An Applied Quantum Hoare Logic

Li Zhou
Department of Computer Science and Technology
Tsinghua University
China
zhou31416@gmail.com

Nengkun Yu
CQSI, FEIT
University of Technology Sydney
Australia
nengkunyu@gmail.com

Mingsheng Ying
CQSI, FEIT, Uinversity of Technology Sydney, Australia
Institute of Software, CAS, China
Tsinghua University, China
Mingsheng.Ying@uts.edu.au

Abstract

We derive a variant of quantum Hoare logic (QHL), called applied quantum Hoare logic (aQHL for short), by: (1) restricting QHL to a special class of preconditions and postconditions, namely projections, which can significantly simplify verification of quantum programs and are much more convenient when used in debugging and testing; and (2) adding several rules for reasoning about robustness of quantum programs, i.e. error bounds of outputs. The effectiveness of aQHL is shown by its applications to verify two sophisticated quantum algorithms: HHL (Harrow-Hassidim-Lloyd) for solving systems of linear equations and qPCA (quantum Principal Component Analysis).

CCS Concepts • **Theory of computation → Hoare logic**; **Program verification**.

Keywords Quantum computation, programming languages, Hoare logic, projections, robustness

ACM Reference Format:
Li Zhou, Nengkun Yu, and Mingsheng Ying. 2019. An Applied Quantum Hoare Logic. In *Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI '19), June 22–26, 2019, Phoenix, AZ, USA.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3314221.3314584

1 Introduction

Quantum programming has already been actively researched for two decades [4, 15, 27, 31–33, 38]. In particular, in the last few years, several mature quantum programming languages and platforms have been introduced, including Quipper [17], Scaffold [1], QWIRE [28], IBM's Qiskit [3], Microsoft's Q# [34], Google's Cirq [16] and Rigetti's Forest [30], perhaps stimulated by rapid progress in the implementation of quantum computing hardware. Now people start to consider how to warrant correctness of quantum programs: debugging, testing or verification?

Quantum Hoare Logic: Indeed, attempts of developing Hoare-like logic for verification of quantum programs have been made in a series of papers [2, 6, 7, 9, 10, 14, 21, 29]. In particular, D'Hondt and Panangaden [13] proposed the notion of quantum weakest precondition, and Ying [37] established quantum Hoare logic (QHL for short) for both partial correctness and total correctness with (relative) completeness. More recently, SDP (Semi-Definite Programming) algorithms for invariant generation and termination analysis of quantum programs were developed in [23, 40].

Hoare Logic in Use: In a retrospective article [19], Hoare described how his logic has been used in industry's development practice: proof and testing are mutually supportive ways of accumulating evidence of the correctness of programs. Assertions sprinkled more or less liberally in the program text, are used not to prove the programs, but rather to help detect and diagnose programming errors. They are evaluated at runtime during overnight tests, and indicate the occurrence of any error as close as possible to the place in the program where it actually occurred.

How to Use Quantum Hoare Logic? We can expect that QHL will be applied in the same way in future quantum software development. Recall from [37] that both partial and total correctness:

$$\models_{\text{par}} \{P\}S\{Q\}, \qquad \models_{\text{tot}} \{P\}S\{Q\}$$

of a quantum program S are defined by an inequality between the expectations $tr(P\rho)$ and $tr(Q[\![S]\!](\rho))$ of precondition and postcondition observables P and Q in the input state ρ and the output state $[\![S]\!](\rho)$, respectively. The attractiveness of QHL comes from its interpretation naturally derived from quantum mechanics and a (relatively) complete axiomatisation. However, there is an obvious gap between its theoretical characterization and practical use; in particular:

1. When combined with testing, even testing a single atomic step: a large number of measurements are needed in order to achieve a good estimation of the expectations $tr(P\rho)$ and $tr(Q[\![S]\!](\rho))$. Even worse, measurements could destroy the state, and we cannot create backup copies of a state in the middle of a quantum

PLDI '19, June 22–26, 2019, Phoenix, AZ, USA

ACM ISBN 978-1-4503-6712-7/19/06…$15.00
https://doi.org/10.1145/3314221.3314584

computation, which is prohibited by the quantum no-cloning theorem.
2. Usually, complicated matrix calculations are involved in applying the inference rules of QHL, especially the loop rule for total correctness, where one needs to compute a ranking function defined by two matrix inequalities (see Definition 2.6). This is why up to now only some simple quantum algorithms can be verified in QHL (by hand).
3. In an attempt to verify a well-known quantum algorithm for machine learning, namely qPCA (quantum Principal Component Analysis) [25], we notice that a quantum algorithm often outputs states that do not exactly but only approximately satisfy the postcondition; that is, they are very close to some states satisfying the postcondition.

Contributions of the Paper: To circumvent the above issues, we derive a variant of QHL, called *applied quantum Hoare logic* (aQHL for short), as follows:

- Only a special class of Hermitian operators, namely projections (or equivalently, closed subspaces of the state Hilbert space), are used as preconditions and postconditions. This restriction allows us to significantly simplify the inference rules for case statements and loops and computation of ranking functions in QHL. These simplifications enable us to verify some sophisticated quantum algorithms; in particular a verification of the famous HHL (Harrow-Hassidim-Lloyd) quantum algorithms for solving systems of linear equations [18] in aQHL is given in Section 5.
- In order to prove that the outputs of a quantum program approximately satisfy a postcondition, we develop several rules for reasoning about robustness of quantum programs, i.e. error bounds of the outputs of programs. Using these new rules, a verification of quantum machine learning algorithm qPCA (quantum Principal Component Analysis) in aQHL is presented in Section 6.

In light of testing and debugging quantum programs, aQHL has one more advantage. In checking a quantum Hoare triple $\{P\}S\{Q\}$ with P and Q being projections, we don't need to calculate expectations $tr(P\rho)$ and $tr(Q[\![S]\!](\rho))$. Instead, we only need to see whether the output state $[\![S]\!](\rho)$ falls into the subspace corresponding to postcondition Q once the input state ρ is in the subspace corresponding to precondition P. This is very similar to the case of checking a Hoare triple for a classical program, and can be done in between two consecutive atomic steps of the program, thus avoids the first issue discussed above. Due to the limit of space, we leave a more detailed discussion on applications of aQHL to quantum program testing and debugging to a forthcoming paper [41].

Organisation of the Paper: The remainder of this Introduction is devoted to a discussion about related work. QHL is briefly reviewed in Section 2. In Section 3, we derive a projective variant aQHL of QHL with simpler inference rules and ranking functions than that of the original QHL. This is achieved by following technical contributions: (1) identifying the termination spaces of quantum programs; and (2) proving two meta-rules (Lifting and reduction; Theorems 3.2 and 3.3) connecting QHL and aQHL. It is worth pointing out that aQHL is (relatively) complete for *projective* preconditions and postconditions. aQHL is expanded in Section 4 by introducing rules for robustness reasoning. Their soundness is proved there. Verifications of HHL and qPCA in aQHL are presented in Sections 5 and 6, respectively. Proof details can be found in the complete version of this paper.

1.1 Related Work and Comparison

The programming language equipped with aQHL is a restricted version of quantum **while**-language in [37], where the measurement in a case statement or a loop is only allowed to be a projective measurement. But such a restriction does not really narrow aQHL's extension of applications because a general measurement can always be implemented by a projective measurement together with a unitary transformation. On the other hand, the expressive power of correctness formulas with projective preconditions and postconditions in aQHL is strictly weaker than that of correctness formulas with general Hermitian operators as preconditions and postconditions in QHL [37]. For current applications, this is not a significant restriction because the majority of the existing quantum algorithms are derived from transformations of pure states. If we directly use the original QHL to verify these algorithms, the involved calculation would be much more complicated, and sometimes even unmanageable by hands.

Note that all projections in (or, closed subspaces of) a Hilbert space form an orthomodular lattice [22], i.e. (the algebraic counterpart of) Birkhoff-von Neumann quantum logic [8], and thus the power of quantum logic can be leveraged in reasoning about quantum programs. The first attempt of using quantum logic in reasoning about quantum programs was made in [9]. Projections were also employed in [39] to develop a predicate transformer semantics of quantum programs. However, no inference rules presented in this paper were derived in [9, 39]. Recently, Unruh [35] developed a quantum relational Hoare logic (qRHL) using subspaces as preconditions and postconditions, but qRHL aims at reasoning about equivalence between two quantum program (and targets applications in security verification of quantum cryptographic protocols) rather than correctness of quantum programs.

Continuity and robustness of classical programs have been systematically studied in [11, 12]. Robustness analysis of quantum programs was first considered in [20]. However,

both the motivation and approach in [20] are different from that in this paper. The proof system developed in [20] is mainly for reasoning about erroneous behaviour of (faulty or noisy implementation of) quantum programs. Our proof rules can be used to verify inexact quantum algorithms like qPCA; in particular, it seems that the verification techniques developed in [20] are unable to verify qPCA. However, the notion of (a, n)-bounded quantum loop proposed in [20] plays an essential role in our rules for robustness reasoning.

2 Quantum Hoare Logic

For convenience of the reader, we briefly review quantum Hoare logic in this section; for more details we refer to [37, 38]. We assume a set *Var* of quantum variables.

Definition 2.1 (Syntax [37]). *The quantum **while**-programs are defined by the grammar:*

$$S ::= \textbf{skip} \mid S_1; S_2 \mid q := |0\rangle \mid \overline{q} := U[\overline{q}] \quad (1)$$

$$\mid \textbf{if } (\square m \cdot M[\overline{q}] = m \to S_m) \textbf{ fi} \quad (2)$$

$$\mid \textbf{while } M[\overline{q}] = 1 \textbf{ do } S \textbf{ od} \quad (3)$$

The program constructs defined above are explained as follows. $q := |0\rangle$ means that quantum variable q is initialised in a basis state $|0\rangle$. $\overline{q} := U[\overline{q}]$ denotes that unitary transformation U is applied to a sequence $\overline{q}$ of quantum variables. In the case statement **if** $\cdots$ **fi**, quantum measurement M is performed on $\overline{q}$ and then a subprogram S_m is chosen for next execution according to the measurement outcome m. In the loop **while** $\cdots$ **od**, measurement M in the guard has only two possible outcomes 0, 1: if the outcome is 0 the loop terminates, and if the outcome is 1 it executes the loop body S and enters the loop again.

For each program S, we write $var(S)$ for its state Hilbert space, i.e. the tensor product of the state Hilbert spaces of the quantum variables occurring in S. We further write $\mathcal{D}(\mathcal{H}_S)$ for the set of partial density operators, i.e. positive operators with traces ≤ 1, in $\mathcal{H}_S$. A configuration is a pair $C = \langle S, \rho\rangle$ where S is a program or the termination symbol $\downarrow$, and $\rho \in \mathcal{D}(\mathcal{H}_S)$ denotes the state of quantum variables.

Definition 2.2 (Operational Semantics [37]). *The operational semantics of quantum **while**-programs is defined as a transition relation $\to$ by the transition rules in Figure 1.*

Let us briefly explain the transitional rules in Figure 1. Essentially, rules (In), (UT), (IF), (L0) and (L1) are determined by the basic postulates of quantum mechanics. The state ρ_0^q in rule (In) is obtained by initialising quantum variable q to basis state $|0\rangle$ by leaving other quantum variables unchanged. Such an initialisation can be realised by quantum operation $\mathcal{E}_I(\rho) = \sum_n |0\rangle_q\langle n|\rho|n\rangle_q\langle 0|$ for all ρ. Unlike the classical case, a quantum state over q and other variables in the register may be entangled and therefore, only setting the state of q being $|0\rangle_q\langle 0|$ is not meaningful. In rule (UT) (and in the sequel), † stands for the adjoint of an operator; in particular,

(Sk) $\langle \textbf{skip}, \rho\rangle \to \langle\downarrow, \rho\rangle$

(In) $\langle q := |0\rangle, \rho\rangle \to \langle\downarrow, \rho_0^q\rangle$

(UT) $\langle \overline{q} := U[\overline{q}], \rho\rangle \to \langle\downarrow, U\rho U^\dagger\rangle$

(SC) $$\frac{\langle S_1, \rho\rangle \to \langle S_1', \rho'\rangle}{\langle S_1; S_2, \rho\rangle \to \langle S_1'; S_2, \rho'\rangle}$$

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

(L0) $\langle \textbf{while } M[\overline{q}] = 1 \textbf{ do } S \textbf{ od}, \rho\rangle \to \langle\downarrow, M_0\rho M_0^\dagger\rangle$

(L1) $\langle \textbf{while } M[\overline{q}] = 1 \textbf{ do } S \textbf{ od}, \rho\rangle \to$
$\langle S; \textbf{while } M[\overline{q}] = 1 \textbf{ do } S \textbf{ od}, M_1\rho M_1^\dagger\rangle$

Figure 1. Transition Rules. In (In), $\rho_0^q = \sum_n |0\rangle_q\langle n|\rho|n\rangle_q\langle 0|$. In (SC), we make the convention $\downarrow; S_2 = S_2$. In (IF), m ranges over every possible outcome of measurement $M = \{M_m\}$.

if the state Hilbert space finite dimensional, it denotes the transpose and conjugate of a matrix. Rule (UT) is simply a rewriting of the postulate for (discrete-time) dynamics of a (closed) quantum system. Transitions in rules (IF), (L0) and (L1) are essentially probabilistic; but we adopt a convention from [33] to present them as a non-probabilistic transition. For example, for each m, the transition in (IF) happens with probability $p_m = tr(M^\dagger M_m\rho)$, and the program state ρ is changed to $\rho_m = M_m\rho M_m^\dagger/p_m$. We can combine probability p_m and density operator ρ_m into a partial density operator $M_m\rho M_m^\dagger = p_m\rho_m$. This convention significantly simplifies the presentation.

Definition 2.3 (Denotational Semantics [37]). *For any quantum **while**-program S, its semantic function is the mapping:*

$$[\![S]\!] : \mathcal{D}(\mathcal{H}_S) \to \mathcal{D}(\mathcal{H}_S)$$

defined by

$$[\![S]\!](\rho) = \sum \{\!|\rho' : \langle S, \rho\rangle \to^* \langle\downarrow, \rho'\rangle|\!\} \quad (4)$$

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

Intuitively, for an input ρ, if for each $k \geq 0$, program S terminates at step k with probability q_k and outputs density operator σ_k, then with the convention above in mind it is easy to see that $[\![S]\!](\rho) = \sum_{k=0}^\infty q_k\sigma_k$.

The properties of quantum program states are described by a special class of observables, called quantum predicates [13]. The Löwner order between operators is defined as follows: $A \sqsubseteq B$ if and only if $B - A$ is positive. Then 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.

Definition 2.4 (Hoare Triple [37]). *A correctness formula (or a Hoare triple) is a statement of the form: $\{A\}S\{B\}$, where S is a quantum **while**-program, and both A, B are quantum*

predicates in $\mathcal{H}_S$, called the precondition and postcondition, respectively.

For any operator A in state Hilbert space $\mathcal{H}_S$, its trace is defined as $tr(A) = \sum_i \langle i|A|i\rangle$, where $\{|i\rangle\}$ is an orthonormal basis of $\mathcal{H}_S$, and $\langle i|$ denotes the adjoint of $|i\rangle$. In the special case of $\dim \mathcal{H}_S < \infty$, A can be seen as a matrix and $tr(A)$ is the sum of the entries on the diagonal of A.

Definition 2.5 (Correctness [37]). 1. *The correctness formula $\{A\}S\{B\}$ is true in the sense of total correctness, written: $\models_{\text{tot}} \{A\}S\{B\}$, if for all $\rho \in \mathcal{D}(\mathcal{H}_S)$ we have:*

$$\text{tr}(A\rho) \leq \text{tr}(B[\![S]\!](\rho)). \tag{5}$$

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

$$\text{tr}(A\rho) \leq \text{tr}(B[\![S]\!](\rho)) + [\text{tr}(\rho) - \text{tr}([\![S]\!](\rho))]. \tag{6}$$

The defining inequalities (5) and (6) of total and partial correctness can be easily understood by noting that the interpretation of $tr(A\rho)$ in physics is the expectation of observable A in state ρ. More precisely, observable A determines a measurement. We repeatedly perform this measurement on quantum systems in the state ρ. Then each possible outcome will be obtained with a certain probability. Then $tr(A\rho)$ is the (statistical) average value of the outcomes in the experiment. Furthermore, $tr(\rho) - tr([\![S]\!](\rho))$ is indeed the probability that with input ρ program S does not terminate.

A Hoare-like logic for quantum **while**-programs was established in [37]. The proof system qPD for partial correctness is presented in Figure 2. Similar to the classical case, the notion of ranking function is needed to guarantee termination.

(Ax.Sk) $\{A\}\mathbf{skip}\{A\}$

(Ax.In) $\left\{\sum_n |n\rangle_q\langle 0|A|0\rangle_q\langle n|\right\} q := |0\rangle\{A\}$

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

(R.SC) $$\frac{\{A\}S_1\{B\} \quad \{B\}S_2\{C\}}{\{A\}S_1; S_2\{C\}}$$

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

(R.LP) $$\frac{\{A\}S\left\{M_0^\dagger BM_0 + M_1^\dagger AM_1\right\}}{\left\{M_0^\dagger BM_0 + M_1^\dagger AM_1\right\}\mathbf{while}\ M[\overline{q}] = 1\ \mathbf{do}\ S\ \mathbf{od}\{B\}}$$

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

Figure 2. Proof System qPD.

Definition 2.6 (Ranking functions [37]). *Consider quantum loop:*

$$\mathbf{while}\ \equiv\ \mathbf{while}\ M[\overline{q}] = 1\ \mathbf{do}\ S\ \mathbf{od}.$$

Let A be a quantum predicate in $\mathcal{H}_{\mathbf{while}}$ and real number $\epsilon > 0$. A function

$$t : \mathcal{D}(\mathcal{H}_{\mathbf{while}}) \to \mathbb{N}\ (\text{nonnegative integers})$$

is called a (A, ϵ)-ranking function of **while** *if it satisfies the following two conditions: for all $\rho \in \mathcal{D}(\mathcal{H}_{\mathbf{while}})$,*

1. $t\left([\![S]\!]\left(M_1\rho M_1^\dagger\right)\right) \leq t(\rho)$; *and*
2. $tr(A\rho) \geq \epsilon$ *implies* $t\left([\![S]\!]\left(M_1\rho M_1^\dagger\right)\right) < t(\rho)$.

The proof system qTD for total correctness is obtained from qPD by replacing rule (R.LP) with (R.LT) in Figure 3.

(R.LT) $$\frac{\begin{array}{l}\bullet\ \{A\}S\{M_0^\dagger BM_0 + M_1^\dagger AM_1\}\\ \bullet\ \text{for any } \epsilon > 0,\ t_\epsilon \text{ is a } (M_1^\dagger AM_1, \epsilon)\text{-ranking}\\ \qquad\text{function of } \mathbf{while}\end{array}}{\{M_0^\dagger BM_0 + M_1^\dagger AM_1\}\mathbf{while}\ M[\overline{q}] = 1\ \mathbf{do}\ S\ \mathbf{od}\{B\}}$$

Figure 3. Proof System qTD.

The soundness and (relative) completeness of both qPD and qTD were proved in [37].

Theorem 2.1 (Soundness and Completeness [37]). *For any quantum program S, and for any quantum predicates A, B:*

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

3 Reasoning about Projective Hoare Triples

Now we start to develop a variant aQHL (applied Quantum Hoare Logic) of QHL (Quantum Hoare Logic) presented in the above section.

3.1 Projective Measurement

To motivate this variant, let us consider a special kind of measurement in quantum physics. A projective measurement on a system with state Hilbert space $\mathcal{H}$ is described by a collection $\{P_m\}$ of projections over $\mathcal{H}$ satisfying $\sum_m P_m = I_{\mathcal{H}}$, where index m stands for the measurement outcomes that may occur in the experiment. If the state of a quantum system was ρ immediately before the measurement is performed on it, then the probability that outcome m occurs is $p_m = \text{tr}(P_m\rho)$, and the state of the system after the measurement is $\rho_m = P_m\rho P_m^\dagger/p_m$. Actually, a general measurement can always be implemented by a projective measurement together with a unitary transformation if an ancillary system is allowed. This fact enables us to restrict all the measurement in a program being projective. For example, in the circuit

model of quantum computation, measurement are usually assumed to be in the computational basis, a special kind of projective measurement.

For a mixed state (density operator) ρ, its support $\mathrm{supp}(\rho)$ is defined as the (topological) closure of the subspace spanned by the eigenvectors of ρ with nonzero eigenvalues. It is easy to see that $\mathrm{supp}(\rho) = \{|\varphi\rangle \in \mathcal{H} : \langle\varphi|\rho|\varphi\rangle = 0\}^{\perp}$, where $^{\perp}$ stands for ortho-complement. An important fact of projective measurements is that, given a state ρ and projection P such that $\mathrm{supp}(\rho) \subseteq P$, if we apply the (yes/no) projective measurement $\{P, I - P\}$ on ρ, the state is not changed. This enables us to check projective Hoare triple between two consecutive atomic steps of a program in testing and debugging.

3.2 Termination Space

Let us first consider termination problem of quantum programs. The interpretation of $tr(\rho) - tr([\![P]\!](\rho))$ given after Definition 2.5 directly leads to the following:

Definition 3.1. *We say that a quantum program S with input state ρ almost surely terminates if* $\mathrm{tr}([\![S]\!](\rho)) = \mathrm{tr}(\rho)$.

The next theorem gives a characterisation of termination:

Theorem 3.1. *For any quantum program S, there exists a closed subspace $T_{[S]}$ such that for any ρ:*

- *S with input ρ almost surely terminate if and only if* $\mathrm{supp}(\rho) \subseteq T_{[S]}$.

Proof. Suppose that the semantic function of S has the Kraus operator-sum representation: $[\![S]\!](\rho) = \sum_i E_i \rho E_i^{\dagger}$. Then for any $\rho \in \mathcal{D}(\mathcal{H}_S)$,

$$\mathrm{tr}([\![S]\!](\rho)) = \mathrm{tr}\Big(\sum_i E_i \rho E_i^{\dagger}\Big) = \mathrm{tr}\Big(\rho \sum_i E_i^{\dagger} E_i\Big).$$

Now we have:

$$\mathrm{tr}(\rho) - \mathrm{tr}([\![S]\!](\rho)) = \mathrm{tr}\Big(\rho\big(I_{\mathcal{H}_S} - \sum_i E_i^{\dagger} E_i\big)\Big)$$

where $I_{\mathcal{H}_S}$ is the identity operator on $\mathcal{H}$. Therefore, $\mathrm{tr}([\![S]\!](\rho)) = \mathrm{tr}(\rho)$ if and only if ρ is orthogonal to the positive semi-definite operator $I_{\mathcal{H}_S} - \sum E_i^{\dagger} E_i$. Define closed subspace

$$T_{[S]} = \Big(I_{\mathcal{H}} - \sum_i E_i^{\dagger} E_i\Big)^{\perp} = \big\{|\varphi\rangle : \big(I_{\mathcal{H}} - \sum_i E_i^{\dagger} E_i\big)|\varphi\rangle = 0\big\}.$$

We can observe that $T_{[S]}$ satisfies the wanted conditions. □

The subspace $T_{[S]}$ in the above theorem is called the termination space of program S.

For a finite-dimensional state Hilbert space, we have an algorithm to compute the termination space. To present it, we need the following simple generalisation of Theorem 3.1:

Proposition 3.1. *For any quantum program S and quantum operation (super-operator) $\mathcal{E}_1$, there exists a closed subspace $Q(S, \mathcal{E}_1)$ such that for any ρ:*

$$\lim_{n\to\infty} \mathrm{tr}((\mathcal{E}_{[\![S]\!]} \circ \mathcal{E}_1)^n(\rho)) = 0 \text{ iff } \mathrm{supp}(\rho) \subseteq Q(S, \mathcal{E}_1).$$

It is easy to see that $T_{[\mathbf{while}]} = Q(S, \mathcal{E}_1)$ for quantum loop

$$\mathbf{while} \equiv \mathbf{while}\ M[\overline{q}] = 1\ \mathbf{do}\ S\ \mathbf{od},$$

where quantum operation: $\mathcal{E}_1(\sigma) = M_1 \sigma M_1^{\dagger}$ for all σ. For simplicity, assume that $[\![S]\!]$ is trace preserving. We notice:

$$\lim_{n\to\infty} \mathrm{tr}((\mathcal{E}_{[\![S]\!]} \circ \mathcal{E}_1)^n(\rho)) = \lim_{n\to\infty} \mathrm{tr}((\mathcal{E}_1^* \circ [\![S]\!]^*)^n(I)\rho),$$

where I is the identity operator and $*$ stands for dual operation. Then it is routine to check that sequence

$$\big\{(\mathcal{E}_1^* \circ [\![S]\!]^*)^n(I)\big\}$$

of bounded operators is non-increasing with the Löwner order. On the other hand, the Löwner partial order in a separable Hilbert space is complete. Thus, this sequence converges to a bounded positive semi-definite operator, call it R. In particular, if the state Hilbert space is finite-dimensional, R can be calculated using Jordan decomposition. Furthermore, we have: $T_{[\mathbf{while}]} = Q(S, \mathcal{E}_1) = R^{\perp}$.

3.3 Correctness of Projective Hoare Triples

From now on, we only consider a special class of quantum Hoare triples $\{P\}S\{Q\}$, where both precondition P and post-condition Q are projections, and all measurements in program S are projective.

There is a one-to-one correspondence between the closed subspaces of a Hilbert space and projectors in it, and moreover, the inclusion between closed subspaces is coincident with the Löwner order between their projectors. So, we will not distinguish a closed subspace from the projection onto it. We write $\mathcal{S}(\mathcal{H})$ for the set of all closed subspaces of Hilbert space $\mathcal{H}$. Furthermore, let $^{\perp}$ stands for the orthocomplement, and for any P, Q, we define:

$$P \wedge Q = P \cap Q, \qquad P \vee Q = \overline{\mathrm{span}(P \cup Q)}$$

where $\overline{T}$ stands for the closure of T and $\mathrm{span}(T)$ for the subspace spanned by T. It is well-known that $(\mathcal{S}(\mathcal{H}), \wedge, \vee, ^{\perp})$ is an orthomodular lattice (or quantum logic) [8, 22], with inclusion $\subseteq$ as its order.

The restriction to projective Hoare triples can significantly simplify the definition of their correctness.

Definition 3.2. *Let $P \in \mathcal{S}(\mathcal{H})$ and $\rho \in \mathcal{D}(\mathcal{H})$. We say that ρ satisfies P, written $\rho \models P$, if* $\mathrm{supp}(\rho) \subseteq P$; *that is,* $P\rho = \rho$.

Definition 3.3.

1. *Projective Hoare triple $\{P\}S\{Q\}$ is true in the sense of partial correctness in aQHL, written:* $\models^{\mathrm{a}}_{\mathrm{par}} \{P\}S\{Q\}$, *if for all ρ:*

$$\rho \models P \Rightarrow [\![S]\!](\rho) \models Q.$$

2. *$\{P\}S\{Q\}$ is true in the sense of total correctness in aQHL, written:* $\models^{\mathrm{a}}_{\mathrm{tot}} \{P\}S\{Q\}$, *if for all ρ:*

$$\rho \models P \Rightarrow [\![S]\!](\rho) \models Q \ \&\ \mathrm{supp}(\rho) \subseteq T_{[S]}.$$

Several simple properties of partial and total correctness in aQHL can be immediately derived from the above definition.

Proposition 3.2.

1. $\models^{\mathrm{a}}_{\mathrm{tot}} \{P\}S\{Q\}$ *iff* $\models^{\mathrm{a}}_{\mathrm{par}} \{P\}S\{Q\}$ & $P \subseteq T_{[S]}$. *Moreover, if* $\models^{\mathrm{a}}_{\mathrm{par}} \{P\}S\{Q\}$, *then* $\models^{\mathrm{a}}_{\mathrm{tot}} \{P \wedge T_{[S]}\}S\{Q\}$.
2. *if* $\models^{\mathrm{a}}_{\mathrm{par}} \{P_i\}S\{Q_i\}$ $(i = 1, 2)$, *then*

$$\models^{\mathrm{a}}_{\mathrm{par}} \{P_1 \vee P_2\}S\{Q_1 \vee Q_2\},\ \models^{\mathrm{a}}_{\mathrm{par}} \{P_1 \wedge P_2\}S\{Q_1 \wedge Q_2\}.$$

The same holds for total correctness.

3.4 Lifting and Reduction

Now we consider the relationship between partial and total correctness $\models_{\mathrm{par}}, \models_{\mathrm{tot}}$ in QHL and $\models^{\mathrm{a}}_{\mathrm{par}}, \models^{\mathrm{a}}_{\mathrm{tot}}$ in aQHL. First of all, correctness in aQHL can be lifted to QHL:

Theorem 3.2 (Lifting Principle)**.** *For any* $P, Q \in \mathcal{S}(\mathcal{H})$*:*

1. *if* $\models^{\mathrm{a}}_{\mathrm{par}} \{P\}S\{Q\}$, *then* $\models_{\mathrm{par}} \{P\}S\{Q\}$*;*
2. *if* $\models^{\mathrm{a}}_{\mathrm{tot}} \{P\}S\{Q\}$, *then* $\models_{\mathrm{tot}} \{P\}S\{Q\}$.

Proof. The semantics $[\![S]\!]$ of a quantum program S is a quantum operation (or super-operator) $\mathcal{E}$ ([38], Proposition 3.3.5). We use $\mathcal{E}^*$ to denote the dual map of $\mathcal{E}$. Note that for any operator A with $0_{\mathcal{H}} \sqsubseteq A \sqsubseteq I_{\mathcal{H}}$, we have: $0_{\mathcal{H}} \sqsubseteq \mathcal{E}^*(A) \sqsubseteq I_{\mathcal{H}}$, where $0_{\mathcal{H}}$ and $I_{\mathcal{H}}$ are the zero and identity operator on the state Hilbert space $\mathcal{H}$.

1. Assume that $\models^{\mathrm{a}}_{\mathrm{par}} \{P\}S\{Q\}$. Then for any $\rho \in \mathcal{D}(\mathcal{H})$ such that $\rho \models P$, we have $[\![S]\!](\rho) \models Q$, or equivalently,

$$\forall\ \rho,\ \mathrm{tr}(P^\perp\rho) = 0 \implies \mathrm{tr}(Q^\perp\mathcal{E}(\rho)) = 0.$$

Furthermore, we obtain:

$$\begin{aligned}
&\forall\ \rho,\ \mathrm{tr}(P^\perp\rho) = 0 \implies \mathrm{tr}(Q^\perp\mathcal{E}(\rho)) = 0\\
\iff\ &\mathrm{supp}(\mathcal{E}^*(Q^\perp)) \subseteq \mathrm{supp}(P^\perp)\\
\iff\ &\mathcal{E}^*(I_{\mathcal{H}} - Q) \sqsubseteq I_{\mathcal{H}} - P\\
\iff\ &\forall\ \rho,\ \mathrm{tr}((I_{\mathcal{H}} - Q)\mathcal{E}(\rho)) \le \mathrm{tr}((I_{\mathcal{H}} - P)\rho)\\
\iff\ &\forall\ \rho,\ \mathrm{tr}(P\rho) \le \mathrm{tr}(Q[\![S]\!](\rho)) + \mathrm{tr}(\rho) - \mathrm{tr}([\![S]\!](\rho))\\
\iff\ &\models_{\mathrm{par}} \{P\}S\{Q\}.
\end{aligned}$$

2. Assume that $\models^{\mathrm{a}}_{\mathrm{tot}} \{P\}S\{Q\}$. Then for any $\rho \models P$, we have $[\![S]\!](\rho) \models Q$ and $\mathrm{tr}(\rho) = \mathrm{tr}([\![S]\!](\rho))$. Consequently, it holds that $\mathrm{tr}(\rho) = \mathrm{tr}(Q[\![S]\!](\rho))$, or equivalently, $\mathrm{tr}(\rho) = \mathrm{tr}(\mathcal{E}^*(Q)\rho)$. In other words, for any $|\psi\rangle \in P$, we have

$$\langle\psi|\mathcal{E}^*(Q)|\psi\rangle = \langle\psi|\psi\rangle.$$

Note the fact that $0_{\mathcal{H}} \sqsubseteq \mathcal{E}^*(Q) \sqsubseteq I_{\mathcal{H}}$. We now prove the following:

Claim: For any $|\phi\rangle \in P^\perp$, $\langle\psi|\mathcal{E}^*(Q)|\phi\rangle = 0$.

If the claim is not true, without any loss of generality, we can suppose that $|\psi\rangle$ and $|\phi\rangle$ are unit vectors and

$$\langle\psi|\mathcal{E}^*(Q)|\phi\rangle = x > 0, \qquad \langle\phi|\mathcal{E}^*(Q)|\phi\rangle = y \ge 0.$$

We restrict our attention to the closed subspace $\mathrm{span}\{|\psi\rangle, |\phi\rangle\}$, and note that $|\psi\rangle \perp |\phi\rangle$. Then in the basis $\mathcal{B} = \{|\psi\rangle, |\phi\rangle\}$, $\mathcal{E}^*(Q)$ has the form: $\begin{pmatrix} 1 & x \\ \bar{x} & y \end{pmatrix}$, which has two eigenvalues:

$$\lambda_1, \lambda_2 = \frac{y + 1 \pm \sqrt{(1-y)^2 + 4|x|^2}}{2}.$$

As $0_{\mathcal{H}} \sqsubseteq \mathcal{E}^*(Q) \sqsubseteq I_{\mathcal{H}}$, we must have $0 \le y \le 1$ and $0 \le \lambda_1, \lambda_2 \le 1$, and therefore $|x| = 0$, which leads to a contradiction. Now, for any state $|\varphi\rangle$, we can always write it as $|\varphi\rangle = |\psi\rangle + |\phi\rangle$, where $|\psi\rangle \in P$ and $|\phi\rangle \in P^\perp$, and then:

$$\begin{aligned}
\langle\varphi|\mathcal{E}^*(Q) - P|\varphi\rangle &= (\langle\psi| + \langle\phi|)(\mathcal{E}^*(Q) - P)(|\psi\rangle + |\phi\rangle)\\
&= \langle\psi|\mathcal{E}^*(Q)|\psi\rangle + \langle\psi|\mathcal{E}^*(Q)|\phi\rangle + \langle\phi|\mathcal{E}^*(Q)|\psi\rangle\\
&\quad + \langle\phi|\mathcal{E}^*(Q)|\phi\rangle - \langle\psi|P|\phi\rangle\\
&= \langle\psi|\psi\rangle + 0 + 0 + \langle\phi|\mathcal{E}^*(Q)|\phi\rangle - \langle\psi|\psi\rangle\\
&= \langle\phi|\mathcal{E}^*(Q)|\phi\rangle \ge 0.
\end{aligned}$$

This implies that $0 \sqsubseteq \mathcal{E}^*(Q) - P$, or $P \sqsubseteq \mathcal{E}^*(Q)$. Therefore, for any $\rho \in \mathcal{D}(\mathcal{H})$, we have:

$$\mathrm{tr}(P\rho) \le \mathrm{tr}(\mathcal{E}^*(Q)\rho) = \mathrm{tr}(Q\mathcal{E}(\rho)) = \mathrm{tr}(Q[\![S]\!](\rho)),$$

which means that $\models_{\mathrm{tot}} \{P\}S\{Q\}$. □

To see how QHL can be *partially* reduced to aQHL, for any positive semi-definite operator A, we write $E(A)$ for the eigenspace of A with eigenvalue 1: $E(A) = \{|\psi\rangle \in \mathcal{H} : A|\psi\rangle = |\psi\rangle\}$. Obviously, $E(A) \sqsubseteq A$, and if A itself is a projector, then $E(A) = A$. Then we have the following:

Theorem 3.3 (Reduction Principle)**.** *For any quantum predicate* P, Q *and program* S*:*

1. *if* $\models_{\mathrm{par}} \{P\}S\{Q\}$, *then* $\models^{\mathrm{a}}_{\mathrm{par}} \{E(P)\}S\{E(Q)\}$*;*
2. *if* $\models_{\mathrm{tot}} \{P\}S\{Q\}$, *then* $\models^{\mathrm{a}}_{\mathrm{tot}} \{E(P)\}S\{E(Q)\}$.

Proof. 1. If $\models_{\mathrm{par}} \{P\}S\{Q\}$, then for any $\rho \models E(P)$, we have $\mathrm{tr}(P\rho) = \mathrm{tr}(\rho)$. So, from the definition of $\models_{\mathrm{par}}$, we have:

$$\begin{aligned}
&\mathrm{tr}(P\rho) \le \mathrm{tr}(Q[\![S]\!](\rho)) + \mathrm{tr}(\rho) - \mathrm{tr}([\![S]\!](\rho))\\
\iff\ &\mathrm{tr}([\![S]\!](\rho)) \le \mathrm{tr}(Q[\![S]\!](\rho))\\
\iff\ &[\![S]\!](\rho) \models E(Q)
\end{aligned}$$

which yields $\models^{\mathrm{a}}_{\mathrm{par}} \{E(P)\}S\{E(Q)\}$

2. If $\models_{\mathrm{tot}} \{P\}S\{Q\}$, then for any $\rho \models E(P)$, we have $\mathrm{tr}(P\rho) = \mathrm{tr}(\rho)$. Therefore:

$$\mathrm{tr}(\rho) = \mathrm{tr}(P\rho) \le \mathrm{tr}(Q[\![S]\!](\rho)) \le \mathrm{tr}([\![S]\!](\rho)) \le \mathrm{tr}(\rho), \qquad (7)$$

Thus, all $\le$'s in equation (7) are saturated, and it holds that $[\![S]\!](\rho) \models E(Q)$ and $\mathrm{tr}([\![S]\!](\rho)) = \mathrm{tr}(\rho)$, which is exactly $\models^{\mathrm{a}}_{\mathrm{tot}} \{E(P)\}S\{E(Q)\}$. □

3.5 Proof System

The lifting and reduction rules proved in the last subsection (Theorems 3.2 and 3.3) can help us to obtain a proof system qPD-a for partial correctness and qTD-a for total correctness in aQHL simpler than that for QHL presented in Section 2.

3.5.1 Proof System for Partial Correctness

The system qPD-a consists of (Ax.Sk), (Ax.UT), (R.SC) and (R.Or) in Figure 2 (restricted to projective Hoare triples) together with (Ax.In-a), (R.IF-a) and (R.LP-a) in Figure 4, which are simplified from (Ax.In), (R.IF) and (R.LP) in QHL.

(Ax.In-a) $\{I_q \otimes \lceil Q \rceil\}\, q := |0\rangle \{Q\}$

(R.IF-a) $\dfrac{\{P_m\}\, S_m \{Q\} \text{ for all } m}{\{\bigvee_m (M_m \wedge P_m)\}\, \textbf{if } (\square m \cdot M[\overline{q}] = m \to S_m)\ \textbf{fi} \{Q\}}$

(R.LP-a) $\dfrac{\{P\} S \{(M_0 \wedge Q) \vee (M_1 \wedge P)\}}{\{(M_0 \wedge Q) \vee (M_1 \wedge P)\}\textbf{while } M[\overline{q}] = 1 \textbf{ do } S \textbf{ od} \{Q\}}$

(R.LT-a) $\dfrac{\begin{array}{c}\{P\} S \{(M_0 \wedge Q) \vee (M_1 \wedge P)\} \\ \text{for all } \epsilon > 0,\ t_\epsilon \text{ is a } ((M_0 \wedge Q) \vee (M_1 \wedge P), \epsilon)\text{-ranking} \\ \text{function of loop } \textbf{while}\end{array}}{\{(M_0 \wedge Q) \vee (M_1 \wedge P)\}\textbf{while } M[\overline{q}] = 1 \textbf{ do } S \textbf{ od} \{Q\}}$

Figure 4. Simplified Axioms and Inference Rules. In (Ax.In-a), $\lceil Q \rceil = \bigvee \{ \text{ closed subspaces } T : |0\rangle\langle 0|_q \otimes T \subseteq Q\}$

3.5.2 Proof System for Total Correctness

The definition of ranking functions can also be significantly simplified. For any $\rho \in \mathcal{D}(\mathcal{H})$ and $M \in \mathcal{S}(\mathcal{H})$, we write $\rho|_M$ for the restriction of ρ on M: $\rho|_M = M\rho M$.

Definition 3.4. *Let quantum loop* **while** $\equiv$ **while** $M[\overline{q}] = 1$ **do** S **od**, $Q \in \mathcal{S}(\mathcal{H}_{\textbf{while}})$ *and* $\epsilon > 0$. *A function* $t : \mathcal{D}(\mathcal{H}_{\textbf{while}}) \mapsto \mathbb{N}$ *is called a* (Q, ϵ)*-ranking function of* **while** *if for all* ρ *with* $\rho \models Q$*:*

1. $[\![S]\!](\rho|_{M_1}) \models Q$;
2. $t([\![S]\!](\rho|_{M_1})) \leq t(\rho)$; *and*
3. $\mathrm{tr}(\rho) \geq \epsilon$ *implies* $t([\![S]\!](\rho|_{M_1})) < t(\rho)$

It is worth comparing the above definition with the original definition of ranking functions (Definition 2.6). First, the conditions in the above definition are simpler than those in Definition 2.6. More importantly, the conditions in Definition 2.6 must be satisfied for all $\rho \in \mathcal{D}(\mathcal{H})$, but in the above definition, we only need to check the conditions for inputs ρ within Q, which might be much smaller than the whole state Hilbert space. As will be seen in verification of the HHL algorithm, this simplification makes finding a ranking function much easier, especially when the semantic function is difficult to represent for general inputs but is simple for those $\rho \models Q$.

The system qTD-a is then obtained from qPD-a by replacing rule (R.LP-a) by (R.LT-a), which is also given in Figure 4. We write $\vdash^{\mathrm{a}}_{\mathrm{qPD}}$ and $\vdash^{\mathrm{a}}_{\mathrm{qTD}}$ for provability in qPD-a and qTD-a, respectively.

3.5.3 Soundness and Completeness

The soundness and (relative) completeness for qPD-a and qTD-a are then established in the following:

Theorem 3.4 (Soundness and Completeness)**.** *For any projective Hoare triples* $\{P\}S\{Q\}$*:*

$$\models^{\mathrm{a}}_{\mathrm{par}} \{P\}S\{Q\} \Leftrightarrow \vdash^{\mathrm{a}}_{\mathrm{qPD}} \{P\}S\{Q\}.$$
$$\models^{\mathrm{a}}_{\mathrm{tot}} \{P\}S\{Q\} \Leftrightarrow \vdash^{\mathrm{a}}_{\mathrm{qTD}} \{P\}S\{Q\}.$$

Proof. This theorem can be derived from Theorem 2.1 using Theorems 3.2 and 3.3. □

It is interesting to compare the above theorem and Theorem 2.1: aQHL also enjoys (relative) completeness. In practical, when only projective preconditions and postconditions are allowed, e.g., in testing and debugging, aQHL is sufficient for use without the need for more complex rules of QHL.

4 Rules for Robustness Reasoning

In the last section, we provided a way to simplify proof rules and ranking functions in QHL so that QHL can be used to verify some larger programs. In this section, we increase applicability of QHL in a different direction. The proof systems given in the last section can be applied very well in reasoning about exact quantum algorithms: on any input data, they output the correct answer with certainty (probability 1) (see for example, [5, 26]). To make it also applicable to inexact quantum algorithms, we extend aQHL with several rules for reasoning about robustness of quantum programs.

For robustness reasoning, we always desire that the outcome ρ of a program is close to the ideal result ρ_I; that is, the distance between ρ and $\rho_I \leq \epsilon$ for some error bound ϵ, for example $\mathrm{tr}|\rho - \rho_I| \leq \epsilon$ using trace norm. It seems that such a condition cannot be expressed as a Hermitian operator in the original QHL. So, we introduce a notion of approximate satisfaction.

4.1 Approximate Satisfaction

Let us first introduce several notations. For any states $\sigma \in \mathcal{D}(\mathcal{H})$, we define the *ball* with centre σ and radius ϵ as:

$$B(\sigma, \epsilon) = \{\rho \in \mathcal{D}(\mathcal{H}) : \mathrm{tr}(\rho) = \mathrm{tr}(\sigma), D(\rho, \sigma) \leq \epsilon\}$$

where $D(\rho, \sigma) = \frac{1}{2}\mathrm{tr}|\rho - \sigma|$ is the trace distance between ρ and σ. We can further define the distance between two subspace P, Q as follows:

$$D(P, Q) = \sup_{\rho \models P} \inf_{\sigma \models Q} D(\rho, \sigma),$$

where ρ and σ range over all density operators.

Definition 4.1. *Let* $P \in \mathcal{S}(\mathcal{H})$ *be a projection in (or a closed subspace of)* $\mathcal{H}$ *and* $\epsilon \geq 0$. *We define convex set:*

$$(P, \epsilon) := \bigcup_{\sigma \models P} B(\sigma, \epsilon) = \{\rho \in \mathcal{D}(\mathcal{H}) : \exists\, \sigma \models P \text{ s.t. } \mathrm{tr}(\rho) = \mathrm{tr}(\sigma) \text{ and } D(\rho, \sigma) \leq \epsilon\}.$$

It is easy to see that inclusion relation $\subseteq$ is a partial order over set $S = \{(P, \epsilon) : 0_{\mathcal{H}} \sqsubset P \sqsubset I_{\mathcal{H}}, 0 \leq \epsilon < 1\} \cup$

$\{(0_{\mathcal{H}}, 0), (I_{\mathcal{H}}, 1)\}$. In particular, for any $(P, \epsilon), (Q, \delta) \in S$, $(P, \epsilon) = (Q, \delta)$ if and only if $P = Q,\ \epsilon = \delta$.

Now we are ready to introduce a key notion of approximate satisfaction in this section:

Definition 4.2. *We say state ρ approximately satisfies (projective) predicate P with error parameter ϵ, written: $\rho \models_\epsilon P$, if $\rho \in (P, \epsilon)$, i.e., there exists a σ with the same trace that $\sigma \models P$ and $D(\rho, \sigma) \leq \epsilon$.*

4.2 Robust (Projective) Hoare Triples

With the preparation in the above subsection, we can define the notion of robust Hoare triple and its truth:

Definition 4.3.
1. *A robust (projective) Hoare triple is a formula of the form: $\{(P, \epsilon)\}S\{(Q, \delta)\}$, where P, Q are projections, S is a program, and $0 \leq \epsilon, \delta < 1$.*
2. *$\{(P, \epsilon)\}S\{(Q, \delta)\}$ is true, written: $\models_{\text{rt}} \{(P, \epsilon)\}S\{(Q, \delta)\}$, if for all $\rho \in D(\mathcal{H})$:*

$$\rho \models_\epsilon P \ \Rightarrow\ [\![S]\!](\rho) \models_\delta Q.$$

Note that predicates P, Q in Definitions 4.1, 4.2 and 4.3 are all projective. Indeed, it is not clear how to define a notion similar to Definition 4.1 for a general Hermitian operator. So, in the next subsection, we are only able to extend projective QHL (rather than the original QHL with general Hermitian operators) for robustness reasoning.

4.3 Inference Rules

To present our rule for reasoning about robust Hoare triples, we need some very technical preparations. First, let recall a definition from [20].

Definition 4.4 (Bounded Loops [20])**.** *Let $0 \leq a \leq 1$ and integer $n \geq 1$. Quantum loop* **while** $M[\overline{q}] = 1$ **do** S **od** *is said to be (a, n)-bounded if $(\mathcal{E}^*)^n(M_1^\dagger M_1) \sqsubseteq a M_1^\dagger M_1$, where linear map $\mathcal{E}$ is defined by $\mathcal{E}(\rho) := [\![S]\!](M_1 \rho M_1^\dagger)$ for all ρ, and $\mathcal{E}^*$ is the dual of $\mathcal{E}$.*

The next definition also requires the notion of quantum program schemes, which can be defined as usual by adding procedure identifiers, ranged over by $X, Y, Z, \ldots$, to the syntax of quantum **while**-programs.

Definition 4.5. *Let X be a procedure identifier and S a program scheme. Then the counter* $\text{Count}(S, X)$ *of X in C is defined by induction as follows:*

1. *if $S \equiv X$, then* $\text{Count}(S, X) = 1$;
2. *if $S \equiv$* **skip**, *$S \equiv q := |0\rangle$ or $S \equiv q := U[q]$, then* $\text{Count}(S, X) = 0$;
3. *if $S \equiv S_1; S_2$, then* $\text{Count}(S, X) = \text{Count}(S_1, X) + \text{Count}(S_2, X)$;
4. *if $S \equiv$* **if** $(\square m \cdot M[\overline{q}] = m \rightarrow S_m)$ **fi**, *then* $\text{Count}(S, X) = \max_m \text{Count}(S_m, X)$;
5. *if $S \equiv$* **while** $M[\overline{q}] = 1$ **do** S' **od** *and S is (a, n)-bounded, then* $\text{Count}(S, X) = \frac{n}{1-a}\text{Count}(S', X)$.

$$\text{(R.We)}\quad \frac{\{(P, \epsilon)\}S\{(Q, \delta)\} \qquad \theta \geq 0}{\{(P, \epsilon + \theta)\}S\{(Q, \delta + \theta)\}}$$

$$\text{(R.No)}\quad \frac{\{(P, \epsilon)\}S\{(Q, \delta)\} \qquad \|[\![X]\!] - [\![Y]\!]\|_\diamond \leq \theta \qquad \text{Count}(S, X) \leq n}{\{(P, \epsilon)\}S[Y/X]\{(Q, \delta + n\theta)\}}$$

$$\text{(R.SN)}\quad \frac{\{(P, \epsilon)\}S_1\{(Q, \delta)\} \qquad \{(Q, \delta)\}S_2\{(R, \theta)\}}{\{(P, \epsilon)\}S_1; S_2\{(R, \theta)\}}$$

$$\text{(R.ON)}\quad \frac{\{(P', \epsilon')\}S\{(Q', \delta')\} \qquad (P, \epsilon) \subseteq (P', \epsilon') \qquad (Q', \delta') \subseteq (Q, \delta)}{\{(P, \epsilon)\}S\{(Q, \delta))\}}$$

Figure 5. Inference Rules for Robust Reasoning. In rule (R.No), $\|[\![X]\!] - [\![Y]\!]\|_\diamond$ is the diamond norm between the semantic functions $[\![X]\!]$ and $[\![Y]\!]$, and S[Y/X] stands for the program scheme obtained by substituting all occurrences of X in S with Y.

Intuitively, Count(S,X) is used to count the number of calls of subprogram X in the whole program S.

Now, we are able to build a proof system qR for robustness reasoning. The system qR consists of all axioms and inference rules of aQHL presented in the last section together with the rules (R.We), (R.No), (R.SN) and (R.ON) in Figure 5. Some of these rules deserve careful explanations. Note that $\rho \models P$ iff $\rho \in (P, 0)$ iff $\rho \models_0 P$. So, we can make the convention $P = (P, 0)$. When ϵ and δ are chosen to be 0, then a special case of rule (R.We) is:

$$\frac{\{P\}S\{Q\} \quad \theta \geq 0}{\{(P, \theta)\}S\{(Q, \theta)\}}$$

Thus, the proof system for projective Hoare triples can be embedded into that for robust Hoare triples. The diamond distance in rule (R.No) is widely used in quantum computation and quantum information science and defined as follows: for any two quantum operations $\mathcal{E}$ and $\mathcal{F}$ on Hilbert space $\mathcal{H}$,

$$\|\mathcal{E} - \mathcal{F}\|_\diamond = \sup_\rho D\big((\mathcal{E} \otimes I_{\mathcal{H}'})(\rho), (\mathcal{F} \otimes I_{\mathcal{H}'})(\rho)\big)$$

where $I_{\mathcal{H}'}$ is the identity operation on $\mathcal{H}'$, the supremum is taken over all separable Hilbert space $\mathcal{H}'$, and density operator ρ is chosen from $\mathcal{D}(\mathcal{H} \otimes \mathcal{H}')$.

We present soundness of the proof system for robustness reasoning as the following:

Theorem 4.1. *(Soundness) The proof system for robustness reasoning is sound for terminating quantum programs: for any terminating quantum program S, projective quantum predicates $P, Q \in \mathcal{S}(\mathcal{H}_{\text{all}})$ and $\epsilon, \varepsilon \geq 0$, we have:*

$$\vdash_{\text{qR}} \{(P, \epsilon)\}S\{(Q, \delta)\} \text{ implies } \models_{\text{rt}} \{(P, \epsilon)\}S\{(Q, \delta)\}.$$

Proof. As usual, it suffices to prove soundness of the rules in Figure 5. But the calculation in the proof; in particular for rule (R.NO), is very involved. □

5 Verification of the HHL Algorithm

The purpose of this and the next section is to show the effectiveness of the proof systems presented in previous sections by employing them to verify the correctness of two major quantum algorithms that form the cornerstone of quantum machine learning.

In this section, we give a verification of quantum algorithm for linear systems of equations (known as the HHL algorithm, named after Harrow, Hassidim and Lloyd) [18], using the proof systems qPD-a and qTD-a established in Section 3. Solving linear systems of equations is a fundamental problem in almost all fields of science: given a matrix A and a vector $\vec{b}$, find a vector $\vec{x}$ such that $A\vec{x} = \vec{b}$. The HHL algorithm [18] was proposed for solving linear systems by providing a state $|x\rangle$ corresponding to $\vec{x}$ rather than give a classical characterisation of $\vec{x}$. When A is sparse and has a low condition number κ, then the algorithm has a runtime of $O(\log(N)\kappa^2)$ where N is the number of linear equations, which offers an exponential speedup over the fastest classical algorithm.

5.1 The HHL Algorithm

To simplify the presentation, let us assume A is Hermitian and full-rank with dimension $N = 2^m$ and therefore it is possible to apply the transform $e^{\iota At_0}$ for a given time t_0. Then A is diagonalizable with the form $A = \sum_{j=1}^{N} \lambda_j |u_j\rangle\langle u_j|$, where λ_j and $|u_j\rangle$ are the corresponding eigenvalues and eigenvectors. To make the algorithm exact, we presume that, for all $0 \leq j \leq N$, $\lambda_j t_0$ is a multiple of 2π: $\delta_j = \frac{\lambda_j t_0}{2\pi} \in \mathbb{N}^+$. We also suppose that the input vector $\vec{b}$ can be prepared efficiently; that is, there is a unitary operator U_b which can efficiently transform $|0\rangle$ into: $|b\rangle = \sum_{i=1}^{N} b_i|i\rangle$. The scale of $\vec{b}$ is not essential as we only care about the solution $\vec{x}$ up to some unimportant scale factor. Thus we can assume $\sum_{i=1}^{N} |b_i|^2 = 1$, and $|b\rangle$ is a unit vector. Moreover, $|b\rangle$ can always be written as a linear combination of $|u_j\rangle$ with complex numbers β_j: $|b\rangle = \sum_{j=1}^{N} \beta_j |u_j\rangle$. The phase estimation algorithm is employed and a control system is needed. We chose a proper dimension T of the control system, $T = 2^n$ where $n = \lceil \max_j \delta_j \rceil$, which ensures that the phase estimation succeed with probability 1 if the initial state of control system is $|0\rangle^{\otimes n}$. Actually, the original HHL algorithm uses a more complex initial state of control system, which minimize a certain quadratic loss function. But as we only want to show the key ideas of the algorithm, a simpler state $|0\rangle^{\otimes n}$ is used as it can make the algorithm output an exact solution state under the assumption of A and t_0 stated above. Given the above A and $|b\rangle$, it is easy to calculate the solution for the linear equation $A|x\rangle = |b\rangle$:

$$|x\rangle = c \sum_{j=1}^{N} \frac{\beta_j}{\lambda_j} |u_j\rangle,$$

up to some unimportant scale factor where c is only used to normalize $|x\rangle$.

$$
\begin{aligned}
&p := |0\rangle^{\otimes n};\\
&q := |0\rangle^{\otimes m};\\
&r := |0\rangle;\\
&\textbf{while } M[r] = 1 \textbf{ do } D \textbf{ od}
\end{aligned}
$$

Figure 6. HHL – quantum algorithm for linear systems of equations.

$$
\begin{aligned}
&q := |0\rangle^{\otimes m};\\
&q := U_b[q];\\
&p := H^{\otimes n}[p];\\
&p, q := U_f[p, q];\\
&p := \mathrm{QFT}^{-1}[p];\\
&p, r := U_c[p, r];\\
&p := \mathrm{QFT}[p];\\
&p, q := U_f^{\dagger}[p, q];\\
&p := H^{\otimes n}[p]
\end{aligned}
$$

Figure 7. Loop body D in HHL algorithm.

5.2 Program HHL

The HHL algorithm can be written as a quantum program in the quantum **while**-language defined in Section 3; see Figure 6. The register p is an n-qubit system with $2^n = T$, which is used as a control system in the phase estimation step, while q is an m-qubit system which stores the vector b, in the sense of a corresponding quantum state $|b\rangle = \sum_i b_i|i\rangle$. The last register r is an one qubit system and it is the indicator of the while loop. The measurement $M = \{M_0, M_1\}$ in the loop is the simplest "yes-no" measurement: $M_0 = |1\rangle_r\langle 1|$ and $M_1 = |0\rangle_r\langle 0|$. The loop body D in Figure 6 is displayed in Figure 7. The unitary operator U_b is a given operator which generates the input vector b; that is,

$$U_b|0\rangle^{\otimes m} = |b\rangle = \sum_{i=1}^{N} b_i|i\rangle.$$

U_f is a controlled unitary operator whose control system is p and target system is q; more precisely,

$$U_f = \sum_{\tau=0}^{T-1} |\tau\rangle_p\langle\tau| \otimes e^{\iota A\tau t_0/T}.$$

QFT and QFT^{-1} are the quantum Fourier transform and the inverse quantum Fourier transform applied to the control register p; more precisely

$$\mathrm{QFT} : |k\rangle \mapsto \frac{1}{\sqrt{T}} \sum_{\tau=0}^{T-1} e^{2\pi\iota\tau k/T}|\tau\rangle, \quad k = 0, 1, \cdots, T-1,$$

$$\mathrm{QFT}^{-1} : |k\rangle \mapsto \frac{1}{\sqrt{T}} \sum_{\tau=0}^{T-1} e^{-2\pi\iota\tau k/T}|\tau\rangle, \quad k = 0, 1, \cdots, T-1.$$

U_c is a controlled unitary which operates on the control register p and target register r. Formally, U_c is the transform with some proper parameter C:

$$U_c : |0\rangle_p|0\rangle_r \mapsto |0\rangle_p|0\rangle_r$$
$$|i\rangle_p|0\rangle_r \mapsto |i\rangle_p \left(\sqrt{1 - \frac{C^2}{i^2}}|0\rangle_r + \frac{C}{i}|1\rangle_r\right) \quad 1 \le i \le T-1$$

5.3 Partial Correctness

Now the partial correctness of program HHL can be stated as the following projective Hoare triple:

$$\models^{\mathrm{a}}_{\mathrm{par}} \{I_p \otimes I_q \otimes I_r\}\mathrm{HHL}\{|0\rangle_p\langle 0| \otimes |x\rangle_q\langle x| \otimes |1\rangle_r\langle 1|\}, \quad (8)$$

in the sense that for any possible input state, the output of the program is always $|0\rangle_p|x\rangle_q|1\rangle_r$, where the state of register q is the desired solution and not entangled with other registers.

Before proving (8), let us first derive:

$$\vdash^{\mathrm{a}}_{\mathrm{qPD}} \{|0\rangle_p\langle 0| \otimes I_q \otimes |0\rangle_r\langle 0|\}D \{|0\rangle_p\langle 0| \otimes (|x\rangle_q\langle x| \otimes |1\rangle_r\langle 1| + I_q \otimes |0\rangle_r\langle 0|)\},$$

for the loop body D in Figure 7. For simplicity, we use following notations:

$$P = |0\rangle_p\langle 0| \otimes I_q \otimes |0\rangle_r\langle 0|;$$
$$Q = |0\rangle_p\langle 0| \otimes |x\rangle_q\langle x| \otimes |1\rangle_r\langle 1|;$$
$$R = |0\rangle_p\langle 0| \otimes (|x\rangle_q\langle x| \otimes |1\rangle_r\langle 1| + I_q \otimes |0\rangle_r\langle 0|);$$
$$|v_j\rangle_r = \sqrt{1 - \frac{C^2}{\delta_j^2}}|0\rangle_r + \frac{C}{\delta_j}|1\rangle_r, \quad \text{for } 1 \le j \le N$$

Using the (Ax.In-a), (Ax.UT) and (R.SC), we have:

$$\vdash^{\mathrm{a}}_{\mathrm{qPD}} \{P\}D\left\{|0\rangle_p\langle 0| \otimes \left[\sum_{j,j'=1}^{N} (\beta_j|u_j\rangle_q|v_j\rangle_r)(\bar{\beta}_{j'q}\langle u_{j'}|_r\langle v_{j'}|)\right]\right\}$$

Moreover, we can also relate the post predicate in the above equation to R. To do this, we first write the explicit form of the state

$$\sum_{j=1}^{N} \beta_j|u_j\rangle_q|v_j\rangle_r = \sum_{j=1}^{N} \beta_j|u_j\rangle_q\left(\sqrt{1 - \frac{C^2}{\delta_j^2}}|0\rangle_r + \frac{C}{\delta_j}|1\rangle_r\right)$$
$$= \sum_{j=1}^{N} c_1|x\rangle_q|1\rangle_r + \sum_{j=1}^{N} \beta_j\sqrt{1 - \frac{C^2}{\delta_j^2}}|u_j\rangle_q|0\rangle_r$$

where c_1 is some constant. Note that $|x\rangle_q|1\rangle_r \in |x\rangle_q\langle x| \otimes |1\rangle_r\langle 1|$ and

$$\sum_{j=1}^{N} \beta_j\sqrt{1 - \frac{C^2}{\delta_j^2}}|u_j\rangle_q|0\rangle_r \in I_q \otimes |0\rangle_r\langle 0|,$$

and two subspaces $|x\rangle_q\langle x| \otimes |1\rangle_r\langle 1|$ and $I_q \otimes |0\rangle_r\langle 0|$ are orthogonal. Then $\sum_{j=1}^{N} \beta_j|u_j\rangle_q|v_j\rangle_r$ is in the subspace of $|x\rangle_q\langle x| \otimes |1\rangle_r\langle 1| + I_q \otimes |0\rangle_r\langle 0|$. So, we have:

$$|0\rangle_p\langle 0| \otimes \left[\sum_{j,j'=1}^{N} (\beta_j|u_j\rangle_q|v_j\rangle_r)(\bar{\beta}_{j'q}\langle u_{j'}|_r\langle v_{j'}|)\right]$$
$$\sqsubseteq |0\rangle_p\langle 0| \otimes (|x\rangle_q\langle x| \otimes |1\rangle_r\langle 1| + I_q \otimes |0\rangle_r\langle 0|) = R, \quad (9)$$

and using rule (R.Or), we obtain: $\vdash^{\mathrm{a}}_{\mathrm{qPD}} \{P\}D\{R\}$. Because $R = (M_0 \wedge Q) \vee (M_1 \wedge P)$, we can further apply (R.LP-a) to conclude:

$$\vdash^{\mathrm{a}}_{\mathrm{qPD}} \{R\}\textbf{while } M[r] = 1 \textbf{ do } D \textbf{ od } \{Q\}$$

The following is immediate from (Ax.In-a) and (R.SC):

$$\vdash^{\mathrm{a}}_{\mathrm{qPD}} \{I_p \otimes I_q \otimes I_r\}p := |0\rangle^{\otimes n}; q := |0\rangle^{\otimes m}; r := |0\rangle \{|0\rangle_p\langle 0| \otimes |0\rangle_q\langle 0| \otimes |0\rangle_r\langle 0|\},$$

and using (R.Or) yields:

$$\vdash^{\mathrm{a}}_{\mathrm{qPD}} \{I_p \otimes I_q \otimes I_r\}p := |0\rangle^{\otimes n}; q := |0\rangle^{\otimes m}; r := |0\rangle \{R\},$$

as $|0\rangle_p\langle 0| \otimes |0\rangle_q\langle 0| \otimes |0\rangle_r\langle 0| \sqsubseteq R$. Now, it follows that

$$\vdash^{\mathrm{a}}_{\mathrm{qPD}} \{I_p \otimes I_q \otimes I_r\}\mathrm{HHL}\{Q\},$$

and we complete the proof of partial correctness.

5.4 Total Correctness

We can further prove the total correctness of HHL:

$$\models^{\mathrm{a}}_{\mathrm{tot}} \{I_p \otimes I_q \otimes I_r\}\mathrm{HHL}\{|0\rangle_p\langle 0| \otimes |x\rangle_q\langle x| \otimes |1\rangle_r\langle 1|\}.$$

which implies that, for any input state, HHL terminates and the state stored in the register q of the output is the desired solution state $|x\rangle$. Actually, we only need to show the existence of (R, ϵ)-ranking function of quantum loop: **while** $\equiv$ **while** $M[r] = 1$ **do** D **od** for any $\epsilon > 0$. Here, we adopt the simplified definition of ranking function for aQHL (Definition 3.4). If we use the original definition of ranking function (Definition 2.6), the calculation will be much more complicated. Let us first define a constant $c_2 \in (0, 1)$:

$$c_2 = \sum_{j=1}^{N} \beta_j\bar{\beta}_j\left(1 - \frac{C^2}{\delta_j^2}\right),$$

$$
\begin{aligned}
X[p,q,r,a] \equiv\ & r := |0\rangle; a := |0\rangle; \\
& r, a := U_\rho[r, a]; \\
& a := |0\rangle; \\
& p, q, r := CU[p, q, r]; \\
& r := |0\rangle;
\end{aligned}
$$

$$
\begin{aligned}
Y[p,q,r,a] \equiv\ & r := |0\rangle; \\
& a := |0\rangle; \\
& p, q := CV[p, q];
\end{aligned}
$$

Figure 8. Subprogram X and ideal subprogram Y. Both of them are terminating.

and a function $f : \mathcal{D}(\mathcal{H}_{all}) \mapsto [0, 1]$:

$$f(\rho) = \mathrm{tr}(\rho I_p \otimes I_q \otimes |0\rangle_r\langle 0|).$$

Then for any $\epsilon > 0$, define function $t_\epsilon : \mathcal{D}(\mathcal{H}_{all}) \mapsto \mathbb{N}$:

$$t_\epsilon(\rho) = \begin{cases} 0, & f(\rho) < \epsilon; \\ \left\lceil \log_{c_2} \dfrac{\epsilon}{f(\rho)} \right\rceil + 1, & f(\rho) \geq \epsilon. \end{cases}$$

It is easy to check that t_ϵ is a (R, ϵ)-ranking function of loop **while**.

6 Verification of qPCA

Principal component analysis (PCA) is one of the fundamental tool used in data analysis, that can be used to reduce a large set of data into a small set while most of the information is still preserved. Mathematically, given a symmetric matrix A, it returns the first n components, that is, the top n largest eigenvalues together with their eigenvectors.

The quantum algorithm for principal component analysis (qPCA) was proposed by Lloyd, Mohseni and Rebentrost in [25], which offers a exponentially speed up over the best classical algorithm when the unknown density matrix is of low-rank. This algorithm opens a new door to the field of quantum machine learning. In this subsection, we present a proof of the correctness of qPCA using aQHL together with the rules for robustness reasoning presented in Section 4.

6.1 The qPCA Algorithm as a Quantum Program

Suppose a given unknown density operator ρ of dimension n_ρ has the spectral decomposition: $\rho = \sum_j r_j |\alpha_j\rangle\langle\alpha_j|$, where r_j and $|\alpha_j\rangle$ are the corresponding eigenvalues and eigenvectors. Let us first consider two different programs X and Y with input registers p, q, r and r_a shown in Figure 8. Here, p is a qubit serving as the control register, q and r are two registers of dimension n_ρ used to store the density operator ρ, and a is the ancilla register which is used to prepare ρ; more precisely, a unitary operator U_ρ satisfies: $U_\rho|0\rangle_r|0\rangle_a = |\psi_\rho\rangle_{r,a}$, where $|\psi_\rho\rangle_{r,a}$ is a purification of ρ, or

$$
\begin{aligned}
\mathrm{qPCA} \equiv\ & p_1 := |0\rangle; \cdots; p_N := |0\rangle; \\
& p_1 := H[p_1]; \cdots; p_N := H[p_N]; \\
& p_1, q, r, a := X[p_1, q, r, a]; \\
& p_2, q, r, a := X^2[p_2, q, r, a]; \\
& \vdots \\
& p_N, q, r, a := X^{2^{(N-1)}}[p_N, q, r, a]; \\
& p_1, p_2, \cdots, p_N := \mathrm{QFT}[p_1, p_2, \cdots, p_N]
\end{aligned}
$$

Figure 9. Program qPCA. X is the subprogram given in Figure 8. The superscript of X denotes the number of repetitions of X.

in other words, $\mathrm{tr}_a(|\psi_\rho\rangle_{r,a}\langle\psi_\rho|) = \rho$. CU is the controlled swap unitary defined by

$$CU[p, q, r] = |0\rangle_p\langle 0| \otimes I_{q,r} + |1\rangle_p\langle 1| \otimes e^{-iS\Delta t}$$

where Δt is a small time interval, $I_{q,r}$ is the identity operator of registers p, r, and S is the swap gate, that is, for any pure states $|\psi\rangle_q$ and $|\phi\rangle_r$: $S|\psi\rangle_q|\phi\rangle_r = |\phi\rangle_r|\psi\rangle_p$. CV is also a controlled unitary which has the form:

$$CV[p, q] = |0\rangle_p\langle 0| \otimes I_q + |1\rangle_p\langle 1| \otimes e^{-i\rho\Delta t}.$$

In [25], it was shown that X and Y are closed in the sense of trace norm

$$\|[\![X]\!] - [\![Y]\!]\|_{\mathrm{tr}} = O(1)\Delta t^2.$$

However, as the whole system may be larger than the subsystems X and Y applied to, it is indeed not appropriate to use trace norm to bound the error. Generally, diamond norm can be as large as m times of trace norm where m is the dimension of the system the quantum operation applied to; that it, $\|[\![X]\!] - [\![Y]\!]\|_\diamond$ may be as large as $n_\rho O(1)\Delta t^2$, which implies that the error depends on the dimension n_ρ of ρ. But a tedious calculation yields $\|[\![X]\!] - [\![Y]\!]\|_\diamond = O(1)\Delta t^2$, which does not depend on n_ρ.

Now we can write the qPCA algorithm as a quantum program in Figure 9. The ideal version qPCA′ is the same as qPCA except that all occurrences of subprogram X is replaced by Y:

$$\mathrm{qPCA}' = \mathrm{qPCA}[Y/X].$$

Here, $p_1, p_2, \cdots, p_N$ are qubit variables, and q, r, a are registers used to store the input state and produce density operator ρ.

6.2 Exact Correctness of qPCA′

Repeatedly applying rules (Ax.In-a), (Ax.UT) and (Ax.SC), we can prove the correctness of qPCA′:

$$
\begin{aligned}
\vdash^{\mathrm{a}}_{\mathrm{qTD}} \{ & I_{\bar{p}} \otimes |\alpha_j\rangle_q\langle\alpha_j| \otimes I_r \otimes I_a\} \mathrm{qPCA}' \\
& \{|\phi\rangle_{\bar{p}}\langle\phi| \otimes |\alpha_j\rangle_q\langle\alpha_j| \otimes |0\rangle_r\langle 0| \otimes |0\rangle_a\langle 0|\}
\end{aligned}
$$

where $\bar{p} = p_1, p_2, \cdots, p_N$, operators $I_{\bar{p}}, I_r, I_a$ are identity mapping over registers $\bar{p}, r$ and a, respectively, $T = 2^N$ and

$$|\phi\rangle = \sum_{\tau=0}^{T-1} \sum_{x=0}^{T-1} \frac{1}{T} e^{-\iota(r_j \Delta t - 2\pi\tau/T)x} |\tau\rangle = \sum_{\tau=0}^{T-1} f(r_j, \tau)|\tau\rangle,$$

$$f(r_j, \tau) = \sum_{x=0}^{T-1} \frac{1}{T} e^{-\iota(r_j \Delta t - 2\pi\tau/T)x}.$$

6.3 Approximate Correctness of qPCA

It is easy to compute that $\mathrm{Count}(\mathrm{qPCA}, X) = T - 1$. Then using rule (R.No), we have the following correctness formula for qPCA:

$$\vdash_{\mathrm{qR}} \{(I_{\bar{p}} \otimes |\alpha_j\rangle_q\langle\alpha_j| \otimes I_r \otimes I_a, 0)\}\mathrm{qPCA}\{(|\phi\rangle_{\bar{p}}\langle\phi| \otimes |\alpha_j\rangle_q\langle\alpha_j| \otimes |0\rangle_r\langle 0| \otimes |0\rangle_a\langle 0|, (T-1)O(1)\Delta t^2)\}.$$

because $\|[\![X]\!] - [\![Y]\!]\|_\diamond = O(1)\Delta t^2$. For eigenvalue r_j of ρ, we assume that

$$\frac{r_j \Delta t T}{2\pi} = \widetilde{\delta_j} = \delta_j + \Delta_j$$

with $\delta_j \in \mathbb{N}$ and $0 \leq |\Delta_j| \leq \frac{1}{2}$. If we only care whether $|\phi\rangle$ is close to the correct eigenvalue, we might consider the subspace Q_j of $\mathcal{H}_{\bar{p}}$:

$$Q_j = \sum_{\delta_j - \Delta < \tau < \delta_j + \Delta} |\tau\rangle_{\bar{p}}\langle\tau|$$

in the sense that, for any state that is in Q_j, if we measure it using the computational basis and obtain the result m_j, then we can use $\frac{2\pi m_j}{\Delta t T}$ to approximate r_j, with an error at most $\frac{2\pi\Delta}{\Delta t T}$. Actually, Q_j satisfies the property:

$$D(|\phi\rangle_{\bar{p}}\langle\phi|, Q_j) \leq \frac{1}{\sqrt{2\Delta - 1}}.$$

Note that $|\alpha_j\rangle\langle\alpha_j| \otimes |0\rangle_r\langle 0| \otimes |0\rangle_a\langle 0|$ is a rank 1 subspace, so

$$D(|\phi\rangle_{\bar{p}}\langle\phi| \otimes |\alpha_j\rangle_q\langle\alpha_j| \otimes |0\rangle_r\langle 0| \otimes |0\rangle_a\langle 0|, Q_j \otimes |\alpha_j\rangle_q\langle\alpha_j| \otimes |0\rangle_r\langle 0| \otimes |0\rangle_a\langle 0|) = D(|\phi\rangle_{\bar{p}}\langle\phi|, Q_j).$$

Using the rule (R.ON) and the soundness of qR (Theorem 4.1), we conclude that:

$$\models_{\mathrm{rt}} \{(I_{\bar{p}} \otimes |\alpha_j\rangle_q\langle\alpha_j| \otimes I_r \otimes I_a, 0)\}\mathrm{qPCA}'\left\{\left(Q_j \otimes |\alpha_j\rangle_q\langle\alpha_j| \otimes |0\rangle_r\langle 0| \otimes |0\rangle_a\langle 0|, (T-1)O(1)\Delta t^2 + \frac{1}{\sqrt{2\Delta - 1}}\right)\right\};$$

that is, for the input state $|\alpha_j\rangle\langle\alpha_j|$ of register q, there exists a density operator $\sigma_j \in \mathcal{D}(\mathcal{H})$ satisfies $\sigma_j \models Q_j$, such that:

$$D([\![\mathrm{qPCA}]\!](|\alpha_j\rangle_q\langle\alpha_j|), \sigma_j \otimes |\alpha_j\rangle_q\langle\alpha_j|) \leq (T-1)O(1)\Delta t^2 + \frac{1}{\sqrt{2\Delta - 1}}.$$

The linearity of the quantum program ensures that, if the input state of register q is ρ itself, then there exists a sequence of density operator $\sigma_j \models Q_j$ which can be used to approximate the eigenvalue r_j with an error at most $\frac{2\pi\Delta}{\Delta t T}$, and the output of the qPCA is closed to the state $\sum_j r_j \sigma_j \otimes |\alpha_j\rangle\langle\alpha_j|$:

$$D([\![\mathrm{qPCA}]\!](\rho), \sum_j r_j \sigma_j \otimes |\alpha_j\rangle\langle\alpha_j|) \leq (T-1)O(1)\Delta t^2 + \frac{1}{\sqrt{2\Delta - 1}}.$$

This gives us a hint to choose a proper order of parameters. For example, if we want to estimate the eigenvalue within an error bound ϵ, and we desire that the program success with a constant probability, then we can choose $\Delta t = \Theta(\epsilon^2)$, $\Delta = O(1)$, $T = \Theta(\epsilon^{-3})$, which is coincident to the parameters in [25]. If we also want that the probability of success of the program is larger than $1 - \epsilon'$ ($\epsilon < \epsilon'$), then we can choose $\Delta t = \Theta(\epsilon^2 \epsilon'^2)$, $\Delta = \Theta(\epsilon'^{-2})$ and $T = \Theta(\epsilon^{-3}\epsilon'^{-4})$.

7 Conclusion

This paper derived an applied variant, called aQHL (applied quantum Hoare logic), of QHL which significantly improves applicability of QHL in two different directions:

- simplifying inference rules of QHL by restricting to projective preconditions and postconditions, so that QHL can be applied to larger quantum programs;
- developing a set of new rules for reasoning about robustness, so that QHL can be more conveniently used to verify inexact quantum programs.

Two sophisticated quantum algorithms, namely HHL for solving systems of linear equations and qPCA (quantum Principal Component Analysis), were verified in aQHL (by hands). In future work, we plan to incorporate the simplified QHL rules and newly introduced rules for robustness reasoning into a theorem prover for QHL (for example, see a recent attempt [24]) so that HHL and qPCA (and more quantum algorithms) can be mechanically verified.

As discussed in the Introduction, aQHL is more suitable for testing and debugging of quantum programs than the original QHL. A scheme for quantum program testing and debugging with projective assertions (quantum predicates) in aQHL will be presented in a forthcoming paper [41]. In particular, we will show that, with robust assertions defined in Section 4 and the gentle measurement lemma [36], it is still applicable when a small system error or noise in implementation occurs.

Acknowledgments

This work was partly supported by the Australian Research Council (Grant No: DE180100156 and DP180100691), the National Key R&D Program of China (Grant No: 2018YFA0306701), and the National Natural Science Foundation of China (Grant No: 61832015).

References

[1] Ali Javadi Abhari, Arvin Faruque, Mohammad Javad Dousti, Lukas Svec, Oana Catu, Amlan Chakrabati, Chen-Fu Chiang, Seth Vanderwilt, John Black, Fred Chong, Margaret Martonosi, Martin Suchara, Ken Brown, Massoud Pedram, and Todd Brun. 2012. *Scaffold: Quantum programming language.* Technical Report TR-934-12. Dept. of Computer Science, Princeton University NJ. ftp://ftp.cs.princeton.edu/reports/2012/934.pdf

[2] Dmitri Akatov. 2005. *The Logic of Quantum Program Verification.* Master's thesis. Oxford University Computing Laboratory. http://www.academia.edu/download/7563948/thesis-1.1.ps

[3] Gadi Aleksandrowicz, Thomas Alexander, Panagiotis Barkoutsos, Luciano Bello, Yael Ben-Haim, David Bucher, Francisco Jose Cabrera-Hernádez, Jorge Carballo-Franquis, Adrian Chen, Chun-Fu Chen, Jerry M. Chow, Antonio D. Córcoles-Gonzales, Abigail J. Cross, Andrew Cross, Juan Cruz-Benito, Chris Culver, Salvador De La Puente González, Enrique De La Torre, Delton Ding, Eugene Dumitrescu, Ivan Duran, Pieter Eendebak, Mark Everitt, Ismael Faro Sertage, Albert Frisch, Andreas Fuhrer, Jay Gambetta, Borja Godoy Gago, Juan Gomez-Mosquera, Donny Greenberg, Ikko Hamamura, Vojtech Havlicek, Joe Hellmers, Łukasz Herok, Hiroshi Horii, Shaohan Hu, Takashi Imamichi, Toshinari Itoko, Ali Javadi-Abhari, Naoki Kanazawa, Anton Karazeev, Kevin Krsulich, Peng Liu, Yang Luh, Yunho Maeng, Manoel Marques, Francisco Jose Martín-Fernández, Douglas T. McClure, David McKay, Srujan Meesala, Antonio Mezzacapo, Nikolaj Moll, Diego Moreda Rodríguez, Giacomo Nannicini, Paul Nation, Pauline Ollitrault, Lee James O'Riordan, Hanhee Paik, Jesús Pérez, Anna Phan, Marco Pistoia, Viktor Prutyanov, Max Reuter, Julia Rice, Abdón Rodríguez Davila, Raymond Harry Putra Rudy, Mingi Ryu, Ninad Sathaye, Chris Schnabel, Eddie Schoute, Kanav Setia, Yunong Shi, Adenilton Silva, Yukio Siraichi, Seyon Sivarajah, John A. Smolin, Mathias Soeken, Hitomi Takahashi, Ivano Tavernelli, Charles Taylor, Pete Taylour, Kenso Trabing, Matthew Treinish, Wes Turner, Desiree Vogt-Lee, Christophe Vuillot, Jonathan A. Wildstrom, Jessica Wilson, Erick Winston, Christopher Wood, Stephen Wood, Stefan Wörner, Ismail Yunus Akhalwaya, and Christa Zoufal. 2019. Qiskit: An Open-source Framework for Quantum Computing. https://doi.org/10.5281/zenodo.2562110

[4] Thorsten Altenkirch and Jonathan Grattage. 2005. A functional quantum programming language. In *Proceedings of the 20th Annual IEEE Symposium on Logic in Computer Science (LICS' 05)*. IEEE, 249–258. https://doi.org/10.1109/LICS.2005.1

[5] Andris Ambainis. 2016. Superlinear advantage for exact quantum algorithms. *SIAM J. Comput.* 45, 2 (2016), 617–631. https://doi.org/10.1137/130939043

[6] Alexandru Baltag and Sonja Smets. 2004. The logic of quantum programs. In *Proceedings of the 2nd International Workshop on Quantum Programming Languages (QPL 2004)*, Peter Selinger (Ed.). 39–56. https://www.mathstat.dal.ca/~selinger/qpl2004/PDFS/04Baltag-Smets.pdf

[7] Alexandru Baltag and Sonja Smets. 2006. LQP: the dynamic logic of quantum information. *Mathematical Structures in Computer Science* 16, 3 (2006), 491–525. https://doi.org/10.1017/S0960129506005299

[8] Garrett Birkhoff and John Von Neumann. 1936. The logic of quantum mechanics. *Annals of Mathematics* 37, 4 (1936), 823–843. https://doi.org/10.2307/1968621

[9] Olivier Brunet and Philippe Jorrand. 2004. Dynamic quantum logic for quantum programs. *International Journal of Quantum Information* 2, 01 (2004), 45–54. https://doi.org/10.1142/S0219749904000067

[10] Rohit Chadha, Paulo Mateus, and Amílcar Sernadas. 2006. Reasoning about imperative quantum programs. *Electronic Notes in Theoretical Computer Science* 158 (2006), 19–39. https://doi.org/10.1016/j.entcs.2006.04.003

[11] Swarat Chaudhuri, Sumit Gulwani, and Roberto Lublinerman. 2010. Continuity analysis of programs. In *Proceedings of the 37th annual ACM SIGPLAN-SIGACT symposium on Principles of programming languages (POPL '10)*. ACM, New York, NY, USA, 57–70. https://doi.org/10.1145/1706299.1706308

[12] Swarat Chaudhuri, Sumit Gulwani, Roberto Lublinerman, and Sara Navidpour. 2011. Proving programs robust. In *Proceedings of the 19th ACM SIGSOFT symposium and the 13th European conference on Foundations of software engineering (ESEC/FSE '11)*. ACM, New York, NY, USA, 102–112. https://doi.org/10.1145/2025113.2025131

[13] Ellie D'hondt and Prakash Panangaden. 2006. Quantum weakest preconditions. *Mathematical Structures in Computer Science* 16, 3 (2006), 429–451. https://doi.org/10.1017/S0960129506005251

[14] Yuan Feng, Runyao Duan, Zhengfeng Ji, and Mingsheng Ying. 2007. Proof rules for the correctness of quantum programs. *Theoretical Computer Science* 386, 1-2 (2007), 151–166. https://doi.org/10.1016/j.tcs.2007.06.011

[15] Simon J Gay. 2006. Quantum programming languages: Survey and bibliography. *Mathematical Structures in Computer Science* 16, 4 (2006), 581–600. https://doi.org/10.1017/S0960129506005378

[16] Google AI Quantum team. 2018. https://github.com/quantumlib/Cirq

[17] Alexander S Green, Peter LeFanu Lumsdaine, Neil J Ross, Peter Selinger, and Benoît Valiron. 2013. Quipper: a scalable quantum programming language. In *Proceedings of the 34th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI '13)*. ACM, New York, NY, USA, 333–342. https://doi.org/10.1145/2499370.2462177

[18] Aram W Harrow, Avinatan Hassidim, and Seth Lloyd. 2009. Quantum algorithm for linear systems of equations. *Physical Review Letters* 103, 15 (2009), 150502. https://doi.org/10.1103/PhysRevLett.103.150502

[19] C.A.R. Hoare. 2009. Viewpoint retrospective: An axiomatic basis for computer programming. *Commun. ACM* 52, 10 (2009), 30–32. https://doi.org/10.1145/1562764.1562779

[20] Shih-Han Hung, Kesha Hietala, Shaopeng Zhu, Mingsheng Ying, Michael Hicks, and Xiaodi Wu. 2019. Quantitative Robustness Analysis of Quantum Programs. *Proc. ACM Program. Lang.* 3, POPL, Article 31 (Jan. 2019), 29 pages. https://doi.org/10.1145/3290344

[21] Yoshihiko Kakutani. 2009. A logic for formal verification of quantum programs. In *Proceedings of the 13th Asian conference on Advances in Computer Science: information Security and Privacy (ASIAN 2009)*, Anupam Datta (Ed.). Springer, Springer Berlin Heidelberg, Berlin, Heidelberg, 79–93. https://doi.org/10.1007/978-3-642-10622-4_7

[22] Gudrun Kalmbach. 1983. *Orthomodular lattices.* Vol. 18. Academic Press.

[23] Yangjia Li and Mingsheng Ying. 2017. Algorithmic Analysis of Termination Problems for Quantum Programs. In *Proceedings of the 45th ACM SIGPLAN Symposium on Principles of Programming Languages (POPL 2018)*. ACM, New York, NY, USA, Article 35, 29 pages. https://doi.org/10.1145/3158123

[24] Tao Liu, Yangjia Li, Shuling Wang, Mingsheng Ying, and Naijun Zhan. 2016. A theorem prover for quantum Hoare logic and its applications. *arXiv preprint arXiv:1601.03835* (2016). https://arxiv.org/abs/1601.03835

[25] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. 2014. Quantum principal component analysis. *Nature Physics* 10, 9 (2014), 631. https://doi.org/10.1038/nphys3029

[26] Ashley Montanaro, Richard Jozsa, and Graeme Mitchison. 2015. On exact quantum query complexity. *Algorithmica* 71, 4 (2015), 775–796. https://doi.org/10.1007/s00453-013-9826-8

[27] Bernhard Ömer. 2003. *Structured quantum programming.* Ph.D. Dissertation. Institute for Theoretical Physics, Vienna University of Technology. http://tph.tuwien.ac.at/~oemer/doc/structquprog.pdf

[28] Jennifer Paykin, Robert Rand, and Steve Zdancewic. 2017. QWIRE: a core language for quantum circuits. In *Proceedings of the 44th ACM SIGPLAN Symposium on Principles of Programming Languages (POPL 2017)*. ACM, New York, NY, USA, 846–858. https://doi.org/10.1145/3009837.3009894

[29] Robert Rand. 2016. Verification logics for quantum programs. http://www.cs.umd.edu/~rrand/wpe.pdf

[30] Rigetti Forest team. 2018. https://www.rigetti.com/forest

[31] Jeff W Sanders and Paolo Zuliani. 2000. Quantum programming. In *International Conference on Mathematics of Program Construction (MPC 2000)*, Roland Backhouse and José Nuno Oliveira (Eds.). Springer, Springer Berlin Heidelberg, Berlin, Heidelberg, 80–99. https://doi.org/10.1007/10722010_6

[32] Peter Selinger. 2004. A brief survey of quantum programming languages. In *International Symposium on Functional and Logic Programming (FLOPS 2004)*, Yukiyoshi Kameyama and Peter J. Stuckey (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 1–6. https://doi.org/10.1007/978-3-540-24754-8_1

[33] Peter Selinger. 2004. Towards a quantum programming language. *Mathematical Structures in Computer Science* 14, 4 (2004), 527–586. https://doi.org/10.1017/S0960129504004256

[34] Krysta Svore, Alan Geller, Matthias Troyer, John Azariah, Christopher Granade, Bettina Heim, Vadym Kliuchnikov, Mariia Mykhailova, Andres Paz, and Martin Roetteler. 2018. Q#: Enabling scalable quantum computing and development with a high-level dsl. In *Proceedings of the Real World Domain Specific Languages Workshop 2018 (RWDSL 2018)*. ACM, New York, NY, USA, Article 7, 10 pages. https://doi.org/10.1145/3183895.3183901

[35] Dominique Unruh. 2019. Quantum Relational Hoare Logic. *Proc. ACM Program. Lang.* 3, POPL, Article 33 (Jan. 2019), 31 pages. https://doi.org/10.1145/3290346

[36] Andreas Winter. 1999. Coding theorem and strong converse for quantum channels. *IEEE Transactions on Information Theory* 45, 7 (1999), 2481–2485. https://doi.org/10.1109/18.796385

[37] Mingsheng Ying. 2011. Floyd–hoare logic for quantum programs. *ACM Transactions on Programming Languages and Systems (TOPLAS)* 33, 6, Article 19 (2011), 49 pages. https://doi.org/10.1145/2049706.2049708

[38] Mingsheng Ying. 2016. *Foundations of Quantum Programming*. Morgan Kaufmann.

[39] Mingsheng Ying, Runyao Duan, Yuan Feng, and Zhengfeng Ji. 2010. Predicate transformer semantics of quantum programs. *Semantic Techniques in Quantum Computation* 8 (2010), 311–360.

[40] Mingsheng Ying, Shenggang Ying, and Xiaodi Wu. 2017. Invariants of quantum programs: characterisations and generation. In *Proceedings of the 44th ACM SIGPLAN Symposium on Principles of Programming Languages (POPL 2017)*. ACM, New York, NY, USA, 818–832. https://doi.org/10.1145/3009837.3009840

[41] Li Zhou, Nengkun Yu, and Mingsheng Ying. In preparation. Testing and debugging of quantum programs. (In preparation).