https://doi.org/10.1007/s00165-018-0465-3
BCS © 2018
Formal Aspects of Computing

Formal Aspects of Computing

Toward automatic verification of quantum programs

Mingsheng Ying [1,2,3]

[1]Centre for Quantum Software and Information, University of Technology Sydney, Sydney NSW 2007, Australia
[2]State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing, China.
[3]Department of Computer Science and Technology, Tsinghua University, Beijing, China

Abstract. This paper summarises the results obtained by the author and his collaborators in a program logic approach to the verification of quantum programs, including quantum Hoare logic, invariant generation and termination analysis for quantum programs. It also introduces the notion of proof outline and several auxiliary rules for more conveniently reasoning about quantum programs. Some problems for future research are proposed at the end of the paper.

Keywords: Quantum programming; Hoare logic; Proof outline; Auxiliary rules; Invariant generation; Termination analysis

1. Introduction

Programming is error-prone. Programming a quantum computer and designing quantum communication protocols are even worse due to the weird nature of quantum systems [Yin16]. Therefore, verification techniques and automatic tools for quantum programs and quantum protocols will be indispensable whence commercial quantum computers and quantum communication systems are available. In the last 10 years, various verification techniques for classical programs have been extended to deal with quantum programs; in particular, several quantum program logics have been proposed:

- Brunet and Jorrand [BJ04] proposed to apply Birkhoff-von Neumann quantum logic in reasoning about quantum programs. In the Birkhoff-von Neumann logic, closed subspaces of the state Hilbert space of a quantum system are used to represent the properties of the system, and logical connectives $\neg$ (negation), $\wedge$ (and), $\vee$ (or) are interpreted as the operations: ortho-complement, meet and join, respectively, in the orthomodular lattice of all closed subspaces of the Hilbert space. The idea of [BJ04] is to add quantum operations (e.g. unitary transformations and quantum measurements) into the language of quantum logic so that it can be employed to describe and reason about the dynamics (evolution) of a quantum system (e.g. a quantum program).
- Chadha, Mateus and Sernadas [CMS06] presented one of the first attempts to develop a Hoare-like logic for quantum programs. The quantum programs considered in [CMS06] can have both classical and quantum

Correspondence and offprint requests to: M. Ying, E-mail: Mingsheng.Ying@uts.edu.au

variables (memories). But only bounded iterations are allowed in these quantum programs. The logic given in [CMS06] is an extension of exogenous quantum logic [MS06], which has terms representing *amplitudes* of quantum states, obtained by introducing the axioms for unitary transformations and quantum measurements. Roughly speaking, the assertions (preconditions and postconditions) in this logic are properties of the real and imaginary components of the amplitudes.

- Baltag and Smets [BS06] chose to develop a quantum generalisation of Propositional Dynamic Logic (PDL), which is an extension of Hoare logic for verification of classical programs. The approach of [BS06] is different from [BJ04, CMS06], and it can be essentially seen as a reinterpretation of the language of PDL where a quantum action is interpreted as a (classical) relation between quantum states and models the input–output relation of a quantum program. In particular, by introducing some axioms to handle separation and locality, their proof system can be used to reason about the behaviours (e.g. entanglement) of compound quantum systems.

- Kakutani [Ka09] proposed a quantum extension of Den Hartog and De Vink's probabilistic Hoare logic [KV02]. The quantum programming language used in [Ka09] is Selinger's QPL [Sel04] without recursive calls but with while loops. The assertions in this logic are probabilistic predicates (or assertions) used in [KV02] but, of course, re-interpreted in the setting of quantum computation (e.g. the probability that qubit variable q returns to 0 upon the measurement in the computational basis is $\frac{1}{4}$).

- Some useful proof rules for reasoning about quantum programs were introduced by Feng et al. [FDJY07]. The programming language employed in [FDJY07] is a natural quantum extension of the **while**-language. For simplicity of presentation, only purely quantum programs without classical variables were considered there. The assertions in these proof rules are physical observables (mathematically modelled by Hermitian operators), or quantum predicates as called in [DP06]. Furthermore, a Hoare-like logic for both partial and total correctness of quantum programs with (relative) completeness was established in [Yin11].

The mathematic foundations, the programming languages and the expressivity of the assertions of the quantum program logics presented in [CMS06, Ka09, Yin11] are carefully compared in a recent survey [Rand17].

Except program logics, model-checking techniques have also been developed for verification of quantum communication protocols and quantum programs. For example, Gay, Papanikolaou and Nagarajan [GPN08] developed a model-checker for verifying properties of the quantum systems that can be modelled in the stabiliser formalism. Feng et al. [FYY13] presented an algorithm for model-checking quantum systems described as super-operator valued Markov chains, and the algorithm was implemented by Feng et al. in [FHTZ15]. The problem of checking linear-time properties of quantum systems was studied in [YLYF14], where linear-time properties are defined to be infinite sequences of sets of atomic propositions modelled as closed subspaces of the state Hilbert spaces, as in the Birkhoff-von Neumann quantum logic.

The main purpose of this paper is to summarise the results obtained by the author and his collaborators in a program logic approach to the verification of quantum programs. The paper is an extension of an invited talk at SETTA'2016 [Yin16a], but the results presented in Sects. 4 and 5 as well as the examples given in Sect. 3 are new and have not been published elsewhere.

The paper is organised as follows: The syntax and semantics of a quantum extension of the **while**-language is defined and a logic of Hoare-style for reasoning about quantum programs is presented in Sect. 2. As is well-known, correctness specification of classical programs can be very tricky. Indeed, correctness specification of quantum programs can be even much trickier. In Sect. 3, we present several simple examples through which the reader can better understand the difference between classical and quantum correctness specifications. The notion of proof outline is introduced and a strong soundness theorem is proved for quantum programs in Sect. 4. Several auxiliary axioms and inference rules are given in Sect. 5, which can help to simplify verifications of quantum programs. Some basic ideas for mechanising quantum program verification based on the quantum Hoare logic are described in Sect. 6. The algorithms for generating invariants and termination analysis of quantum programs are summarised in Sects. 7 and 8, respectively. Some topics for future studies are pointed out in the concluding section. For convenience of the reader, several basic properties of operators in Hilbert spaces needed in this paper are listed in Appendix A. Several simple quantum relations (i.e. quantum predicates in a tensor product of state Hilbert spaces) were employed in Sect. 3, a brief discussion on composition of quantum relations that can be used to generate more sophisticated quantum relations from these simple ones is given in Appendix B. For readability, the proofs of the results in Sects. 4 and 5 are deferred to Appendix C.

2. Hoare logic for quantum programs

Hoare logic is a cornerstone in verification of classical programs. As mentioned in Sect. 1, several attempts [BJ04, CMS06, BS06, FDJY07, Ka09, Yin11] have been made to build a Hoare-like logic for quantum programs. In this section, we focus on the one presented in [Yin11] because it is the only quantum Hoare logic with (relative) completeness in the existing literature.

2.1. Syntax and semantics

Let us first briefly recall the syntax of a quantum extension of the **while**-language. It is assumed that the reader is familiar with the basic notions of quantum computing. (A reader not familiar with them can have a quick look at Section 2 of [Yin11] or Section 1.1 of [YYW17]; for more details, we refer to [NC00] or [Yin16].) We assume a countably infinite set Var of quantum variables. For each $q \in Var$, we write $\mathcal{H}_q$ for its state Hilbert space.

Definition 1 (*Syntax* [Yin11, Yin16]) The quantum **while**-programs are defined by the grammar:

$$P ::= \mathbf{skip} \mid P_1;\ P_2 \mid q :=\mid 0\rangle \mid \overline{q} := U[\overline{q}] \tag{1}$$

$$\mid \mathbf{if}\ (\square m \cdot M[\overline{q}] = m \to P_m)\ \mathbf{fi} \tag{2}$$

$$\mid \mathbf{while}\ M[\overline{q}] = 1\ \mathbf{do}\ P\ \mathbf{od} \tag{3}$$

The command "$q :=\mid 0\rangle$" is an initialisation that sets quantum variable q to a basis state $\mid 0\rangle$. The statement "$\overline{q} := U[\overline{q}]$" means that unitary transformation U is performed on quantum register $\overline{q}$, leaving the states of the variables not in $\overline{q}$ unchanged. The construct "$\mathbf{if} \cdots \mathbf{fi}$" is a quantum generalisation of case or switch statement. In executing it, measurement $M = \{M_m\}$ is performed on $\overline{q}$, and then a subprogram P_m is selected to be executed next according to the outcomes m of measurement. The statement "$\mathbf{while} \cdots \mathbf{od}$" is a quantum generalisation of **while**-loop. The measurement in it has only two possible outcomes 0, 1. If the outcome 0 is observed, then the program terminates, and if the outcome 1 occurs, the program executes the loop body P and continues the loop.

To further illustrate these program constructs, let us see two simple examples, both of which are quantum generalisations of certain probabilistic programs. The first is taken from [YYW17], and it is the quantum version of the example of three dials in [KMMM10].

Example 1 (Three Quantum Dials) Consider a slot machine that has three dials d_1, d_2, d_3 and two suits $\heartsuit$ and $\diamondsuit$. It spins the dials independently so that they come to rest on each of the suits with equal probability. This machine can be modelled as a probabilistic program:

$$flip \equiv (d_1 := \heartsuit \oplus_{\frac{1}{2}} d_1 := \diamondsuit);\ (d_2 := \heartsuit \oplus_{\frac{1}{2}} d_2 := \diamondsuit);\ (d_3 := \heartsuit \oplus_{\frac{1}{2}} d_3 := \diamondsuit)$$

where $P_1 \oplus_p P_2$ stands for a probabilistic choice which chooses to execute P with probability p and to execute Q with probability $1 - p$.

We can define a quantum variant of $flip$ as follows:

$$qflip \equiv H[d_1];\ H[d_2];\ H[d_3]$$

where H is the Hadamard gate in the 2-dimensional Hilbert space $\mathcal{H}_2$ with $\{\mid \heartsuit\rangle, \mid \diamondsuit\rangle\}$ as an orthonormal basis. The program $qflip$ also spins the dials, but does it in a quantum way modelled by the Hadamard "coin-tossing" operator H.

It is worth pointing out an interesting difference between the probabilistic and quantum setting: spinning the dials with equal probability can be implemented in many different ways in the quantum world; e.g. a quantum gate different from the Hadamard gate.

The second example is a quantum generalisation of random walk that every one working on probabilistic algorithms or programming must be familiar with. Here is a simplified version (taken from [YYFD13]) of the one-dimensional quantum walks defined in [ABNVW01].

Example 2 (Quantum Walk) Let $\mathcal{H}_c$ be the 2-dimensional Hilbert space with orthonormal basis states $\mid L\rangle$ and $\mid R\rangle$, indicating directions Left and Right, respectively. It is the state space of a quantum coin. Let $\mathcal{H}_p$ be the n-dimensional Hilbert space with orthonormal basis states $\mid 0\rangle, \mid 1\rangle, \ldots, \mid n-1\rangle$, where vector $\mid i\rangle$ denotes position i for each $0 \leq i < n$. The state space of the walk is $\mathcal{H} = \mathcal{H}_c \otimes \mathcal{H}_p$. The initial state is $\mid L\rangle \mid 0\rangle$. Each step of the walk consists of:

1. Measure the position of the system to see whether it is 1. If the outcome is "yes", then the walk terminates, otherwise, it continues. The measurement is mathematically modelled by $M = \{M_{yes}, M_{no}\}$, where
$$M_{yes} = |1\rangle\langle 1|, \ M_{no} = I_p - M_{yes} = \sum_{i\neq 1} |i\rangle\langle i|$$
and I_p is the identity operator in the position space $\mathcal{H}_p$;
2. The Hadamard "coin-tossing" operator H is applied in the coin space $\mathcal{H}_c$;
3. The shift operator S defined by
$$S|L,i\rangle = |L, i\ominus 1\rangle, \qquad S|R,i\rangle = |R, i\oplus 1\rangle$$
for $i = 0, 1, \ldots, n-1$ is performed on the space $\mathcal{H}$. Intuitively, the system walks one step left or right according to the direction state. Here, $\oplus$ and $\ominus$ stand for addition and subtraction modulo n, respectively. The operator S can be equivalently written as
$$S = \sum_{i=0}^{n-1} |L\rangle\langle L| \otimes |i\ominus 1\rangle\langle i| + \sum_{i=0}^{n-1} |R\rangle\langle R| \otimes |i\oplus 1\rangle\langle i| .$$

This walk can be written as the quantum **while**-loop:

$$QW \equiv \ \ c := |L\rangle; \ p := |0\rangle; \ \textbf{while}\ M[p] = no\ \textbf{do}\ c := H[c]; \ \ c, p := S[c, p]\ \textbf{od}$$

It is worth noticing an essential difference between the quantum walk and a classical random walk: the coin (or direction) variable c can be in a superposition of $|L\rangle$ and $|R\rangle$ like $|+\rangle = \frac{1}{\sqrt{2}}(|L\rangle + |R\rangle)$, and thus the walker is moving left and right "simultaneously"; for example,
$$\frac{1}{\sqrt{2}}(|L\rangle + |R\rangle)|i\rangle \to \frac{1}{\sqrt{2}}(|L\rangle|i\ominus 1\rangle + |R\rangle|i\oplus 1\rangle).$$
This means that if the walker is currently at position i, then after one step she/he will be at both position $i\ominus 1$ and $i\oplus 1$.

Similar to the case of classical **while**-programs, the operational semantics of quantum programs can be defined as a transition relation between configurations. For each quantum program P, we write $var(P)$ for the set of quantum variables occurring in P. Let
$$\mathcal{H}_P = \bigotimes_{q\in var(P)} \mathcal{H}_q$$
be the state Hilbert space of P. A partial density operator is defined as a positive operator ρ with trace $tr(\rho) \leq 1$. We write $\mathcal{D}(\mathcal{H}_P)$ for the set of partial density operators in $\mathcal{H}_P$. A configuration is a pair $\langle P, \rho\rangle$, where P is a program or the termination symbol $\downarrow$, and $\rho \in \mathcal{D}(\mathcal{H}_P)$ denotes the state of quantum variables.

Definition 2 (*Operational Semantics* **[Yin11, Yin16]**) The operational semantics of quantum programs is defined by the transition rules in Fig. 1.

It is worth pointing out that the transition rules (In), (UT), (IF), (L0) and (L1) are decreed by the basic postulates of quantum mechanics. A major difference between the operational semantics of classical and quantum programs should be noticed. During the execution of a classical case statement or a **while**-loop, checking its guard does not change the state of program variables. In contrast, checking the guard of a quantum case statement or a quantum loop requires to perform a measurement on (the system denoted by) the program variables, which, again decreed by a basic postulate of quantum mechanics, changes the state of these variables. From the operational semantics, we see that the control flow of a quantum program is determined by the outcomes of the quantum measurements occurring in the case statements or loops. Note that the outcomes of quantum measurements are classical information, so the control flows of quantum programs considered in this paper are classical. Quantum programs with quantum control flows were studied in [AG05] and Chapters 6 and 7 of [Yin16].

$$
\begin{array}{ll}
\text{(Sk)} & \langle \mathbf{skip}, \rho\rangle \to \langle \downarrow, \rho\rangle \\
\text{(In)} & \langle q := |0\rangle, \rho\rangle \to \langle \downarrow, \rho_0^q\rangle \\
\text{(UT)} & \langle \overline{q} := U[\overline{q}], \rho\rangle \to \langle \downarrow, U\rho U^\dagger\rangle \\
\text{(SC)} & \dfrac{\langle P_1, \rho\rangle \to \langle P_1', \rho'\rangle}{\langle P_1; P_2, \rho\rangle \to \langle P_1'; P_2, \rho'\rangle} \\
\text{(IF)} & \langle \mathbf{if}\ (\square m \cdot M[\overline{q}] = m \to P_m)\ \mathbf{fi}, \rho\rangle \to \langle P_m, M_m \rho M_m^\dagger\rangle \\
\text{(L0)} & \langle \mathbf{while}\ M[\overline{q}] = 1\ \mathbf{do}\ P\ \mathbf{od}, \rho\rangle \to \langle \downarrow, M_0 \rho M_0^\dagger\rangle \\
\text{(L1)} & \langle \mathbf{while}\ M[\overline{q}] = 1\ \mathbf{do}\ P\ \mathbf{od}, \rho\rangle \to \langle P; \mathbf{while}\ M[\overline{q}] = 1\ \mathbf{do}\ P\ \mathbf{od}, M_1 \rho M_1^\dagger\rangle
\end{array}
$$

Fig. 1. Transition Rules for Quantum **while**-Programs. In (IN), $\rho_0^q = |\,0\rangle_q\langle 0\,| \otimes \rho \otimes |\,0\rangle_q\langle 0\,| + |\,0\rangle_q\langle 1\,| \otimes \rho \otimes |\,1\rangle_q\langle 0\,|$ if $type(q) = \mathbf{Bool}$ and $\rho_0^q = \sum_{n=-\infty}^{\infty} |\,0\rangle_q\langle n\,| \otimes \rho \otimes |\,n\rangle_q\langle 0\,|$ if $type(q) = \mathbf{Int}$. In (SC), we make the convention $\downarrow;\ P_2 = P_2$. In (IF), m ranges over every possible outcome of measurement $M = \{M_m\}$

Definition 3 (*Denotational Semantics* [Yin11, Yin16]) For any program P, its semantic function is the mapping:

$$[\![P]\!] : \mathcal{D}(\mathcal{H}_P) \to \mathcal{D}(\mathcal{H}_P)$$
$$[\![P]\!](\rho) = \sum \{\!|\ \rho' : \langle P, \rho\rangle \to^* \langle \downarrow, \rho'\rangle\ |\!\}$$

for every $\rho \in \mathcal{D}(\mathcal{H}_P)$, where $\to^*$ is the reflexive and transitive closure of $\to$, and $\{\!|\ \cdot\ |\!\}$ denotes a multi-set.

The defining equation of $[\![P]\!](\rho)$ deserves an explanation. It is clear from rules (IF), (L0) and (L1) that transition relation $\to$ is usually not deterministic; that is, $\{\!|\ \rho' : \langle P, \rho\rangle \to^* \langle \downarrow, \rho'\rangle\ |\!\}$ may have more than one element. This is different from the case of classical **while**-programs. The summation in the right-hand side of the equation comes from the essential linearity of quantum mechanics. It is proved that $[\![P]\!](\cdot)$ is a quantum operation (or a super-operator) (for the definition of quantum operation, see [NC00], Chapter 8 or [Yin16], Chapter 2).

To illustrate the above two definitions, let reconsider the three quantum dials in Example 1.

Example 3 (Semantics of Three Quantum Dials - A Continuation of Example 1) A state of probabilistic program $flip$ is a configuration of the slot machine, i.e. a mapping from dials to suits. The semantics of $flip$ is a function that maps each initial state to a uniform distribution of states in which every configurations has probability $\frac{1}{8}$.

In contrast, the state Hilbert space of quantum program $qflip$ is then $\mathcal{H}_2^{\otimes 3}$. For instance, if we write:

$$|\,+\rangle = \frac{1}{\sqrt{2}}(|\,\heartsuit\rangle + |\,\diamondsuit\rangle), \qquad |\,-\rangle = \frac{1}{\sqrt{2}}(|\,\heartsuit\rangle - |\,\diamondsuit\rangle)$$

for the equal superpositions of $|\,\heartsuit\rangle$ and $|\,\diamondsuit\rangle$, then:

$$[\![qflip]\!](|\,+, -, +\rangle) = |\,\heartsuit, \diamondsuit, \heartsuit\rangle;$$

if we write:

$$|\,W\rangle = \frac{1}{\sqrt{3}}(|\,\heartsuit, \heartsuit, \diamondsuit\rangle + |\,\heartsuit, \diamondsuit, \heartsuit\rangle + |\,\diamondsuit, \heartsuit, \heartsuit\rangle)$$

for the Werner state, a typical entangled state of three qubits, then

$$[\![qflip]\!](|\,W\rangle) = \frac{1}{2\sqrt{6}}(3\,|\,\heartsuit, \heartsuit, \heartsuit\rangle + |\,\heartsuit, \heartsuit, \diamondsuit\rangle + |\,\heartsuit, \diamondsuit, \heartsuit\rangle - |\,\heartsuit, \diamondsuit, \diamondsuit\rangle + |\,\diamondsuit, \heartsuit, \heartsuit\rangle - |\,\diamondsuit, \heartsuit, \diamondsuit\rangle - |\,\diamondsuit, \diamondsuit, \heartsuit\rangle - 3\,|\,\diamondsuit, \diamondsuit, \diamondsuit\rangle).$$

(Note that Definitions 2 and 3 are given in a more general way; i.e. in terms of density operators. Here, for simplicity, a pure state $|\,\psi\rangle$ is identified with the corresponding density operator $\rho = |\,\psi\rangle\langle\psi\,|$.)

The Löwner order between operators in the state Hilbert space $\mathcal{H}_P$ of a quantum program is defined as follows:

$$A \sqsubseteq B \text{ if } B - A \text{ is a positive operator.}$$

This order can be naturally lifted to an order between quantum operations (or super-operators). It can be shown that the set $\mathcal{D}(\mathcal{H}_P)$ of partial density operators in $\mathcal{H}_P$ with the Löwner order $\sqsubseteq$ is a complete partial

order (CPO), and all quantum operations in $\mathcal{H}_P$ form a CPO too. Then a fixed point characterisation for the denotational semantics of quantum **while**-loops can be derived (see Proposition 5.1(6) and Corollary 5.1 in [Yin11], or Proposition 3.3.2 and Corollary 3.3.1 in [Yin16]).

2.2. Partial correctness and total correctness

Now we review the notions of partial and total correctness for quantum programs. Recall from [MM05] that probabilistic predicates (or assertions, e.g. preconditions and postconditions) are defined to be bounded real-valued functions interpreted as the expectations of random variables. As introduced in [DP06], a quantum predicate in a Hilbert space $\mathcal{H}$ is an observable (a Hermitian operator) A in $\mathcal{H}$ with $0 \sqsubseteq A \sqsubseteq I$, where 0 and I are the zero operator and the identity operator in $\mathcal{H}$, respectively, and $\sqsubseteq$ stands for the Löwner order. In the quantum foundations literature, a quantum predicate is called an *effect*.

Definition 4 (*Correctness Formula, Hoare Triple* **[Yin11, Yin16]**) A correctness formula (or a Hoare triple) is a statement of the form

$$\{A\}P\{B\}$$

where P is a quantum program, and both A, B are quantum predicates in $\mathcal{H}_P$, called the precondition and postcondition, respectively.

The appearance of a quantum Hoare triple is exactly the same as that of its classical counterpart. But in a classical Hoare triple, precondition A and postcondition B stands for (first-order) logical formulas, and in a quantum Hoare triple, A and B are two operators that are interpreted as observables in physics.

Definition 5 (*Partial and Total Correctness* **[Yin11, Yin16]**) 1. The correctness formula $\{A\}P\{B\}$ is true in the sense of total correctness, written $\models_{tot} \{A\}P\{B\}$, if for all input states $\rho \in \mathcal{D}(\mathcal{H}_P)$ we have:

$$\mathrm{tr}(A\rho) \leq \mathrm{tr}(B[\![P]\!](\rho)). \tag{4}$$

2. The correctness formula $\{A\}P\{B\}$ is true in the sense of partial correctness, written $\models_{par} \{A\}P\{B\}$, if for all input states $\rho \in \mathcal{D}(\mathcal{H}_P)$ we have:

$$\mathrm{tr}(A\rho) \leq \mathrm{tr}(B[\![P]\!](\rho)) + [\mathrm{tr}(\rho) - \mathrm{tr}([\![P]\!](\rho))]. \tag{5}$$

A key to understanding the above definition is the observation that $tr(A\rho)$ is the expected (average) value of observable A when the system is in state ρ. The following lemma shows that the defining inequalities (4) and (5) for total and partial correctness can be restated in a form of Löwner order, which is often easier to manipulate because the universal quantifier over density operator ρ is eliminated.

Let us consider the three quantum dials in Examples 1 and 3 once again to illustrate the notion of correctness introduced in the above definition.

Example 4 (Correctness of Three Quantum Dials) We write:

$$|GHZ\rangle = \frac{1}{\sqrt{2}}(|\heartsuit,\heartsuit,\heartsuit\rangle + |\diamondsuit,\diamondsuit,\diamondsuit\rangle)$$

for the GHZ (Greenberger-Horne-Zeilinger) state, another typical entangled state of three qubits,

$$|\Phi\rangle = \frac{1}{2}(|\heartsuit,\heartsuit,\heartsuit\rangle + |\heartsuit,\diamondsuit,\diamondsuit\rangle + |\diamondsuit,\heartsuit,\diamondsuit\rangle + |\diamondsuit,\diamondsuit,\heartsuit\rangle),$$

and $|\Psi\rangle = |\heartsuit,\heartsuit,\heartsuit\rangle$. Then $A = |\Phi\rangle\langle\Phi|$, $B = |\Psi\rangle\langle\Psi|$ and $C = |GHZ\rangle\langle GHZ|$ are all quantum predicates. It is easy to check that

$$\models_{tot} \{A\}qflip\{C\}, \qquad \models_{tot} \{\frac{1}{4}B\}qflip\{C\}.$$

This means that if the input is state $|\Phi\rangle$, then program $qflip$ will certainly output the GHZ state; and if the input is state $|\Psi\rangle$, it will output a state $|\Gamma\rangle$ that is similar to the GHZ state in the sense:

$$\Pr(|\Gamma\rangle \text{ and the GHZ state cannot be discriminated}) \geq \frac{1}{4}.$$

Note that partial and total correctness are the same for $qflip$ because it does not contain any loop.

$$\text{(Ax.Sk)}\quad \{A\}\mathbf{Skip}\{A\}$$

$$\text{(Ax.In.B)}\quad \{|0\rangle_q\langle 0|A|0\rangle_q\langle 0| + |1\rangle_q\langle 0|A|0\rangle_q\langle 1|\}q := |0\rangle\{A\}$$

$$\text{(Ax.In.I)}\quad \left\{\sum_{n=-\infty}^{\infty} |n\rangle_q\langle 0|A|0\rangle_q\langle n|\right\} q := |0\rangle\{A\}$$

$$\text{(Ax.UT)}\quad \{U^\dagger AU\}\overline{q} := U\,[\overline{q}]\,\{A\}$$

$$\text{(R.SC)}\quad \frac{\{A\}P_1\{B\}\qquad \{B\}P_2\{C\}}{\{A\}P_1; P_2\{C\}}$$

$$\text{(R.IF)}\quad \frac{\{A_m\}P_m\{B\} \text{ for all } m}{\left\{\sum_m M_m^\dagger A_m M_m\right\}\mathbf{if}\ (\square m \cdot M[\overline{q}] = m \to P_m)\ \mathbf{fi}\{B\}}$$

$$\text{(R.LP)}\quad \frac{\{B\}P\left\{M_0^\dagger AM_0 + M_1^\dagger BM_1\right\}}{\{M_0^\dagger AM_0 + M_1^\dagger BM_1\}\mathbf{while}\ M[\overline{q}] = 1\ \mathbf{do}\ P\ \mathbf{od}\{A\}}$$

$$\text{(R.Or)}\quad \frac{A \sqsubseteq A'\quad \{A'\}P\{B'\}\quad B' \sqsubseteq B}{\{A\}P\{B\}}$$

Fig. 2. Proof System qPD for Quantum **while**-Programs. In (Ax.In.B), $type(q) = \mathbf{Boolean}$. In (Ax.In.I), $type(q) = \mathbf{Int}$

$$\text{(R.LT)}\quad \frac{\begin{array}{l}\bullet\ \{Q\}S\{M_0^\dagger PM_0 + M_1^\dagger QM_1\}\\ \bullet\ \text{for any } \epsilon > 0,\ t_\epsilon \text{ is a } (M_1^\dagger QM_1, \epsilon)\text{−ranking function of loop } \mathbf{while}\ M[\overline{q}] = 1\ \mathbf{do}\ S\ \mathbf{od}\end{array}}{\{M_0^\dagger PM_0 + M_1^\dagger QM_1\}\mathbf{while}\ M[\overline{q}] = 1\ \mathbf{do}\ S\ \mathbf{od}\{P\}}$$

Fig. 3. Proof System qTD for Quantum **while**-Programs

The quantum predicates A, B, C in the above example are very simple and defined by a particular input/output state. More examples of correctness specifications will be given in Sect. 3.

Lemma 1 1. $\models_{tot} \{A\}P\{B\}$ if and only if $A \sqsubseteq [\![P]\!]^*(B)$.

2. $\models_{par} \{A\}P\{B\}$ if and only if:

$$A \sqsubseteq [\![P]\!]^*(B) + (I - [\![P]\!]^*(I))$$

where I is the identity operator in $\mathcal{H}_P$, and $[\![P]\!]^*$ stands for the dual of super-operator $[\![P]\!]$ (see Definition 13 in Appendix A).

Proof Immediate from Lemma 4.1.2 in [Yin16] and Lemma 3 in Appendix A. □

The above lemma will be extensively used in the proofs of the results in Sect. 5.

2.3. Quantum Hoare logic

A Hoare-like logic for quantum programs is established in [Yin11]. It consists of:

- A proof system qPD for partial correctness: The axioms and inference rules of qPD are presented in Fig. 2.
- A proof system qTD for total correctness: qTD is obtained from qPD by replacing rule (R.LP) with the rule (R.LT) given in Fig. 3. The notion of ranking function used in rule (R.LT) is defined in the following:

Definition 6 ([Yin11, Yin16]) Let A be a quantum predicate in $\mathcal{H}_{var(P)\cup\overline{q}}$ and real number $\epsilon > 0$. A function

$$t : \mathcal{D}(\mathcal{H}_{var(P)\cup\overline{q}}) \to \mathbb{N}\ \text{(nonnegative integers)}$$

is called a (A, ϵ)-ranking function of quantum loop "$\mathbf{while}\ M[\overline{q}] = 1\ \mathbf{do}\ P\ \mathbf{od}$" if it satisfies the following two conditions: for all $\rho \in \mathcal{D}(\mathcal{H}_{var(P)\cup\overline{q}})$,

1. $t\left([\![S]\!]\left(M_1\rho M_1^\dagger\right)\right) \le t(\rho)$; and
2. $tr(P\rho) \ge \epsilon$ implies $t\left([\![S]\!]\left(M_1\rho M_1^\dagger\right)\right) < t(\rho)$.

It is interesting to carefully compare the Hoare rule for partial correctness of classical loops:

$$\frac{\{\varphi \wedge b\}P\{\varphi\}}{\{\varphi\}\mathbf{while}\ b\ \mathbf{do}\ P\ \mathbf{od}\{\varphi \wedge \neg b\}}$$

and its variant for total correctness with the corresponding rules (R.LP) and (R.LT) for quantum loops, respectively. At the first glance, the rules for classical loops and those for quantum loops are very different. But actually there is a certain similarity between them: in the rules for classical loops, $\varphi \wedge b$ (resp. $\varphi \wedge \neg b$) can be thought of as the restriction of φ under b (resp. $\neg b$). If we set:

$$D = M_0^\dagger A M_0 + M_1^\dagger B M_1,$$

then B can be seen as the restriction of D under M_1 and A the restriction of D under M_0, and M_0 can be considered as the negation of M_1. Furthermore, as we will see at the end of Sect. 7, rules (R.LP) and (R.LT) can also be understood in the context of inductive assertion.

The soundness and (relative) completeness of both proof system qPD and qTD were proved in [Yin11, Yin16].

Theorem 1 (Soundness and Completeness **[Yin11, Yin16])** For any quantum **while**-program P, and for any quantum predicates A, B,

$$\models_{par} \{A\}P\{B\} \Leftrightarrow \vdash_{qPD} \{A\}P\{B\},$$
$$\models_{tot} \{A\}P\{B\} \Leftrightarrow \vdash_{qTD} \{A\}P\{B\}.$$

3. Examples of quantum correctness specification

In this section, let us see several simple examples, which shows some interesting and tricky differences between specifying correctness of classical programs and that of quantum programs.

3.1. Unary quantum predicates

A quantum predicate about a single quantum system q; i.e. an observable (between the zero and identity operators) in state Hilbert space $\mathcal{H}_q$ can be understood as a unary quantum predicate.

Example 5 An equality like $x = c$ with x being a variable and c a constant often appears in precondition or postcondition of a classical program correctness formula, e.g.

$$\models_{tot} \{x = 1\}x := x + 1\{x = 2\}$$

Let q be a quantum variable and $|\psi\rangle$ a given (constant) state in $\mathcal{H}_q$. Then equality "$q = |\psi\rangle$" can be expressed as quantum predicate $A = |\psi\rangle\langle\psi|$ in $\mathcal{H}_q$, and we have:

$$\models_{tot} \{U^\dagger |\psi\rangle\langle\psi| U\}\ q := U[q]\ \{A\}, \tag{6}$$
$$\models_{tot} \{A\}\ q := U[q]\ \{U|\psi\rangle\langle\psi|U^\dagger\} \tag{7}$$

for any unitary operator U in $\mathcal{H}_q$. The precondition in (6) and the postcondition in (7) express equalities "$q = U^\dagger|\psi\rangle$", "$q = U|\psi\rangle$", respectively. More generally, for any closed subspace X of $\mathcal{H}_q$, Membership "$q \in X$" can be expressed as quantum predicate P_X (the projection onto X) and we have:

$$\models_{tot} \{P_{U^\dagger(X)}\}\ q := U[q]\ \{P_X\}, \tag{8}$$
$$\models_{tot} \{P_X\}\ q := U[q]\ \{P_{U(X)}\} \tag{9}$$

The precondition in (8) and the postcondition in (9) express memberships "$q \in U^\dagger(X)$", "$q \in U(X)$", respectively.

3.2. Quantum relations

A quantum predicate about multiple quantum systems $q_1, \ldots, q_n$; i.e. an observable (between the zero and identity operators) in the tensor product:

$$\bigotimes_{i=1}^{n} \mathcal{H}_{q_i},$$

can be understood as a quantum relation among $q_1, \ldots, q_n$. In particular, a quantum predicate A in $\mathcal{H}_1 \otimes \mathcal{H}_2$ can be seen as a quantum (binary) relation between $\mathcal{H}_1$ and $\mathcal{H}_2$. The following are two examples showing some quantum relations used in preconditions and postconditions. A further discussion about quantum relation is given in Appendix B.

Example 6 An equality like $x = y$ between two variables x, y also often appears in precondition or postcondition of a classical program correctness formula, e.g.

$$\models_{tot} \{x = y\}\ x := x + 1;\ y := y + 1\ \{x = y\}$$

Let p, q be two quantum variables with the same state Hilbert space $\mathcal{H}$. Then equality "$p = q$" can be expressed as the symmetrisation operator:

$$S_+ = \frac{1}{2}(I + \mathrm{SWAP})$$

where I is the identity operator in $\mathcal{H} \otimes \mathcal{H}$, and operator SWAP in $\mathcal{H} \otimes \mathcal{H}$ is defined by

$$\mathrm{SWAP} \mid \varphi, \psi\rangle = \mid \psi, \varphi\rangle$$

for all $\mid \varphi\rangle, \mid \psi\rangle \in \mathcal{H}$, together with linearity. It is easy to see that $\mathrm{SWAP} \cdot S_+ = S_+ \cdot \mathrm{SWAP} = \mathrm{SWAP} \cdot S_+ \cdot \mathrm{SWAP} = S_+$. Furthermore, for any unitary operator U in $\mathcal{H}$, it holds that:

$$\models_{tot} \{S_+\}\ p := U[p];\ q := U[q]\ \{S_+\}$$

It is interesting to note that a similar conclusion holds for the anti-symmetrisation operator:

$$S_- = \frac{1}{2}(I + \mathrm{SWAP})$$

that is, we have $\mathrm{SWAP} \cdot S_- = S_- \cdot \mathrm{SWAP} = -S_-$, $\mathrm{SWAP} \cdot S_- \cdot \mathrm{SWAP} = S_-$ and

$$\models_{tot} \{S_-\}\ p := U[p];\ q := U[q]\ \{S_-\}$$

Example 7 Consider the following correctness formula for classical programs:

$$\models_{tot} \{X = x \wedge Y = y\}\ R := X;\ X := Y;\ Y := R\ \{X = y \wedge Y = x\}$$

It means that after executing the program, the values of variables X and Y are exchanged. To properly specify the precondition and postcondition, auxiliary (ghost) variables x, y are introduced. In quantum computing, statements $R := X$, $X := Y$ and $Y := R$ cannot be directly realised as prohibited by the no-cloning theorem. But we can consider the following modification: let X, Y, R be three quantum variables with the same state Hilbert space $\mathcal{H}$. Then we have:

$$\models_{tot} \{A\}R, X := \mathrm{SWAP}[R, X];\ Y, Y := \mathrm{SWAP}[X, Y];\ Y, R := \mathrm{SWAP}[Y, R]\ \{(\mathrm{SWAP} \otimes I)A(\mathrm{SWAP} \otimes I)\} \quad (10)$$

for any quantum predicate A in $\mathcal{H} \otimes \mathcal{H} \otimes \mathcal{H}$, where the first, second and third $\mathcal{H}$ are the state Hilbert spaces of X, Y, R respectively, and I is the identity operator in $\mathcal{H}$. Note that we do not use any auxiliary (ghost) variable in correctness formula (10). In particular, if

$$A = \mid \varphi\rangle\langle\varphi \mid \otimes \mid \psi\rangle\langle\psi \mid \otimes I,$$

where $\mid \varphi\rangle, \mid \psi\rangle \in \mathcal{H}$, and I is the identity operator in $\mathcal{H}$, then it holds that

$$\begin{aligned} \models_{tot} \{\mid \varphi\rangle\langle\varphi \mid \otimes \mid \psi\rangle\langle\psi \mid \otimes I\}R, X := \mathrm{SWAP}[R, X];\ Y, Y := \mathrm{SWAP}[X, Y];\\ Y, R := \mathrm{SWAP}[Y, R]\ \{\mid \psi\rangle\langle\psi \mid \otimes \mid \varphi\rangle\langle\varphi \mid \otimes I\} \end{aligned} \quad (11)$$

Intuitively, the precondition and postcondition of (11) means "$X = \mid \varphi\rangle$ and $Y = \mid \psi\rangle$", "$X = \mid \psi\rangle$ and $Y = \mid \varphi\rangle$", respectively. Here, auxiliary variables $\mid \varphi\rangle, \mid \psi\rangle$ are employed.

All of the examples given in the previous subsection and this subsection can be proved using axiom (Ax.UT) and rule (R.SC) in Fig. 2. A more interesting example using rule (R.IF) will be presented in Sect. 4.

3.3. Logical connectives

Logical connectives are widely used in specifying (the preconditions and postconditions of) classical programs. Here, we show that several quantum counterparts of them can be used in specifying correctness of quantum programs. However, it is not the case that every logical connective has an appropriate quantum counterpart.

Example 8 The predicate "True" and "False" are described by the identity and zero operators I and 0, respectively, in the state Hilbert space $\mathcal{H}_P$ of the program P under consideration.

- $\models_{par} \{I\}P\{B\}$ means that for any density operator ρ,
$$tr(B[\![P]\!](\rho)) = tr([\![P]\!](\rho));$$
that is, the probability that postcondition B is satisfied by the output is equal to the probability that the program terminates.
- $\models_{par} \{A\}P\{I\}$ always holds, because $A \sqsubseteq I$ implies that for any partial density operator ρ,
$$tr(A\rho) \leq tr(\rho) = tr([\![P]\!](\rho)) + [tr(\rho) - tr([\![P]\!](\rho))].$$
- $\models_{tot} \{I\}P\{B\}$ means that for any density operator ρ,
$$tr(B[\![P]\!](\rho)) = 1.$$
Since $B \sqsubseteq I$, we have $tr(B[\![P]\!](\rho)) \leq tr([\![P]\!](\rho))$ and thus $tr([\![P]\!](\rho)) = 1$; that is, program P always terminates, and output $[\![P]\!](\rho)$ satisfies postcondition B.
- $\models_{tot} \{A\}P\{I\}$ means that for any density operator, $tr(A\rho) \leq tr([\![P]\!](\rho))$; intuitively, if precondition A is satisfied by input ρ to P, then P terminates.

Example 9 1. Negation in classical logic has a quantum counterpart: for each quantum predicate in Hilbert space $\mathcal{H}$, $I - A$ can be used as the negation of A. In fact, for any density operator ρ in $\mathcal{H}$,
$$tr[(I - A)\rho] = 1 - tr(A\rho).$$

2. Conjunction in classical logic does not always have an appropriate quantum counterpart; some difficulties caused by this fact in reasoning about quantum programs were discussed in [YCFD07]. But if A_1 and A_2 are quantum predicates in $\mathcal{H}_1$, $\mathcal{H}_2$, respectively, then $A_1 \otimes A_2$ can be used as the conjunction (in $\mathcal{H}_1 \otimes \mathcal{H}_2$) of A_1 and A_2 because for any density operator ρ_1 in $\mathcal{H}_1$ and ρ_2 in $\mathcal{H}_2$, it holds that
$$tr((A_1 \otimes A_2)(\rho_1 \otimes \rho_2)) = tr(A_1\rho_1) \cdot tr(A_2\rho_2);$$
for example, see Eq. (11).
3. For a family $\{A_i\}$of quantum predicates in the same Hilbert space $\mathcal{H}$, and a probability distribution $\{p_i\}$, we often use the convex combination $\sum_i p_i A_i$ as a connective; for example, see rules (R.CC) and (R.Inv) in Fig. 7.
4. In a sense, quantum predicate $\sum_m M_m^\dagger A_m M_m$ in rule (R.IF) in Fig. 2 can be understood as the disjunction of (the conjunction of) "the outcome of measurement M is m" and quantum predicate A_m.

4. Proof outlines

The notion of proof outline can be introduced to structure a correctness proof of a classical **while**-program, according to the structure of the program itself, so that the proof is easier to follow (see [ABO09], Section 3.4). It is also a basis for defining non-interference in reasoning about parallel programs with shared variables. In this section, we generalise this notion to the quantum case.

Definition 7 Let S be a quantum **while**-programs.

1. A proof outline for partial correctness of S is a formula $\{A\}P^*\{B\}$ formed by the formation axioms and rules in Fig. 4, where P^* results from P by interspersing quantum predicates.
2. A proof outline for total correctness of S is defined by introducing ranking function into rule (R.LP') (see Definition 6 and rule (R.LT)).

Let us consider a simple example to illustrate the above definition.

$$\text{(Ax.Sk}') \quad \{A\}\mathbf{Skip}\{A\}$$

$$\text{(Ax.In.B}') \quad \{|0\rangle_q\langle 0|A|0\rangle_q\langle 0| + |1\rangle_q\langle 0|A|0\rangle_q\langle 1|\}q := |0\rangle\{A\}$$

$$\text{(Ax.In.I}') \quad \left\{\sum_{n=-\infty}^{\infty} |n\rangle_q\langle 0|A|0\rangle_q\langle n|\right\} q := |0\rangle\{A\}$$

$$\text{(Ax.UT}') \quad \{U^\dagger AU\}\overline{q} := U\,[\overline{q}]\,\{A\}$$

$$\text{(R.SC}') \quad \frac{\{A\}P_1^*\{B\} \qquad \{B\}P_2^*\{C\}}{\{A\}P_1^*; \{B\}P_2^*\{C\}}$$

$$\text{(R.IF}') \quad \frac{\{A_{m_i}\}P_{m_i}^*\{B\}\ (i=1,...,k)}{\left\{\sum_{i=1}^k M_{m_i}^\dagger A_{m_i} M_{m_i}\right\}\ \mathbf{if}\ M[\overline{q}] = m_1 \to \{A_{m_1}\}\,P_{m_1}^* \ \square\ \cdots\cdots\cdots\cdots\ \square\ M[\overline{q}] = m_k \to \{A_{m_k}\}\,P_{m_k}^*\ \mathbf{fi}\ \{B\}}$$

$$\text{(R.LP}') \quad \frac{\{B\}P^*\left\{M_0^\dagger AM_0 + M_1^\dagger BM_1\right\}}{\{M_0^\dagger AM_0 + M_1^\dagger BM_1\}\ \mathbf{while}\ M[\overline{q}] = 1\ \mathbf{do}\ \{B\}\ P^*\ \left\{M_0^\dagger AM_0 + M_1^\dagger BM_1\right\}\ \mathbf{od}\ \{A\}}$$

$$\text{(R.Or}') \quad \frac{A \sqsubseteq A' \quad \{A'\}P^*\{B'\} \quad B' \sqsubseteq B}{\{A\}\{A'\}P\{B'\}\{B\}}$$

$$\text{(R.Del)} \quad \frac{\{A\}P^*\{B\}}{\{A\}P^{**}\{B\}}$$

Fig. 4. Formation Axioms and Rules for Partial Correctness of Quantum **while**-Programs. In (R.IF'), $\{m_1, \ldots, m_k\}$ is the set of all possible outcomes of measurement M. In (R.Del), S^{**} is obtained by deleting some quantum predicates from S^*

Example 10 (Teleportation) Quantum teleportation is a protocol that can send quantum states only using a classical communication channel. Suppose that Alice possesses two qubits p, q and Bob possesses qubit r, and there is entanglement, i.e. the EPR (Einstein–Podolsky–Rosen) pair:

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

between q and r. Then Alice can send a quantum state $|\,\psi\rangle = \alpha\,|\,0\rangle + \beta\,|\,1\rangle$ to Bob, i.e. from p to r, by two-bit classical communication (for detailed description, see [NC00], Section 1.3.7). This protocol can be written as quantum program QTEL in Fig. 5. The correctness of QTEL can be described as the Hoare triple:

$$\left\{|\,\psi\rangle_p\langle\psi\,| \otimes |\,\beta_{00}\rangle_{q,r}\langle\beta_{00}\,|\right\}\ \text{QTEL}\ \left\{I_p \otimes I_q \otimes |\,\psi\rangle_r\langle\psi\,|\right\}.$$

A proof outline for the correctness of QTEL is also presented in Fig. 5.

We can define the notion of subprogram of a quantum program in a familiar way; namely by induction on the length of the program. Then a special form of proof outline is defined in the following:

Definition 8 A proof outline $\{A\}P^*\{B\}$ of P is called standard if every subprogram Q of P is proceeded by exactly one quantum predicate, denoted $pre(Q)$, in P^*.

The next proposition shows that standard proof outlines are indeed sufficient for reasoning about correctness of quantum programs.

Proposition 1 1. If $\{A\}P^*\{B\}$ is a proof outline for partial correctness (respectively, total correctness), then $\vdash_{qPD} \{A\}P\{B\}$ (respectively, $\vdash_{qTD} \{A\}P\{B\}$).

2. If $\vdash_{qPD} \{A\}P\{B\}$ (respectively, $\vdash_{qTD} \{A\}P\{B\}$), then there is a standard proof outline $\{A\}P^*\{B\}$ for partial correctness (respectively, total correctness).

$$
\begin{aligned}
&\{|\psi\rangle_p\langle\psi| \otimes |\beta_{00}\rangle_{q,r}\langle\beta_{00}|\}\\
&\sqsubseteq \{|\beta_{00}\rangle_{p,q}\langle\beta_{00}| \otimes |\psi\rangle_r\langle\psi| + |\beta_{10}\rangle_{p,q}\langle\beta_{10}| \otimes |\psi_1\rangle_r\langle\psi_1|\\
&\qquad\qquad + |\beta_{01}\rangle_{p,q}\langle\beta_{01}| \otimes |\psi_2\rangle_r\langle\psi_2| + |\beta_{11}\rangle_{p,q}\langle\beta_{11}| \otimes |\psi_3\rangle_r\langle\psi_3|\}\\
\mathrm{QTEL} \equiv\ & p, q := \mathrm{CNOT}[p, q];\\
&\{|+\rangle_p\langle+| \otimes |0\rangle_q\langle 0| \otimes |\psi\rangle_r\langle\psi| + |-\rangle_p\langle-| \otimes |0\rangle_q\langle 0| \otimes |\psi_1\rangle_r\langle\psi_1|\\
&\qquad\qquad + |+\rangle_p\langle+| \otimes |1\rangle_q\langle 1| \otimes |\psi_2\rangle_r\langle\psi_2| + |-\rangle_p\langle-| \otimes |1\rangle_q\langle 1| \otimes |\psi_3\rangle_r\langle\psi_3|\}\\
&p := H[p];\\
&\{|0\rangle_p\langle 0| \otimes |0\rangle_q\langle 0| \otimes |\psi\rangle_r\langle\psi| + |1\rangle_p\langle 1| \otimes |0\rangle_q\langle 0| \otimes |\psi_1\rangle_r\langle\psi_1|\\
&\qquad\qquad + |0\rangle_p\langle 0| \otimes |1\rangle_q\langle 1| \otimes |\psi_2\rangle_r\langle\psi_2| + |1\rangle_p\langle 1| \otimes |1\rangle_q\langle 1| \otimes |\psi_3\rangle_r\langle\psi_3|\}\\
&\textbf{if } M[q] = 0 \to \{|0\rangle_p\langle 0| \otimes I_q \otimes |\psi\rangle_r\langle\psi| + |1\rangle_p\langle 1| \otimes I_q \otimes |\psi_1\rangle_r\langle\psi_1|\}\ \textbf{skip}\\
&\square \qquad 1 \to \{|0\rangle_p\langle 0| \otimes I_q \otimes |\psi_2\rangle_r\langle\psi_2| + |1\rangle_p\langle 1| \otimes I_q \otimes |\psi_3\rangle_r\langle\psi_3|\}\ r := X[r]\\
&\textbf{fi};\\
&\{|0\rangle_p\langle 0| \otimes I_q \otimes |\psi\rangle_r\langle\psi| + |1\rangle_p\langle 1| \otimes I_q \otimes |\psi_1\rangle_r\langle\psi_1|\}\\
&\textbf{if } M[p] = 0 \to \{I_p \otimes I_q \otimes |\psi\rangle_r\langle\psi|\}\ \textbf{skip}\\
&\square \qquad 1 \to \{I_p \otimes I_q \otimes |\psi_1\rangle_r\langle\psi_1|\}\ r := Z[r]\\
&\textbf{fi}\ \{I_p \otimes I_q \otimes |\psi\rangle_r\langle\psi|\}
\end{aligned}
$$

Fig. 5. Quantum teleportation program and correctness proof outline. (1) Quantum states: $|\psi_1\rangle = \alpha\,|0\rangle - \beta\,|1\rangle$, $|\psi_2\rangle = \beta\,|0\rangle + \alpha\,|1\rangle$, $|\psi_3\rangle = -\beta\,|0\rangle + \alpha\,|1\rangle$. (2) Entangled states: $|\beta_{01}\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$, $|\beta_{10}\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle)$, $|\beta_{01}\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$. (3) Measurement in the computational basis: $M = \{M_0, M_1\}$, where $M_0 = |0\rangle\langle 0|$, $M_1 = |1\rangle\langle 1|$

Proof Induction on the lengths of proof and formation. □

As in classical programming, another usage of proof outline is that it enables us to establish a strong soundness of quantum Hoare logic. To this end, we need the following:

Definition 9 Let P be a quantum **while**-program and T a subprogram of P. Then $at(T, P)$ is inductively defined as follows:

1. If $T \equiv P$, then $at(T, P) \equiv P$;
2. If $P \equiv P_1;\ P_2$, then $at(T, P) \equiv at(T, P_1);\ P_2$ when T is a subprogram of P_1, and $at(T, P) \equiv at(T, P_2)$ when T is a subprogram of P_2;
3. If $P \equiv \textbf{if } (\square m \cdot M[\overline{q}] = m \to P_m)\ \textbf{fi}$, then for each m, whenever T is a subprogram of P_m, $at(T, P) \equiv at(T, P_m)$;
4. If $P \equiv \textbf{while } M[\overline{q}] = 1\ \textbf{do } P'\ \textbf{od}$ and T is a subprogram of P', then $at(T, P) \equiv at(T, P');\ P$.

Intuitively, $at(T, P)$ denotes the remainder of P to be executed when the control is at subprogram T.

Definitions 7–9 are straightforward generalisations of the corresponding concepts for classical programs (see for example [ABO09], Section 3.4). However, the strong soundness (Theorem 3.3 in [ABO09]) cannot be straightforwardly generalised to the quantum case. To present the strong soundness for quantum programs, we also need to extend the transition relation between configurations given in Definition 2 to a transition relation between configuration ensembles. We define a configuration ensemble as a multi-set $\{|\ \langle P_i, \rho_i\rangle\ |\}$ of configurations with

$$\sum_i tr(\rho_i) \leq 1.$$

For simplicity, we identify a singleton $\{|\ \langle P, \rho\rangle\ |\}$ with the configuration $\langle P, \rho\rangle$.

Definition 10 The transition relation between configuration ensembles is of the form:

$$\{|\ \langle P_i, \rho_i\rangle\ |\} \to \{|\ \langle Q_j, \sigma_j\rangle\ |\} \tag{12}$$

and defined by rules (Sk), (In), (UT), (SC) in Fig. 1 together with the rules presented in Fig. 6.

Now we are ready to present the strong soundness of the quantum Hoare logic.

$$\text{(IF}') \quad \langle \textbf{if } (\square m \cdot M[\overline{q}] = m \to P_m) \textbf{ fi}, \rho\rangle \to \{|\langle P_m, M_m \rho M_m^\dagger\rangle|\}$$

$$\text{(L}') \quad \langle \textbf{while } M[\overline{q}] = 1 \textbf{ do } P \textbf{ od}, \rho\rangle \to \{|\langle \downarrow, M_0 \rho M_0^\dagger\rangle, \langle P; \textbf{while } M[\overline{q}] = 1 \textbf{ do } P \textbf{ od}, M_1 \rho M_1^\dagger\rangle|\}$$

$$\text{(MS)} \quad \frac{\langle P, \rho\rangle \in \mathcal{A} \quad \langle P, \rho\rangle \to \mathcal{B}}{\mathcal{A} \to (\mathcal{A} \setminus \{|\langle P, \rho\rangle|\}) \cup \mathcal{B}}$$

Fig. 6. Extended transition rules for quantum **while**-programs. In (MU), $\mathcal{A}$, $\mathcal{B}$ are configuration ensembles

Theorem 2 (Strong Soundness) Let $\{A\}P^*\{B\}$ be a standard proof outline for partial correctness. If

$$\langle P, \rho\rangle \to^* \{|\ \langle P_i, \rho_i\rangle\ |\},$$

then:

1. for each i, $P_i \equiv at(T_i, P)$ for some subprogram T_i of P or $P_i \equiv \downarrow$; and
2. $tr(A\rho) \leq \sum_i tr\,(B_i \rho_i)$, where

$$B_i = \begin{cases} B & \text{if } P_i \equiv \downarrow, \\ pre\,(T_i) & \text{if } P_i \equiv at\,(T_i, P)\,. \end{cases}$$

Proof See Appendix C.1. □

A difference between the above theorem and the strong soundness for classical **while**-programs (see [ABO09], Theorem 3.3) should be noticed. The summation in the right-hand side of the inequality in clause 2 of the above theorem indicates that we have to consider all the reached programs P_i collectively in the quantum case. The reason behind it is that certain nondeterminism is introduced in the operational semantics of case statements and **while**-loops in quantum computation.

5. Auxiliary axioms and rules

Several auxiliary axioms and rules introduced in [Gor75, Har79] (see also [ABO09], Section 3.8) are very useful for simplifying the presentation of correctness proofs of classical programs. In this section, we generalise some of them into the quantum case. For this purpose, we first introduce several notations. Let $X \subseteq Y \subseteq Var$, and let A be an operator in

$$\mathcal{H}_X = \bigotimes_{q \in X} \mathcal{H}_q.$$

Then operator:

$$cl_Y(A) = A \otimes I_{\mathcal{H}_{Y \setminus X}}$$

is called the cylindric extension of A in $\mathcal{H}_Y$. If $X,\ Y \subseteq Var$ and $X \cap Y = \emptyset$. Then partial trace tr_Y is a mapping from operators in $\mathcal{H}_{X \cup Y}$ to operators in $\mathcal{H}_X$ defined by

$$tr_Y(|\ \varphi\rangle\langle\psi\ | \otimes |\ \varphi'\rangle\langle\psi'\ |) = \langle\psi'\ |\ \varphi'\rangle \cdot |\ \varphi\rangle\langle\psi\ |$$

for every $|\ \varphi\rangle, |\ \psi\rangle$ in $\mathcal{H}_X$ and $|\ \varphi'\rangle, |\ \psi'\rangle$ in $\mathcal{H}_Y$.

Now we can present the auxiliary axioms and rules for quantum programs in Fig. 7.

It is interesting to compare our auxiliary axioms and rules for quantum programs with those for classical programs:

- The appearance of axiom (Ax.Inv) is the same as axiom (INVARIANCE) in Section 3.8 of [ABO09].
- Obviously, rule (R.SO) is quantum generalisations of rule (SUBSTITUTION) in [ABO09], with the substitution $\overline{z} := \overline{t}$ being replaced by an arbitrary super-operator $\mathcal{E}$.
- It is easy to see that rules (R.CC) and (R.Inv) are generalisations of rules (CONJUNCTION) and (INVARIANCE), respectively, in [ABO09], with logical conjunction being replaced by a probabilistic (convex) combination.
- It is more interesting to note that rule (R.TI) is a quantum generalisation of two rules (DISJUNCTION) and (∃-INTRODUCTION) in [ABO09], where partial trace is considered as a quantum counterpart of logical disjunction and existence quantifier.

$$\text{(Ax.Inv)} \quad \{A\}\, P\, \{A\}$$

$$\text{(R.TI)} \quad \frac{\{A\} P \{B\}}{\{tr_W A\}\, P\, \{B\}}$$

$$\text{(R.CC)} \quad \frac{\{A_i\}\, P\, \{B_i\} \;\; (i = 1, ..., m)}{\{\sum_{i=1}^m p_i A_i\}\, P\, \{\sum_{i=1}^m p_i B_i\}}$$

$$\text{(R.Inv)} \quad \frac{\{A\} P \{B\}}{\{pA + qC\} P \{pB + qC\}}$$

$$\text{(R.SO)} \quad \frac{\{A\} P \{B\}}{\{\mathcal{E}^*(A)\}\, P\, \{\mathcal{E}^*(B)\}}$$

$$\text{(R.Lim)} \quad \frac{\lim_{n\to\infty} A_n = A \quad \{A_n\}\, P\, \{B_n\} \quad \lim_{n\to\infty} B_n = B}{\{A\} P \{B\}}$$

Fig. 7. Auxiliary axioms and rules. In (Ax.Inv), $var(P) \cap V = \emptyset$ and $A = cl_{V \cup var(P)}(B)$ for some $V \subseteq Var$ and for some quantum predicate B in $\mathcal{H}_V$. In (R.TI), $V, W \subseteq Var$, $V \cap W = \emptyset$, A, B are quantum predicates in $\mathcal{H}_{V \cup W}$ and $\mathcal{H}_V$, respectively, and $var(P) \subseteq V$. In (R.CC), $p_i \geq 0$ $(i = 1, \ldots, m)$ and $\sum_{i=1}^m p_j \leq 1$. In (R.Inv), $p, q \geq 0$, $p + q \leq 1$, and C is a quantum predicate in $\mathcal{H}_V$ for some $V \subseteq Var$ with $V \cap var(P) = \emptyset$. In (R.SO), $\mathcal{E}$ is a quantum operation (or super-operator) in $\mathcal{H}_V$ for some $V \subseteq Var$ with $V \cap var(P) = \emptyset$. In (R.Lim), $\{A_n\}$ and $\{B_n\}$ are increasing and decreasing, respectively sequences with respect to the Löwner order

The following theorem establishes soundness of the auxiliary axioms and rules in Fig. 7.

Theorem 3 1. The axiom (Ax.Inv) is sound for partial correctness.
2. The rules (R.TI), (R.CC), (R.Inv), (R.SO) and (R.Lim) are sound both for partial and total correctness.

Proof See Appendix C.2 □

Before concluding this section, let us see a simple example to show how the auxiliary axioms and rules can be used to derive some new properties of a quantum program.

Example 11 Let q be a qubit variable. Then by (Ax.UT) in Fig. 3 we obtain:

$$\{|-\rangle\langle -|\}\, q := H[q]\, \{|1\rangle\langle 1|\} \tag{13}$$

where H is the Hadamard gate and $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. The rule (R.SO) in Fig. 7 is especially useful for reasoning about quantum computation with noise. The amplitude damping channel is an important type of quantum noise where energy is lost from a quantum system with a certain probability γ. It can be modelled by quantum operation:

$$\mathcal{E}(\rho) = E_0 \rho E_0 + E_1 \rho E_1$$

for any density operator ρ, where

$$E_0 = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-\gamma} \end{pmatrix}, \qquad E_1 = \begin{pmatrix} 1 & \sqrt{\gamma} \\ 0 & 0 \end{pmatrix}.$$

Then applying (R.SO) to (13) yields:

$$\left\{ \frac{1+\gamma}{2} |0\rangle\langle 0| - \frac{\sqrt{1-\gamma}}{2}(|0\rangle\langle 1| + |1\rangle\langle 0|) + \frac{1-\gamma}{2} |1\rangle\langle 1| \right\}\, q := H[q]\, \{\gamma |0\rangle\langle 0| + (1-\gamma)|1\rangle\langle 1|\}.$$

6. Mechanising quantum program verification

Proving correctness of classical programs using Hoare logic is very fiddly and tedious. As we saw from the example given in Fig. 5, it is even worse to prove correctness of quantum programs because a huge amount of vector and matrix calculations is involved. Thus, automatic tools will be very helpful.

Similar to the case of classical programs, a verifier can prove the correctness:

$$\vdash_{par} \{A\} P \{B\} \text{ (or } \vdash_{tot} \{A\} P \{B\})$$

of a quantum program P in the following three steps:

- *Annotating the program*: insert quantum predicates at the intermediate points (in a way similar to a proof outline described in Sect. 4). Intuitively, an inserted statement is intended to hold when the control reaches the corresponding point.
- *Generating verification conditions (VCs)*: a set of mathematical statements, usually of the form $A_i \sqsubseteq B_i$ (see Lemma 1), is generated from the annotated program, where A_i, B_i are matrices, and $\sqsubseteq$ stands for the Löwner order. It is required that if all the generated VCs are true, then $\vdash_{par} \{A\}P\{B\}$ (or $\vdash_{tot} \{A\}P\{B\}$).
- *Proving VCs*: prove that matrix $B_i - A_i$ is positive semi-definite for each VC $A_i \sqsubseteq B_i$.

Recently, a theorem prover was built by Liu, Li, Wang et al. in [LLW16] for the quantum Hoare logic based on the proof assistant Isabelle/HOL. It has been used to verify several quantum programs, including Grover search and phase estimation, which is a key in Shor's factoring algorithm. We can expect that more automatic tools for verification of quantum programs and quantum cryptographic protocols will be built after quantum computers be commercialised.

7. Invariants of quantum programs

It has been well-understood since the very beginning that invariant generation is crucial for automatic verification of programs, and the problem of invariant generation has been intensively studied for classical programs. At this moment, invariants are provided by humans to the theorem prover for quantum programs developed in [LLW16] rather than generated by the system itself. But the issue of generating quantum invariants was recently considered in [YYW17]. In this section, we briefly review the main results in [YYW17].

We first observed that the control flow of a quantum program can be represented by a super-operator-valued transition system (SVTS):

Definition 11 (*Super-Operator-Valued Transition Systems* **[YYW17]**) An SVTS is a 5-tuple:

$$\mathcal{S} = \langle \mathcal{H}, L, l_0, \mathcal{T}, \Theta \rangle,$$

where:

1. $\mathcal{H}$ is a Hilbert space;
2. L is a finite set of locations;
3. $l_0 \in L$ is the initial location;
4. Θ is a quantum predicate in $\mathcal{H}$ denoting the initial condition; and
5. $\mathcal{T}$ is a set of transitions. Each transition $\tau \in \mathcal{T}$ is written as $\tau = l \xrightarrow{\mathcal{E}} l'$ with $l, l' \in L$ and $\mathcal{E}$ being a super-operator in $\mathcal{H}$. For each $l \in L$, it is required that

$$\mathcal{E}_l = \sum \{| \mathcal{E} : l \xrightarrow{\mathcal{E}} l' \in \mathcal{T} |\}$$

is trace-preserving, i.e. $tr(\mathcal{E}_l(\rho)) = tr(\rho)$ for all ρ.

Now we assume that the control flow SVTS of quantum program P is $\mathcal{S}_P$ with state Hilbert space $\mathcal{H}_P$ (see [YYW17] for detailed construction of $\mathcal{S}_P$). A set Π of paths in $\mathcal{H}_P$ is said to be prime if for each

$$\pi = l_1 \xrightarrow{\mathcal{E}_1} \cdots \xrightarrow{\mathcal{E}_{n-1}} l_n \in \Pi,$$

its proper initial segments

$$l_1 \xrightarrow{\mathcal{E}_1} \cdots \xrightarrow{\mathcal{E}_{k-1}} l_k \notin \Pi$$

for all $k < n$. We write $\mathcal{E}_\pi$ for the composition of $\mathcal{E}_1, \ldots, \mathcal{E}_{n-1}$ and $\mathcal{E}_\Pi = \sum \{| \mathcal{E}_\pi : \pi \in \Pi |\}$. Then we can define invariants for quantum programs:

Definition 12 (*Invariants of Quantum Programs* **[YYW17]**) Let $\mathcal{S}_P = \langle \mathcal{H}_P, L, l_0, \mathcal{T}, \Theta \rangle$ be the control flow SVTS of quantum program P and $l \in L$. An invariant at location $l \in L$ is a quantum predicate O in $\mathcal{H}_P$ satisfying the condition: for any density operator ρ and prime set Π of paths from l_0 to l, we have:

$$\mathrm{tr}(\Theta\rho) \leq 1 - \mathrm{tr}\left(\mathcal{E}_\Pi(\rho)\right) + \mathrm{tr}\left(O\mathcal{E}_\Pi(\rho)\right).$$

It was shown in [YYW17] that invariants can be used to establish partial correctness of quantum programs. We can also introduce the notion of inductive assertion for quantum programs. It is not easy to identify the invariant in the proof rules (R.LP) and (R.LT) for quantum loops, at least not as explicit as in the Hoare rule for classical loops. Example 4.1 in [YYW17] indicates that the quantum predicate $M_0^\dagger A M_0 + M_1^\dagger B M_1$ in rules (R.LP) and (R.LT) can be viewed as an inductive assertion (invariant). Furthermore, by generalising the constraint-based technique of Colón et al. [CSS03], [SSM04], it was demonstrated in [YYW17] that invariant generation for quantum programs can be reduced to an SDP (Semidefinite Programming) problem. In particular, the method proposed in [YYW17] was actually applied to generate the invariants of the quantum walk in Example 2 and quantum Metropolis sampling [TOVPV11].

8. Terminations analysis of quantum programs

As is well-known, termination analysis is a key step in proving total correctness of programs. Termination of quantum programs has been researched along the following two lines:

- Algorithmic analysis of termination for quantum programs was first considered in [YF10] where the Jordan decomposition of complex matrices was employed as the main tool. In [YYFD13], the author and his collaborators introduced quantum Markov chains as a semantic model of quantum programs. Then in a series of their papers [YFYY13], [YY12], [LYY14], termination of quantum programs was reduced to the reachability problem of quantum Markov chains, which is in turn tackled by developing a theory of quantum graphs (see [Yin16], Section 5.2). Indeed, this line of research also paves a way to model-checking quantum systems.
- The notion of ranking function was defined in [Yin11] in order to present the proof rule (R.LT) for total correctness of quantum loops. It was recalled as Definition 6 in this paper. In the last few years, (super)martingales have been employed as a powerful mathematical tools for termination analysis of probabilistic programs [CS13], [FH15], [CFNH16]. Recently, a notion of quantum (super)martingales was introduced in [LY18] as a generalisation of Definition 6 and used to characterise the termination of quantum programs. The synthesis problem of (super)martingales for quantum programs was also investigated there. The basic idea is that the fundamental Gleason theorem [Gle57] in quantum foundations can be used to determine the template of ranking functions, and then the synthesis problem of quantum (super)martingales with certain templates can also be reduced to an SDP problem. It seems that a further development of this line of research requires us to systematically establish a mathematical theory of quantum (super)martingales first.

The termination of the quantum walk in Example 2 was analysed by both of the above approaches [YYFD13, LY18], and the second approach was also used in [LY18] to analyse the termination of quantum Bernoulli factory, a quantum algorithm for random number generation [DJR15].

9. Conclusion

We conclude this paper by pointing out several problems for future research:

- *Combining Classical Computation and Quantum Computation*: We have been focusing on pure quantum programs without classical computation. This allows us to have a clean theory of quantum programming. But almost all existing quantum algorithms involves both quantum and classical computation, and the state-of-art quantum programming languages like Quipper [GLR13], LIQUi|> [WS14], Q# [SGT18], Scaffold [JPK15] and QWire [PRZ17] include both classical and quantum variables. So, it is desirable to generalise the verification techniques discussed in this paper to quantum programs involving classical computation. Indeed, a kind of correctness formulas (Hoare triples) involving both classical and quantum variables were already introduced in [YF11], but a Hoare-like logic for programs with both classical and quantum variables is still to be developed.
- *Parallel and Distributed Quantum Programs*: Distributed quantum computing has been studied for 20 years since [Gro97] and [CB97], including finding quantum algorithms for solving paradigmatic problems from classical distributed computing [TKM05], experiments toward physical implementation of distributed quantum computing [SMB06] and architecture of distributed quantum hardware systems [MMN08]. In particular, since practical quantum computers with large qubit capacity are still out of the current technology's reach, it is an attractive idea to use the physical resources of two or more small-capacity quantum computers to realise a large-capacity quantum computing system. So, another interesting problem is how the program logic and related techniques considered in this paper can be extended for reasoning about parallel and distributed

quantum programs. The notion of proof outline for quantum programs introduced in Sect. 4 should be helpful in dealing with the (non)interference of the variables shared by component quantum programs.

- *Verification of Quantum Cryptographic Protocols*: Over the last 10 years, quantum communication has blossomed into a viable commercial technology. But verifying correctness and security of quantum communication protocols is a notoriously difficult problem. Process algebra approach has been introduced for verification of quantum cryptographic protocols [AGN14, FY15, KKK16]. On the other hand, a (probabilistic) relational Hoare logic (pRHL) and a machine-checked framework for reasoning about security and privacy in classical computing and communicating systems were established by Barthe et al. [BFG14, BKO13]. The success of this line of research suggests us to develop verification techniques for quantum cryptographic protocols by extending our quantum Hoare logic. Indeed, the first attempt to develop a quantum relational Hoare logic (qRHL) was already made by Unruh [Unruh18], where a tool for the verification based on qRHL was implemented and successfully used to the security proof of several quantum cryptographic protocols.

Acknowledgements

The author likes to thank Professors Martin Fränzle, Deepak Kapur and Naijun Zhan for inviting me to give a talk at SETTA'2016. This work was partly supported by the Australian Research Council (Grant No: DP160101652) and the Key Research Program of Frontier Sciences, Chinese Academy of Sciences.

A. Basic properties of operators in Hilbert spaces

For convenience of the reader, we first review some basic properties of operators and super-operators needed in the proofs presented in Appendix C. The following lemma gives a characterisation of the Löwner order in terms of trace.

Lemma 2 Let A, B be observables (i.e. Hermitian operators) in Hilbert space $\mathcal{H}$. Then $A \sqsubseteq B$ if and only if $tr(A\rho) \leq tr(B\rho)$ for all density operators in $\mathcal{H}$.

The notion of dual super-operator is introduced in the next definition. It will be extensively used in Appendix C.

Definition 13 Let $\mathcal{E}$ be a quantum operation (i.e. super-operator) in Hilbert space $\mathcal{H}$ with the Kraus representation $\mathcal{E} = \sum_i E_i \circ E_i^\dagger$. Then its (Schrödinger–Heisenberg) dual is the super-operator $\mathcal{E}^*$ defined by

$$\mathcal{E}^*(A) = \sum_i E_i^\dagger A E_i$$

for any observable A in $\mathcal{H}$.

The next lemma presents a characterisation of duality between super-operators in terms of trace.

Lemma 3 For any quantum operation $\mathcal{E}$, observable A and density operator ρ in $\mathcal{H}$, we have:

$$tr(A\mathcal{E}(\rho)) = tr(\mathcal{E}^*(A)\rho).$$

In particular, $tr(\mathcal{E}(\rho)) = tr(\mathcal{E}^*(I)\rho)$, where I is the identity operator in $\mathcal{H}$.

Positivity of operators and the Löwner order between operators in a tensor product of Hilbert spaces are considered in the following:

Lemma 4 1. If A_1, A_2 are positive operators in $\mathcal{H}_1$ and $\mathcal{H}_2$, respectively, then $A_1 \otimes A_2$ is a positive operator in $\mathcal{H}_1 \otimes \mathcal{H}_2$.

2. For any operators A_1, B_1 in $\mathcal{H}_1$ and A_2, B_2 in $\mathcal{H}_2$, $A_1 \sqsubseteq B_1$ and $A_2 \sqsubseteq B_2$ implies $A_1 \otimes A_2 \sqsubseteq B_1 \otimes B_2$.

The proofs of all of the above lemmas can be found in any standard textbook on the theory of operators in Hilbert spaces.

B. Compositions of quantum relations

In Sect. 3, we already saw some simple quantum relations (i.e. quantum predicates in the tensor product of more than one state Hilbert spaces) in specifying correctness of quantum programs. The notion of composition of classical relations is well-defined and widely used to construct complicated relations from simple ones. However, it is highly nontrivial to find an appropriate definition of the notion of composition of quantum relations. Here, we give several tentative definitions:

Definition 14 Let A be a quantum relation between $\mathcal{H}_1$ and $\mathcal{H}_2$ (i.e. an observable in $\mathcal{H}_1 \otimes \mathcal{H}_2$ between the zero and identity operators) and B a quantum relation between $\mathcal{H}_2$ and $\mathcal{H}_3$. We define three kinds of their compositions:

1. Circle composition:

$$A \circ_{\mathcal{B}} B = \frac{1}{d_2} \sum_i \langle ii \mid A \otimes B \mid ii \rangle$$

   where and in the sequel $d_2 = \dim \mathcal{H}_2$ and $\mathcal{B} = \{| i\rangle\}$ is an orthonormal basis of $\mathcal{H}_2$.
2. Bullet composition:

$$A \bullet_{\mathcal{B}} B = \langle \Psi \mid A \otimes B \mid \Psi \rangle$$

   where $| \Psi\rangle_{\mathcal{B}} = \frac{1}{\sqrt{d_2}} \sum_i | ii\rangle$ is the (unnormalised) maximal entanglement defined by an orthonormal basis $\mathcal{B} = \{| i\rangle\}$ of $\mathcal{H}_2$.
3. Diamond composition:

$$A \diamond_v B = tr_{\mathcal{H}_2^{\otimes 2}} [S_v (A \otimes B) S_v]$$

   where $v \in \{+, -\}$, $S_\pm$ are the symmetrisation and anti-symmetrisation operators (see Example 6), and $tr_{\mathcal{H}_2^{\otimes 2}}$ stands for tracing out the middle two $\mathcal{H}_2$'s in $\mathcal{H}_1 \otimes \mathcal{H}_2 \otimes \mathcal{H}_2 \otimes \mathcal{H}_3$.

The algebraic structure of quantum relations equipped with the above composition operations and corresponding transitive closures is a very interesting topic for future research.

C. Proofs of Theorems

In this section, we provide the proofs of theorems omitted in the main part of this paper.

C.1. Proof of Theorem 2

To prove the strong soundness, suppose that

$$\langle P, \rho \rangle \to^n \{| \langle P_i, \rho_i \rangle |\}.$$

We proceed by induction on the length n of computation.

Induction Basis: For $n = 0$, $\{| \langle P_i, \rho_i \rangle |\}$ is a singleton $\{| \langle P_1, \rho_1 \rangle |\}$ with $P_1 \equiv P$ and $\rho_1 \equiv \rho$. Then we can choose $T_1 \equiv P$ and it holds that $P_1 \equiv at(T_1, P)$. Note that in the proof outline $\{A\}P^*\{B\}$, we have $A \sqsubseteq pre(P) = B_1$. Thus,

$$tr(A\rho) \le tr(B_1 \rho_1) = \sum_i tr(B_i \rho_i).$$

Induction Step: Now we assume that

$$\langle P, \rho \rangle \to^{n-1} \{| \langle P_i, \rho_i \rangle |\} \to \{| \langle P_i, \rho_i \rangle \mid i \neq i_0 |\} \cup \{| \langle Q_j, \sigma_j \rangle |\}$$

where

$$\langle P_{i_0}, \rho_{i_0} \rangle \to \{| \langle Q_j, \sigma_j \rangle |\}$$

is derived by one of the rules used in defining transition relation (12). Then we need to consider the following cases:

► Case 1. The last step uses rule (IF′). Then P_{i_0} can be written in the following form:

$$P_{i_0} \equiv \textbf{if } (\square m \cdot M[\overline{q}] = m \to R_m) \textbf{ fi},$$

and for each j, $Q_j \equiv R_m \equiv at(R_m, P)$ and $\sigma_j = M_m \rho_{i_0} M_m^\dagger$ for some m. On the other hand, a segment of the proof outline $\{A\}P^*\{B\}$ must be derived by the following inference:

$$\frac{\{A_m\}R_m^*\{C\} \text{ for every } m}{\left\{\sum_m M_m^\dagger A_m M_m\right\} \textbf{ if } \left(\square m \cdot M[\overline{q}] = m \to \{A_m\}\, R_m^*\right) \textbf{ fi}\{C\}}$$

and

$$B_{i_0} = pre\,(P_{i_0}) \sqsubseteq \sum_m M_m^\dagger A_m M_m, \quad A_m = pre\,(R_m)\,.$$

Therefore,

$$\begin{aligned} tr\,(B_{i_0}\rho_{i_0}) &\le tr\left(\sum_m M_m^\dagger A_m M_m \rho_{i_0}\right) \\ &= \sum_m tr\left(M_m^\dagger A_m M_m \rho_{i_0}\right) \\ &= \sum_m tr\left(A_m M_m \rho_{i_0} M_m^\dagger\right) \\ &= \sum_j tr\left(pre\left(Q_j\right)\sigma_j\right). \end{aligned}$$

By the induction hypothesis, we obtain:

$$\begin{aligned} tr(A\rho) &\le \sum_{i\neq i_0} tr\,(B_i\rho_i) + tr\,(B_{i_0}\rho_{i_0}) \\ &\le \sum_{i\neq i_0} tr\,(B_i\rho_i) + \sum_j tr\left(pre\left(Q_j\right)\sigma_j\right). \end{aligned}$$

So, the conclusion is true in this case.

► Case 2. The last step uses rule (L′). Then P_{i_0} must be in the following form:

$$P_{i_0} \equiv \textbf{while } M[\overline{q}] = 1 \textbf{ do } R \textbf{ od}$$

and

$$\{|\,\langle Q_j, \sigma_j\rangle\,|\} = \{|\,\langle Q_0, \sigma_0\rangle, \langle Q_1, \sigma_1\rangle\,|\}$$

with $Q_0 \equiv \textbf{skip}$, $\sigma_0 = M_0\rho_{i_0}M_0^\dagger$, $Q_1 \equiv R;\ P_{i_0}$ and $\sigma_1 = M_1\rho_{i_0}M_1^\dagger$. A segment of $\{A\}P^*\{B\}$ must be derived by the following inference:

$$\frac{\{D\}R^*\{M_0CM_0^\dagger + M_1DM_1^\dagger\}}{\begin{array}{r}\{M_0CM_0^\dagger + M_1DM_1^\dagger\}\ \textbf{while } M[\overline{q}] = 0 \textbf{ do } \{C\}\ \textbf{skip}\ \{C\} \\ = 1 \textbf{ do } R^*\{M_0CM_0^\dagger + M_1DM_1^\dagger\} \textbf{ od } \{C\}\end{array}}$$

and $B_{i_0} \sqsubseteq M_0CM_0^\dagger + M_1DM_1^\dagger$. Then $Q_0 \equiv at(\textbf{skip}, P)$, $pre(Q_0) = C$, $Q_1 \equiv at(R, P)$ and $pre(Q_1) = D$. It follows that

$$\begin{aligned} tr\,(B_{i_0}\rho_{i_0}) &\le tr\left[\left(M_0CM_0^\dagger + M_1DM_1^\dagger\right)\rho_{i_0}\right] \\ &= tr\left(M_0CM_0^\dagger\rho_{i_0}\right) + tr\left(M_1DM_1^\dagger\rho_{i_0}\right) \end{aligned}$$

$$= tr\left(CM_0^\dagger \rho_{i_0} M_0\right) + tr\left(DM_1^\dagger \rho_{i_0} M_1\right)$$
$$= tr\left(pre(Q_0)\sigma_0\right) + tr\left(pre(Q_1)\sigma_1\right).$$

Furthermore, by the induction hypothesis, we have:

$$tr(A\rho) \le \sum_{i\neq i_0} tr\left(B_i\rho_i\right) + tr\left(B_{i_0}\rho_{i_0}\right)$$
$$\le \sum_{i\neq i_0} tr\left(B_i\rho_i\right) + \sum_j tr\left(pre\left(Q_j\right)\sigma_j\right).$$

Thus, the conclusion is true in this case.

► Case 3. The last step uses rule (Sk), (In) or (UT). Similar but easier.

C.2. Proof of Theorem 3

We only prove the theorem for partial correctness; the case of total correctness is simpler.

1. We first prove that rule (Ax.Inv) is sound for partial correctness. Since $var(P)\cap V=\emptyset$, $[\![P]\!]$ can be seen as a super-operator $\mathcal{E}$ in $\mathcal{H}_{V^c}$. Then when considering $[\![P]\!]$ as a super-operator in $\mathcal{H}$, we have:

$$\begin{aligned}
&[\![P]\!]^*(A) + \left[I - [\![P]\!]^*(I)\right] \\
&\quad = B\otimes \mathcal{E}^*(I_{V^c}) + [I - I_V\otimes \mathcal{E}^*\left(I_{V^c}\right)] \\
&\quad = B\otimes \mathcal{E}^*\left(I_{V^c}\right) + I_V\otimes [I_{V^c} - \mathcal{E}^*\left(I_{V^c}\right)] \\
&\quad \sqsupseteq B\otimes \mathcal{E}^*\left(I_{V^c}\right) + B\otimes [I_{V^c} - \mathcal{E}^*\left(I_{V^c}\right)] \\
&\quad = B\otimes I_{V^c} = A
\end{aligned}$$

because $B\sqsubseteq I_{V^c}$ and $\mathcal{E}^*\left(I_{V^c}\right)\sqsubseteq I_{V^c}$, where I stands for the identity operator in $\mathcal{H}$.

2. Now we prove the soundness of rule (R.TI) for partial correctness. Assume that $\models_{par}\{A\}P\{B\}$. Then it holds that

$$A\sqsubseteq [\![P]\!]^*\left(B\otimes I_W\right) + \left(I - [\![P]\!]^*(I)\right)$$

where I is the identity operator in $\mathcal{H}_{V\cup W}$. Note that $var(P)\subseteq V$. Then we have:

$$tr_W\left([\![P]\!]^*\left(B\otimes I_W\right)\right) = [\![P]\!]^*(B)$$

where the occurrences of $[\![P]\!]$ in the left-hand side and right-hand side are seen as a super-operator in $\mathcal{H}_{V\cup W}$ and one in $\mathcal{H}_V$, respectively. Similarly, we have $tr_W\left([\![P]\!]^*(I)\right) = [\![P]\!]^*(I_V)$. Therefore, it follows that

$$\begin{aligned}
tr_W A &\sqsubseteq tr_W [\![P]\!]^*\left(B\otimes I_W\right) + tr_W\left(I - [\![P]\!]^*(I)\right) \\
&= [\![P]\!]^*(B) + \left(I_V - [\![P]\!]^*\left(I_V\right)\right)
\end{aligned}$$

and $\models_{par}\{tr_W A\}\,P\{B\}$.

3. We prove the soundness of (R.CC) for partial correctness. If for every $i=1,\ldots,m$, $\models_{par}\{A_i\}P\{B_i\}$, then by Lemma 1 we have:

$$A_i\sqsubseteq [\![P]\!]^*(B_i) + \left[I - [\![P]\!]^*(I)\right].$$

Consequently, it holds that

$$\begin{aligned}
\sum_{i=1}^m p_i A_i &\sqsubseteq \sum_{i=1}^m p_j\left([\![P]\!]^*(A_j) + [I - [\![P]\!]^*(I)]\right) \\
&= [\![P]\!]^*\left(\sum_{i=1}^m p_i B_i\right) + [I - [\![P]\!]^*(I)]
\end{aligned}$$

because $[\![P]\!]^*(\cdot)$ is linear, and $I - [\![P]\!]^*(I)$ is positive. Thus, we have:

$$\models_{par} \left\{ \sum_{i=1}^{m} p_i A_i \right\} P \left\{ \sum_{i=1}^{m} p_i B_i \right\}.$$

4. We prove the soundness of (R.Inv) for partial correctness. From $\models_{par} \{A\}P\{B\}$, i.e. $A \sqsubseteq [\![P]\!]^*(B) + (I - [\![P]\!]^*(I))$, we derive that

$$\begin{aligned} pA + qC &\sqsubseteq p[\![P]\!]^*(B) + p\,(I - [\![P]\!]^*(I)) \\ &= (p[\![P]\!]^*(B) + qC) + p\,(I - [\![P]\!]^*(I)) \\ &\sqsubseteq (p[\![P]\!]^*(B) + qC) + (I - [\![P]\!]^*(I)) \\ &= [\![P]\!]^*(pB + qC) + (I - [\![P]\!]^*(I)) \end{aligned}$$

because $p \leq 1$ and $V \cap var(P) = \emptyset$ implies $[\![P]\!]^*(C) = C$. Therefore, we have $\models_{par} \{pA + qC\}P\{pB + qC\}$.

5. We prove that rule (R.SO) is sound for partial correctness. Suppose that $\models_{par} \{A\}P\{B\}$, i.e. $A \sqsubseteq [\![P]\!]^*(B) + (I - [\![P]\!]^*(I))$. Note that $\mathcal{E}$ is a super-operator in $\mathcal{H}_V$ and $V \cap var(P) = \emptyset$. Then $\mathcal{E}^*$ and $[\![P]\!]^*$ commutes, i.e. $\mathcal{E}^* \circ [\![P]\!]^* = [\![P]\!]^* \circ \mathcal{E}^*$. Moreover, it holds that $\mathcal{E}^*(I) \sqsubseteq I$ and thus

$$(\mathcal{I} - [\![P]\!]^*)(\mathcal{E}^*(I)) \sqsubseteq (\mathcal{I} - [\![P]\!]^*)(I)$$

where $\mathcal{I}$ is the identity super-operator. Therefore, we obtain:

$$\begin{aligned} \mathcal{E}^*(A) &\sqsubseteq \mathcal{E}^*\left[[\![P]\!]^*(B) + (I - [\![P]\!]^*(I))\right] \\ &= \mathcal{E}^*([\![P]\!]^*(B)) + \left[\mathcal{E}^*(I) - \mathcal{E}^*([\![P]\!]^*(I))\right] \\ &= [\![P]\!]^*(\mathcal{E}^*(B)) + (\mathcal{I} - [\![P]\!]^*)(\mathcal{E}^*(I)) \\ &\sqsubseteq [\![P]\!]^*(\mathcal{E}^*(B)) + (\mathcal{I} - [\![P]\!]^*)(I) \\ &= [\![P]\!]^*(\mathcal{E}^*(B)) + (I - [\![P]\!]^*(I)) \end{aligned}$$

and $\models_{par} \{\mathcal{E}^*(A)\}P\{\mathcal{E}^*(B)\}$.

6. Finally, we prove the soundness of (R.Lim) for partial correctness. The existence of $\lim_{n\to\infty} A_n$ and $\lim_{n\to\infty} B_n$ is guaranteed by Lemma 4.1.3 in [Yin16]. Assume that $\models_{par} \{A_n\}P\{B_n\}$. Then

$$A_n \sqsubseteq [\![P]\!]^*(B_n) + \left[I - [\![P]\!]^*(I)\right]$$

and the continuity of super-operator $[\![P]\!]^*$ yields:

$$\begin{aligned} A = \lim_{n\to\infty} A_n &\sqsubseteq \lim_{n\to\infty} \left([\![P]\!]^*(B_n) + \left[I - [\![P]\!]^*(I)\right]\right) \\ &= [\![P]\!]^*(\lim_{n\to\infty} B_n) + \left[I - [\![P]\!]^*(I)\right] \\ &= [\![P]\!]^*(B) + \left[I - [\![P]\!]^*(I)\right]. \end{aligned}$$

Thus, $\models_{par} \{A\}P\{B\}$.

References

[AG05] Altenkirch T, Grattage J (2005) A functional quantum programming language. In: Proceedings of the 20th IEEE symposium on logic in computer science (LICS), pp 249–258

[ABNVW01] Ambainis A, Bach E, Nayak A, Vishwanath A, Watrous J (2001) One-dimensional quantum walks. In: Proceedings of the 33rd ACM symposium on theory of computing (STOC), pp 37–49

[ABO09] Apt KR, de Boer FS, Olderog E-R (2009) Verification of sequential and concurrent programs. Springer, London

[AGN14] Ardeshir-Larijani E, Gay SJ, Nagarajan R (2014) Verification of concurrent quantum protocols by equivalence checking. In: Proceedings of the 20th international conference on tools and algorithms for the construction and analysis of systems (TACAS), pp 500–514

[BS06] Baltag A, Smets S (2006) LQP: the dynamic logic of quantum information. Math Struct Comput Sci 16:491–525

[BFG14] Barthe G, Fournet C, Grégoire B, Strub P-Y, Swamy N, Zanella Béguelin S (2014) Probabilistic relational verification for cryptographic implementations. In: Proceedings of the 41st annual ACM symposium on principles of programming languages (POPL), pp 193–206

[BKO13] Barthe G, Köpf B, Olmedo F, Zanella Béguelin Z (2013) Probabilistic relational reasoning for differential privacy. In: ACM transactions on programming languages and systems, vol 35, No. 9

[BJ04] Brunet O, Jorrand P (2004) Dynamic quantum logic for quantum programs. Int J Quantum Inf 2:45–54

[CMS06] Chadha R, Mateus P, Sernadas A (2006) Reasoning about imperative quantum programs. Electron Notes Theor Comput Sci 158:19–39

[CS13] Chakarov A, Sankaranarayanan S (2013) Probabilistic program analysis with martingales. In: Proceedings of the 25th international conference on computer aided verification (CAV). Springer LNCS 8044, pp 511–526

[CFNH16] Chatterjee K, Fu HF, Novotný P, Hasheminezhad R (2016) Algorithmic analysis of qualitative and quantitative termination problems for affine probabilistic programs. In: Proceedings of the 43rd annual ACM symposium on principles of programming languages (POPL), pp 327–342

[CB97] Cleve R, Buhrman H (1997) Substituting quantum entanglement for communication. Phys Rev A 56:1201–1204

[CSS03] Colón MA, Sankaranarayanan S, Sipma HB (2003) Linear invariant generation using non-linear constraint solving. In: Proceedings of the 15th international conference on computer aided verification (CAV). Springer LNCS, pp 420–433

[DJR15] Dale H, Jennings D, Rudolph T (2015) Provable quantum advantage in randomness processing. Nat Commun 6:8203

[DP06] D'Hondt E, Panangaden P (2006) Quantum weakest preconditions. Math Struct Comput Sci 16:429–451

[FDJY07] Feng Y, Duan RY, Ji ZF, Ying MS (2007) Proof rules for the correctness of quantum programs. Theor Comput Sci 386:151–166

[FHTZ15] Feng Y, Hahn EM, Turrini A, Zhang LJ (2015) QPMC: a model checker for quantum programs and protocols. In: Proceedings of the 20th international symposium on formal methods (FM). Springer LNCS 9109, pp 265–272

[FY15] Feng Y, Ying MS (2015) Toward automatic verification of quantum cryptographic protocols. In: Proceedings of the 26th international conference on concurrency theory (CONCUR), pp 441–455

[FYY13] Feng Y, Yu NK, Ying MS (2013) Model checking quantum Markov chains. J Comput Syst Sci 79:1181–1198

[FH15] Fioriti LMF, Hermanns H (2015) Probabilistic termination: soundness, completeness, and compositionality. In: Proceedings of the 42nd annual ACM symposium on principles of programming languages (POPL), pp 489–501

[GPN08] Gay SJ, Papanikolaou N, Nagarajan R (2008) QMC: a model checker for quantum systems. In: Proceedings of the 20th international conference on computer aided verification (CAV). Springer LNCS 5123, pp 543–547

[Gle57] Gleason AM (1957) Measures on the closed subspaces of a Hilbert space. J Math Mech 6:885–893

[Gor75] Gorelick GA (1975) A complete axiomatic system for proving assertions about recursive and non-recursive programs. Technical Report, Department of Computer Science, University of Toronto

[GLR13] Green A, Lumsdaine PL, Ross NJ, Selinger P, Valiron B (2013) Quipper: a scalable quantum programming language. In: Proceedings of the 34th ACM conference on programming language design and implementation (PLDI), pp 333–342

[Gro97] Grover LK (1997) Quantum telecomputation. arXiv:quant-ph/9704012

[Har79] Harel D (1979) First-order dynamic logic, LNCS 68. Springer

[KV02] Den Hartog J, de Vink EP (2002) Verifying probabilistic programs using a Hoare like logic. Int J Found Comput Sci 13:315–340

[Kle99] Leymann T (1999) Hoare logic and auxiliary variables. Formal Asp Comput 11:541–566

[KMMM10] Katoen J-P, McIver A, Meinicke L, Morgan CC (2010) Linear-invariant generation for probabilistic programs—automated support for proof-based methods. In: Proceedings 17th international symposium on static analysis (SAS). Springer LNCS 6337, pp 390–406

[KKK16] Kubota T, Kakutani Y, Kato G, Kawano Y, Sakurada H (2016) Semi-automated verification of security proofs of quantum cryptographic protocols. J Symb Comput 73:192–220

[JPK15] JavadiAbhari A, Patil S, Kudrow D, Heckey J, Lvov A, Chong FT, Martonosi M (2015) ScaffCC: scalable compilation and analysis of quantum programs. Parallel Comput 45:2–17

[Ka09] Kakutani Y (2009) A logic for formal verification of quantum programs. In: Proceedings of the 13th Asian computing science conference (ASIAN 2009). Springer LNCS 5913, pp 79–93

[LY18] Li YJ, Ying MS (2018) Algorithmic analysis of termination problems for quantum programs. In: Proceedings of the 45th annual ACM symposium on principles of programming languages (POPL), pp 35.1–35.29

[LYY14] Li YJ, Yu NK, Ying MS (2014) Termination of nondeterministic quantum programs. Acta Inf 51:1–24

[LLW16] Liu T, Li YJ, Wang SL, et al A theorem prover for quantum Hoare logic and its applications. arXiv:1601.03835

[MS06] Mateus P, Sernadas A (2006) Weakly complete axiomatization of exogenous quantum propositional logic. Inf Comput 204:771–794

[MMN08] van Meter R, Munro WJ, Nemoto K, Itoh KM (2008) Arithmetic on a distributed-memory quantum multicomputer. ACM J Emerg Technol Comput Syst 3, Art. No. 17

[MM05] McIver A, Morgan C (2005) Abstraction, refinement and proof for probabilistic systems. Springer, Berlin

[NC00] Nielsen MA, Chuang IL (2000) Quantum computation and quantum information. Cambridge University Press, Cambridge

[PRZ17] Paykin J, Rand R, Zdancewic S (2017) QWIRE: a core language for quantum circuits. In: Proceedings of the 44th annual ACM symposium on principles of programming languages (POPL), pp 846–858

[Rand17] Rand R Verification logics for quantum programs. http://www.cis.upenn.edu/~rrand/wpe.pdf

[SSM04] Sankaranarayanan S, Sipma HB, Manna Z (2004) Non-linear loop invariant generation using Gröbner bases. In: Proceedings of the 31st ACM symposium on principles of programming languages (POPL), pp 318–329

[Sel04] Selinger P (2004) Towards a quantum programming language. Math Struct Comput Sci 14:527–586

[SMB06] Serafini A, Mancinians S, Bose S (2006) Distributed quantum computation via optical fibers. Phys Rev Lett 96, Art. No. 010503

[SGT18] Svore K, Geller A, Troyer M, Azariah J, Granade C, Heim B, Kliuchnikov V, Mykhailova M, Paz A, Roetteler M (2018) Q#: enabling scalable quantum computing and development with a high-level DSL. In: Proceedings of the real world domain specific languages workshop 2018, Art. no. 8

[TKM05] Tani S, Kobayashi H, Matsumoto K (2012) Exact quantum algorithms for the leader election problem. ACM Trans Comput Theory 4, Art. No. 1

[TOVPV11] Temme K, Osborne TJ, Vollbrecht KG, Poulin D, Verstraete F (2011) Quantum metropolis sampling. Nature 471:87–90

[Unruh18] Unruh D Quantum relational Hoare logic. https://arxiv.org/pdf/1802.03188.pdf
[WS14] Wecker D, Svore KM (2014) LIQUi|>: a software design architecture and domain-specific language for quantum computing. arXiv:1402.4467
[Yin11] Ying MS (2011) Floyd-Hoare logic for quantum programs. ACM Trans Program Lang Syst 39, Art. no. 19
[Yin16] Ying MS (2016) Foundations of quantum programs. Morgan-Kaufmann
[Yin16a] Ying MS (2016) Toward automatic verification of quantum programs (extended abstract). In: Proceedings of the 2nd international symposium on dependable software engineering: theories, tools, and applications (SETTA)
[YCFD07] Ying MS, Chen JX, Feng Y, Duan RY (2007) Commutativity of quantum weakest preconditions. Inf Process Lett 104:152–158
[YF10] Ying MS, Feng Y (2010) Quantum loop programs. Acta Inf 47:221–250
[YF11] Ying MS, Feng Y (2011) A flow-chart language for quantum programming. IEEE Trans Softw Eng 37:466–485
[YLYF14] Ying MS, Li YJ, Yu NK, Feng Y (2014) Model-checking linear-time properties of quantum systems. ACM Trans Comput Logic 15, Art. no. 22
[YYW17] Ying MS, Ying SG, Wu XD (2017) Invariants of quantum programs: characterisations and generation. In: Proceedings of the 44th annual ACM symposium on principles of programming languages (POPL), pp 818–832
[YYFD13] Ying MS, Yu NK, Feng Y, Duan RY (2013) Verification of quantum programs. Sci Comput Program 78:1679–1700
[YFYY13] Ying SG, Feng Y, Yu NK, Ying MS (2013) Reachability analysis of quantum Markov chains. In: Proceedings of the 24th Int Conf Concurr Theory (CONCUR), pp 334–348
[YY12] Yu NK, Ying MS (2012) Reachability and termination analysis of concurrent quantum programs. In: Proceedings of the 23th international conference on concurrency theory (CONCUR), pp 69–83

Received 30 May 2017
Accepted in revised form 20 July 2018 by Naijun Zhan, Martin Fraenzle, Deepak Kapur, and Heike Wehrheim