International Journal of Theoretical Physics (2019) 58:3611–3626
https://doi.org/10.1007/s10773-019-04226-0

Entanglement in Quantum Process Algebra

Yong Wang[1]

Received: 3 April 2019 / Accepted: 17 July 2019 / Published online: 13 August 2019

Abstract
We explicitly model entanglement in quantum processes by treating entanglement as a kind of parallelism. We introduce a shadow constant quantum operation and a so-called entanglement merge into quantum process algebra qACP. The transition rules of the shadow constant quantum operation and entanglement merge are designed. We also do a sound and complete axiomatization modulo the so-called quantum bisimilarity for the shadow constant quantum operation and entanglement merge. Then, this new type entanglement merge is extended into the full qACP. The new qACP has wide use in verification for quantum protocols, since most quantum protocols have mixtures with classical and quantum information, and also there are many quantum protocols adopting entanglement.

Keywords Quantum mechanics · Entanglement · Quantum processes · Process algebra

1 Introduction

To unify quantum computing and classical computing under the same process algebra framework [1–5], is attractive and has an important significance, because most quantum communication protocols involve quantum information and classical information, quantum computing and classical computing. There are several so-called quantum process algebra, such as CQP (Communicating Quantum Processes) [8, 9], QPAlg (Quantum Process Algebra) [10–13], qCCS [7, 14, 15, 17, 18], qACP [19]. These works try to give quantum protocols and quantum computing a process algebra foundation, some are for pure quantum computing, and the other unify quantum computing and classical computing.

There is one core concept called entanglement which is unique in quantum protocols and quantum computing. Unfortunately, this mechanism has not been modeled in quantum process algebra until now, though there are a few theoretical works on entanglement, such as types for quantum computing [20].

In this paper, we give entanglement a process algebra foundation by treating entanglement as a kind of parallelism. Based on our previous work qACP, we introduce a shadow constant quantum operation Ⓢ and a new kind of entanglement merge $\between$ to model

✉ Yong Wang
wangy@bjut.edu.cn

[1] College of Computer Science and Technology, Faculty of Information Technology, Beijing University of Technology, Beijing, 100124, China

entanglement in quantum protocols and quantum computing. We extend the new kind of parallelism into the whole qACP to make that it can verify quantum protocols involving quantum information with entanglement and classical information mixed.

This work uses some results of the previous works, especially qCCS [14] and qACP [20], in the following ways. (1) We still use the concept of a quantum process configuration $\langle p, \varrho \rangle$ [7, 7–11, 14, 15, 18, 20], which is usually consisted of a process term p and state information ϱ of all (public) quantum information variables. (2) Like qCCS [14] and qACP [20], 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 bisimilarity are avoided.

This paper is organized as follows. We do not introduce some preliminaries, including quantum mechanics, equational logic, structural operational semantics, please refer to [6] and [7] for details. In Section 2, we introduce quantum process algebra qACP. We model entanglement as a kind of parallelism in Section 3 and extend this new kind of parallelism into the whole qACP in Section 4. In Section 5, we verify a quantum protocol which mixes quantum information (with entanglement) and classical information. Finally, we conclude this paper in Section 6.

2 Preliminaries

For convenience of the reader, we introduce quantum process algebra qACP [20] briefly.

ACP [5] is a kind of process algebra which focuses on the specification and manipulation of process terms by use of a collection of operator symbols. In ACP, there are several kind of operator symbols, such as basic operators to build finite processes (called BPA), communication operators to express concurrency (called PAP), deadlock constants and encapsulation enable us to force actions into communications (called ACP), liner recursion to capture infinite behaviors (called ACP with linear recursion), the special constant silent step and abstraction operator (called ACP_τ with guarded linear recursion) allows us to abstract away from internal computations.

Bisimulation or rooted branching bisimulation based structural operational semantics is used to formally provide each process term used the above operators and constants with a process graph. The axiomatization of ACP (according the above classification of ACP, the axiomatizations are $\mathcal{E}_{\mathrm{BPA}}$, $\mathcal{E}_{\mathrm{PAP}}$, $\mathcal{E}_{\mathrm{ACP}}$, $\mathcal{E}_{\mathrm{ACP}}$ + RDP (Recursive Definition Principle) + RSP (Recursive Specification Principle), $\mathcal{E}_{\mathrm{ACP}_\tau}$ + RDP + RSP + CFAR (Cluster Fair Abstraction Rule) respectively) imposes an equation logic on process terms, so two process terms can be equated if and only if their process graphs are equivalent under the semantic model.

ACP can be used to formally reason about the behaviors, such as processes executed sequentially and concurrently by use of its basic operator, communication mechanism, and recursion, desired external behaviors by its abstraction mechanism, and so on.

ACP is organized by modules and can be extended with fresh operators to express more properties of the specification for system behaviors. These extensions are required both the equational logic and the structural operational semantics to be extended. Then the extension can use the whole outcomes of ACP, such as its concurrency, recursion, abstraction, etc.

qACP [20] is the first axiomatization attempt for quantum processes. A weak bisimilarity (quantum branching bisimulation equivalence) is established for quantum processes. This weak bisimilarity is in a non-probabilistic way that follows [14] and can be used to model silent step and abstract internal actions. qACP still uses the framework of a quantum process configuration $\langle p, \varrho \rangle$, but treating 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. 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. qACP establishes the relationship between quantum bisimilarity and classical bisimilarity, including strong bisimilarity and weak bisimilarity, which makes an axiomatization of quantum processes possible. qACP establishes 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. 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.

3 Modeling Entanglement in qACP

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 $\xrightarrow{\alpha} \surd$ represents successful termination after execution of the quantum operation α, 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.

3.1 Entanglement in Quantum Mechanics and Quantum Computing

Quantum information are carried by particles. The simplest non-trivial quantum system is the quantum bit or qubit. A qubit's state space is the 2-dimensional space which is denoted as Q. The space Q is equipped with a standard basis composed with $|0\rangle$ and $|1\rangle$. The tensor product of Q is $Q \otimes Q$ for the space of two qubits and its standard basis composed with the four vectors $|00\rangle$, $|01\rangle$, $|10\rangle$ and $|11\rangle$. Another important basis for $Q \otimes Q$ is called *Bell states* or *EPR states*, which contains the four vectors:

$$\beta_1 = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle),$$

$$\beta_2 = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle),$$

$$\beta_3 = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle),$$

$$\beta_4 = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle).$$

The elements of Bell states are entangled states, which represent systems which are correlated with each other. And many quantum protocols and quantum computation can derive extra power of entanglement, since it is unique for quantum computing.

3.2 Modeling Entanglement as a Kind of Parallelism – Entanglement Merge

We consider entanglement as a kind of parallelism, i.e., information formed by entangled particles may be distributed over a long distance, and quantum operations manipulated on one particle not only change the information represented by this particle, but also those represented by other particles entangled with this particular particle dramatically without any interactions among them. This new kind of parallelism does not need any information exchange and any information channel.

So, we extend the Basic Quantum Process Algebra (BQPA) to form a new Algebra of Quantum Communicating Processes (AQCP) which is also called AQCP.

3.2.1 Shadow Constant

Since process algebra, exactly ACP or qACP, is a kind of algebraic manipulation on actions or quantum operations, and information are hidden by actions and quantum operations. Quantum operation manipulated on one particle will change the quantum states of other entangled particles simultaneously, but, the absence of any quantum operation on other entangled particles will disturb the principles of structural operational semantics on which qACP is based. To conquer this problem, we introduce a special constant quantum operation which is called shadow constant Ⓢ. Now, the set A of all quantum operations is extended to $A \cup \{Ⓢ\}$. The shadow constant Ⓢ is always depended on some entangled particles, when a quantum operation α is manipulated on one particle, then there will be shadow operations $Ⓢ_\alpha$ manipulated on the other entangled particles.

Actually, when one quantum operation α is manipulated on one particle, the states of the other entangled particles are changed without any quantum operation. So, the behavior of the shadow operation Ⓢ is doing nothing, as the following transition rule says. This is why the shadow constant Ⓢ is called a *shadow*, especially, $Ⓢ_\upsilon$ is the shadow of υ.

$$\frac{}{\langle Ⓢ, \varrho\rangle \rightarrow \langle \surd, \varrho\rangle}$$

$$\frac{}{\langle Ⓢ_\upsilon, \varrho\rangle \rightarrow \langle \surd, \varrho\rangle}$$

Obviously, we can get the following two conclusions.

Theorem 1 *BQPA with shadow constant is a conservative extension of BQPA.*

Proof Since the corresponding TSS of BQPA is source-dependent [20], and the transition rules for the shadow constant Ⓢ contain only a fresh constant in their source, so the corresponding TSS of BQPA with shadow constant is a conservative extension of that of BQPA. That means that BQPA with shadow constant is a conservative extension of BQPA. □

Theorem 2 *Quantum bisimulation equivalence is a congruence with respect to BQPA with shadow constant.*

Proof The structural part of QTSSs for BQPA with shadow constant and BQPA are all in panth format [20], so bisimulation equivalence that they induce is a congruence. According

to the definition of quantum bisimulation, quantum bisimulation equivalence that QTSSs for BQPA with shadow constant induce is also a congruence. □

The axioms for shadow constant is shown in Table 1.

We can easily get the following two theorems.

Theorem 3 *$\mathcal{E}_{\mathrm{BQPA}}$ + SC1 - SC3 is sound for BQPA with shadow constant modulo quantum bisimulation equivalence.*

Proof Since quantum bisimulation is both an equivalence and a congruence for BQPA with shadow constant, 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 $\mathcal{E}_{\mathrm{BQPA}}$ + SC1 - SC3 and σ a closed substitution that maps the variable in s and t to basic quantum process terms, then we need to check that $\langle \sigma(s), \varrho \rangle \underline{\leftrightarrow} \langle \sigma(t), \varsigma \rangle$.

Since axioms in $\mathcal{E}_{\mathrm{BQPA}}$ + SC1 - SC3 are sound modulo bisimulation equivalence, according to the definition of quantum bisimulation, we only need to check if $\varrho' = \varsigma'$ when $\varrho = \varsigma$, 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 1 meets the above condition. □

Theorem 4 *$\mathcal{E}_{\mathrm{BQPA}}$ + SC1 - SC3 is complete for BQPA with shadow constant modulo quantum bisimulation equivalence.*

Proof To prove that $\mathcal{E}_{\mathrm{BQPA}}$ + SC1 - SC3 is complete for BQPA with shadow constant modulo quantum bisilumation equivalence, it means that $\langle s, \varrho \rangle \underline{\leftrightarrow} \langle t, \varsigma \rangle$ implies $s = t$.

It can be easily proved that $\mathcal{E}_{\mathrm{BQPA}}$ + SC1 - SC3 is complete for BQPA with shadow constant modulo bisimulation equivalence, that is, $s \underline{\leftrightarrow} t$ implies $s = t$.

(1) The axioms SC1-SC3 are turned into rewriting rules directly from left to right, and added to the three rewriting rules in the proof the completeness of $\mathcal{E}_{\mathrm{BPA}}$ (see [5]). The resulting TRS is terminating modulo AC (Associativity and Commutativity) of + operator through defining new weight functions on process terms.

$$weight(\textcircled{S}) \triangleq 2$$

$$weight(\upsilon) \triangleq 2$$

$$weight(s + t) \triangleq weight(s) + weight(t)$$

$$weight(s \cdot t) \triangleq weight(s)^2 \cdot weight(t)$$

We can get that each application of a rewriting rule strictly decreases the weight of a process term, and that moreover process terms that are equivalent modulo AC of + have the same weight. Hence, the TRS is terminating modulo AC of + .

Table 1 Axioms for shadow constant

No.	Axiom
SC1	$x + \textcircled{S} = \textcircled{S}$
SC2	$\textcircled{S} \cdot x = x$
SC3	$x \cdot \textcircled{S} = x$

(2) We will show that the normal form n are not of the form $s + Ⓢ, Ⓢ \cdot s, s \cdot Ⓢ$. The proof is based on induction with respect to the size of the normal form n.

- If n is an atomic action, then it does not contain the shadow constant Ⓢ.
- n cannot be of the form $s + Ⓢ, s \cdot Ⓢ, Ⓢ \cdot s$, because in that case, the directed version of SC1, SC2 and SC3 would apply to it, contradicting the fact that n is a normal form.

We proved that normal forms are all basic process terms.

(3) We proceed to prove that the axiomatization $\mathcal{E}_{\mathrm{BQPA}}$ + SC1 - SC3 is complete for BQPA with shadow constant modulo bisimulation equivalence. Let the process terms s and t be bisimilar. The TRS is terminating modulo AC of the + , so it reduces s and t to normal forms n and n', respectively. Since the rewrite rules and equivalence modulo AC of the + can be derived from $\mathcal{E}_{\mathrm{BQPA}}$ + SC1 - SC3, $s = n$ and $t = n'$. Soundness of $\mathcal{E}_{\mathrm{BQPA}}$ +SC1 - SC3 then yields $s \underline{\leftrightarrow} n$ and $t \underline{\leftrightarrow} n'$, so $n \underline{\leftrightarrow} s \underline{\leftrightarrow} t \underline{\leftrightarrow} n'$. We shown that the normal forms n and n' are basic process terms. Then it follows that $n \underline{\leftrightarrow} n'$ implies $n =_{AC} n'$. Hence, $s = n =_{AC} n' = t$.

$\langle s, \varrho\rangle \underline{\leftrightarrow} \langle t, \varsigma\rangle$ with $\varrho = \varsigma$ means that $s \underline{\leftrightarrow} t$ with $\varrho = \varsigma$ and $\varrho' = \varsigma'$, 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 $\mathcal{E}_{\mathrm{BQPA}}$ + SC1 - SC3 for BQPA with shadow constant modulo bisimulation equivalence determines that $\mathcal{E}_{\mathrm{BQPA}}$ + SC1 - SC3 is complete for BQPA with shadow constant modulo quantum bisimulation equivalence. □

3.2.2 Entanglement Merge

In AQCP, there are two kind of merges: left merge $\lfloor\!\lfloor$ and communication merge $|$. For parallelism, these two kind of merges remain in the new AQCP. To model entanglement, another new kind of merge called entanglement merge should be added. In this kind of merge, there is not any information exchange via any channel.

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.

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

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

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

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

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.

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

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

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

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

And also the merge $\langle s \parallel t, \varrho\rangle$, in which there is entanglement between s and t, 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 expressed by the following eight transition rules.

$$\frac{\langle x,\varrho\rangle \xrightarrow{\upsilon} \langle\surd,\varrho'\rangle \quad \langle y,\varrho\rangle \xrightarrow{\textcircled{S}_\upsilon} \langle\surd,\varrho'\rangle}{\langle x \parallel y,\varrho\rangle \xrightarrow{\upsilon} \langle\surd,\varrho'\rangle}$$

$$\frac{\langle x,\varrho\rangle \xrightarrow{\textcircled{S}_\upsilon} \langle\surd,\varrho'\rangle \quad \langle y,\varrho\rangle \xrightarrow{\upsilon} \langle\surd,\varrho'\rangle}{\langle x \parallel y,\varrho\rangle \xrightarrow{\upsilon} \langle\surd,\varrho'\rangle}$$

$$\frac{\langle x,\varrho\rangle \xrightarrow{\upsilon} \langle\surd,\varrho'\rangle \quad \langle y,\varrho\rangle \xrightarrow{\textcircled{S}_\upsilon} \langle y',\varrho'\rangle}{\langle x \parallel y,\varrho\rangle \xrightarrow{\upsilon} \langle y',\varrho'\rangle}$$

$$\frac{\langle x,\varrho\rangle \xrightarrow{\textcircled{S}_\upsilon} \langle\surd,\varrho'\rangle \quad \langle y,\varrho\rangle \xrightarrow{\upsilon} \langle y',\varrho'\rangle}{\langle x \parallel y,\varrho\rangle \xrightarrow{\upsilon} \langle y',\varrho'\rangle}$$

$$\frac{\langle x,\varrho\rangle \xrightarrow{\upsilon} \langle x',\varrho'\rangle \quad \langle y,\varrho\rangle \xrightarrow{\textcircled{S}_\upsilon} \langle\surd,\varrho'\rangle}{\langle x \parallel y,\varrho\rangle \xrightarrow{\upsilon} \langle x',\varrho'\rangle}$$

$$\frac{\langle x,\varrho\rangle \xrightarrow{\textcircled{S}_\upsilon} \langle x',\varrho'\rangle \quad \langle y,\varrho\rangle \xrightarrow{\upsilon} \langle\surd,\varrho'\rangle}{\langle x \parallel y,\varrho\rangle \xrightarrow{\upsilon} \langle x',\varrho'\rangle}$$

$$\frac{\langle x,\varrho\rangle \xrightarrow{\upsilon} \langle x',\varrho'\rangle \quad \langle y,\varrho\rangle \xrightarrow{\textcircled{S}_\upsilon} \langle y',\varrho'\rangle}{\langle x \parallel y,\varrho\rangle \xrightarrow{\upsilon} \langle x' \parallel y',\varrho'\rangle}$$

$$\frac{\langle x,\varrho\rangle \xrightarrow{\textcircled{S}_\upsilon} \langle x',\varrho'\rangle \quad \langle y,\varrho\rangle \xrightarrow{\upsilon} \langle y',\varrho'\rangle}{\langle x \parallel y,\varrho\rangle \xrightarrow{\upsilon} \langle x' \parallel y',\varrho'\rangle}$$

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 three extra operator that are called left merge $\between$ and communication merge $|$, and also entanglement merge $\between\!\!\!\!\!\!\between$. We call BQPA extended with the merge operator $\|$, the left merge operator $\between$, the communication merge operator $|$ and the entanglement merge $\between\!\!\!\!\!\!\between$ as Quantum Process Algebra with Parallelism and entanglement, which is also called QPAP.

The left merge $\between$ and communication merge $|$ are the same as those in QPAP in qACP. The eight transition rules of entanglement merge $\between\!\!\!\!\!\!\between$ are as follows.

$$\frac{\langle x,\varrho\rangle \xrightarrow{\upsilon} \langle\surd,\varrho'\rangle \quad \langle y,\varrho\rangle \xrightarrow{\circledS_\upsilon} \langle\surd,\varrho'\rangle}{\langle x \between\!\!\!\!\!\!\between y,\varrho\rangle \xrightarrow{\upsilon} \langle\surd,\varrho'\rangle}$$

$$\frac{\langle x,\varrho\rangle \xrightarrow{\circledS_\upsilon} \langle\surd,\varrho'\rangle \quad \langle y,\varrho\rangle \xrightarrow{\upsilon} \langle\surd,\varrho'\rangle}{\langle x \between\!\!\!\!\!\!\between y,\varrho\rangle \xrightarrow{\upsilon} \langle\surd,\varrho'\rangle}$$

$$\frac{\langle x,\varrho\rangle \xrightarrow{\upsilon} \langle\surd,\varrho'\rangle \quad \langle y,\varrho\rangle \xrightarrow{\circledS_\upsilon} \langle y',\varrho'\rangle}{\langle x \between\!\!\!\!\!\!\between y,\varrho\rangle \xrightarrow{\upsilon} \langle y',\varrho'\rangle}$$

$$\frac{\langle x,\varrho\rangle \xrightarrow{\circledS_\upsilon} \langle\surd,\varrho'\rangle \quad \langle y,\varrho\rangle \xrightarrow{\upsilon} \langle y',\varrho'\rangle}{\langle x \between\!\!\!\!\!\!\between y,\varrho\rangle \xrightarrow{\upsilon} \langle y',\varrho'\rangle}$$

$$\frac{\langle x,\varrho\rangle \xrightarrow{\upsilon} \langle x',\varrho'\rangle \quad \langle y,\varrho\rangle \xrightarrow{\circledS_\upsilon} \langle\surd,\varrho'\rangle}{\langle x \between\!\!\!\!\!\!\between y,\varrho\rangle \xrightarrow{\upsilon} \langle x',\varrho'\rangle}$$

$$\frac{\langle x,\varrho\rangle \xrightarrow{\circledS_\upsilon} \langle x',\varrho'\rangle \quad \langle y,\varrho\rangle \xrightarrow{\upsilon} \langle\surd,\varrho'\rangle}{\langle x \between\!\!\!\!\!\!\between y,\varrho\rangle \xrightarrow{\upsilon} \langle x',\varrho'\rangle}$$

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

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

We can get the following conclusions.

Theorem 5 *QPAP is a conservative extension of BQPA with shadow constant.*

Proof Since the corresponding TSS of BQPA with shadow constant is source-dependent [20], and the transition rules for merge operator $\|$, left merge operator $\between$, communication merge $|$ and entanglement merge $\between\!\!\!\!\!\!\between$ contain only a fresh operator in their source, so the corresponding TSS of QPAP is a conservative extension of that of BQPA with

shadow constant. That means that QPAP is a conservative extension of BQPA with shadow constant. □

Theorem 6 *Quantum bisimulation equivalence is a congruence with respect to QPAP.*

Proof The structural part of QTSSs for QPAP and BQPA with shadow constant are all in panth format [20], 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. □

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

Theorem 7 $\mathcal{E}_{\mathrm{QPAP}}$ *is sound for QPAP modulo quantum bisimulation equivalence.*

Proof Since quantum bisimulation is both an equivalence and a congruence for QPAP [20], 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

Table 2 Axioms for QPAP

No.	Axiom
QM1	$x \parallel y = (x \lfloor\!\lfloor y + y \lfloor\!\lfloor x) + x \mid y + x \between y$
QLM2	$\upsilon \lfloor\!\lfloor y = \upsilon \cdot y$
QLM3	$(\upsilon \cdot x) \lfloor\!\lfloor y = \upsilon \cdot (x \parallel y)$
QLM4	$(x + y) \lfloor\!\lfloor z = x \lfloor\!\lfloor z + y \lfloor\!\lfloor z$
QCM5	$\nu \mid \mu = \gamma(\nu, \mu)$
QCM6	$\nu \mid (\mu \cdot y) = \gamma(\nu, \mu) \cdot y$
QCM7	$(\nu \cdot x) \mid \mu = \gamma(\nu, \mu) \cdot x$
QCM8	$(\nu \cdot x) \mid (\mu \cdot y) = \gamma(\nu, \mu) \cdot (x \parallel y)$
QCM9	$(x + y) \mid z = x \mid z + y \mid z$
QCM10	$x \mid (y + z) = x \mid y + x \mid z$
QEM11	$\upsilon \between \textcircled{S}_\upsilon = \upsilon$
QEM12	$\textcircled{S}_\upsilon \between \upsilon = \upsilon$
QEM13	$\upsilon \between (\textcircled{S}_\upsilon \cdot y) = \upsilon \cdot y$
QEM14	$\textcircled{S}_\upsilon \between (\upsilon \cdot y) = \upsilon \cdot y$
QEM15	$(\upsilon \cdot x) \between \textcircled{S}_\upsilon = \upsilon \cdot x$
QEM16	$(\textcircled{S}_\upsilon \cdot x) \between \upsilon = \upsilon \cdot x$
QEM17	$(\upsilon \cdot x) \between (\textcircled{S}_\upsilon \cdot y) = \upsilon \cdot (x \parallel y)$
QEM18	$(\textcircled{S}_\upsilon \cdot x) \between (\upsilon \cdot y) = \upsilon \cdot (x \parallel y)$
QEM19	$(x + y) \between z = x \between z + y \between z$
QEM20	$x \between (y + z) = x \between y + x \between z$
QEM21	$x + \textcircled{S} = x$
QEM22	$x \cdot \textcircled{S} = x$
QEM23	$\textcircled{S} \cdot x = x$

maps the variable in s and t to basic quantum process terms, then we need to check that $\langle \sigma(s), \varrho\rangle \underline{\leftrightarrow} \langle \sigma(t), \varsigma\rangle$.

Since axioms in $\mathcal{E}_{\mathrm{QPAP}}$ (same as $\mathcal{E}_{\mathrm{PAP}}$) are sound for QPAP modulo bisimulation equivalence, according to the definition of quantum bisimulation, we only need to check if $\varrho' = \varsigma'$ when $\varrho = \varsigma$, 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 8 *$\mathcal{E}_{\mathrm{QPAP}}$ 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 $\langle s, \varrho\rangle \underline{\leftrightarrow} \langle t, \varsigma\rangle$ implies $s = t$.

It can be easily proved that $\mathcal{E}_{\mathrm{QPAP}}$ (same as $\mathcal{E}_{\mathrm{PAP}}$) is complete for PAP modulo bisimulation equivalence, that is, $s \underline{\leftrightarrow} t$ implies $s = t$.

(1) The axioms QM1, QLM2-QLM4, QCM5-QCM10, QEM11-QEM23 are turned into rewriting rules directly from left to right, and added to the 20 rewriting rules in the proof the completeness of $\mathcal{E}_{\mathrm{BPA}}$ (see [5]). The resulting TRS is terminating modulo AC (Associativity and Commutativity) of + operator through defining new weight functions on process terms.

$$
\begin{aligned}
&weight(s \parallel t) \triangleq 4 \cdot (weight(s) \cdot weight(t))^2 + 1\\
&weight(s \between t) \triangleq (weight(s) \cdot weight(t))^2\\
&weight(s \mid t) \triangleq (weight(s) \cdot weight(t))^2\\
&weight(s \between t) \triangleq (weight(s) \cdot weight(t))^2
\end{aligned}
$$

We can get that each application of a rewriting rule strictly decreases the weight of a process term, and that moreover process terms that are equivalent modulo AC of + have the same weight. Hence, the TRS is terminating modulo AC of + .

(2) We will show that the normal form n are not of the form $s \parallel t$, $s \between t$, $s \mid t$ and $s \between t$. The proof is based on induction with respect to the size of the normal form n.

- If n is an atomic action, then it does not contain the shadow constant Ⓢ.
- Suppose $n =_{AC} s + t$ or $n =_{AC} s \cdot t$. Then by induction, the normal forms s and t do not contain †, so n does not contain any parallel operator.
- n cannot be of the form $s \parallel t$, because in that case, the directed version of QM1 would apply to it, contradicting the fact that n is a normal form.
- n cannot be of the form $s \between t$, because in that case, the directed version of QLM2-QLM4 would apply to it, contradicting the fact that n is a normal form.
- n cannot be of the form $s \mid t$, because in that case, the directed version of QCM5-QCM10 would apply to it, contradicting the fact that n is a normal form.
- n cannot be of the form $s \between t$, because in that case, the directed version of QEM11-QEM20 would apply to it, contradicting the fact that n is a normal form.

We proved that normal forms are all basic process terms.

(3) We proceed to prove that the axiomatization $\mathcal{E}_{\mathrm{QPAP}}$ is complete for QPAP modulo bisimulation equivalence. Let the process terms s and t be bisimilar. The TRS is terminating modulo AC of the + , so it reduces s and t to normal forms n and n', respectively. Since the rewrite rules and equivalence modulo AC of the + can be derived from $\mathcal{E}_{\mathrm{QPAP}}$, $s = n$ and $t = n'$. Soundness of $\mathcal{E}_{\mathrm{QPAP}}$ then yields $s \underline{\leftrightarrow} n$ and $t \underline{\leftrightarrow} n'$, so $n \underline{\leftrightarrow} s \underline{\leftrightarrow} t \underline{\leftrightarrow} n'$.

Table 3 Two extra axioms for AQCP

No.	Axiom
QEM24	$\delta \between x = \delta$
QEM25	$x \between \delta = \delta$

We shown that the normal forms n and n' are basic process terms. Then it follows that $n \underline{\leftrightarrow} n'$ implies $n =_{AC} n'$. Hence, $s = n =_{AC} n' = t$.

$\langle s, \varrho\rangle \underline{\leftrightarrow} \langle t, \varsigma\rangle$ with $\varrho = \varsigma$ means that $s \underline{\leftrightarrow} t$ with $\varrho = \varsigma$ and $\varrho' = \varsigma'$, 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 QPAP modulo bisimulation equivalence determines that $\mathcal{E}_{\mathrm{QPAP}}$ is complete for QPAP modulo quantum bisimulation equivalence. □

For deadlock constant δ and encapsulation operator ∂_H, two extra axioms should be added, as Table 3 shows.

We can easily get that the new axiomatization $\mathcal{E}_{\mathrm{AQCP}}$ is sound for AQCP modulo quantum bisimulation equivalence, and the new $\mathcal{E}_{\mathrm{AQCP}}$ is complete for AQCP modulo quantum bisimulation equivalence.

4 qACP with Entanglement Merge

Now, we consider the influence of the new AQCP with entanglement to the whole qACP, i.e., AQCP with guarded recursion and $AQCP_\tau$ with guarded recursion, which are based on AQCP.

Guarded recursion defines infinite computation through guarded recursion specifications. Extension to guarded recursion based on the new AQCP has almost no influence comparing with that in qACP. The axiomatization $\mathcal{E}_{\mathrm{AQCP}}$ + RDP + RSP is sound and complete for AQCP with linear recursion modulo quantum bisimulation equivalence.

Similarly, the new AQCP does not influence $AQCP_\tau$ with guarded recursion, i.e., $\mathcal{E}_{\mathrm{AQCP}_\tau}$ + RSP + RDP + CFAR is sound and complete for $AQCP_\tau$ with guarded linear recursion, modulo quantum rooted branching bisimulation equivalence.

But, entanglement merge $\between$ makes entanglement explicit in qACP. Based on the framework of quantum process configuration $\langle p, \varrho\rangle$, by introducing silent step τ and abstraction operator τ_I, the definition of ϱ only records the so-called public quantum variables and claim that a τ operation only manipulates on entangled quantum variables which should be included in the so-called private variables. Now, we explicitly define a new entanglement merger to model entanglement in quantum processes and this declaration can be moved away.

Since, shadow constant quantum operation and entanglement merge are defined for quantum operations, i.e., they are only valid for quantum operations. A quantum operation α can only effect with its shadow constant $Ⓢ_\alpha$, any other mismatch, such as α and β, α and $Ⓢ_\beta$, a classical action a and a quantum operation α, will all cause a deadlock δ. This leads that qACP with entanglement merge also unify quantum and classical computing in a high level of computational logic, the same as qACP does.

From now on, we call qACP which represents not only the original qACP, but also qACP with entanglement merge.

5 Verification for Quantum Protocols with Entanglement – the E91 Protocol

With support of Entanglement merge $\langle\rangle$, now, qACP can be used to verify quantum protocols utilizing entanglement. The E91 protocol [16] is the first quantum protocol which utilizes entanglement and mixes quantum and classical information. In this section, we take an example of verification for the E91 protocol.

The E91 protocol is used to create a private key between two parities, Alice and Bob. Firstly, we introduce the basic E91 protocol briefly, which is illustrated in Fig. 1.

1. Alice generates a string of EPR pairs q with size n, i.e., $2n$ particles, and sends a string of qubits q_b from each EPR pair with n to Bob through a quantum channel Q, remains the other string of qubits q_a from each pair with size n.
2. Alice create two string of bits with size n randomly, denoted as B_a and K_a.
3. Bob receives q_b and randomly generates a string of bits B_b with size n.
4. Alice measures each qubit of q_a according to a basis by bits of B_a. And the measurement results would be K_a, which is also with size n.
5. Bob measures each qubit of q_b 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 E91 protocol in an abstract way with more technical details as Fig. 1 illustrates.

Now, $M[q_a; K_a]$ denotes the Alice's measurement operation of q_a, and $\textcircled{S}_{M[q_a;K_a]}$ denotes the responding shadow constant; $M[q_b; K_b]$ denotes the Bob's measurement operation of q_b, and $\textcircled{S}_{M[q_b;K_b]}$ denotes the responding shadow constant. Alice sends q_b to Bob through the quantum channel Q by quantum communicating action $send_Q(q_b)$ and Bob receives q_b through Q by quantum communicating action $receive_Q(q_b)$. 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

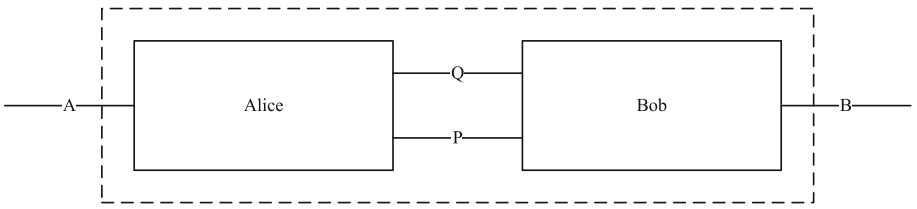

Fig. 1 The E91 protocol

through channel A by communicating action $receive_A(D_i)$ and sends results D_o through channel B by communicating action $send_B(D_o)$.

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

$$\begin{aligned}
A &= \sum_{D_i \in \Delta_i} receive_A(D_i) \cdot A_1 \\
A_1 &= send_Q(q_b) \cdot A_2 \\
A_2 &= M[q_a; K_a] \cdot A_3 \\
A_3 &= \textcircled{S}_{M[q_b; K_b]} \cdot A_4 \\
A_4 &= receive_P(B_b) \cdot A_5 \\
A_5 &= send_P(B_a) \cdot A_6 \\
A_6 &= cmp(K_{a,b}, K_a, K_b, B_a, B_b) \cdot A
\end{aligned}$$

where Δ_i is the collection of the input data.

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

$$\begin{aligned}
B &= receive_Q(q_b) \cdot B_1 \\
B_1 &= \textcircled{S}_{M[q_a; K_a]} \cdot B_2 \\
B_2 &= M[q_b; K_b] \cdot B_3 \\
B_3 &= send_P(B_b) \cdot B_4 \\
B_4 &= receive_P(B_a) \cdot B_5 \\
B_5 &= cmp(K_{a,b}, K_a, K_b, B_a, B_b) \cdot B_6 \\
B_6 &= \sum_{D_o \in \Delta_o} send_B(D_o) \cdot B
\end{aligned}$$

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. The quantum operation and its shadow constant pair will lead entanglement occur, otherwise, a deadlock δ will occur. We define the following communication functions.

$$\begin{aligned}
\gamma(send_Q(q_b), receive_Q(q_b)) &\triangleq c_Q(q_b) \\
\gamma(send_P(B_b), receive_P(B_b)) &\triangleq c_P(B_b) \\
\gamma(send_P(B_a), receive_P(B_a)) &\triangleq c_P(B_a)
\end{aligned}$$

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 $H = \{send_Q(q_b), receive_Q(q_b), send_P(B_b), receive_P(B_b), send_P(B_a), receive_P(B_a), M[q_a; K_a], \textcircled{S}_{M[q_a; K_a]}, M[q_b; K_b], \textcircled{S}_{M[q_b; K_b]}\}$ and $I = \{c_Q(q_b), c_P(B_b), c_P(B_a), M[q_a; K_a], M[q_b; K_b], cmp(K_{a,b}, K_a, K_b, B_a, B_b)\}$.

Then we get the following conclusion.

Theorem 9 *The basic E91 protocol $\tau_I(\partial_H(A \parallel B))$ exhibits desired external behaviors.*

Proof

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

$$\partial_H(A_1 \parallel B) = c_Q(q_b) \cdot \partial_H(A_2 \parallel B_1)$$
$$\partial_H(A_2 \parallel B_1) = M[q_a; K_a] \cdot \partial_H(A_3 \parallel B_2)$$
$$\partial_H(A_3 \parallel B_2) = M[q_b; K_b] \cdot \partial_H(A_4 \parallel B_3)$$
$$\partial_H(A_4 \parallel B_3) = c_P(B_b) \cdot \partial_H(A_5 \parallel B_4)$$
$$\partial_H(A_5 \parallel B_4) = c_P(B_a) \cdot \partial_H(A_6 \parallel B_5)$$
$$\partial_H(A_6 \parallel B_5) = cmp(K_{a,b}, K_a, K_b, B_a, B_b) \cdot \partial_H(A \parallel B_5)$$
$$\partial_H(A \parallel B_5) = cmp(K_{a,b}, K_a, K_b, B_a, B_b) \cdot \partial_H(A \parallel B_6)$$
$$\partial_H(A \parallel B_6) = \sum_{D_o \in \Delta_o} send_B(D_o) \cdot \partial_H(A \parallel B)$$

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

$$\{X_1 = \sum_{D_i \in \Delta_i} receive_A(D_i) \cdot X_2, X_2 = c_Q(q_b) \cdot X_3,$$
$$X_3 = M[q_a; K_a] \cdot X_4, X_4 = M[q_b; K_b] \cdot X_5, X_5 = c_P(B_b) \cdot X_6, X_6 = c_P(B_a) \cdot X_7,$$
$$X_7 = cmp(K_{a,b}, K_a, K_b, B_a, B_b) \cdot X_8, X_8 = cmp(K_{a,b}, K_a, K_b, B_a, B_b) \cdot X_9,$$
$$X_9 = \sum_{D_o \in \Delta_o} send_B(D_o) \cdot X_1\}$$

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

$$\begin{aligned}
\tau_I(\langle X_1 | E \rangle) &= \sum_{D_i \in \Delta_i} receive_A(D_i) \cdot \tau_I(\langle X_2 | E \rangle) \\
&= \sum_{D_i \in \Delta_i} receive_A(D_i) \cdot \tau_I(\langle X_3 | E \rangle) \\
&= \sum_{D_i \in \Delta_i} receive_A(D_i) \cdot \tau_I(\langle X_4 | E \rangle) \\
&= \sum_{D_i \in \Delta_i} receive_A(D_i) \cdot \tau_I(\langle X_5 | E \rangle) \\
&= \sum_{D_i \in \Delta_i} receive_A(D_i) \cdot \tau_I(\langle X_6 | E \rangle) \\
&= \sum_{D_i \in \Delta_i} receive_A(D_i) \cdot \tau_I(\langle X_7 | E \rangle) \\
&= \sum_{D_i \in \Delta_i} receive_A(D_i) \cdot \tau_I(\langle X_8 | E \rangle) \\
&= \sum_{D_i \in \Delta_i} receive_A(D_i) \cdot \tau_I(\langle X_9 | E \rangle) \\
&= \sum_{D_i \in \Delta_i} \sum_{D_o \in \Delta_o} receive_A(D_i) \cdot send_B(D_o) \cdot \tau_I(\langle X_1 | E \rangle)
\end{aligned}$$

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 E91 protocol $\tau_I(\partial_H(A \parallel B))$ exhibits desired external behaviors. □

6 Conclusions

We explicitly model entanglement in quantum processes by introducing a shadow constant quantum operation Ⓢ and a so-called entanglement merge ⟆ into quantum process algebra qACP. The new qACP has wide use in verification for quantum protocols, since most quantum protocols have mixtures with classical and quantum information, and also there are many quantum protocols adopting entanglement.

To maintain the principle of structural operational semantics on which qACP is based, the shadow constant quantum operation is really a kind of placeholder, and the entanglement merge ⟆ actually does a synchronization between two interleaving processes at the point of the quantum operation and its shadows. During verification for quantum protocols, the synchronization point and the shadow constant quantum operations are put in place during the modeling phase.

But, (1) This synchronization and the shadow constant (though it is only a shadow) are not existing actually in quantum protocols and quantum computing; (2) qACP is a kind of high level computational logic, though quantum and classical computing are unified under this high level computational logic, but the hidden quantum information and more technical details can not be observed. In future, more suitable theory should be pursued to satisfy the above two requirements.

References

1. Baeten, J.C.M.: A brief history of process algebra. Theor. Comput. Sci. Process Algebra **335**(2–3), 131–146 (2005)
2. Milner, R.: Communication and Concurrency. Prentice Hall (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 (2007)
6. Plotkin, G.D.: A structural approach to operational semantics. Aarhus University, Tech Report DAIMIFN-19 (1981)
7. Feng, Y., Duan, R.Y., Ji, Z.F., Ying, M.S.: Probabilistic bisimulations for quantum processes. Inf. Comput. **2007**(205), 1608–1639 (2007)
8. Gay, S.J., Nagarajan, R.: Communicating quantum processes. In: Proceedings of the 32nd ACM Symposium on Principles of Programming Languages, pp. 145–157. ACM Press, Long Beach (2005)
9. Gay, S.J., Nagarajan, R.: Typechecking communicating quantum processes. Math. Struct. Comput. Sci. **2006**(16), 375–406 (2006)
10. Jorrand, P., Lalire, M.: Toward a quantum process algebra. In: Proceedings of the 1st ACM Conference on Computing Frontiers, pp. 111–119. ACM Press, Ischia (2005)
11. Jorrand, P., Lalire, M.: From quantum physics to programming languages: A process algebraic approach. Lect. Notes Comput. Sci **2005**(3566), 1–16 (2005)
12. Lalire, M.: Relations among quantum processes: Bisimilarity and congruence. Math. Struct. Comput. Sci. **2006**(16), 407–428 (2006)
13. 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. TUCS General Publications (2004)
14. Ying, M., Feng, Y., Duan, R., Ji, Z.: An algebra of quantum processes. ACM Trans. Comput. Logic (TOCL) **10**(3), 1–36 (2009)
15. 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)
16. Ekert, A.K.: Quantum cryptography based on Bell's theorem. Phys. Rev. Lett. **67**(6), 661–663 (1991)
17. Deng, Y., Feng, Y.: Open bisimulation for quantum processes. Manuscript, arXiv:1201.0416 (2012)

18. Feng, Y., Deng, Y., Ying, M.: Symbolic bisimulation for quantum processes. Manuscript, arXiv:1202.3484 (2012)
19. Wang, Y.: An axiomatization for quantum processes to unifying quantum and classical computing. Manuscript, arXiv:1311.2960 (2013)
20. Duncan, R.: Types for Quantum Computing. Ph.D. Dessertation, Oxford University (2006)