom
A1	$x + y = y + x$
A2	$(x + y) + z = x + (y + z)$
A3	$x + x = x$
A4	$(x + y) \cdot z = x \cdot z + y \cdot z$
A5	$(x \cdot y) \cdot z = x \cdot (y \cdot z)$

9 Verification for Quantum Protocols – the BB84 Protocol

The unifying of qACP and classical ACP under the framework of quantum process configuration $\langle p, \varrho \rangle$ makes verification for quantum protocols possible, not only the pure quantum protocol, but also protocol that mixes quantum information and classical information.

The famous BB84 protocol [18] is the first quantum key distribution protocol, in which quantum information and classical information are mixed. We take an example of the BB84 protocol to illustrate the usage of qACP in verification of quantum protocols.

The BB84 protocol is used to create a private key between two parities, Alice and Bob. Firstly, we introduce the basic BB84 protocol briefly, which is illustrated in Fig. 1.

1. Alice create two string of bits with size n randomly, denoted as B_a and K_a.
2. Alice generates a string of qubits q with size n, and the i th qubit in q is $|x_y\rangle$, where x is the i th bit of B_a and y is the i th bit of K_a.
3. Alice sends q to Bob through a quantum channel Q between Alice and Bob.
4. Bob receives q and randomly generates a string of bits B_b with size n.
5. Bob measures each qubit of q according to a basis by bits of B_b. And the measurement results would be K_b, which is also with size n.
6. Bob sends his measurement bases B_b to Alice through a public channel P.
7. Once receiving B_b, Alice sends her bases B_a to Bob through channel P, and Bob receives B_a.
8. Alice and Bob determine that at which position the bit strings B_a and B_b are equal, and they discard the mismatched bits of B_a and B_b. Then the remaining bits of K_a and K_b, denoted as K_a' and K_b' with $K_{a,b} = K_a' = K_b'$.

We re-introduce the basic BB84 protocol in an abstract way with more technical details as Fig. 1 illustrates.

Now, we assume a special measurement operation $Rand[q;B_a]$ which create a string of n random bits B_a from the q quantum system, and the same as $Rand[q;K_a]$, $Rand[q';B_b]$. $M[q;K_b]$ denotes the Bob's measurement operation of q. The generation of n qubits q through two quantum operations $Set_{K_a}[q]$ and $H_{B_a}[q]$. Alice sends q to Bob through the quantum channel Q by quantum communicating action $send_Q(q)$ and Bob receives q through Q by quantum communicating action $receive_Q(q)$. Bob sends B_b to Alice through the public channel P by classical communicating action $send_P(B_b)$ and Alice receives B_b through channel P by classical communicating action $receive_P(B_b)$, and the same as $send_P(B_a)$ and $receive_P(B_a)$. Alice and Bob generate the private key $K_{a,b}$ by a classical comparison action $cmp(K_{a,b}, K_a, K_b, B_a, B_b)$. Let Alice and Bob be a system AB and let interactions between Alice and Bob be internal actions. AB receives external input D_i through channel A by communicating action $receive_A(D_i)$ and sends results D_o through channel B by communicating action $send_B(D_o)$.
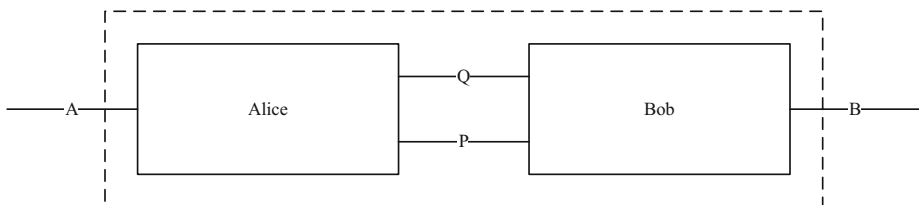

Fig. 1 The BB84 protocol

Then the state transition of Alice can be described by qACP as follows.

$$A = \sum_{D_i \in \Delta_i} receive_A(D_i) \cdot A_1$$
$$A_1 = Rand[q; B_a] \cdot A_2$$
$$A_2 = Rand[q; K_a] \cdot A_3$$
$$A_3 = Set_{K_a}[q] \cdot A_4$$
$$A_4 = H_{B_a}[q] \cdot A_5$$
$$A_5 = send_Q(q) \cdot A_6$$
$$A_6 = receive_P(B_b) \cdot A_7$$
$$A_7 = send_P(B_a) \cdot A_8$$
$$A_8 = cmp(K_{a,b}, K_a, K_b, B_a, B_b) \cdot A$$

where Δ_i is the collection of the input data.

And the state transition of Bob can be described by qACP as follows.

$$B = receive_Q(q) \cdot B_1$$
$$B_1 = Rand[q'; B_b] \cdot B_2$$
$$B_2 = M[q; K_b] \cdot B_3$$
$$B_3 = send_P(B_b) \cdot B_4$$
$$B_4 = receive_P(B_a) \cdot B_5$$
$$B_5 = cmp(K_{a,b}, K_a, K_b, B_a, B_b) \cdot B_6$$
$$B_6 = \sum_{D_o \in \Delta_o} send_B(D_o) \cdot B$$

where Δ_o is the collection of the output data.

The send action and receive action of the same data through the same channel can communicate each other, otherwise, a deadlock δ will be caused. We define the following communication functions.

$$\gamma(send_Q(q), receive_Q(q)) \triangleq c_Q(q)$$
$$\gamma(send_P(B_b), receive_P(B_b)) \triangleq c_P(B_b)$$
$$\gamma(send_P(B_a), receive_P(B_a)) \triangleq c_P(B_a)$$

Let A and B in parallel, then the system AB can be represented by the following process term.

$$\tau_I(\partial_H(A \parallel B))$$

where $H = \{send_Q(q), receive_Q(q), send_P(B_b), receive_P(B_b), send_P(B_a), receive_P(B_a)\}$ and $I = \{Rand[q; B_a], Rand[q; K_a], Set_{K_a}[q], H_{B_a}[q], Rand[q'; B_b], M[q; K_b], c_Q(q), c_P(B_b), c_P(B_a), cmp(K_{a,b}, K_a, K_b, B_a, B_b)\}$.

Then we get the following conclusion.

Theorem 9.1 *The basic BB84 protocol $\tau_I(\partial_H(A \parallel B))$ exhibits desired external behaviors.*

Proof

$$\partial_H(A \parallel B) = \sum_{D_i \in \Delta_i} receive_A(D_i) \cdot \partial_H(A_1 \parallel B)$$

$$\partial_H(A_1 \parallel B) = Rand[q; B_a] \cdot \partial_H(A_2 \parallel B)$$

$$\partial_H(A_2 \parallel B) = Rand[q; K_a] \cdot \partial_H(A_3 \parallel B)$$

$$\partial_H(A_3 \parallel B) = Set_{K_a}[q] \cdot \partial_H(A_4 \parallel B)$$

$$\partial_H(A_4 \parallel B) = H_{B_a}[q] \cdot \partial_H(A_5 \parallel B)$$

$$\partial_H(A_5 \parallel B) = c_Q(q) \cdot \partial_H(A_6 \parallel B_1)$$

$$\partial_H(A_6 \parallel B_1) = Rand[q'; B_b] \cdot \partial_H(A_6 \parallel B_2)$$

$$\partial_H(A_6 \parallel B_2) = M[q; K_b] \cdot \partial_H(A_6 \parallel B_3)$$

$$\partial_H(A_6 \parallel B_3) = c_P(B_b) \cdot \partial_H(A_7 \parallel B_4)$$

$$\partial_H(A_7 \parallel B_4) = c_P(B_a) \cdot \partial_H(A_8 \parallel B_5)$$

$$\partial_H(A_8 \parallel B_5) = cmp(K_{a,b}, K_a, K_b, B_a, B_b) \cdot \partial_H(A \parallel B_5)$$

$$\partial_H(A \parallel B_5) = cmp(K_{a,b}, K_a, K_b, B_a, B_b) \cdot \partial_H(A \parallel B_6)$$

$$\partial_H(A \parallel B_6) = \sum_{D_o \in \Delta_o} send_B(D_o) \cdot \partial_H(A \parallel B)$$

Let $\partial_H(A \parallel B) = \langle X_1 | E \rangle$, where E is the following guarded linear recursion specification:

$$\{X_1 = \sum_{D_i \in \Delta_i} receive_A(D_i) \cdot X_2, X_2 = Rand[q; B_a] \cdot X_3, X_3 = Rand[q; K_a] \cdot X_4,$$

$$X_4 = Set_{K_a}[q] \cdot X_5, X_5 = H_{B_a}[q] \cdot X_6, X_6 = c_Q(q) \cdot X_7,$$

$$X_7 = Rand[q'; B_b] \cdot X_8, X_8 = M[q; K_b] \cdot X_9, X_9 = c_P(B_b) \cdot X_{10}, X_{10} = c_P(B_a) \cdot X_{11},$$

$$X_{11} = cmp(K_{a,b}, K_a, K_b, B_a, B_b) \cdot X_{12}, X_{12} = cmp(K_{a,b}, K_a, K_b, B_a, B_b) \cdot X_{13},$$

$$X_{13} = \sum_{D_o \in \Delta_o} send_B(D_o) \cdot X_1\}$$

Then we apply abstraction operator τ_I into $\langle X_1 | E \rangle$.

$$\tau_I(\langle X_1 | E \rangle) = \sum_{D_i \in \Delta_i} receive_A(D_i) \cdot \tau_I(\langle X_2 | E \rangle)$$

$$= \sum_{D_i \in \Delta_i} receive_A(D_i) \cdot \tau_I(\langle X_3 | E \rangle)$$

$$= \sum_{D_i \in \Delta_i} receive_A(D_i) \cdot \tau_I(\langle X_4 | E \rangle)$$

$$= \sum_{D_i \in \Delta_i} receive_A(D_i) \cdot \tau_I(\langle X_5 | E \rangle)$$

$$= \sum_{D_i \in \Delta_i} receive_A(D_i) \cdot \tau_I(\langle X_6 | E \rangle)$$

$$= \sum_{D_i \in \Delta_i} receive_A(D_i) \cdot \tau_I(\langle X_7 | E \rangle)$$

$$= \sum_{D_i \in \Delta_i} receive_A(D_i) \cdot \tau_I(\langle X_8 | E \rangle)$$

$$= \sum_{D_i \in \Delta_i} receive_A(D_i) \cdot \tau_I(\langle X_9|E\rangle)$$

$$= \sum_{D_i \in \Delta_i} receive_A(D_i) \cdot \tau_I(\langle X_{10}|E\rangle)$$

$$= \sum_{D_i \in \Delta_i} receive_A(D_i) \cdot \tau_I(\langle X_{11}|E\rangle)$$

$$= \sum_{D_i \in \Delta_i} receive_A(D_i) \cdot \tau_I(\langle X_{12}|E\rangle)$$

$$= \sum_{D_i \in \Delta_i} receive_A(D_i) \cdot \tau_I(\langle X_{13}|E\rangle)$$

$$= \sum_{D_i \in \Delta_i} \sum_{D_o \in \Delta_o} receive_A(D_i) \cdot send_B(D_o) \cdot \tau_I(\langle X_1|E\rangle)$$

We get $\tau_I(\langle X_1|E\rangle) = \sum_{D_i \in \Delta_i} \sum_{D_o \in \Delta_o} receive_A(D_i) \cdot send_B(D_o) \cdot \tau_I(\langle X_1|E\rangle)$, that is, $\tau_I(\partial_H(A \parallel B)) = \sum_{D_i \in \Delta_i} \sum_{D_o \in \Delta_o} receive_A(D_i) \cdot send_B(D_o) \cdot \tau_I(\partial_H(A \parallel B))$. So, the basic BB84 protocol $\tau_I(\partial_H(A \parallel B))$ exhibits desired external behaviors. $\qquad\square$

10 Extensions – Renaming Operator

One of the most fascinating characteristics is the modularity of ACP, that is, ACP can be extended easily. Through out this paper, we can see that qACP also inherents the modularity characteristics of ACP. By introducing new operators or new constants, qACP can have more properties. It is already proved that ACP or qACP possibly has the same expressive power as a Turing machine [7]. Though extensions can not improve the expressive power of qACP, but they provide qACP an elegant fashion to express a new property.

In this section, we take an example of renaming operator which is used to rename the atomic quantum operations.

10.1 Transition Rules of Renaming Operators

Renaming operator $\varrho_f(t)$ renames all actions in process term t, and the change of the quantum state is consistent, which is expressed by the following two transition rules.

$$\frac{\langle x, \varrho\rangle \xrightarrow{\upsilon} \langle\sqrt{}, \varrho'\rangle}{\langle \varrho_f(x), \varrho\rangle \xrightarrow{f(\upsilon)} \langle\sqrt{}, \varrho'\rangle}$$

$$\frac{\langle x, \varrho\rangle \xrightarrow{\upsilon} \langle x', \varrho'\rangle}{\langle \varrho_f(x), \varrho\rangle \xrightarrow{f(\upsilon)} \langle \varrho_f(x'), \varrho'\rangle}$$

Theorem 10.1 AQCP$_\tau$ *with guarded linear recursion and renaming operators is a conservative extension of* AQCP$_\tau$ *with guarded linear recursion.*

Proof The corresponding TSS of AQCP$_\tau$ with guarded linear recursion and renaming operators is a conservative extension of that of AQCP$_\tau$ with guarded linear recursion. That

means that $AQCP_\tau$ with guarded linear recursion and renaming operators is a conservative extension of $AQCP_\tau$ with guarded linear recursion.   □

Theorem 10.2 *Quantum rooted branching bisimulation equivalence is a congruence with respect to* $AQCP_\tau$ *with guarded linear recursion and renaming operators.*

Proof The structural part of QTSSs for $AQCP_\tau$ with guarded linear recursion and renaming operators are all in RBB cool format by incorporating the successful termination predicate $\downarrow$ in the transition rules, so rooted branching bisimulation equivalence that they induce is a congruence. According to the definition of quantum rooted branching bisimulation, quantum rooted branching bisimulation equivalence that QTSSs for $AQCP_\tau$ with guarded linear recursion and renaming operators induce is also a congruence.   □

10.2 Axioms for Renaming Operators

The axioms for renaming operator is shown in Table 9.

Theorem 10.3 $\mathcal{E}_{AQCP_\tau}$ + *RSP* + *RDP* + *CFAR* + *QRN1-QRN4 is sound for* $AQCP_\tau$ *with guarded linear recursion and renaming operators, modulo quantum rooted branching bisimulation equivalence.*

Proof Since quantum rooted branching bisimulation is both an equivalence and a congruence for $AQCP_\tau$ with guarded linear recursion and renaming operators, only the soundness of the first clause in the definition of the relation = is needed to be checked. That is, if $s = t$ is an axiom in $\mathcal{E}_{AQCP_\tau}$ + RSP + RDP + CFAR + QRN1-QRN4 and σ a closed substitution that maps the variable in s and t to basic quantum process terms, then we need to check that $\langle \sigma(s), \varrho \rangle \underline{\leftrightarrow}_{rb} \langle \sigma(t), \sigma \rangle$.

Since axioms in $\mathcal{E}_{AQCP_\tau}$ + RSP + RDP + CFAR + QRN1-QRN4 (same as $\mathcal{E}_{ACP_\tau}$ + RSP + RDP + CFAR + QRN1-QRN4) are sound for ACP_τ with guarded linear recursion and renaming operators modulo rooted branching bisimulation equivalence, according to the definition of quantum rooted branching bisimulation, we only need to check if $\varrho' = \sigma'$ when $\varrho = \sigma$, where ϱ evolves into ϱ' after execution of $\sigma(s)$ and σ evolves into σ' after execution of $\sigma(t)$. We can find that every axiom in Table 9 meets the above condition.   □

Theorem 10.4 $\mathcal{E}_{AQCP_\tau}$ + *RSP* + *RDP* + *CFAR* + *QRN1-QRN4 is complete for* $AQCP_\tau$ *with guarded linear recursion and renaming operators, modulo quantum rooted branching bisimulation equivalence.*

Proof To prove that $\mathcal{E}_{AQCP_\tau}$ + RSP + RDP + CFAR + QRN1-QRN4 is complete for $AQCP_\tau$ with guarded linear recursion and renaming operators modulo quantum rooted branching bisilumation equivalence, it means that $\langle s, \varrho \rangle \underline{\leftrightarrow}_{rb} \langle t, \sigma \rangle$ implies $s = t$.

Table 9 Axioms for renaming

No.	Axiom
QRN1	$\varrho_f(\upsilon) = f(\upsilon)$
QRN2	$\varrho_f(\delta) = \delta$
QRN3	$\varrho_f(x + y) = \varrho_f(x) + \varrho_f(y)$
QRN4	$\varrho_f(x \cdot y) = \varrho_f(x) \cdot \varrho_f(y)$

It was already proved that $\mathcal{E}_{\mathrm{AQCP}_\tau}$ + RSP + RDP + CFAR + QRN1-QRN4 (same as $\mathcal{E}_{\mathrm{ACP}_\tau}$ + RSP + RDP + CFAR + QRN1-QRN4) is complete for ACP_τ with guarded linear recursion and renaming operators modulo rooted branching bisimulation equivalence, that is, $s \underline{\leftrightarrow}_{rb} t$ implies $s = t$. $\langle s, \varrho \rangle \underline{\leftrightarrow}_{rb} \langle t, \sigma \rangle$ with $\varrho = \sigma$ means that $s \underline{\leftrightarrow}_{rb} t$ with $\varrho = \sigma$ and $\varrho' = \sigma'$, where ϱ evolves into ϱ' after execution of s and σ evolves into σ' after execution of t, according to the definition of quantum rooted branching bisimulation equivalence. The completeness of $\mathcal{E}_{\mathrm{AQCP}_\tau}$ + RSP + RDP + CFAR + QRN1-QRN4 for ACP_τ with guarded linear recursion and renaming operators modulo rooted branching bisimulation equivalence determines that $\mathcal{E}_{\mathrm{AQCP}_\tau}$ + RSP + RDP + CFAR + QRN1-QRN4 is complete for AQCP_τ with guarded linear recursion and renaming operators modulo quantum rooted branching bisimulation equivalence. □

We can see that qACP with renaming operator and ACP with renaming operator can also be unified under the framework of quantum process configuration $\langle p, \varrho \rangle$.

11 Conclusions

In this paper, we extend the traditional structural operational semantics under the framework of quantum process configuration $\langle p, \varrho \rangle$ to support quantum processes. Based on the relationship between quantum bisimilarity and classical bisimilarity, we establish a series of axiomatization for quantum processes called qACP. We also unify qACP and classical ACP under the framework of quantum process configuration $\langle p, \varrho \rangle$. It makes qACP can adapt to all quantum communication protocols.

Now, we point out some future directions. (1) Quantum entanglement makes the processing of the silent step τ somewhat strange. The nature of influence of quantum entanglement for computation, especially for parallelism and concurrency, should be considered carefully and deeply in future, because quantum entanglement is unique for quantum mechanics. (2) Other novel framework representing quantum processes should be proposed, not only the quantum process configuration $\langle p, \varrho \rangle$. New framework will unify quantum computing and classical computing in a new way, which maybe capture the nature of quantum computing more naturally. (3) qACP inherits the modularity of ACP and makes it can be extended in an elegant fashion, in future, more properties can be extended in qACP. (4) The axiomatization of qACP can be used to verify most quantum communication protocols easily and widely in future.

References

1. Baeten, J.C.M.: A brief history of process algebra. Theor. Comput. Sci. In. Process. Algebra **335**(2–3), 131–146 (2005)
2. Milner, R.: Communication and Concurrency. Prentice Hall, Upper Saddle River (1989)
3. Milner, R., Parrow, J., Walker, D.: A calculus of mobile processes, Parts I and II. Inf. Comput. **1992**(100), 1–77 (1992)
4. Hoare, C.A.R.: Communicating sequential processes. http://www.usingcsp.com/ (1985)
5. Fokkink, W.: Introduction to Process Algebra, 2nd edn. Springer, Berlin (2007)
6. Plotkin, G.D.: A structural approach to operational semantics. Aarhus University, Tech. Report DAIMIFN-19 (1981)
7. Baeten, J.C.M., Bergstra, J.A., Klop, J.W.: On the consistency of Koomen's fair abstraction rule. Theor. Comput. Sci. **51**(1/2), 129–176 (1987)

8. Feng, Y., Duan, R.Y., Ji, Z.F., Ying, M.: Probabilistic bisimulations for quantum processes. Inf. Comput. **205**(2007), 1608–1639 (2007)
9. Gay, S.J., Nagarajan, R.: Communicating quantum processes. In: Proceedings of the 32nd ACM Symposium on Principles of Programming Languages, Long Beach, California, USA, pp. 145–157. ACM Press (2005)
10. Gay, S.J., Nagarajan, R.: Typechecking communicating quantum processes. Math. Struct. Comput. Sci. **2006**(16), 375–406 (2006)
11. Jorrand, P., Lalire, M.: Toward a quantum process algebra. In: Proceedings of the 1st ACM Conference on Computing Frontiers, Ischia, Italy, pp. 111–119. ACM Press (2005)
12. Jorrand, P., Lalire, M.: From quantum physics to programming languages: a process algebraic approach. Lect. Notes Comput. Sci **2005**(3566), 1–16 (2005)
13. Lalire, M.: Relations among quantum processes: bisimilarity and congruence. Math. Struct. Comput. Sci. **2006**(16), 407–428 (2006)
14. Lalire, M., Jorrand, P.: A process algebraic approach to concurrent and distributed quantum computation: operational semantics. In: Proceedings of the 2nd International Workshop on Quantum Programming Languages, pp. 109–126 (2004)
15. Ying, M., Feng, Y., Duan, R., Ji, Z.: An algebra of quantum processes. ACM Trans. Comput. Log. (TOCL) **10**(3), 1–36 (2009)
16. Hennessy, M., Lin, H.: Symbolic bisimulations. Theor. Comput. Sci. **138**(2), 353–389 (1995)
17. Feng, Y., Duan, R., Ying, M.: Bisimulations for quantum processes. In: Proceedings of the 38th ACM Symposium on Principles of Programming Languages (POPL 11), pp. 523–534. ACM Press (2011)
18. Bennett, C.H., Brassard, G.: Quantum cryptography: Public-key distribution and coin tossing. In: Proceedings of the IEEE International Conference on Computer, Systems and Signal Processing, pp. 175–179 (1984)
19. Deng, Y., Feng, Y.: Open bisimulation for quantum processes. Manuscript, arXiv:1201.0416 (2012)
20. Feng, Y., Deng, Y., Ying, M.: Symbolic bisimulation for quantum processes. Manuscript, arXiv:1202.3484 (2012)
21. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information. Cambridge University Press, Cambridge (2000)