Relational Proofs for Quantum Programs

GILLES BARTHE, Max Planck Institute for Security and Privacy, Germany and IMDEA Software Institute, Spain
JUSTIN HSU, University of Wisconsin–Madison, USA
MINGSHENG YING, University of Technology Sydney, Australia, Institute of Software, Chinese Academy of Sciences, China, and Tsinghua University, China
NENGKUN YU, University of Technology Sydney, Australia
LI ZHOU, Max Planck Institute for Security and Privacy, Germany and Tsinghua University, China

Relational verification of quantum programs has many potential applications in quantum and post-quantum security and other domains. We propose a relational program logic for quantum programs. The interpretation of our logic is based on a quantum analogue of probabilistic couplings. We use our logic to verify non-trivial relational properties of quantum programs, including uniformity for samples generated by the quantum Bernoulli factory, reliability of quantum teleportation against noise (bit and phase flip), security of quantum one-time pad and equivalence of quantum walks.

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

Additional Key Words and Phrases: quantum programming, verification, relational properties, coupling

ACM Reference Format:
Gilles Barthe, Justin Hsu, Mingsheng Ying, Nengkun Yu, and Li Zhou. 2020. Relational Proofs for Quantum Programs. *Proc. ACM Program. Lang.* 4, POPL, Article 21 (January 2020), 29 pages. https://doi.org/10.1145/3371089

1 INTRODUCTION

Program verification is traditionally focused on proving properties of a single program execution. In contrast, relational verification aims to prove properties about two program executions. In some cases, such as program refinement and program equivalence, the goal is to relate executions of two different programs on equal or related inputs. However, some properties consider two executions of the same program (with related inputs); examples include information flow policies (*non-interference*: two runs of a program on states that only differ in their secret have equal visible effects) and robustness (*k-Lipschitz continuity*: running a program on two initial states at distance d yields two final states at distance at most $k \cdot d$). In the probabilistic setting, relational verification can also show that a program outputs a uniform distribution, or that two programs yield "approximately equal" distributions. By taking suitable instantiations of approximate equality, relational verification

Authors' addresses: Gilles Barthe, Max Planck Institute for Security and Privacy, Germany , IMDEA Software Institute, Spain; Justin Hsu, Department of Computer Sciences, University of Wisconsin–Madison, USA; Mingsheng Ying, Centre for Quantum Software and Information, University of Technology Sydney, Australia , State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, China , Tsinghua University, China; Nengkun Yu, Centre for Quantum Software and Information, University of Technology Sydney, Australia; Li Zhou, Max Planck Institute for Security and Privacy, Germany , Department of Computer Science and Technology, Tsinghua University, China.

2475-1421/2020/1-ART21
https://doi.org/10.1145/3371089

has found success in cryptography [Barthe et al. 2009], machine learning [Barthe et al. 2018] and differential privacy [Barthe et al. 2016, 2012].

This paper develops a relational program logic, called rqPD, for a core quantum programming language. Our logic is based on the interpretation of predicates as physical observables, mathematically modelled as Hermitian operators [D'Hondt and Panangaden 2006], and is inspired by the qPD program logic [Ying 2011, 2016] for quantum programs. Concretely, our judgments have the form:

$$P_1 \sim P_2 : A \Rightarrow B$$

where P_1 and P_2 are quantum programs, and precondition A and postcondition B are Hermitian operators over the tensor product Hilbert spaces of P_1 and P_2. We define an interpretation of these judgments, develop a rich set of sound proof rules, and show how these rules can be used to verify relational properties for quantum programs.

Technical challenges and solutions. The central challenge for building a useful relational logic is to find an interpretation of judgments that captures properties of interest, while guaranteeing soundness of a convenient set of proof rules. This challenge is not unique to quantum programs. In the probabilistic setting [Barthe et al. 2015, 2009], one solution is to interpret judgments in terms of probabilistic couplings, a standard abstraction from probability theory [Lindvall 2002; Thorisson 2000; Villani 2008]. The connection with probabilistic couplings has many advantages: (i) it builds the logic on an abstraction that has proven to be useful for probabilistic reasoning; (ii) it identifies natural extensions of the logic; and (iii) it suggests other applications and properties that can be handled by similar techniques. Unfortunately, the quantum setting raises additional challenges. Notably, we may need to reason about entangled quantum states. There are some existing proposals of analogue of probabilistic couplings in the quantum setting (see [Kümmerer and Schwieger 2016; Winter 2016]). In particular, Zhou et al. [2019a] addressed this issue by developing a notion of quantum coupling, and validated their definition by showing an analogue of Strassen's theorem [Strassen 1965].[1] In this work, we base our notion of valid judgment on this definition of quantum coupling.

Once the interpretation of the logic is fixed, the next challenge is to define a useful set of proof rules. As in other relational logics, we need structural rules and three kinds of construct-specific rules. *Synchronous* rules apply when P_1 and P_2 have the same top-level construct and operates on both programs, whereas the *left* and *right* rules only operate on one of the two programs. In the quantum setting, the main difficulties are:

- *Structural rules*: many useful rules are not sound in the quantum setting or require further hypotheses; in particular, in the presence of entanglement — an indispensable resource in quantum computation and communication. We generalise the core judgment to track and enforce the hypotheses required to preserve soundness.
- *Construct-specific rules*: all proof rules of quantum Hoare logic qPD can be directly generalised into quantum relational logic rqPD (see Subsection 5.2). Although these directly generalised rules are useful, they do not fully capture the essence of quantum relational reasoning (see Subsection 5.3). Synchronous proof rules for classical control-flow constructs, i.e. conditionals and loops, generally require that the two programs follow the same control flow path, so that they execute in lockstep. In order to retain soundness in our setting, we introduce a measurement condition ensuring that corresponding branches in the control flow are taken with equal probabilities.

[1] Informally, Strassen's theorem states that there exists a B-coupling between two distributions μ and μ' over sets X and X' respectively iff for every subset Y in the support of μ, $\mu(Y) \leq \mu(B(Y))$, where $B(Y)$ is the set-theoretic image of Y under B.

Simplifying side conditions. Checking measurement conditions is often challenging. To make this step easier, we introduce a simplified version of rqPD where assertions are modelled as projective predicates, or equivalently, (closed) subspaces of the state Hilbert space — a special case of Hermitian operators. The restriction to projective predicates leads to simpler inference rules and easier program verification, at the cost of expressiveness [Zhou et al. 2019b]. In particular, checking measurement conditions reduces to showing that a program condition lies in a subspace (with probability 1), a task that is often simpler. We provide a formal comparison between the original logic system rqPD and its simplified version, and leverage this comparison to relate our work with a recent proposal for a quantum relational Hoare logic with projective predicates [Unruh 2019b].

Applications. To test its effectiveness, we apply rqPD to verify non-trivial relational properties of several quantum programs, including uniformity for samples generated by the quantum Bernoulli factory, reliability of quantum teleportation against noise (bit and phase flip), equivalence of quantum walks and security of quantum one-time pad. Using our simplified rqPD, we are able to verify the relational properties of some more sophisticated quantum programs, for example, equivalence of quantum walks with different coin tossing operators.

Working Example. We will use the following pair of simple quantum programs as our working example to illustrate our basic ideas along the way:

Example 1.1. *Let q be a qubit (quantum bit). Consider two programs:*

$$P_1 \equiv q := |0\rangle; q := H[q]; Q_1, \qquad P_2 \equiv q := |0\rangle; Q_2; q := H[q].$$

In both, q is initialised in a basis state $|0\rangle$. P_1 applies the Hadamard gate H to q and executes Q_1, while P_2 first executes Q_2 and then applies H to q. The subprograms Q_1, Q_2 are as follows:

$$Q_1 \equiv \textbf{if } (\mathcal{M}[q] = 0 \to q := X[q] \square 1 \to q := H[q]) \textbf{ fi}$$
$$Q_2 \equiv \textbf{if } (\mathcal{M}'[q] = 0 \to q := Z[q] \square 1 \to q := H[q]) \textbf{ fi}$$

where $\mathcal{M}, \mathcal{M}'$ are the measurement in the computational basis $|0\rangle, |1\rangle$ and the measurement in basis $|\pm\rangle = \frac{1}{\sqrt{2}}(|0\rangle \pm |1\rangle)$ respectively. Intuitively, Q_1 first performs $\mathcal{M}$ on q, then applies either the Pauli gate X or Hadamard gate H, depending on whether the measurement outcome is 0 or 1. But Q_2 uses the outcomes of a different measurement $\mathcal{M}$ to choose between the Pauli gate Z and Hadamard gate H.

Obviously, programs P_1, P_2 have similar structures. The logic rqPD developed in this paper will enable us to specify and prove some interesting symmetry between them.

2 MATHEMATICAL PRELIMINARIES

We assume basic familiarity with Hilbert spaces, see Nielsen and Chuang [2002] for an introduction.

Quantum states. The state space of a quantum system is a Hilbert space $\mathcal{H}$. In this paper, we only consider finite-dimensional $\mathcal{H}$. A *pure state* of the quantum system is modelled by a (column) vector in $\mathcal{H}$ of length 1; we use the Dirac notation ($|\varphi\rangle$, $|\psi\rangle$) to denote pure states. For example, qubit q in Example 1.1 has the 2-dimensional Hilbert space as its state space; it can be in basis states $|0\rangle, |1\rangle$ as well as in their superpositions $|+\rangle, |-\rangle = \frac{1}{\sqrt{2}}(|0\rangle \pm |1\rangle)$. An operator A in an d-dimensional Hilbert space $\mathcal{H}$ is represented as an $d \times d$ matrix. Its *trace* is defined as $tr(A) = \sum_i \langle i|A|i\rangle$, where $\{|i\rangle\}$ is an orthonormal basis of $\mathcal{H}$. A positive operator ρ in $\mathcal{H}$ is called a *partial density operator* if its trace satisfies $tr(\rho) \leq 1$; if $tr(\rho) = 1$, then ρ is called a *density operator*. A *mixed state* of a quantum system is a distribution over pure states. If state $|\psi_i\rangle$ has probability p_i, the mixed state can be represented by a density operator $\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i|$, where row vector $\langle\psi_i|$ stands for the

conjugate transpose of $|\psi_i\rangle$. For example, if qubit q is in state $|0\rangle$ with probability $\frac{2}{3}$ and in $|+\rangle$ with probability $\frac{1}{3}$, then its state can be described by density operator

$$\rho = \frac{2}{3}|0\rangle\langle 0| + \frac{1}{3}|+\rangle\langle +| = \frac{1}{6}\begin{pmatrix} 5 & 1 \\ 1 & 1 \end{pmatrix}. \tag{1}$$

We write $\mathcal{D}^{\leq}(\mathcal{H})$ and $\mathcal{D}(\mathcal{H})$ for the set of partial density operators and the set of density operators in $\mathcal{H}$, respectively. For any $\rho \in \mathcal{D}^{\leq}(\mathcal{H})$, the *support* $\mathrm{supp}(\rho)$ of ρ is defined as the span of the eigenvectors of ρ with nonzero eigenvalues.

Operations on states. A basic operation on a (closed) quantum system is modelled as a *unitary operator* U such that $U^\dagger U = I_{\mathcal{H}}$, where † stands for conjugate and transpose. For example, the Pauli and Hadamard gates used in Example 1.1 are:

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \qquad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \qquad H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix},$$

and X, Z, H transform states $|0\rangle, |1\rangle$ to $X|0\rangle = |1\rangle$, $X|1\rangle = |0\rangle$; $Z|0\rangle = |0\rangle$, $Z|1\rangle = -|1\rangle$; $H|0\rangle = |+\rangle$, $H|1\rangle = |-\rangle$, respectively.

Another basic operation is *measurement*. A *physical observable* is modelled by an operator A in $\mathcal{H}$ that is Hermitian, i.e., $A^\dagger = A$. An operator P is a *projection* onto a (closed) subspace of $\mathcal{H}$ if and only if it is Hermitian (i.e. $P^\dagger = P$) and idempotent (i.e. $P^2 = P$). Quantum measurements are constructed from observables A. An *eigenvector* of A is a non-zero vector $|\psi\rangle \in \mathcal{H}$ such that $A|\psi\rangle = \lambda|\psi\rangle$ for some complex number λ (indeed, λ must be real when A is Hermitian). In this case, λ is called an *eigenvalue* of A. For each eigenvalue λ, the set $\{|\psi\rangle : A|\psi\rangle = \lambda|\psi\rangle\}$ of eigenvectors corresponding to λ and zero vector is a (closed) subspace of $\mathcal{H}$. We write P_λ for the projection onto this subspace. By the spectral decomposition [Nielsen and Chuang 2002, Theorem 2.1], A can be decomposed as a sum $A = \sum_\lambda \lambda P_\lambda$ where λ ranges over all eigenvalues of A. Moreover, $\mathcal{M} = \{P_\lambda\}_\lambda$ is a (projective) measurement.

If we perform $\mathcal{M}$ on the quantum system in state ρ, then outcome λ is obtained with probability $p_\lambda = tr(P_\lambda^\dagger P_\lambda \rho) = tr(P_\lambda \rho)$, and after that, the system will be in state $(P_\lambda \rho P_\lambda)/p_\lambda$. Therefore, the expectation of A in state ρ is $[\![A]\!]_\rho = \sum_\lambda p_\lambda \cdot \lambda = \sum_\lambda \lambda tr(P_\lambda \rho) = tr(A\rho)$. For instance, the measurements in Example 1.1 are defined as $\mathcal{M} = \{M_0, M_1\}$, $\mathcal{M}' = \{M_0', M_1'\}$ with

$$M_0 = |0\rangle\langle 0| = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \qquad M_1 = |1\rangle\langle 1| = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix},$$

$$M_0' = |+\rangle\langle +| = \frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}, \qquad M_1' = |-\rangle\langle -| = \frac{1}{2}\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}.$$

If we perform $\mathcal{M}'$ on a qubit in (mixed) state ρ given in equation (1), then the probability that we get outcome "1" is

$$p(1) = tr(M_1'\rho) = tr\left[\frac{1}{2}\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \cdot \frac{1}{6}\begin{pmatrix} 5 & 1 \\ 1 & 1 \end{pmatrix}\right] = \frac{1}{12} \cdot tr\begin{pmatrix} 4 & 0 \\ -4 & 0 \end{pmatrix} = \frac{1}{3}$$

and after that, the qubit's state will change to

$$M_1'\rho M_1'/p(1) = \frac{1}{2}\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \cdot \frac{1}{6}\begin{pmatrix} 5 & 1 \\ 1 & 1 \end{pmatrix} \cdot \frac{1}{2}\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \div \frac{1}{3} = \frac{1}{2}\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}.$$

Similarly, the probability of outcome "0" is $p(0) = \frac{2}{3}$, and then the state changes to $|+\rangle\langle +|$.

We will use observables as predicates in our logic. To compare two operators A and B in a Hilbert space $\mathcal{H}$, we will use the *Löwner order* between operators defined as follows: $A \sqsubseteq B$ if and only if $B - A$ is positive. A quantum predicate [D'Hondt and Panangaden 2006] (or an *effect*) 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.

Tensor Products of quantum states. Let $\mathcal{H}_1, \mathcal{H}_2$ be the state Hilbert spaces of two quantum systems considered in isolation. Then the composite system has state space modeled by the tensor product $\mathcal{H}_1 \otimes \mathcal{H}_2$. The notion of *partial trace* is needed to extract the state of a subsystem. Formally, the partial trace over $\mathcal{H}_1$ is a mapping $tr_1(\cdot)$ from operators on $\mathcal{H}_1 \otimes \mathcal{H}_2$ to operators in $\mathcal{H}_2$ defined by the following equation: $tr_1(|\varphi_1\rangle\langle\psi_1| \otimes |\varphi_2\rangle\langle\psi_2|) = \langle\psi_1|\varphi_1\rangle \cdot |\varphi_2\rangle\langle\psi_2|$ for all $|\varphi_1\rangle, |\psi_1\rangle \in \mathcal{H}_1$ and $|\varphi_2\rangle, |\psi_2\rangle \in \mathcal{H}_2$ together with linearity. The partial trace $tr_2(\cdot)$ over $\mathcal{H}_2$ can be defined symmetrically. Suppose that we have a composite system of two subsystems with state spaces $\mathcal{H}_1, \mathcal{H}_2$, respectively, and it is in (mixed) state ρ. Then the states of the first and second subsystems can be described by $tr_2(\rho)$, $tr_1(\rho)$, respectively. For example, if the subsystems are both qubits, and they are maximally entangled; i.e. in state $|\Phi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ or equivalently

$$|\Phi\rangle\langle\Phi| = \frac{1}{2}(|0\rangle\langle 0| \otimes |0\rangle\langle 0| + |0\rangle\langle 1| \otimes |0\rangle\langle 1| + |1\rangle\langle 0| \otimes |1\rangle\langle 0| + |1\rangle\langle 1| \otimes |1\rangle\langle 1|) \tag{2}$$

then the partial traces $tr_1(|\Phi\rangle\langle\Phi|) = \frac{1}{2}(|0\rangle\langle 0| + |1\rangle\langle 1|)$ and $tr_2(|\Phi\rangle\langle\Phi|) = \frac{1}{2}(|0\rangle\langle 0| + |1\rangle\langle 1|)$ describe states of the second and first subsystems, respectively.

3 QUANTUM COUPLINGS AND LIFTINGS

3.1 Quantum Couplings

To relate pairs of quantum programs, our program logic will rely on a quantum version of probabilistic coupling. In the probabilistic world, a coupling for two discrete distributions μ_1 and μ_2 over sets A_1 and A_2 is a discrete distribution μ over $A_1 \times A_2$ such that the first and second marginals of μ are equal to μ_1 and μ_2 respectively. A coupling μ is an R-lifting for μ_1 and μ_2 if additionally its support is included in R, i.e. every element outside R has probability zero.

In order to define the quantum analogue of couplings, we apply a correspondence between the probabilistic and quantum worlds [Nielsen and Chuang 2002]:

probability distributions $\Leftrightarrow$ *density operators* *marginal distributions* $\Leftrightarrow$ *partial traces*

This leads to the following definition of quantum coupling.

Definition 3.1 (Coupling). *Let $\rho_1 \in \mathcal{D}^{\leq}(\mathcal{H}_1)$ and $\rho_2 \in \mathcal{D}^{\leq}(\mathcal{H}_2)$. Then $\rho \in \mathcal{D}^{\leq}(\mathcal{H}_1 \otimes \mathcal{H}_2)$ is called a coupling for $\langle\rho_1, \rho_2\rangle$ if $tr_1(\rho) = \rho_2$ and $tr_2(\rho) = \rho_1$.*

Proposition 3.1 (Trace equivalence). *If ρ is a coupling for $\langle\rho_1, \rho_2\rangle$, then they have the same trace: $tr(\rho) = tr(\rho_1) = tr(\rho_2)$.*

The following are examples of quantum couplings. They are quantum generalisations of several typical examples of (discrete) probabilistic couplings (see [Barthe et al. 2019]). From these simple examples, we can see a close and natural correspondence as well as some essential differences between probabilistic coupling and their quantum counterparts. Our first example shows that couplings always exist.

Example 3.1. *Let $\rho_1 \in \mathcal{D}(\mathcal{H}_1)$ and $\rho_2 \in \mathcal{D}(\mathcal{H}_2)$ be density operators. The tensor product $\rho_\otimes = \rho_1 \otimes \rho_2$ is a coupling for $\langle\rho_1, \rho_2\rangle$.*

Just like the case for probabilistic couplings, there can be more than one quantum coupling between two operators.

Example 3.2. *Let $\mathcal{H}$ be a d-dimensional Hilbert space. Let $\mathcal{B} = \{|i\rangle\}$ be an orthonormal basis of $\mathcal{H}$. Then the uniform density operator over $\mathcal{H}$ is $\mathbf{Unif}_{\mathcal{H}} = \frac{1}{d}\sum_i |i\rangle\langle i|$. For each unitary operator U in $\mathcal{H}$,*

we write $U(\mathcal{B}) = \{U|i\rangle\}$, which is also an orthonormal basis of $\mathcal{H}$. Then $\rho_U = \frac{1}{d}\sum_i (|i\rangle U|i\rangle)(\langle i|\langle i|U^\dagger)$ is a coupling for $\langle \mathbf{Unif}_{\mathcal{H}}, \mathbf{Unif}_{\mathcal{H}}\rangle$. Indeed, the arbitrariness of U shows that there are (uncountably) infinitely many couplings for $\langle \mathbf{Unif}_{\mathcal{H}}, \mathbf{Unif}_{\mathcal{H}}\rangle$. For instance, the maximally entangled state $|\Phi\rangle\langle\Phi|$ in equation (2) is such a coupling for

$$\left\langle \mathbf{Unif}_{\mathcal{H}_1} = \frac{1}{2}(|0\rangle\langle 0| + |1\rangle\langle 1|), \mathbf{Unif}_{\mathcal{H}_2} = \frac{1}{2}(|0\rangle\langle 0| + |1\rangle\langle 1|) \right\rangle.$$

Example 3.3. *Let ρ be a partial density operator in $\mathcal{H}$. Then by the spectral decomposition theorem [Nielsen and Chuang 2002, Theorem 2.1], ρ can be written as $\rho = \sum_i p_i|i\rangle\langle i|$ for some orthonormal basis $\mathcal{B} = \{|i\rangle\}$ and $p_i \geq 0$ with $\sum_i p_i \leq 1$. We define $\rho_{\mathrm{id}(\mathcal{B})} = \sum_i p_i|ii\rangle\langle ii|$. Then it is easy to see that $\rho_{\mathrm{id}(\mathcal{B})}$ is a coupling for $\langle \rho, \rho\rangle$. An essential difference between this example and its classical counterpart (see [Hsu 2017] Example 2.1.5) is that ρ might be decomposed with other orthonormal bases, say $\mathcal{D} = \{|j\rangle\}$: $\rho = \sum_j q_j|j\rangle\langle j|$. In general, $\rho_{\mathrm{id}(\mathcal{B})} \neq \rho_{\mathrm{id}(\mathcal{D})}$, and we can define a different coupling for $\langle \rho, \rho\rangle$: $\rho_{\mathrm{id}(\mathcal{D})} = \sum_j q_j|jj\rangle\langle jj|$.*

3.2 Quantum Lifting

Although there can be many couplings for two operators, it is usually not simple to find one suited to our application. As said at the beginning of this section, lifting can help for this purpose. The definition of liftings smoothly generalises to the quantum case.

Definition 3.2. *Let $\rho_1 \in \mathcal{D}^{\leq}(\mathcal{H}_1)$ and $\rho_2 \in \mathcal{D}^{\leq}(\mathcal{H}_2)$, and let $\mathcal{X}$ be (the projection onto) a (closed) subspace of $\mathcal{H}_1 \otimes \mathcal{H}_2$. Then $\rho \in \mathcal{D}^{\leq}(\mathcal{H}_1 \otimes \mathcal{H}_2)$ is called a witness of the lifting $\rho_1 \mathcal{X}^{\#} \rho_2$ if:*

1. *ρ is a coupling for $\langle \rho_1, \rho_2\rangle$;*
2. *$\mathrm{supp}(\rho) \subseteq \mathcal{X}$.*

Example 3.4. *The following are examples of quantum liftings.*

1. *The coupling ρ_U for the uniform density operator and itself in Example 3.2 is a witness for the lifting $\mathbf{Unif}_{\mathcal{H}}\ \mathcal{X}(\mathcal{B}, U)^{\#}\ \mathbf{Unif}_{\mathcal{H}}$, where $\mathcal{X}(\mathcal{B}, U) = \mathrm{span}\{|i\rangle U|i\rangle\}$ is a subspace of $\mathcal{H} \otimes \mathcal{H}$.*
2. *The coupling $\rho_{\mathrm{id}(\mathcal{B})}$ in Example 3.3 is a witness of the lifting $\rho(=_{\mathcal{B}})^{\#}\rho$, where $(=_{\mathcal{B}}) \equiv \mathrm{span}\{|ii\rangle\}$ defined by the orthonormal basis $\mathcal{B} = \{|i\rangle\}$ is a subspace of $\mathcal{H} \otimes \mathcal{H}$. It is interesting to note that the maximally entangled state $|\Psi\rangle = \frac{1}{\sqrt{d}}\sum_i |ii\rangle$ is in $=_{\mathcal{B}}$.*
3. *The coupling $\rho_{\mathrm{id}(\mathcal{B})}$ in Example 3.3 is a witness of the lifting $\rho(=_{sym})^{\#}\rho$, defining $=_{sym}$ to be the symmetrisation operator, i.e. $(=_{sym}) \equiv \frac{1}{2}(I \otimes I + S)$, where S is the SWAP operator defined by $S|\varphi, \psi\rangle = |\psi, \varphi\rangle$ for any $|\varphi\rangle, |\psi\rangle \in \mathcal{H}$ together with linearity. Operator S is independent of the basis, and given any orthonormal basis $\{|i\rangle\}$ of $\mathcal{H}$, S has the following form: $S = \sum_{ij} |i\rangle\langle j| \otimes |j\rangle\langle i|$.*
4. *The coupling $\rho_1 \otimes \rho_2$ in Example 3.1 is a witness of the lifting $\rho_1\ (\mathcal{H}_1 \otimes \mathcal{H}_2)^{\#}\ \rho_2$.*

The two operators $=_{\mathcal{B}}$ and $=_{sym}$ in the above example represents two different kind of symmetry between two quantum systems with the same state Hilbert space $\mathcal{H}$. They will be used to describe relational properties of the two quantum programs P_1, P_2 in Example 1.1. Liftings of equality are especially interesting for verification, since they can be interpreted as relating equivalent quantum systems. The following proposition characterizes these liftings.

Proposition 3.2. *Let $\rho_1, \rho_2 \in \mathcal{D}^{\leq}(\mathcal{H})$. The following statements are equivalent:*

1. *$\rho_1 = \rho_2$;*
2. *there exists an orthonormal basis $\mathcal{B}$ s.t. $\rho_1(=_{\mathcal{B}})^{\#}\rho_2$;*
3. *$\rho_1(=_{sym})^{\#}\rho_2$.*

We see from Example 3.4 and Proposition 3.2 that in the quantum world, equality has different generalisations $=_{\mathcal{B}}$ and $=_{sym}$. Our logic will establish the existence of a lifting of equality, which then implies equality of density operators, i.e., equivalence of quantum states.

The notion of quantum lifting can be further generalised to a quantitative version, which will be more convenient in defining the semantics of our logic. We first recall a notation introduced in D'Hondt and Panangaden [2006]: $\rho \models_\lambda A$ means $tr(A\rho) \geq \lambda$. It can be understood as a quantitative satisfaction relation between a state ρ and an observable A with a real number $\lambda > 0$ as a threshold.

Definition 3.3. *Let $\rho_1 \in \mathcal{D}^{\leq}(\mathcal{H}_1)$ and $\rho_2 \in \mathcal{D}^{\leq}(\mathcal{H}_2)$, let A be an observable in $\mathcal{H}_1 \otimes \mathcal{H}_2$, and let $\lambda > 0$. Then $\rho \in \mathcal{D}^{\leq}(\mathcal{H}_1 \otimes \mathcal{H}_2)$ is called a witness of the λ-lifting $\rho_1 A^{\#} \rho_2$ if:*

(1) *ρ is a coupling for $\langle \rho_1, \rho_2 \rangle$;*
(2) *$\rho \models_\lambda A$.*

It is obvious that whenever A is the projection onto subspace $\mathcal{X}$ and $\lambda = tr(\rho)$, then the above definition degenerates to Definition 3.2.

3.3 Separable versus Entangled Liftings

Entanglement presents a major difference between classical and quantum systems and is responsible for most of the advantages of quantum computing and communication over their classical counterparts. A partial density operator $\rho \in \mathcal{D}^{\leq}(\mathcal{H}_1 \otimes \mathcal{H}_2)$ is said to be *separable* if there exist $\rho_{mi} \in \mathcal{D}^{\leq}(\mathcal{H}_i)$ $(i = 1, 2)$ such that $\rho = \sum_m (\rho_{m1} \otimes \rho_{m2})$. A (mixed) state ρ in $\mathcal{H}_1 \otimes \mathcal{H}_2$ is said to be *entangled* if it is not separable. Indeed, the notions of separability and entanglement can be defined for a general positive operator (rather than density operator).

The following proposition shows that entanglement can provide a stronger witness of lifting even with respect to a *separable* observable A; that is, sometimes an entangled witness is possible but separable witness does not exist (the proof is given in [Barthe et al. 2019]).

Proposition 3.3. *There are states ρ_i in $\mathcal{H}_i$ $(i = 1, 2)$, separable observable A over $\mathcal{H}_1 \otimes \mathcal{H}_2$, entangled state ρ in $\mathcal{H}_1 \otimes \mathcal{H}_2$, and $\lambda > 0$ such that:*

(1) *ρ is a witness of λ-lifting $\rho_1 A^{\#} \rho_2$; and*
(2) *any separable state σ in $\mathcal{H}_1 \otimes \mathcal{H}_2$ is not a witness of λ-lifting $\rho_1 A^{\#} \rho_2$.*

4 QUANTUM PROGRAMMING LANGUAGE

We recall the syntax and semantics for a quantum programming language given in [Ying 2011, 2016]. Let *Var* be a set of quantum variables. For each $q \in Var$, we write $\mathcal{H}_q$ for its state Hilbert space.

Definition 4.1 (Syntax). *Quantum programs are defined by the following syntax:*

$$P ::= \textbf{skip} \mid P_1; P_2 \mid q := |0\rangle \mid \overline{q} := U[\overline{q}] \mid \textbf{if } (\square m \cdot \mathcal{M}[\overline{q}] = m \rightarrow P_m) \textbf{ fi} \mid \textbf{while } \mathcal{M}[\overline{q}] = 1 \textbf{ do } P \textbf{ od}$$

The initialisation $q := |0\rangle$ sets quantum variable q to a basis state $|0\rangle$. The statement $\overline{q} := U[\overline{q}]$ means that unitary transformation U is applied to register $\overline{q}$. The **if**-statement is a quantum generalisation of a classical case statement. In executing it, measurement $\mathcal{M} = \{M_m\}$ is performed on $\overline{q}$, and then a subprogram P_m is selected to be executed next according to the outcome m of measurement. The **while**-statement is a quantum generalisation of the classical while loop. The measurement in it has only two possible outcomes 0, 1; if the outcome 0 is observed then the program terminates, otherwise the program executes the subprogram P and continues.

We write $var(P)$ for the set of quantum variables occurring in a quantum program P. Then tensor product $\mathcal{H}_P = \bigotimes_{q \in var(P)} \mathcal{H}_q$ is the state Hilbert space of P. A *configuration* is a pair $C = \langle P, \rho \rangle$, where P is a program or the termination symbol $\downarrow$, and $\rho \in \mathcal{D}^{\leq}(\mathcal{H}_P)$ is a partial density operator modeling the state of quantum variables.

Definition 4.2 (Operational Semantics). *The operational semantics of quantum programs is defined as a transition relation $\rightarrow$ by the transition rules in Fig. 1.*

$$\text{(Sk)}\ \langle \mathbf{skip}, \rho\rangle \to \langle \downarrow, \rho\rangle \qquad \text{(In)}\ \langle q := |0\rangle, \rho\rangle \to \langle \downarrow, \rho_0^q\rangle$$

$$\text{(UT)}\ \langle \overline{q} := U[\overline{q}], \rho\rangle \to \langle \downarrow, U\rho U^\dagger\rangle \qquad \text{(SC)}\ \frac{\langle P_1, \rho\rangle \to \langle P_1', \rho'\rangle}{\langle P_1; P_2, \rho\rangle \to \langle P_1'; P_2, \rho'\rangle}$$

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

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

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

Fig. 1. Transition Rules. Symbol $\downarrow$ stands for termination. In rule (In), $\rho_0^q = \sum_i |0\rangle_q\langle i|\rho|i\rangle_q\langle 0|$ for a given orthonormal basis $\{|i\rangle\}$ of $\mathcal{H}_q$. In (IF), m ranges over all possible outcomes of measurement $\mathcal{M}$.

The transitions in rules (IF), (L0) and (L1) are essentially probabilistic. In both **if** and **while** statements, a measurement is performed at the beginning, and then the program enters different branches based on the measurement outcome. For each outcome m, the transition in (IF) happens with probability $p_m = tr(M_m^\dagger M_m \rho)$, and the program state ρ is changed to $\rho_m = M_m \rho M_m^\dagger / p_m$. In rule (L0) and (L1) the outcome "0" occurs with the probability $p_0 = tr(M_0 \rho M_0^\dagger)$, and the program terminates in state $M_0 \rho M_0^\dagger / p_0$; otherwise, with the probability $p_1 = tr(M_1 \rho M_1^\dagger)$, the outcome "1" occurs, the program state is changed to $M_1 \rho M_1^\dagger / p_1$, and then the program executes the loop body P and goes back to the beginning of the loop. We follow a convention suggested by Selinger [2004a] to 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 is useful for presenting the operational semantics as a non-probabilistic transition system, simplifying the presentation.

Definition 4.3 (Denotational Semantics). *For any quantum program P, its semantic function is the mapping $[\![P]\!] : \mathcal{D}^{\leq}(\mathcal{H}_P) \to \mathcal{D}^{\leq}(\mathcal{H}_P)$ defined as follows: for every $\rho \in \mathcal{D}^{\leq}(\mathcal{H}_P)$,*

$$[\![P]\!](\rho) = \sum \{\!|\rho' : \langle P, \rho\rangle \to^* \langle \downarrow, \rho'\rangle|\!\}, \tag{3}$$

where $\to^*$ is the reflexive and transitive closure of $\to$, and $\{\!|\cdot|\!\}$ denotes a multi-set.

For instance, let us consider program Q_2 in our working example 1.1 with input ρ given in equation (1). According to Definition 4.2, it has two transitions:

$$\langle Q_2, \rho\rangle \to \langle q := Z[q], \rho_0\rangle \to \langle \downarrow, \rho_0'\rangle, \qquad \langle Q_2, \rho\rangle \to \langle q := H[q], \rho_1\rangle \to \langle \downarrow, \rho_1'\rangle,$$

where:

$$\rho_0 = M_0' \rho M_0' = \frac{1}{3}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}, \qquad \rho_0' = Z\rho_0 Z = \frac{1}{3}\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix},$$

$$\rho_1 = M_1' \rho M_1' = \frac{1}{6}\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}, \qquad \rho_1' = H\rho_1 H = \frac{1}{3}\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}.$$

According to Definition 4.3, the output is $[\![Q_2]\!](\rho) = \rho_0' + \rho_1' = \frac{1}{3}\begin{pmatrix} 1 & -1 \\ -1 & 2 \end{pmatrix}$. Furthermore, one can show that for any possible input ρ with trace one, programs P_1, P_2 in Example 1.1 have the same output: $[\![P_1]\!](\rho) = [\![P_2]\!](\rho) = \frac{1}{4}\begin{pmatrix} 1 & -1 \\ -1 & 3 \end{pmatrix}$.

The soundness of some of the proof rules in probabilistic relational Hoare logic requires programs to terminate [Barthe et al. 2009]. The same is true in the quantum setting.

Definition 4.4. *A quantum program P is called* lossless*, written $\models P$ lossless, if its semantics function $[\![P]\!]$ is* trace-preserving*; that is, $tr([\![P]\!](\rho)) = tr(\rho)$ for all $\rho \in \mathcal{D}^{\leq}(\mathcal{H}_P)$.*

For example, programs P_1 and P_2 in Example 1.1 are both lossless.

Remark 4.1. *The lossless property of quantum loop* **while** $\mathcal{M}[\overline{q}] = 1$ **do** P **od** *was previously studied [Ying et al. 2013]. Let the semantic function of loop body P be given in Kraus operator-sum form: $[\![P]\!](\rho) = \sum_i E_i \rho E_i^\dagger$. We define (super-)operator $\mathcal{E}$ by $\mathcal{E}(\rho) = \sum_i (M_1^\dagger E_i^\dagger)\rho(E_i M_1)$ for every ρ. A square matrix X is called an eigenvector of $\mathcal{E}$ corresponding to an eigenvalue λ if $\mathcal{E}(X) = \lambda X$. It was shown that the loop is lossless if and only if any eigenvector of $\mathcal{E}$ corresponding to an eigenvalue with modulus 1 is traceless.*

5 RELATIONAL PROGRAM LOGIC

We adopt standard conventions and notations for relational program logics. For each quantum variable $q \in Var$, we assume two tagged copies $q\langle 1\rangle$ and $q\langle 2\rangle$, and their state Hilbert spaces are the same as that of q: $\mathcal{H}_{q\langle 1\rangle} = \mathcal{H}_{q\langle 2\rangle} = \mathcal{H}_q$. For $i = 1, 2$, if $X \subseteq Var$, then we write $X\langle i\rangle = \{q\langle i\rangle \mid q \in X\}$. Furthermore, for every quantum program P with $var(P) \subseteq Var$, we write $P\langle i\rangle$ for the program obtained by replacing each quantum variable q in P with $q\langle i\rangle$. Also, for each operator A in $\mathcal{H}_X = \bigotimes_{q\in X} \mathcal{H}_q$, we write $A\langle i\rangle$ for the corresponding operator of A in $\mathcal{H}_{X\langle i\rangle} = \bigotimes_{q\in X} \mathcal{H}_{q\langle i\rangle}$. For simplicity, we will drop the tags whenever they can be understood from the context; for example, we often simply write $A \otimes B$ instead of $A\langle 1\rangle \otimes B\langle 2\rangle$.

5.1 Judgments and Satisfaction

Judgments in our logic are of the form

$$\Gamma \vdash P_1 \sim P_2 : A \Rightarrow B \tag{4}$$

where P_1 and P_2 are quantum programs, A and B are quantum predicates in $\mathcal{H}_{P_1\langle 1\rangle} \otimes \mathcal{H}_{P_2\langle 2\rangle}$, and Γ is a set of measurement or separability conditions. If $\Gamma = \{\Sigma_1, ..., \Sigma_n\}$, then for any $\rho \in \mathcal{D}^{\leq}\left(\mathcal{H}_{P_1\langle 1\rangle} \otimes \mathcal{H}_{P_2\langle 2\rangle}\right)$, $\rho \models \Gamma$ means $\rho \models \Sigma_i$ for all $i = 1, ..., n$. We defer the definition of measurement or separability condition Σ_i for now, simply assuming a given notion of satisfaction $\rho \models \Sigma_i$. In particular, if $\Gamma = \emptyset$, then we simply write $\vdash P_1 \sim P_2 : A \Rightarrow B$ for $\Gamma \vdash P_1 \sim P_2 : A \Rightarrow B$.

Definition 5.1. *The judgment $\Gamma \vdash P_1 \sim P_2 : A \Rightarrow B$ is valid, written:*

$$\Gamma \models P_1 \sim P_2 : A \Rightarrow B$$

if for every $\rho \in \mathcal{D}^{\leq}\left(\mathcal{H}_{P_1\langle 1\rangle} \otimes \mathcal{H}_{P_2\langle 2\rangle}\right)$ such that $\rho \models \Gamma$, there exists a quantum coupling σ for $\left\langle [\![P_1]\!](tr_{\langle 2\rangle}(\rho)), [\![P_2]\!](tr_{\langle 1\rangle}(\rho))\right\rangle$ such that

$$tr(A\rho) \leq tr(B\sigma) + tr(\rho) - tr(\sigma). \tag{5}$$

We will often use $\rho \models P_1 \sim P_2 : A \Rightarrow B$ as shorthand.

The above definition differs from validity in probabilistic relational Hoare logic in several ways. Except the set Γ of measurability and separability conditions (explained below), lifting does not appear explicitly. However, the existence of a lifting can be established from inequality (5) under mild conditions, as we now explain. First, we note that $tr(\rho) - tr(\sigma)$ captures the non-termination probability of the programs, as in the (non-relational) quantum program logic qPD. To see a clearer probabilistic-quantum correspondence, let us consider the simple case where both P_1 and P_2 are lossless. Then $tr(\rho) - tr(\sigma) = 0$ and inequality (5) is simplified to $tr(A\rho) \leq tr(B\sigma)$, or equivalently: for any $\lambda > 0$, $\rho \models_\lambda A \Rightarrow \sigma \models_\lambda B$. This is a real number-valued analogue of boolean-valued proposition "$\rho \in A \Rightarrow \sigma \in B$". Therefore, for any $\lambda > 0$, if $\rho \models_\lambda A$, then $\sigma \models_\lambda B$ and combined with

$\{=_{\mathcal{B}}\}\{I \otimes I\}$ (Conseq)

$q := |0\rangle; \sim q := |0\rangle;$ (Init)

$\{I \otimes I\}$ $\{\mathrm{SC} : \mathcal{M}' \approx \mathcal{M}'\}$

$q := H[q]; \sim \mathbf{skip};$ (UT-L)

$\{I \otimes I\}$ $\{\mathrm{SC} : \mathcal{M} \approx \mathcal{M}'\}$

$Q_1; \sim Q_2;$ (IF1)

$\left\{B = \frac{1}{2}[I \otimes I + (I \otimes H)S(I \otimes H)]\right\}$

$\mathbf{skip}; \sim q := H[q];$ (UT-R)

$\left\{(=_{sym}) = \frac{1}{2}[I \otimes I + S]\right\}$

Derivation of $Q_1 \sim Q_2$ (IF1)

$\mathcal{M} \approx \mathcal{M}' \models I \otimes I \Rightarrow \{A_{00}, A_{11}\}$

$\left\{A_{00} = \frac{1}{2}[I \otimes I + (X \otimes ZH)S(X \otimes HZ)]\right\}$

$q := X[q]; \sim q := Z[q];$ (UT)

$\left\{B = \frac{1}{2}[I \otimes I + (I \otimes H)S(I \otimes H)]\right\}$

$\left\{A_{11} = \frac{1}{2}[I \otimes I + (H \otimes HH)S(H \otimes HH)]\right\}$

$q := H[q]; \sim q := H[q];$ (UT)

$\left\{B = \frac{1}{2}[I \otimes I + (I \otimes H)S(I \otimes H)]\right\}$

Fig. 2. Verification of working example 1.1 in rqPD: $\mathrm{P}_1 \sim \mathrm{P}_2$. The proof outline is shown in the left column with side-condition labeled by SC, and the derivation of $Q_1 \sim Q_2$ is displayed in the right column with measurement condition.

the assumption that σ is a coupling for $\left\langle [\![P_1]\!](tr_{\langle 2\rangle}(\rho)), [\![P_2]\!](tr_{\langle 1\rangle}(\rho))\right\rangle$, we see that σ is a witness for λ-lifting $[\![P_1]\!](tr_{\langle 2\rangle}(\rho))B^{\#}[\![P_2]\!](tr_{\langle 1\rangle}(\rho))$.

An interesting symmetry between programs P_1, P_2 in our working example 1.1 can be expressed as the following judgment:

$$\vdash P_1 \sim P_2 : (=_{\mathcal{B}}) \Rrightarrow (=_{sym}). \tag{6}$$

where precondition $=_{\mathcal{B}}$ is the equality defined by the computational basis $\mathcal{B} = \{|0\rangle, |1\rangle\}$ of a qubit; i.e. $(=_{\mathcal{B}}) = span\{|00\rangle, |11\rangle\} = |00\rangle\langle 00| + |11\rangle\langle 11|$ [see Example 3.4 2)], and postcondition $=_{sym}$ is the projector onto the symmetric space [see Example 3.4 3)]. The validity of this judgment can be checked by the denotational semantics of P_1, P_2. We first observe that for any $\rho \in \mathcal{D}^{\leq}\left(\mathcal{H}_{P_1} \otimes \mathcal{H}_{P_2}\right)$, $tr(=_{\mathcal{B}} \rho) \leq tr(\rho)$. Moreover, we have:

$$[\![P_1]\!](tr_{\langle 2\rangle}(\rho)) = [\![P_2]\!](tr_{\langle 1\rangle}(\rho)) = \frac{1}{4}\begin{pmatrix} 1 & -1 \\ -1 & 3 \end{pmatrix} \times tr(\rho)$$

by noting that $tr(tr_{\langle 2\rangle}(\rho)) = tr(tr_{\langle 1\rangle}(\rho)) = tr(\rho)$. As shown in Example 3.4 (3), lifting $[\![P_1]\!](tr_{\langle 2\rangle}(\rho))$ $(=_{sym})^{\#}[\![P_2]\!](tr_{\langle 1\rangle}(\rho))$ holds and, suppose σ is a witness, then we have $tr(=_{sym} \sigma) = tr(\sigma)$ and therefore, $tr(=_{\mathcal{B}} \rho) \leq tr(=_{sym} \sigma)$ as $tr(\rho) = tr(\sigma)$ according to Proposition 3.1, which actually implies the validity of judgment (6).

In the remainder of this section, we gradually develop the proof system for our logic rqPD. At the same time, we will see how the proof rules in rqPD can be used to verify judgment (6). For readability, we first give a proof outline of judgment (6) in Fig. 2, where a judgment $\Gamma \vdash P_1 \sim P_2 : A \Rrightarrow B$ derived by an inference rule R in rqPD is displayed as

$$\begin{array}{l} \{A\}\ \{\mathrm{SC} : \Gamma\} \\ \qquad P_1 \sim P_2 \qquad (R) \\ \{B\} \end{array}$$

5.2 Basic Construct-Specific Rules

As usual, the proof system consists of two categories of rules: *construct-specific rules* and *structural rules*. Let us start from a set of construct-specific rules that can be directly adapted from quantum Hoare logic qPD [Ying 2011, 2016]. They include two-side rules and one-side rules given in Figs. 3 and 4, respectively. It is worth noting that all of these rules do not introduce any measurement or separability condition. These rules are easy to understand if compared with the corresponding rules of qPD, which are displayed in [Barthe et al. 2019], and those of probabilistic logic pRHL. For the working example 1.1, these rules are used to prove the following judgments in Fig. 2:

(1) $\vdash q := |0\rangle \sim q := |0\rangle : (=_{\mathcal{B}}) \Rightarrow I \otimes I$ by rule (Conseq) and (Init);
(2) $\vdash q := H[q] \sim \mathbf{skip} : I \otimes I \Rightarrow I \otimes I$ by rule (UT-L);
(3) $\vdash \mathbf{skip} \sim q := H[q] : B \Rightarrow (=_{sym})$ by rule (UT-R); and
(4) $\vdash q := X[q] \sim q := Z[q] : A_{00} \Rightarrow B$ and $\vdash q := H[q] \sim q := H[q] : A_{11} \Rightarrow B$ by rule (UT).

As we will see in Section 6, these basic rules are already enough to verify interesting relational properties of quantum programs, including security of quantum one-time pad.

$$\text{(Skip)}\quad \vdash \mathbf{Skip} \sim \mathbf{Skip} : A \Rightarrow A$$

$$\text{(Init)}\quad \vdash q_1 := |0\rangle \sim q_2 := |0\rangle : \sum_{i,j} (|i\rangle_{q_1\langle 1\rangle}\langle 0| \otimes |j\rangle_{q_2\langle 2\rangle}\langle 0|) A (|0\rangle_{q_1\langle 1\rangle}\langle i| \otimes |0\rangle_{q_2\langle 2\rangle}\langle j|) \Rightarrow A$$

$$\text{(UT)}\quad \vdash \overline{q_1} := U_1[\overline{q_1}] \sim \overline{q_2} := U_2[\overline{q_2}] : \left(U_1^\dagger \otimes U_2^\dagger\right) A (U_1 \otimes U_2) \Rightarrow A$$

$$\text{(SC)}\quad \frac{\vdash P_1 \sim P_2 : A \Rightarrow B \qquad \vdash P_1' \sim P_2' : B \Rightarrow C}{\vdash P_1; P_1' \sim P_2; P_2' : A \Rightarrow C}$$

$$\text{(IF)}\quad \frac{\begin{array}{c}\vdash P_{1m} \sim P_{2n} : B_{mn} \Rightarrow C \text{ for every } (m,n) \in S \qquad \forall\, m,n :\ \models P_{1m}, P_{2n} \text{ lossless} \\ \vdash \mathbf{if}\ (\square m \cdot \mathcal{M}_1[\overline{q}] = m \to P_{1m})\ \mathbf{fi} \sim \mathbf{if}\ (\square n \cdot \mathcal{M}_2[\overline{q}] = n \to P_{2n})\ \mathbf{fi} : \\ \sum_{(m,n)\in S} \left(M_{1m}^\dagger \otimes M_{2n}^\dagger\right) B_{mn} (M_{1m} \otimes M_{2n}) \Rightarrow C\end{array}}{\models \mathbf{while}\ \mathcal{M}_i[\overline{q}] = 1\ \mathbf{do}\ P_i\ \mathbf{od} \text{ lossless } (i = 1,2)}$$

$$\text{(LP)}\quad \frac{\vdash P_1 \sim P_2 : B \Rightarrow (M_{10} \otimes M_{20})^\dagger A (M_{10} \otimes M_{20}) + (M_{11} \otimes M_{21})^\dagger B (M_{11} \otimes M_{21})}{\begin{array}{c}\vdash \mathbf{while}\ \mathcal{M}_1[\overline{q}] = 1\ \mathbf{do}\ P_1\ \mathbf{od} \sim \mathbf{while}\ \mathcal{M}_2[\overline{q}] = 1\ \mathbf{do}\ P_2\ \mathbf{od} : \\ (M_{10} \otimes M_{20})^\dagger A (M_{10} \otimes M_{20}) + (M_{11} \otimes M_{21})^\dagger B (M_{11} \otimes M_{21}) \Rightarrow A\end{array}}$$

Fig. 3. Two-sided rqPD rules. The set S in rule (IF) is a subset of the Cartesian product of the possible outcomes of measurements $\mathcal{M}_1$ and $\mathcal{M}_2$.

Remark 5.1. *Note that in rule (IF) the branches of two case statements are not required to match exactly. Whenever there is an one-to-one correspondence between the outcomes of measurement $\mathcal{M}_1$ and $\mathcal{M}_2$, then (IF) can be simplified to (IF-w) in Fig. 5.*

5.3 Measurement Conditions

The straightforward generalisations of the proof rules for case statements and loops in qPD given in the above subsection are not strong enough for more complicated applications of rqPD. In particular, they do not reveal the subtle differences between the relational and non-relational properties of quantum programs. To understand this point, let us take a closer look at derivation of the judgment about two **if** statements Q_1 and Q_2 in Fig. 2. Let us first list all derivable judgments of possible combinations of branches as follows:

$$\text{(Init-L)} \quad \vdash q_1 := |0\rangle \sim \mathbf{skip} : \sum_i \left(|i\rangle_{q_1\langle 1\rangle}\langle 0|\right) A \left(|0\rangle_{q_1\langle 1\rangle}\langle i|\right) \Rightarrow A$$

$$\text{(UT-L)} \quad \vdash \overline{q_1} := U_1\,[\overline{q_1}] \sim \mathbf{skip} : U_1^\dagger A U_1 \Rightarrow A$$

$$\text{(IF-L)} \quad \frac{\vdash P_{1m} \sim P : B_m \Rightarrow C \text{ for every } m}{\vdash \mathbf{if}\ (\square m \cdot M_1[\overline{q}] = m \rightarrow P_{1m})\ \mathbf{fi} \sim P : \sum_m M_{1m}^\dagger B_m M_{1m} \Rightarrow C}$$

$$\text{(LP-L)} \quad \frac{\models \mathbf{while}\ \mathcal{M}_1[\overline{q}] = 1\ \mathbf{do}\ P_1\ \mathbf{od}\ \text{lossless} \qquad \vdash P_1 \sim \mathbf{skip} : B \Rightarrow M_{10}^\dagger A M_{10} + M_{11}^\dagger B M_{11}}{\vdash \mathbf{while}\ \mathcal{M}_1[\overline{q}] = 1\ \mathbf{do}\ P_1 \mathbf{od} \sim \mathbf{skip} : M_{10}^\dagger A M_{10} + M_{11}^\dagger B M_{11} \Rightarrow A}$$

Fig. 4. One-sided rqPD rules. We omitted the right-sides rules, which are symmetric to the ones here.

$$\text{(IF-w)} \quad \frac{\vdash P_{1m} \sim P_{2m} : B_m \Rightarrow C \text{ for every } m \qquad \forall\, m :\ \models P_{1m}, P_{2m} \text{ lossless}}{\begin{array}{c}\vdash \mathbf{if}\ (\square m \cdot M_1[\overline{q}] = m \rightarrow P_{1m})\ \mathbf{fi} \sim \mathbf{if}\ (\square m \cdot M_2[\overline{q}] = m \rightarrow P_{2m})\ \mathbf{fi} : \\ \sum_m \left(M_{1m}^\dagger \otimes M_{2m}^\dagger\right) B_m\,(M_{1m} \otimes M_{2m}) \Rightarrow C\end{array}}$$

Fig. 5. A weak rule for case statements.

(1) $\vdash q := X[q] \sim q := Z[q] : A_{00} \Rightarrow B$ and $\vdash q := H[q] \sim q := H[q] : A_{11} \Rightarrow B$;
(2) $\vdash q := X[q] \sim q := H[q] : A_{01} \Rightarrow B$ and $\vdash q := H[q] \sim q := Z[q] : A_{10} \Rightarrow B$

where A_{00}, A_{11} and B are given as in Fig. 2 and

$$A_{01} = \frac{1}{2}[I \otimes I + (X \otimes HH)S(X \otimes HH)], \qquad A_{10} = \frac{1}{2}[I \otimes I + (H \otimes ZH)S(H \otimes HZ)].$$

Applying rule (IF) directly we obtain: $\vdash Q_1 \sim Q_2 :\ A \Rightarrow B$, where

$$A = \sum_{i,j=0}^{1} (M_i \otimes M'_j)^\dagger A_{ij} (M_i \otimes M'_j) = \begin{pmatrix} 7/8 & 1/8 & 0 & 0 \\ 1/8 & 7/8 & 0 & 0 \\ 0 & 0 & 7/8 & -1/8 \\ 0 & 0 & -1/8 & 7/8 \end{pmatrix}.$$

Then, using rule (UT-L), (UT-R) and (Init) for the rest parts of the programs, we are only able to derive $\vdash P_1 \sim P_2 :\ \frac{7}{8} I \otimes I \Rightarrow =_{sym}$. However, $=_{\mathcal{B}} \not\sqsubseteq \frac{7}{8} I \otimes I$, so the rule (IF) is too weak to derive judgment (6) as we desire. A similar argument shows that rules (IF-L) and (IF-R) are also too weak.

We have more examples (e.g., Example 6.1) showing that some important relational properties cannot be verified simply using rules (IF) and (LP). The reason can be seen from the soundness proof of (IF) and (LP) [Barthe et al. 2019], where we only use a part of the output states to construct the coupling, so for a given postcondition, the derivable preconditions are sometimes too weak. To resolve this issue, we need to capture more relational information between two programs. A crucial issue in developing inference rules for relational reasoning is to guarantee that two programs P_1 and P_2 under comparison execute in lockstep. In probabilistic relational Hoare logic, a side-condition $\Theta \Rightarrow e_1 = e_2$ is introduced for this purpose, where Θ is the precondition, e_1 and e_2 are the guards in P_1 and P_2, respectively. In the quantum case, branching (control flow) is determined by the measurement outcomes. So, more sophisticated rules for case analysis, loops, and conditionals in rqPD involve *measurement conditions*.

Definition 5.2. *Let $\mathcal{M}_1 = \{M_{1m}\}$ and $\mathcal{M}_2 = \{M_{2m}\}$ be two measurements with the same set $\{m\}$ of possible outcomes in $\mathcal{H}_{P_1}$ and $\mathcal{H}_{P_2}$, respectively, and let $\rho \in \mathcal{D}^{\leq}\left(\mathcal{H}_{P_1\langle 1\rangle} \otimes \mathcal{H}_{P_2\langle 2\rangle}\right)$. Then we say that*

ρ satisfies $\mathcal{M}_1 \approx \mathcal{M}_2$, written $\rho \models \mathcal{M}_1 \approx \mathcal{M}_2$, if $\mathcal{M}_1$ and $\mathcal{M}_2$ produce equal probability distributions when applied to ρ. That is, for all m, we have: $tr(M_{1m} tr_{\langle 2\rangle}(\rho) M_{1m}^\dagger) = tr(M_{2m} tr_{\langle 1\rangle}(\rho) M_{2m}^\dagger)$.

Intuitively, the above measurement conditions mean that P_1 and P_2 enter the corresponding branches with the same probability (and thus execute in lockstep).

The above definition is enough for relating measurements in case statements. But when dealing with loops, we have to consider the measurements in the loop guards together with the loop bodies. To address this issue, we further introduce the following definition:

Definition 5.3. *Let P_1 and P_2 be two programs, and let $\mathcal{M}_1 = \{M_{10}, M_{11}\}$, $\mathcal{M}_2 = \{M_{20}, M_{21}\}$ be boolean-valued measurements in $\mathcal{H}_{P_1}, \mathcal{H}_{P_2}$, respectively. Then for any $\rho \in \mathcal{D}^{\leq}\left(\mathcal{H}_{P_1\langle 1\rangle} \otimes \mathcal{H}_{P_2\langle 2\rangle}\right)$, we say that ρ satisfies $(\mathcal{M}_1, P_1) \approx (\mathcal{M}_2, P_2)$, written*

$$\rho \models (\mathcal{M}_1, P_1) \approx (\mathcal{M}_2, P_2),$$

if $\mathcal{M}_1$ and $\mathcal{M}_2$ produce equal probability distributions in iterations of P_1 and P_2, respectively; that is, for all $n \geq 0$:

$$tr[\mathcal{E}_{10} \circ (\llbracket P_1 \rrbracket \circ \mathcal{E}_{11})^n (tr_{\langle 2\rangle}(\rho))] = tr[\mathcal{E}_{20} \circ (\llbracket P_2 \rrbracket \circ \mathcal{E}_{21})^n (tr_{\langle 1\rangle}(\rho))] \tag{7}$$

where $\mathcal{E}_{ij}(\cdot) = M_{ij}(\cdot)M_{ij}^\dagger$ for $i = 1, 2$ and $j = 0, 1$.

In the above definition, equation (7) is required to hold for all $n \geq 0$ (and thus, for infinitely many n). But the next lemma shows that it can be actually checked within a finite number of steps when the state Hilbert spaces are finite-dimensional, as in our setting. Therefore, an algorithm for checking the measurement condition $(\mathcal{M}_1, P_1) \approx (\mathcal{M}_2, P_2)$ can be designed and incorporated into the tools (e.g. theorem prover) implementing our logic in the future.

Lemma 5.1. *Let $d_i = \dim \mathcal{H}_{P_i}$ $(i = 1, 2)$. If (7) holds for any $0 \leq n \leq d_1^2 + d_2^2 - 1$, then it holds for all $n \geq 0$.*

Note that a branching structure appears after a measurement is performed. To describe it, we introduce the following:

Definition 5.4. *Let $\mathcal{M}_1 = \{M_{1m}\}$ and $\mathcal{M}_2 = \{M_{2m}\}$ be as in Definition 5.2, and let A and B_m be quantum predicates in $\mathcal{H}_{P_1\langle 1\rangle} \otimes \mathcal{H}_{P_2\langle 2\rangle}$. We define:*

$$\mathcal{M}_1 \approx \mathcal{M}_2 \models A \Rightarrow \{B_m\}$$

if for any $\rho \models \mathcal{M}_1 \approx \mathcal{M}_2$, and for each m, there exists a coupling σ_m for $\left\langle M_{1m} tr_{\langle 2\rangle}(\rho) M_{1m}^\dagger, M_{2m} tr_{\langle 1\rangle}(\rho) M_{2m}^\dagger \right\rangle$ such that

$$tr(A\rho) \leq tr\Big(\sum_m B_m \sigma_m\Big). \tag{8}$$

For the working example 1.1, one may check $\mathcal{M} \approx \mathcal{M}' \models I \otimes I \Rightarrow \{A_{00}, A_{11}\}$ as shown in Fig. 2. To see this, suppose $\rho \models \mathcal{M} \approx \mathcal{M}'$. For $m = 0$, $M_0 tr_{\langle 2\rangle}(\rho) M_0 = p_0 |0\rangle\langle 0|$ and $M_0' tr_{\langle 1\rangle}(\rho) M_0' = p_0 |+\rangle\langle +|$ with parameter $p_0 = tr(M_0 tr_{\langle 2\rangle}(\rho) M_0)$, and it is straightforward to check $\sigma_0 = p_0 |0\rangle\langle 0| \otimes |+\rangle\langle +|$ is a witness of lifting $(M_0 tr_{\langle 2\rangle}(\rho) M_0) A_{00}{}^{\#} (M_0' tr_{\langle 1\rangle}(\rho) M_0')$, which leads to $tr(A_{00}\sigma_0) = p_0$. Similar arguments hold for $m = 1$, with $p_1 = tr(M_1 tr_{\langle 2\rangle}(\rho) M_1)$, witness $\sigma_1 = p_1 |1\rangle\langle 1| \otimes |-\rangle\langle -|$ and $tr(A_{11}\sigma_1) = p_1$. Observe that $tr(I \otimes I\rho) = tr(\rho)$ and $tr(\rho) = p_0 + p_1$. Thus, we conclude that $tr(I \otimes I\rho) = tr(A_{00}\sigma_0) + tr(A_{11}\sigma_1)$.

A one-side variant of the above definition will also be useful.

$$\text{(IF1)}\ \frac{\mathcal{M}_1 \approx \mathcal{M}_2 \models A \Rightarrow \{B_m\} \quad \vdash P_{1m} \sim P_{2m} : B_m \Rightarrow C \text{ for every } m}{\mathcal{M}_1 \approx \mathcal{M}_2 \vdash \textbf{if } (\square m \cdot \mathcal{M}_1[\overline{q}] = m \rightarrow P_{1m}) \textbf{ fi} \sim \textbf{if } (\square m \cdot \mathcal{M}_2[\overline{q}] = m \rightarrow P_{2m}) \textbf{ fi} : A \Rightarrow C}$$

$$\text{(LP1)}\ \frac{\mathcal{M}_1 \approx \mathcal{M}_2 \models A \Rightarrow \{B_0, B_1\} \quad \vdash P_1 \sim P_2 : B_1 \Rightarrow A}{(\mathcal{M}_1, P_1) \approx (\mathcal{M}_2, P_2) \vdash \textbf{while } \mathcal{M}_1[\overline{q}] = 1 \textbf{ do } P_1 \textbf{od} \sim \textbf{while } \mathcal{M}_2[\overline{q}] = 1 \textbf{ do } P_2 \textbf{ od} : A \Rightarrow B_0}$$

$$\text{(IF1-L)}\ \frac{\mathcal{M}_1 \approx I_2 \models A \Rightarrow \{B_m\} \quad \vdash P_{1m} \sim P : B_m \Rightarrow C \text{ for every } m}{\vdash \textbf{if } (\square m \cdot \mathcal{M}_1[\overline{q}] = m \rightarrow P_{1m}) \textbf{ fi} \sim P : A \Rightarrow C}$$

$$\models \textbf{while } \mathcal{M}_1[\overline{q}] = 1 \textbf{ do } P_1 \textbf{ od} \text{ lossless}$$

$$\text{(LP1-L)}\ \frac{\mathcal{M}_1 \approx I_2 \models A \Rightarrow \{B_0, B_1\} \quad \vdash P_1 \sim \textbf{skip} : B_1 \Rightarrow A}{\vdash \textbf{while } \mathcal{M}_1[\overline{q}] = 1 \textbf{ do } P_1 \textbf{ od} \sim \textbf{skip} : A \Rightarrow B_0}$$

Fig. 6. More rules for case statements and loops. We omitted the right-sides rules, which are symmetric to the ones displayed here. In (LP1-L), M_{10} and M_{11} only apply on the Hilbert space of the left program, that is, e.g., M_{10} is an abbreviation of $M_{10} \otimes I_2$.

Definition 5.5. *Let $\mathcal{M}_1 = \{M_{1m}\}$, A and B_m be as in Definition 5.4. We define*

$$\mathcal{M}_1 \approx I_2 \models A \Rightarrow \{B_m\},$$

where I_2 stands for the identity operator in $\mathcal{H}_{P_2}$, if for any $\rho \in \mathcal{D}^{\leq}\left(\mathcal{H}_{P_1\langle 1\rangle} \otimes \mathcal{H}_{P_2\langle 2\rangle}\right)$, and for each m, there exist $\rho_{2m} \in \mathcal{D}^{\leq}\left(\mathcal{H}_{P_2\langle 2\rangle}\right)$ and a coupling σ_m for $\left\langle M_{1m} tr_{\langle 2\rangle}(\rho) M_{1m}^{\dagger}, \rho_{2m} \right\rangle$ such that $\sum_m \rho_{2m} = tr_{\langle 1\rangle}(\rho)$ and

$$tr(A\rho) \leq tr\Big(\sum_m B_m \sigma_m\Big). \tag{9}$$

Similarly, we can define $I_1 \approx \mathcal{M}_2 \models A \Rightarrow \{B_m\}$, where I_1 is the identity operator in $\mathcal{H}_{P_1}$.

Now we are ready to present our new rules for case statements and loops in Fig. 6. As pointed out in the Introduction, synchronous rules in non-probabilistic relational Hoare logic RHL and probabilistic logic pRHL for control-flow constructs (conditionals and loops) require that the two programs under comparison execute in lockstep. The control flows of quantum programs studied in this paper are determined by the outcome of measurements. Thus, measurement conditions $\mathcal{M}_1 \approx \mathcal{M}_2$ and $(\mathcal{M}_1, P_1) \approx (\mathcal{M}_2, P_2)$ in our rules (IF1) and (LP1) and their one-side variants are introduced to warrant that the two programs execute in lockstep; more precisely, they enter the same branch in their control flows with equal probabilities.

Using rule (IF1), we are able to derive $\mathcal{M} \approx \mathcal{M}' \vdash Q_1 \sim Q_2 : I \otimes I \Rightarrow B$ for our working example, shown in Fig. 2. Also in Example 6.1, correctness of Quantum Bernoulli Factory is verified using (LP1) while (LP) is too weak to derive the desired judgment.

Comparison between Rules (IF), (LP) and (IF1), (LP1): A careful comparison between the rules without and with measurement conditions is helpful for us to determine where the rules with measurement conditions are needed.

(1) First, we notice that the appearance of the special case (IF-w) of (IF) is similar to (IF1). Indeed, whenever the measurement conditions are true and each branch is terminating, then (IF1) degenerates to (IF-w) provided we set: $A = \sum_m \left(M_{1m}^{\dagger} \otimes M_{2m}^{\dagger}\right) B_m \left(M_{1m} \otimes M_{2m}\right)$. However, this choice of A is much weaker than the best possible choice. To see this, suppose ρ is a coupling of inputs that satisfy the premises, and let $\rho_{1m} = M_{1m} tr_2(\rho) M_{1m}^{\dagger}$ and $\rho_{2m} = M_{2m} tr_1(\rho) M_{2m}^{\dagger}$ for all m. Actually, (IF-w) uses $\sum_m \rho_{1m} \otimes \rho_{2m}$ as part of the coupling of the output states to derive the precondition. However, this state represents only $1/d$ of the output

in general, where d is the dimension of the quantum register being measured. More precisely, the set $\{(M_{1m} \otimes M_{2m})\}_m$ is a part of quantum measurement $\mathcal{M}_1 \otimes \mathcal{M}_2 = \{(M_{1m} \otimes M_{2n})\}_{m,n}$ and only contains about $1/d$ measurement operators of $\mathcal{M}_1 \otimes \mathcal{M}_2$. Consequently, the trace of the coupling state (probability of occurrence) is smaller than possible, which leads to a weaker precondition.

(2) The above defect was remedied in the general rule (IF) by allowing all possible combinations (m, n) rather than only diagonal (m, m). But there is another issue that sometimes prevents (IF) to derive relational properties as strong as those by (IF1). As can be seen in its soundness proof, (IF) simply relates two programs in a manner of tensor product, which does not captures the possible correlation between these programs. Recall that in probabilistic logic pRHL, coupling was introduced to warrant that two programs be executed in a lockstep manner so that sharing randomness can be achieved. The rule (IF1) is proposed for the same purpose and can be used to reason about stronger relational properties of quantum programs, as shown in Example 1.1 as well as Example 6.1. On the other hand, whenever a strong correlation between two programs does not exist or is unnecessary for our purpose (see for instance, Example 6.5 - quantum one-time pad), we prefer to use (IF) because it is simpler.

(3) The same argument applies to the rules (LP) and (LP1) for loops.

5.4 Separability Conditions

We now turn to the structural rules for our logic rqPD. The rules (Conseq), (Weaken) and (Case) of probabilistic relation Hoare logic (pRHL) can be straightforwardly generalised to the quantum setting and are shown in Fig. 7. In (Conseq), we use the Löwner order between quantum predicates (Hermitian operators) in place of boolean implication. The meanings of rules (Weaken) and (Case) are obvious. However, the (Frame) rule requires special care.

$$\text{(Conseq)}\ \frac{\Gamma \vdash P_1 \sim P_2 : A' \Rrightarrow B' \quad A \sqsubseteq A' \quad B' \sqsubseteq B}{\Gamma \vdash P_1 \sim P_2 : A \Rrightarrow B}$$

$$\text{(Weaken)}\ \frac{\Gamma \subseteq \Gamma' \quad \Gamma \vdash P_1 \sim P_2 : A \Rrightarrow B}{\Gamma' \vdash P_1 \sim P_2 : A \Rrightarrow B}$$

$$\text{(Case)}\ \frac{\Gamma \vdash P_1 \sim P_2 : A_i \Rrightarrow B\ (i = 1, ..., n) \quad \{p_i\} \text{ is a probability distribution}}{\Gamma \vdash P_1 \sim P_2 : \sum_{i=1}^n p_i A_i \Rrightarrow B}$$

$$\text{(Frame)}\ \frac{\Gamma \vdash P_1 \sim P_2 : A \Rrightarrow B}{\Gamma \cup \{[V, var(P_1, P_2)]\} \vdash P_1 \sim P_2 : A \otimes C \Rrightarrow B \otimes C}$$

Fig. 7. Structural rqPD rules. In (Conseq), $\sqsubseteq$ stands for the Löwner order between operators. In (Frame), $V \bigcap var(P_1, P_2) = \emptyset$ and C is a quantum predicate in $\mathcal{H}_V$.

A typical difficulty in reasoning about a quantum system is entanglement between its subsystems. The notions of bipartite separability and entanglement considered in Subsection 3.3 can be generalised to the case of more than two subsystems. A partial density operator ρ in $\bigotimes_{i=1}^n \mathcal{H}_i$ is separable between $\mathcal{H}_i$ $(i = 1, ..., n)$ if there exist partial density operators $\rho_{mi} \in \mathcal{D}^{\leq}(\mathcal{H}_i)$ such that $\rho = \sum_m \left(\bigotimes_{i=1}^n \rho_{mi}\right)$. The following separability condition can be introduced to specify that certain entanglement is not provided (as a resource) or not allowed (e.g. between an adversary and a storage containing sensitive information).

Definition 5.6. *Let P_1, P_2 be two programs and $\Sigma = [X_1, ..., X_n]$ a partition of $var(P_1\langle 1\rangle) \cup var(P_2\langle 2\rangle)$. Then we say that a state $\rho \in \mathcal{D}^{\leq}\left(\mathcal{H}_{P_1\langle 1\rangle} \otimes \mathcal{H}_{P_2\langle 2\rangle}\right)$ satisfies separability condition Σ, written $\rho \models \Sigma$, if ρ is separable between $\mathcal{H}_{X_i}$ $(i = 1, ..., n)$.*

$$\text{(SC+)}\quad \frac{\Gamma \vdash P_1 \sim P_2 : A \Rightarrow B \qquad \Delta \vdash P_1' \sim P_2' : B \Rightarrow C \qquad \Gamma \overset{(P_1,P_2)}{\models} \Delta}{\Gamma \vdash P_1; P_1' \sim P_2; P_2' : A \Rightarrow C}$$

Fig. 8. Strong sequential rule

With the above definition, we can define the (Frame) rule for quantum programs in Fig. 7 where a separability condition between the programs P_1, P_2 and the new predicate C. Recall that in probabilistic logic pRHL, the frame rule allows an assertion C to be carried from the precondition through to the postcondition. The validity of the frame rule is based on the assumption that the two programs P_1 and P_2 cannot modify the (free) variables in C; or mathematically speaking, $var(P_1, P_2) \cap var(C) = \emptyset$. In the quantum setting, however, the syntactic disjointness between $var(P_1, P_2)$ and $var(C)$ is not enough. Indeed, an entanglement can occur between them even if they are disjoint, and some properties of the subsystem denoted by the variables in C can be changed by certain actions, say measurements of P_1 or P_2. So, the separability condition $\Gamma = [V, var(P_1, P_2)]$ is introduced in the conclusion part of the frame rule to exclude such an entanglement between P_1, P_2 and C, where V is the set of quantum variables appearing in C.

5.5 Entailment between Side-Conditions

In the above two subsections, measurement and separability conditions are introduced into our logic rqPD. But the (SC) rule for sequential composition in Fig. 3 does not contain these conditions. So, it must be accordingly strengthened to accommodate the propagation of measurement and separability conditions. To this end, we need the following:

Definition 5.7. *Let Γ, Δ be two sets of measurement or separability conditions, and P_1, P_2 two programs. We say that Δ is couple-entailed by Γ through (P_1, P_2), written*

$$\Gamma \overset{(P_1,P_2)}{\models} \Delta, \tag{10}$$

if for any $\rho \models \Gamma$, whenever σ is a coupling for $\left\langle [\![P_1]\!]\left(tr_{\langle 2\rangle}(\rho)\right), [\![P_2]\!]\left(tr_{\langle 1\rangle}(\rho)\right)\right\rangle$, then $\sigma \models \Delta$.

Using the above definition, a strengthened version of (SC) is presented as rule (SC+) in Fig. 8.

Let us go back to the working example 1.1. After a direct calculation, we are able to show

$$\emptyset \overset{(q:=|0\rangle, q:=|0\rangle)}{\models} \mathcal{M}' \approx \mathcal{M}' \quad \text{and} \quad \mathcal{M}' \approx \mathcal{M}' \overset{(q:=H[q], \mathbf{skip})}{\models} \mathcal{M} \approx \mathcal{M}'.$$

Now, using rule (SC+) we can combine all the segment judgments shown in last few sections together to reason about judgment (6), as shown in Fig. 2.

In general, it is not easy to use this rule because it is hard to check a side condition of the form (10). However, in most of the applications, we do not need the full power of (SC+) because most of conditionals and loops can be dealt with by using (IF) and (LP) where no measurement conditions are introduced and thus $\Gamma = \Delta = \emptyset$, and (IF1) and (LP1) are only employed for a few times. In particular, if we only need (IF1) or (LP1) to reason about a single conditional or loop, then $\Delta = \emptyset$ and side condition (10) is trivially valid; for instance, Example 6.1 is actually this case. The difficulty of applying (SC+) will arise only when (IF1) and (LP1) are needed to reason about many conditionals and loops. In the case of finite-dimensional state Hilbert spaces, for a large class of quantum programs, this difficulty can be significantly eased by a back-tracking strategy for collecting a set of measurement or separability conditions at the beginning of a sequence of consecutive judgments in order to warrant that all side-conditions of the form (10) are valid. An elaboration of this strategy is given in [Barthe et al. 2019].

5.6 Auxiliary Rules for General Quantum Operations

Our programming language only contains two simple kinds of quantum operations: unitary transformations and quantum measurements. Also, the post-measurement states are recorded in the semantics so that the dimension of the state Hilbert space is fixed. In applications, however, it is often convenient to apply general quantum operations; for example, quantum noises and communication channels. Principally, a general quantum operation can be implemented by unitary transformations and measurements through introducing ancilla systems and discarding post-measurement states (see [Nielsen and Chuang 2002], Section 8.2.2 for the system-environment model). But the implementations are usually quite complicated. So, for convenience, we choose to expand the programming language by explicitly adding program constructs of the form:

$$P ::= \overline{q} := \mathcal{E}[\overline{q}] \tag{11}$$

where $\mathcal{E}$ is a general quantum operation. Mathematically, $\mathcal{E}$ is modelled as trace-preserving super-operator with Hilbert space $\mathcal{H}_{\overline{q}}$ as its domain, but its codomain can be a different Hilbert space, even with a different dimension, e.g. $\mathcal{H}_{\overline{q}\setminus\overline{q'}}$ or $\mathcal{H}_{\overline{q}\cup\overline{q'}}$. It is well-known that a super-operator $\mathcal{E}$ can be represented by a set of operators $\{E_i\}$ (Kraus operator-sum representation): $\mathcal{E}(\rho) = \sum_i E_i \rho E_i^\dagger$ for every state ρ in $\mathcal{H}_{\overline{q}}$.

The operational semantics of program $\overline{q} := \mathcal{E}[\overline{q}]$ is defined by the following transition rule:

$$\langle \overline{q} := \mathcal{E}[\overline{q}], \rho \rangle \to \langle \downarrow, \mathcal{E}(\rho) \rangle.$$

Based on this, the denotational semantics of quantum programs containing super-operators can be defined in the same way as Definition 4.3, provided allowing that the domain and codomain of the semantic function of a quantum program can be different.

We present three inference rules for general quantum operations in Fig. 9, generalising rules (UT), (UT-L) and (UT-R), respectively.

$$\text{(SO)} \quad \vdash \overline{q_1} := \mathcal{E}_1[\overline{q_1}] \sim \overline{q_2} := \mathcal{E}_2[\overline{q_2}] : \left(\mathcal{E}_1^* \otimes \mathcal{E}_2^*\right)(A) \Rightarrow A$$
$$\text{(SO-L)} \quad \vdash \overline{q_1} := \mathcal{E}_1[\overline{q_1}] \sim \mathbf{skip} : \mathcal{E}_1^*(A) \Rightarrow A$$

Fig. 9. Rules for trace-preserving super-operators (quantum operations). We use $\mathcal{E}^*$ to denote the dual of super-operator $\mathcal{E}$; that is, $\mathcal{E}^*(A) = \sum_i E_i^\dagger A E_i$ if $\mathcal{E}(\rho) = \sum_i E_i \rho E_i^\dagger$. Rule (SO-R) is symmetric to (SO-L).

Remark 5.2. *It is worth mentioning that allowing different dimensions of the domain and codomain of $\mathcal{E}$ in (11) has a benefit; that is, it enables us to introduce auxiliary quantum variables or discard a quantum variable. The construct of introducing auxiliary variables can be defined as cylinder extension, i.e. tensor product with the identity operator of the state Hilbert space of the auxiliary variables (divided by its dimension for normalisation), and the construct of discarding a quantum variable $q \in \overline{q}$ is indeed included in Selinger's quantum programming language QPL [Selinger 2004b]. It can be defined as a partial trace $\mathbf{Tr}[q]$, with its semantics described as a special super-operator: $\mathcal{E}(\rho) = \sum_i \langle i|\rho|i\rangle \in \mathcal{D}(\mathcal{H}_{\overline{q}\setminus\{q\}})$ for any $\rho \in \mathcal{D}(\mathcal{H}_{\overline{q}})$, where $\{|i\rangle\}$ is an orthonormal basis of $\mathcal{H}_q$. Then rules (SO) and its one-side variants (SO-L), (SO-R) warrant that introducing auxiliary variables and discarding a variable can be safely done in relational reasoning.*

5.7 Soundness Theorem

We can prove that our proof system is sound with respect to validity of judgments. The proof of soundness is given in the complete version of this paper [Barthe et al. 2019].

Theorem 5.1 (Soundness). *Derivable judgments are valid:*

$$\Gamma \vdash P_1 \sim P_2 : A \Rightarrow B \implies \Gamma \models P_1 \sim P_2 : A \Rightarrow B$$

Completeness of relational logics is a challenging problem. In the deterministic setting, relational Hoare logic can be shown to be relatively complete for terminating programs provided it includes sufficiently many one-sided rules. Relative completeness fails for probabilistic relational Hoare logic; a further potential complication is that the coupling method — upon which probabilistic relational Hoare logic builds — is itself not complete for proving convergence Markov chains [Anil Kumar and Ramesh 2001].

6 EXAMPLES

In this section, we give several examples to illustrate the power of rqPD. We mainly show how their relational properties can be formally specified as rqPD judgments. Due to the limited space, only a proof outline of the first example is given, and the formal derivations of other judgments are deferred to [Barthe et al. 2019].

6.1 Symmetry between Simple Programs

Let us start from our working example 1.1. In last section, we already proved judgment (6) in our logic rqPD. A symmetry between programs P_1 and P_2 modelled by judgment (6) is more interesting than their similarity we can observe at the first glance. It worth noting that two different kinds of "quantum equality" are used in the precondition and postcondition. To understand this judgment better, let recall Proposition 3.2, which give us an intuition of what $=_{\mathcal{B}}$ and $=_{sym}$ mean. The judgment tell us that, if the inputs of P_1 and P_2 are the same, then the outputs are also same, or in other words, program P_1 and P_2 are actually equal.

Remark 6.1. *As discussed before, rule (IF1) is necessary to derive judgment (6) while using rule (IF-w) or more general (IF) is impossible to prove it. A strong correlation between two programs can be detected only if we run them in a lockstep manner. This is why rule (IF1) works. However, (IF) only requires that two programs run simultaneously while lockstep is not guaranteed. Therefore, it is not surprising to see (IF) fails here.*

6.2 Uniformity

An elegant characterisation of uniform probability distribution with coupling was given in [Barthe et al. 2017]. Unfortunately, the characterisation does not directly carry over to the quantum setting. In this subsection, we show how an alternative approach based on quantum coupling can be used to describe uniformity in quantum systems. Let $\mathcal{H}$ be a Hilbert space and $\mathcal{B} = \{|i\rangle\}$ be an orthonormal basis of $\mathcal{H}$. For each i, we write $M_i = |i\rangle\langle i|$. Then the measurement in basis $\mathcal{B}$ is defined as $\mathcal{M}_{\mathcal{B}} = \{M_i\}$.

Definition 6.1. *A density operator ρ in $\mathcal{H}$ with $d = \dim \mathcal{H}$ is called* uniform *in basis $\mathcal{B}$ if the outcome of measurement $\mathcal{M}_{\mathcal{B}}$ on ρ is uniformly distributed; i.e. for every i,*

$$p_i = tr(M_i \rho) = \langle i|\rho|i\rangle = \frac{1}{d}.$$

The following proposition gives a characterization of uniformity of a program's outputs in terms of quantum coupling.

Proposition 6.1 (Uniformity by coupling). *Let P be a quantum program, $\mathcal{B} = \{|i\rangle\}$ be an orthonormal basis of $\mathcal{H}_P$ and $d = \dim \mathcal{H}_P$. Then the following three statements are equivalent:*

(1) *for any input density operator ρ in $\mathcal{H}_P$, output $[\![P]\!](\rho)$ is uniform in basis $\mathcal{B}$;*

(2) *for any basis state* $|i\rangle$ *in* $\mathcal{B}$*,*

$$\models P \sim P : \frac{I \otimes I}{d} \Rightarrow |i\rangle\langle i| \otimes I \tag{12}$$

where I *is the identity operator in* $\mathcal{H}_P$*;*

(3) *for any basis state* $|i\rangle$ *in* $\mathcal{B}$*,*

$$[var(P\langle 1\rangle), var(P\langle 2\rangle)] \models P \sim P :=^e_C \Rightarrow |i\rangle\langle i| \otimes I \tag{13}$$

where $C = \{|j\rangle\}$ *is an arbitrary orthonormal basis of* $\mathcal{H}_P$*, and the equality operator* $=^e_C$ *is defined to be* $|\Phi\rangle\langle\Phi|$*, where* Φ *is the maximally entangled state* $|\Phi\rangle = \frac{1}{\sqrt{d}}\sum_j |jj\rangle$*. More precisely,* $=^e_C$ *is (the projection onto) the one-dimensional subspace spanned by the maximally entangle state* $|\Phi\rangle$*. It is interesting to see that in judgment (13), a separability condition is enforced on inputs, but entanglement appears in the precondition* $=^e_C$*. This suggests that entanglement cannot be avoid in such a characterisation of uniformity.*

Now, we consider uniformity for a concrete quantum protocol. The Bernoulli factory (BF) [Keane and O'Brien 1994] is a protocol for random number generation. It uses a coin with an unknown probability p of heads to simulate a new coin that has probability $f(p)$ of heads for a given function $f : [0,1] \to [0,1]$. The Quantum Bernoulli factory (QBF) [Dale et al. 2015] also generates classical randomness (e.g., a biased coin with probability $f(p)$), but it uses quantum coins instead of classical coins. Interestingly, QBF can simulate a strictly larger class of functions f than those simulated by BF. As a direct application of the above proposition, we can verify a simplified version of quantum Bernoulli factory in our logic.

Example 6.1 (Simplified Quantum Bernoulli Factory). *Suppose we have a two-qubit system with state Hilbert space* $\mathcal{H}_{q_x} \otimes \mathcal{H}_{q_y}$ *and an initial state* $|0\rangle_{q_x}|0\rangle_{q_y}$*. We are allowed to perform projective measurement* $\mathcal{M} = \{M_0, M_1\}$*:*

$$M_0 = |0\rangle_{q_x}\langle 0| \otimes |1\rangle_{q_y}\langle 1| + |1\rangle_{q_x}\langle 1| \otimes |0\rangle_{q_y}\langle 0|, \quad M_1 = |0\rangle_{q_x}\langle 0| \otimes |0\rangle_{q_y}\langle 0| + |1\rangle_{q_x}\langle 1| \otimes |1\rangle_{q_y}\langle 1|$$

and apply a given—but unknown—one-qubit unitary transformation U *such that* $0 < |\langle 0|U|0\rangle| < 1$ *on system* x *or* y*.*[2] *How can we produce the uniform state* $\frac{1}{2}I_{q_x}$*? The following quantum program accomplishes this task:*

$$\text{QBF} \equiv q_x := |0\rangle;\ q_y := |0\rangle;\ \textbf{while}\ \mathcal{M}[q_x, q_y] = 1\ \textbf{do}\ q_x := U[q_x];\ q_y := U[q_y]\ \textbf{od};\ \textbf{Tr}[q_y]$$

where **Tr** *stands for the partial trace over system* q_y*.*

Note that state $\frac{1}{2}I_{q_x}$ is the only density operators being uniform in any orthonormal basis $\mathcal{B}$. With Proposition 6.1, QBF can be verified by proving that for any $|\psi\rangle \in \mathcal{H}_{q_x}$:

$$\models \text{QBF} \sim \text{QBF} : \frac{1}{2}I_{q_x} \otimes I_{q_y} \otimes I_{q'_x} \otimes I_{q'_y} \Rightarrow |\psi\rangle_{q_x}\langle\psi| \otimes I_{q'_x}. \tag{14}$$

Since this judgment is valid for all $|\psi\rangle$, the output is uniform in all basis so the output state must be $\frac{1}{2}I_{q_x}$.

It is worth pointing out that rule (LP1) plays an essential role in the proof. All registers are initialised before the loop and therefore, we are able to run two of the same QBF in a lockstep manner. Thus, rule (LP) is too weak to derive judgment (14). Rule (SO) is also needed in the verification of (14) because $\textbf{Tr}[q_y]$ appears at the end of QBF. Indeed, if we do not trace out system q_y at the end, then QBF outputs the Bell state $\frac{1}{\sqrt{2}}(|0\rangle_{q_x}|1\rangle_{q_y} + |1\rangle_{q_x}|0\rangle_{q_y})$. This fact can also be realized in our logic. Moreover, it implies that our program QBF is not a trivial generalisation of classical Bernoulli factory because it is capable of producing the maximally entangled state.

[2] In the classical BF, this condition means that the coin must be non-trivial—it cannot always return 0 or always return 1.

6.3 Quantum Teleportation

Now we consider a more sophisticated example. Quantum teleportation [Bennett et al. 1993] is arguably the most famous quantum communication protocol. With it, quantum information (e.g. the exact state of an atom or a photon) can be transmitted from one location to another, only through classical communication, but with the help of previously shared entanglement between the sender and the receiver. The correctness of quantum teleportation has been formally verified by several different methods in the literature, e.g. using categorical formalism of quantum mechanics [Abramsky and Coecke 2004]. Our logic provides a new way for verifying the correctness of quantum teleportation; more importantly, it can be used to verify the reliability of quantum teleportation against various kinds of quantum noise. To the best of our knowledge, this is the first formal verification of its reliability.

Example 6.2. *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 an arbitrary qubit 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 [Nielsen and Chuang 2002], Section 1.3.7). If we regard p as the input state and r the output state, then this protocol can be modeled by a quantum program:*

$$\begin{aligned}\mathrm{QTEL} \equiv\ & q := |0\rangle;\ r := |0\rangle;\ q := H[q];\ q, r := \mathrm{CNOT}[q, r];\ p, q :=\ \mathrm{CNOT}[p, q];\ p := H[p];\\ & \textbf{if } (\mathcal{M}[q] = 0 \rightarrow \textbf{skip} \square 1 \rightarrow r := X[r])\ \textbf{fi};\\ & \textbf{if } (\mathcal{M}[p] = 0 \rightarrow \textbf{skip} \square 1 \rightarrow r := Z[r])\ \textbf{fi}\end{aligned}$$

where H is the Hadamard gate, X and Z are the Pauli gates, CNOT *is the controlled-NOT:*

$$\mathrm{CNOT} = \begin{pmatrix} I & 0 \\ 0 & X \end{pmatrix},$$

$I, 0$ are the 2×2 unit and zero matrices, respectively, and $\mathcal{M}$ is the measurement in the computational basis, i.e. $\mathcal{M} = \{M_0, M_1\}$, where $M_0 = |0\rangle\langle 0|$, $M_1 = |1\rangle\langle 1|$.

6.3.1 Correctness of Quantum Teleportation. In this subsection, we show how our logic can be used to verify the correctness of quantum teleportation. The correctness of QTEL can be described as the judgment:

$$\models \mathrm{QTEL} \sim \textbf{skip} : (=_{\mathcal{B}}) \Rightarrow (=_{\mathcal{B}}), \tag{15}$$

where $\mathcal{B} = \{|\psi\rangle, |\phi\rangle\}$ is an arbitrary othornormal basis of the state Hilbert space of a qubit, and $(=_{\mathcal{B}}) = |\psi\rangle|\psi\rangle\langle\psi|\langle\psi| + |\phi\rangle|\phi\rangle\langle\phi|\langle\phi|$ is the projector onto the subspace $\mathrm{span}\{|\psi\rangle|\psi\rangle, |\phi\rangle|\phi\rangle\}$ [see Example 3.4 (2)]. Indeed, for any input states ρ, there always exists an orthonormal basis $\mathcal{B}$ such that $\rho(=_{\mathcal{B}})^{\#}\rho$, and we assume that a witness for this lifting is σ. From judgment (15), we know that there exists a coupling σ' for $\big\langle [\![\mathrm{QTEL}]\!](\rho), [\![\textbf{skip}]\!](\rho)\big\rangle$ such that $tr(=_{\mathcal{B}} \sigma') \geq tr(=_{\mathcal{B}} \sigma) = 1$. So, σ' is a witness of lifting: $[\![\mathrm{QTEL}]\!](\rho)(=_{\mathcal{B}})^{\#}[\![\textbf{skip}]\!](\rho)$, which, together with Proposition 3.2, implies $[\![\mathrm{QTEL}]\!](\rho) = [\![\textbf{skip}]\!](\rho) = \rho$.

Interestingly, the correctness of QTEL can also be described as the following judgment:

$$\models \mathrm{QTEL} \sim \textbf{skip} : (=_{sym}) \Rightarrow (=_{sym}) \tag{16}$$

using a different equality $=_{sym}$, that is, the projector onto the symmetric subspace [see Example 3.4 (3)]. A similar argument shows that for any input ρ, we have $[\![\mathrm{QTEL}]\!](\rho) = [\![\textbf{skip}]\!](\rho) = \rho$.

The proof of these two judgments are somewhat easy. Unlike the previous two examples, the basic construct-specific rule (IF-L) is enough to derive the results.

6.3.2 Reliability of Quantum Teleportation. In this subsection, we further show that our logic can be used to deduce not only correctness but also reliability of quantum teleportation when its actual implementation suffers certain physical noise.

Quantum noise are usually modelled by super-operators, a more general class of quantum operations than unitary transformations.

Example 6.3 (Noise of Qubits, [Nielsen and Chuang 2002], Section 8.3).

(1) *The bit flip noise flips the state of a qubit from* $|0\rangle$ *to* $|1\rangle$ *and vice versa with probability* $1-p$*, and can be modelled by super-operator:*

$$\mathcal{E}_{BF}(\rho) = E_0\rho E_0 + E_1\rho E_1 \tag{17}$$

for all ρ*, where*

$$E_0 = \sqrt{p}I = \sqrt{p}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad E_1 = \sqrt{1-p}X = \sqrt{1-p}\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

(2) *The phase flip noise can be modelled by the super-operator* $\mathcal{E}_{PF}$ *with*

$$E_0 = \sqrt{p}I = \sqrt{p}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad E_1 = \sqrt{1-p}Z = \sqrt{1-p}\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

(3) *The bit-phase flip noise is modelled by the super-operator* $\mathcal{E}_{BPF}$ *with*

$$E_0 = \sqrt{p}I = \sqrt{p}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad E_1 = \sqrt{1-p}Y = \sqrt{1-p}\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}.$$

where X, Y, Z *are Pauli matrices.*

We sometimes write the bit flip super-operators as $\mathcal{E}_{BF}(p)$ in order to explicitly specify the flip probability p. The same convention is applied to the phase flip $\mathcal{E}_{PF}$ and bit-phase flip $\mathcal{E}_{BPF}$.

Example 6.4. *If the bit flip noise occurs after the Hadamard gates on both qubit* p *and* q*, then the teleportation programs becomes:*

$$\begin{aligned} \mathrm{QTEL}_{BF} \equiv\ & q := |0\rangle;\ r := |0\rangle;\ q := H[q]; q := \mathcal{E}_{BF}[q];\ q, r :=\ \mathrm{CNOT}[q, r]; \\ & p, q :=\ \mathrm{CNOT}[p, q];\ p := H[p]; p := \mathcal{E}_{BF}[p]; \\ & \textbf{if}\ (\mathcal{M}[q] = 0 \rightarrow \textbf{skip} \square\ 1 \rightarrow r := X[r])\ \textbf{fi}; \\ & \textbf{if}\ (\mathcal{M}[p] = 0 \rightarrow \textbf{skip} \square\ 1 \rightarrow r := Z[r])\ \textbf{fi} \end{aligned}$$

where $\mathcal{E}_{\mathrm{BF}}$ *is the bit flip super-operator. Moreover, we write* QTEL_{PF} *and* QTEL_{BPF} *for the phase flip and bit-phase flip occurring at the same positions.*

Now the reliability of QTEL with the different noises—bit flip, phase flip and bit-phase flip—is modelled by the judgments:

$$\models \mathrm{QTEL}_{BF} \sim \mathrm{QTEL} : \mathcal{E}_{PF}(p)(|\psi\rangle_p\langle\psi|) \otimes |\psi\rangle_{p'}\langle\psi| \Rightarrow |\psi\rangle_r\langle\psi| \otimes |\psi\rangle_{r'}\langle\psi|, \tag{18}$$

$$\models \mathrm{QTEL}_{PF} \sim \mathrm{QTEL} : \mathcal{E}_{PF}(p)(|\psi\rangle_p\langle\psi|) \otimes |\psi\rangle_{p'}\langle\psi| \Rightarrow |\psi\rangle_r\langle\psi| \otimes |\psi\rangle_{r'}\langle\psi|, \tag{19}$$

$$\models \mathrm{QTEL}_{BPF} \sim \mathrm{QTEL} : \mathcal{E}_{PF}(p^2 + (1-p)^2)(|\psi\rangle_p\langle\psi|) \otimes |\psi\rangle_{p'}\langle\psi| \Rightarrow |\psi\rangle_r\langle\psi| \otimes |\psi\rangle_{r'}\langle\psi|. \tag{20}$$

To understand these judgments better, let us choose pure state $|\psi\rangle$ as the input of both QTEL_{BF} and QTEL as an example. The correctness of QTEL has been verified and therefore, $[\![\mathrm{QTEL}]\!](|\psi\rangle\langle\psi|) = |\psi\rangle\langle\psi|$. We assume that $[\![\mathrm{QTEL}_{BF}]\!](|\psi\rangle\langle\psi|) = \rho$. There exists a unique coupling $\rho \otimes |\psi\rangle\langle\psi|$ for the outputs $\langle\rho, |\psi\rangle\langle\psi|\rangle$, and according to the judgment (18), we know that:

$$tr(\mathcal{E}_{PF}(p)(|\psi\rangle_p\langle\psi|) \otimes |\psi\rangle_{p'}\langle\psi| \cdot |\psi\rangle_p\langle\psi| \otimes |\psi\rangle_{p'}\langle\psi|) \leq tr(|\psi\rangle_r\langle\psi| \otimes |\psi\rangle_{r'}\langle\psi| \cdot \rho \otimes |\psi\rangle_{r'}\langle\psi|);$$

that is, $\langle\psi|\rho|\psi\rangle \geq p + (1-p)|\langle\psi|Z|\psi\rangle|^2$. Whenever p is close to 1, then ρ is also close to $|\psi\rangle\langle\psi|$, and this is what reliability actually means.

Judgemens (18), (19) and (20) can be verified in our logic rqPD using rule (SO-L) for general quantum operations and rule (IF) for all pairwise comparisons.

6.4 Quantum One-Time Pad

In this subsection, we show that our logic can be used to specify and verify correctness and security of a basic quantum encryption scheme, namely the quantum one-time pad (QOTP) [Boykin and Roychowdhury 2003; Mosca et al. 2000]. Similar to the classical one-time pad, it uses a one-time pre-shared secret key to encrypt and decrypt the quantum data, providing the information-theoretic security. We first consider the simplest case, for protecting one-qubit data.

Example 6.5. *The QOTP scheme includes three parts: key generation* **KeyGen**, *encryption* **Enc** *and decryption* **Dec**, *which can be written as programs:*

$$\begin{aligned}
\mathbf{KeyGen} \equiv\ & a := |0\rangle; b := |0\rangle;\ a := H[a]; b := H[b]; \\
& \mathbf{if}\ \mathcal{M}[a,b] = 00 \to \mathbf{skip} \square\ 01 \to \mathbf{skip} \\
& \qquad \square\ 10 \to \mathbf{skip} \square\ 11 \to \mathbf{skip} \\
& \mathbf{fi} \\
\mathbf{Enc} \equiv \mathbf{Dec} \equiv\ & \mathbf{if}\ \mathcal{M}[a,b] = 00 \to \mathbf{skip} \qquad \square\ 01 \to p = Z[p] \\
& \qquad \square\ 10 \to p = X[p] \square\ 11 \to p = Z[p]; p = X[p] \\
& \mathbf{fi} \\
\mathbf{DisKey} \equiv\ & \mathbf{Tr}[a]; \mathbf{Tr}[b]
\end{aligned}$$

Here, registers a and b are used as the secret key, and measurement

$$\mathcal{M} = \{M_{00} = |00\rangle_{ab}\langle 00|, M_{01} = |01\rangle_{ab}\langle 01|, M_{10} = |10\rangle_{ab}\langle 10|, M_{11} = |11\rangle_{ab}\langle 11|\}$$

is introduced to detect the value of secret key, which has two-bit classical outcome. Register p is the input quantum data which we want to protect. H is the Hadamard gate and X, Z are Pauli gates as usual. As the secret key is not considered when analysing the correctness and security of the protocol, we further introduce **DisKey** *to discard the key.*

6.4.1 Correctness of Quantum One-Time Pad. The correctness of QOTP can be formulated as the following judgment:

$$\vdash \mathbf{KeyGen}; \mathbf{Enc}; \mathbf{Dec}; \mathbf{DisKey} \ \sim\ \mathbf{skip} : (=_{sym}) \Rightarrow (=_{sym}). \tag{21}$$

where $=_{sym}$ represents the projector onto the symmetric space. By an argument similar to that for the correctness of quantum teleportation in Section 6.3.1, we can show that if judgment (21) is valid, then for any possible input ρ of register p,

$$[\![\mathbf{KeyGen}; \mathbf{Enc}; \mathbf{Dec}; \mathbf{DisKey}]\!](\rho) = [\![\mathbf{skip}]\!](\rho) = \rho;$$

that is, the input state is the same as the output after QOTP.

Judgment (21) can be verified mainly with rule (IF-L). But note that the initialisations of registers a and b are regarded as the creation of new local qubits. So, rules (SO-L) and (SO-R) are needed here instead of (Init-L).

6.4.2 Security of Quantum One-Time Pad. Using the characterisation of uniformity given in Section 6.2, we may verify the security of QOTP by proving that for any $|\psi\rangle \in \mathcal{H}_p$:

$$\vdash \mathbf{KeyGen}; \mathbf{Enc}; \mathbf{DisKey} \sim \mathbf{KeyGen}; \mathbf{Enc}; \mathbf{DisKey} : \frac{I_p \otimes I_{p'}}{2} \Rightarrow |\psi\rangle_p\langle\psi| \otimes I_{p'}. \tag{22}$$

In fact, if judgment (22) is valid for all $|\psi\rangle$, then the output is uniform in all bases; that is, the output of register p is $\frac{1}{2}I_p$, which is actually the maximally mixed state and nothing can be inferred from it. To derive this judgment, we need rule (SO) and (IF-w).

6.4.3 General Quantum One-Time Pad. Now let us generalise Example 6.5 to the general case for protecting n-qubit data. In this case, QOTP can be written as the following quantum program:

$$\begin{aligned}
\mathbf{KeyGen}(n) \equiv\ & a_1 := |0\rangle; \cdots; a_n := |0\rangle; b_1 := |0\rangle; \cdots; b_n := |0\rangle; \\
& a_1 := H[a_1]; \cdots; a_n := H[a_n]; b_1 := H[b_1]; \cdots; b_n := H[b_n]; \\
& \mathbf{if}\ (\square x_1 z_1 \cdot \mathcal{M}[a_1, b_1] = x_1 z_1 \rightarrow \mathbf{skip})\ \mathbf{fi}; \\
& \vdots \\
& \mathbf{if}\ (\square x_n z_n \cdot \mathcal{M}[a_n, b_n] = x_n z_n \rightarrow \mathbf{skip})\ \mathbf{fi} \\
\mathbf{Enc}(n) \equiv \mathbf{Dec}(n) \equiv\ & \mathbf{if}\ (\square x_1 z_1 \cdot \mathcal{M}[a_1, b_1] = x_1 z_1 \rightarrow p_1 = Z^{z_1}[p_1];\ p_1 = X^{x_1}[p_1])\ \mathbf{fi}; \\
& \vdots \\
& \mathbf{if}\ (\square x_n z_n \cdot \mathcal{M}[a_n, b_n] = x_n z_n \rightarrow p_n = Z^{z_n}[p_n];\ p_n = X^{x_n}[p_n])\ \mathbf{fi} \\
\mathbf{DisKey}(n) \equiv\ & \mathbf{Tr}[a_1]; \cdots; \mathbf{Tr}[a_n]; \mathbf{Tr}[b_1] \cdots; \mathbf{Tr}[b_n]
\end{aligned}$$

Again, if we regard register $\bar{p} = p_1, \cdots, p_n$ as the input and output of QOTP and consider the trivial program **skip** with the duplicated register $\bar{p}' = p'_1, \cdots, p'_n$, then the judgment

$$\vdash \mathbf{KeyGen}(n); \mathbf{Enc}(n); \mathbf{Dec}(n); \mathbf{DisKey}(n) \sim \mathbf{skip} : (=_{sym}) \Rightarrow (=_{sym}). \tag{23}$$

is derivable using the basic rules of logic rqPD, where $=_{sym}$ is the projector onto the symmetric space between $\bar{p}$ and $\bar{p}'$. Indeed, this judgment implies the correctness of QOTP; that is, the input and output quantum data on register $\bar{p} = p_1, \cdots, p_n$ (might be entangled) are conserved. Similarly, we can also prove that for any $|\psi\rangle \in \mathcal{H}_{\bar{p}}$:

$$\models \mathbf{KeyGen}(n); \mathbf{Enc}(n); \mathbf{DisKey}(n) \sim \mathbf{KeyGen}(n); \mathbf{Enc}(n); \mathbf{DisKey}(n) : \frac{I_{\bar{p}} \otimes I_{\bar{p}'}}{2^n} \Rightarrow |\psi\rangle_{\bar{p}}\langle\psi| \otimes I_{\bar{p}'}. \tag{24}$$

The above judgment actually implies the output after the encryption is the maximally mixed state and it is impossible for the eavesdropper to obtain any information about the quantum data.

7 REASONING ABOUT PROJECTIVE PREDICATES

The logic rqPD was developed for reasoning about the equivalence between quantum programs with respect to general preconditions and postconditions represented by Hermitian operators. But in some applications, it is more convenient to use a simplified version of rqPD with preconditions and postconditions being projective predicates (equivalently, subspaces of the state Hilbert spaces). In this section, we present such a simplified version of rqPD and give an example to show its utility. As one may expect, a price for this simplification is a weaker expressive power of the logic. The coefficients $\frac{1}{d}$ and $\frac{1}{2}$ in the preconditions of judgments (12) and (14) are not expressible in rqPD with projective predicates, indicating that the expressive power of rqPD with projective predicates is *strictly* weaker than that of full rqPD.

7.1 Inference Rules

In this subsection, we develop inference rules for judgments with projective preconditions and postconditions. We consider judgments of the form:

$$P_1 \sim P_2 : A \Rightarrow B \tag{25}$$

where A, B are two projections in (or equivalently, subspaces of) $\mathcal{H}_{P_1\langle 1\rangle} \otimes \mathcal{H}_{P_2\langle 2\rangle}$.

Definition 7.1. *Judgment (25) is projectively valid, written:*

$$\models_P P_1 \sim P_2 : A \Rightarrow B,$$

if for any $\rho_1 \in \mathcal{D}^{\leq}\left(\mathcal{H}_{P_1\langle 1\rangle}\right)$ and $\rho_2 \in \mathcal{D}^{\leq}\left(\mathcal{H}_{P_2\langle 2\rangle}\right)$ such that $\rho_1\ A^{\#}\ \rho_2$, there exists a lifting of B relating the output quantum states: $[\![P_1]\!](\rho_1)\ B^{\#}\ [\![P_2]\!](\rho_2)$.

The following proposition clarifies the relationship between projective validity and the notion of validity introduced in Definition 5.1.

Proposition 7.1. *For any two program P_1 and P_2, and projective predicates A and B:*

(1) $\models P_1 \sim P_2 : A \Rightarrow B \Rightarrow \models_P P_1 \sim P_2 : A \Rightarrow B$;
(2) $\models P_1 \sim P_2 : A \Rightarrow B \not\Leftarrow \models_P P_1 \sim P_2 : A \Rightarrow B$.

To present rules for proving projectively valid judgments, we need the following modifications of Definitions 5.4 and 5.5.

Definition 7.2. *Let $\mathcal{M}_1 = \{M_{1m}\}$ and $\mathcal{M}_2 = \{M_{2m}\}$ be two measurements with the same set $\{m\}$ of possible outcomes in $\mathcal{H}_{P_1}$ and $\mathcal{H}_{P_2}$, and let A and B_m be projective predicates in $\mathcal{H}_{P_1\langle 1\rangle} \otimes \mathcal{H}_{P_2\langle 2\rangle}$. Then the assertion*

$$\models_P \mathcal{M}_1 \approx \mathcal{M}_2 : A \Rightarrow \{B_m\} \tag{26}$$

holds if for any $\rho_1 \in \mathcal{D}^{\leq}\left(\mathcal{H}_{P_1\langle 1\rangle}\right)$ and $\rho_2 \in \mathcal{D}^{\leq}\left(\mathcal{H}_{P_2\langle 2\rangle}\right)$ such that $\rho_1\ A^{\#}\ \rho_2$, there exists a sequence of lifting of B_m relating the post-measurement states with the same outcomes: for all m,

$$(M_{1m}\rho_1 M_{1m}^{\dagger})\ B_m^{\#}\ (M_{2m}\rho_2 M_{2m}^{\dagger}).$$

Definition 7.3. *Let $\mathcal{M}_1 = \{M_{1m}\}$ be a measurements in $\mathcal{H}_{P_1}$, and let A and B_m be projective predicates in $\mathcal{H}_{P_1\langle 1\rangle} \otimes \mathcal{H}_{P_2\langle 2\rangle}$. Then the assertion*

$$\models_P \mathcal{M}_1 \approx I_2 : A \Rightarrow \{B_m\} \tag{27}$$

holds if for any $\rho_1 \in \mathcal{D}^{\leq}\left(\mathcal{H}_{P_1\langle 1\rangle}\right)$ and $\rho_2 \in \mathcal{D}^{\leq}\left(\mathcal{H}_{P_2\langle 2\rangle}\right)$ such that $\rho_1\ A^{\#}\ \rho_2$, there exist ρ_{2m} such that $\sum_m \rho_{2m} = \rho_2$ and for all m,

$$(M_{1m}\rho_1 M_{1m}^{\dagger})\ B_m^{\#}\ \rho_{2m}.$$

Now the proof system for judgments with projective preconditions and postconditions consists of rules (Skip), (UT), (SC), (UT-L/R), (Conseq), (Equiv) and (Frame) in Figs. 4, 6 and 7 with $\vdash$, $\models$ being replaced by $\vdash_P$ and $\models_P$, respectively, together with the rules given in Fig. 10.

Let us carefully compare this simplified proof system for projective predicates with the original rqPD for general predicates of Hermitian operators:

- In rules (Init-P), (Init-P-L), (Init-P-R), (SO-P), (SO-P-L) and (SO-P-R), we have to use the operation proj$(\cdot)$ because the operators in its operand there are not necessarily projective.
- Rule (Case) has no counterpart for projective predicates because probabilistic combination $\sum_i p_i A_i$ of a family of projective predicates A_i is usually not projective.

$$\text{(Init-P)}\ \vdash_P q_1 := |0\rangle \sim q_2 := |0\rangle : A \Rightarrow |0\rangle_{q_1\langle 1\rangle}\langle 0| \otimes |0\rangle_{q_2\langle 2\rangle}\langle 0| \otimes \mathrm{proj}(tr_{\mathcal{H}_{q_1\langle 1\rangle}\otimes\mathcal{H}_{q_2\langle 2\rangle}}(A))$$

$$\text{(Init-P-L)}\ \vdash_P q_1 := |0\rangle \sim \mathbf{skip} : A \Rightarrow |0\rangle_{q_1\langle 1\rangle}\langle 0| \otimes \mathrm{proj}(tr_{\mathcal{H}_{q_1\langle 1\rangle}}(A))$$

$$\text{(IF-P)}\ \frac{\models_P \mathcal{M}_1 \approx \mathcal{M}_2 : A \Rightarrow \{B_m\} \quad \vdash_P P_{1m} \sim P_{2m} : B_m \Rightarrow C \text{ for every } m}{\vdash_P \mathbf{if}\ (\square m \cdot \mathcal{M}_1[\overline{q}] = m \rightarrow P_{1m})\ \mathbf{fi} \sim \mathbf{if}\ (\square m \cdot \mathcal{M}_2[\overline{q}] = m \rightarrow P_{2m})\ \mathbf{fi} : A \Rightarrow C}$$

$$\text{(IF-P-L)}\ \frac{\models_P \mathcal{M}_1 \approx I_2 : A \Rightarrow \{B_m\} \quad \vdash_P P_{1m} \sim P : B_m \Rightarrow C \text{ for every } m}{\vdash_P \mathbf{if}\ (\square m \cdot \mathcal{M}_1[\overline{q}] = m \rightarrow P_{1m})\ \mathbf{fi} \sim P : A \Rightarrow C}$$

$$\text{(LP-P)}\ \frac{\models_P \mathcal{M}_1 \approx \mathcal{M}_2 : A \Rightarrow \{B_0, B_1\} \quad \vdash_P P_1 \sim P_2 : B_1 \Rightarrow A}{\vdash_P \mathbf{while}\ \mathcal{M}_1[\overline{q}] = 1\ \mathbf{do}\ P_1\ \mathbf{od} \sim \mathbf{while}\ \mathcal{M}_2[\overline{q}] = 1\ \mathbf{do}\ P_2\ \mathbf{od} : A \Rightarrow B_0}$$

$$\text{(LP-P-L)}\ \frac{\models \mathbf{while}\ \mathcal{M}_1[\overline{q}] = 1\ \mathbf{do}\ P_1\ \mathbf{od} \text{ lossless} \quad \models_P \mathcal{M}_1 \approx I_2 : A \Rightarrow \{B_0, B_1\} \quad \vdash_P P_1 \sim \mathbf{skip} : B_1 \Rightarrow A}{\vdash_P \mathbf{while}\ \mathcal{M}_1[\overline{q}] = 1\ \mathbf{do}\ P_1\ \mathbf{od} \sim \mathbf{skip} : A \Rightarrow B_0}$$

$$\text{(SO-P)}\ \vdash_P \overline{q_1} := \mathcal{E}_1\,[\overline{q_1}] \sim \overline{q_2} := \mathcal{E}_2\,[\overline{q_2}] : A \Rightarrow \mathrm{proj}((\mathcal{E}_1 \otimes \mathcal{E}_2)\,(A))$$

$$\text{(SO-P-L)}\ \vdash_P \overline{q_1} := \mathcal{E}_1\,[\overline{q_1}] \sim \mathbf{skip} : A \Rightarrow \mathrm{proj}(\mathcal{E}_1(A))$$

Fig. 10. Rules for Projective Predicates. For any positive operator A on Hilbert space $\mathcal{H}$, we write $\mathrm{proj}(A)$ for the projection onto $\mathrm{supp}(A)$, the subspace spanned by the eigenvectors of A with nonzero eigenvalues. The quantum operations appeared in (SO-P) and (SO-P-L) are all trace-preserving. We omit the right-side counterparts of (Init-P-L), (IF-P-L), (LP-P-L) and (SO-P-L).

- The main simplification occurs in the rules for control-flow constructs (i.e. conditionals and loops). We only consider rule (IF-P); the same explanation applies to other control-flow rules. First, the measurement condition $\models_P \mathcal{M}_1 \approx \mathcal{M}_2 : A \Rightarrow \{B_m\}$ in the premise of (IF-P) is weaker than the measurement condition $\mathcal{M}_1 \approx \mathcal{M}_2 \models A \Rightarrow \{B_m\}$ in the premise of (IF). Second, the measurement condition $\mathcal{M}_1 \approx \mathcal{M}_2$ is the conclusion of (IF) is removed in (IF-P). As already pointed out in the Introduction, this is biggest reward of the projective simplification of our logic.

Proposition 7.2. *The proof system for judgments with projective preconditions and postconditions are sound.*

As will be seen in the next subsection, this simplified proof system, in particular, the simplified rules for control-flow constructs, whenever they are applicable, can significantly ease the verification of relational properties of quantum programs. On the other hand, some relational properties of quantum programs, e.g. judgments (12), (14) and (18-20), can be verified by the original rqPD but not by this simplified system. Even for the same quantum programs, the original rqPD usually can prove stronger relational properties in the case where the weakest preconditions or strongest postconditions are not projective.

7.2 Example: Quantum Walks

In this subsection, we present an example to show the effectiveness of the inference rules given in the previous subsection. Quantum (random) walks [Kempe 2003; Venegas-Andraca 2012] are quantum analogues of random walks, and have been widely used in the design of quantum algorithms, including quantum search and quantum simulation. There are two key ideas in defining a quantum walk that are fundamentally different from that of a classical random walk: (1) a "quantum coin" is introduced to govern the movement of the walker, which allows the walker to move to two different directions, say left and right, simultaneously; (2) an absorbing boundary is realised by a

quantum measurement. Here, we show how our logic can be applied to verify a certain equivalence of two one-dimensional quantum walks with absorbing boundaries: the quantum coins used in these two quantum walks are different, but they terminate at the same position.

Example 7.1. *Let $\mathcal{H}_c$ be the coin space, the 2-dimensional Hilbert space with orthonormal basis state $|L\rangle$ (or $|0\rangle_c$) and $|R\rangle$ (or $|1\rangle_c$), indicating directions Left and Right, respectively. Let $\mathcal{H}_p$ be the $(n+1)$-dimensional Hilbert space with orthonormal basis states $|0\rangle, |1\rangle, ..., |n-1\rangle, |n\rangle$, where vector $|i\rangle$ denotes position i for each $0 \leq i \leq n$; in particular, positions 0 and n are the absorbing boundaries. The state space of the walk is then $\mathcal{H} = \mathcal{H}_c \otimes \mathcal{H}_p$. Each step of the walk consists of:*

1. *Measure the position of the system to see whether it is 0 or n. If the outcome is "yes", then the walk terminates; otherwise, it continues. The measurement can be described as $\mathcal{M} = \{M_{yes}, M_{no}\}$, where the measurement operators are: $M_{yes} = |0\rangle\langle 0| + |n\rangle\langle n|$, $M_{no} = I_p - M_{yes} = \sum_{i=1}^{n-1} |i\rangle\langle i|$, and I_p is the identity in position space $\mathcal{H}_p$;*
2. *Apply a "coin-tossing" operator C is in the coin space $\mathcal{H}_c$.*
3. *Apply a shift operator $S = \sum_{i=1}^{n-1} |L\rangle\langle L| \otimes |i-1\rangle\langle i| + \sum_{i=1}^{n-1} |R\rangle\langle R| \otimes |i+1\rangle\langle i|$ in the space $\mathcal{H}$. Intuitively, operator S moves the position one step to the left or to the right according to the direction state.*

A major difference between a quantum walk and a classical random walk is that a superposition of movement to the left and a movement to the right can happen in the quantum case. The quantum walk can be written as a quantum program with the initial state in $\mathcal{H}_c \otimes \mathcal{H}_p$ as the input:

$$\textbf{while } \mathcal{M}[p] = no \textbf{ do } c := C[c];\ c, p := S[c, p] \textbf{ od} \tag{28}$$

We consider two frequently used "coin-tossing" operators here: the Hadamard operator H and the balanced operator: $Y = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & \boldsymbol{i} \\ \boldsymbol{i} & 1 \end{pmatrix}$. We use $\textbf{while}(H)$ and $\textbf{while}(Y)$ to denote program (28) with $C = H$ or Y, respectively. What interests us is: with what kind of initial states do the quantum walks with different "coin-tossing" operators H and Y produce the same output position? To this end, we add a measurement to determine the exact position of the walks after their termination, and discard the coin. So, programs $\textbf{while}(H)$ and $\textbf{while}(Y)$ are modified to:

$$QW(H) \equiv \textbf{while}(H);\ \textbf{if } (\square i \cdot \mathcal{M}'[p] = i \rightarrow \textbf{skip}) \textbf{ fi};\ \textbf{Tr}[c],$$
$$QW(Y) \equiv \textbf{while}(Y);\ \textbf{if } (\square i \cdot \mathcal{M}'[p] = i \rightarrow \textbf{skip}) \textbf{ fi};\ \textbf{Tr}[c]$$

where measurement $\mathcal{M}' = \{M_i'\}$ with $M_i' = |i\rangle\langle i|$ for $i = 0, 1, \cdots, n$.

Before formulating our result in our logic, let us fix the notations. Whenever comparing programs $QW(H)$ and $QW(Y)$ and using, say x, to denote a variable in the former, then we shall use x' for the same variable in the latter. For simplicity, we use $|d, i\rangle_{c,p}$ as an abbreviation of $|d\rangle_c|i\rangle_p$, $I_{c,p}$ is the identity over the whole space $\mathcal{H}_c \otimes \mathcal{H}_p$, and $S_{c,p;c',p'}$ is the SWAP operator between two systems $\mathcal{H}_c \otimes \mathcal{H}_p$ and $\mathcal{H}_{c'} \otimes \mathcal{H}_{p'}$. Furthermore, we introduce the following unitary operator U and projective predicates $(=_{sym})$ and $(=^p_{sym})$:

$$U : |d, i\rangle_{c,p} \mapsto (-1)^{\frac{i+d+3}{2}} |d, i\rangle_{c,p} \qquad (=_{sym}) = \frac{1}{2}(I_{c,p} \otimes I_{c',p'} + S_{c,p;c',p'})$$
$$(=^p_{sym}) = \frac{1}{2}\Big(\sum_{i,i'=0,n} |i\rangle_p\langle i| \otimes |i'\rangle_{p'}\langle i'| + \sum_{i,i'=0,n} |i\rangle_p\langle i'| \otimes |i\rangle_{p'}\langle i'| \Big),$$

In [Barthe et al. 2019], we show how to derive the following judgment in the projective version of rqPD:

$$\models_P QW(H) \sim QW(Y) : U_{c',p'}(=_{sym})U^\dagger_{c',p'} \Rightarrow (=^p_{sym}). \tag{29}$$

This judgment means that if walks $QW(H)$ and $QW(Y)$ start from states ρ_1 and $\rho_2 = U\rho_1 U^\dagger$, respectively, then they terminate at exactly the same position.

8 RELATED WORK

The formal verification of quantum programs is an active area of research, and many expressive formalisms have been proposed in the literature [Chadha et al. 2006; D'Hondt and Panangaden 2006; Feng et al. 2007; Kakutani 2009; Ying 2011, 2016]. However, previous work largely considers single program executions. Security of quantum one-time pad (our Example 6.5) was verified in [Unruh 2019a] using a variant of quantum Hoare logic rather than relational logic. Other formalisms explicitly target equivalence of quantum programs [Ardeshir-Larijani et al. 2013; Feng and Ying 2015; Kubota et al. 2013]. However, these works are based on bisimulations and symbolic methods, which have a more limited scope and are less powerful than general relational program logics. Finally, some works develop specialized methods for proving concrete properties of quantum programs; for instance, Hung et al [Hung et al. 2018] reason about quantitative robustness of quantum programs. It would be interesting to cast the latter into our more general framework. This seems possible although may not be straightforward; indeed, in Subsection 6.3.2, we showed that our logic can be used to reason about the reliability of quantum teleportation against several kinds of quantum noise.

Our work is most closely related to the quantum relational Hoare logic recently proposed by Li and Unruh [2019]; Unruh [2019b]. Both works are inspired by probabilistic relational Hoare logic [Barthe et al. 2009] and share the long-term objective of providing a convenient framework for formal verification of quantum cryptography. However, the two works explore different points in the design space of relational logics for quantum programs. There are several fundamental differences between our logic and Unruh's one, including expressiveness, entanglement in defining the validity of judgments and inference rules. A careful comparison of them is given in [Barthe et al. 2019].

9 CONCLUSION

We have introduced a relational program logic for a core quantum programming language; our logic is based on a quantum analogue of probabilistic couplings, and is able to verify several non-trivial examples of relational properties for quantum programs.

There are several promising directions for future work. First, we would like to further develop the theory of quantum couplings, and in particular to define a quantum version of approximate couplings. An extension apRHL of probabilistic relational Hoare logic pRHL was defined in [Barthe et al. 2013] for verification of differential privacy. A surprising connection between quantum differential privacy and gentle measurements recently observed by Aaronson and Rothblum [2019] presents a further possible application of a quantum counterpart of apRHL in quantum physics. Second, we would like to explore variants and applications of our logic to other areas, including the convergence of quantum Markov chains, quantum cryptography, and translation validation of quantum programs; in particular, the correctness of optimising quantum compilers for NISQ (Noisy Intermediate Quantum) devices.

ACKNOWLEDGMENTS

This work is partially supported by the University of Wisconsin, a Facebook TAV award, 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). We are grateful to the Max Planck Institute for Software Systems for hosting some of the authors.

REFERENCES

Scott Aaronson and Guy N. Rothblum. 2019. Gentle Measurement of Quantum States and Differential Privacy. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing (STOC 2019)*. ACM, New York, NY, USA, 322–333. https://doi.org/10.1145/3313276.3316378

Samson Abramsky and Bob Coecke. 2004. A Categorical Semantics of Quantum Protocols. In *19th IEEE Symposium on Logic in Computer Science (LICS 2004), 14-17 July 2004, Turku, Finland, Proceedings*. 415–425. https://doi.org/10.1109/LICS.2004.1319636

V.S. Anil Kumar and H. Ramesh. 2001. Coupling vs. conductance for the Jerrum–Sinclair chain. *Random Structures & Algorithms* 18, 1 (2001), 1–17. https://doi.org/10.1002/1098-2418(200101)18:1<1::AID-RSA1>3.0.CO;2-7

Ebrahim Ardeshir-Larijani, Simon J. Gay, and Rajagopal Nagarajan. 2013. Equivalence Checking of Quantum Protocols. In *Tools and Algorithms for the Construction and Analysis of Systems - 19th International Conference, TACAS 2013, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2013, Rome, Italy, March 16-24, 2013. Proceedings (Lecture Notes in Computer Science)*, Nir Piterman and Scott A. Smolka (Eds.), Vol. 7795. Springer, 478–492. https://doi.org/10.1007/978-3-642-36742-7_33

Gilles Barthe, Thomas Espitau, Benjamin Grégoire, Justin Hsu, Léo Stefanesco, and Pierre-Yves Strub. 2015. Relational Reasoning via Probabilistic Coupling. In *Logic for Programming, Artificial Intelligence, and Reasoning - 20th International Conference, LPAR-20 2015, Suva, Fiji, November 24-28, 2015, Proceedings (Lecture Notes in Computer Science)*, Martin Davis, Ansgar Fehnker, Annabelle McIver, and Andrei Voronkov (Eds.), Vol. 9450. Springer, 387–401. https://doi.org/10.1007/978-3-662-48899-7_27

Gilles Barthe, Thomas Espitau, Benjamin Grégoire, Justin Hsu, and Pierre-Yves Strub. 2017. Proving uniformity and independence by self-composition and coupling. In *LPAR-21, 21st International Conference on Logic for Programming, Artificial Intelligence and Reasoning, Maun, Botswana, 7-12th May 2017 (EPiC Series)*, Thomas Eiter and David Sands (Eds.), Vol. 46. EasyChair, 385–403. http://www.easychair.org/publications/paper/340344

Gilles Barthe, Thomas Espitau, Benjamin Grégoire, Justin Hsu, and Pierre-Yves Strub. 2018. Proving expected sensitivity of probabilistic programs. *PACMPL* 2, POPL (2018), 57:1–57:29. https://doi.org/10.1145/3158145

Gilles Barthe, Marco Gaboardi, Benjamin Grégoire, Justin Hsu, and Pierre-Yves Strub. 2016. Proving Differential Privacy via Probabilistic Couplings. In *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '16, New York, NY, USA, July 5-8, 2016*, Martin Grohe, Eric Koskinen, and Natarajan Shankar (Eds.). ACM, 749–758. https://doi.org/10.1145/2933575.2934554

Gilles Barthe, Benjamin Grégoire, and Santiago Zanella Béguelin. 2009. Formal certification of code-based cryptographic proofs. In *Proceedings of the 36th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2009, Savannah, GA, USA, January 21-23, 2009*, Zhong Shao and Benjamin C. Pierce (Eds.). ACM, 90–101. https://doi.org/10.1145/1480881.1480894

Gilles Barthe, Justin Hsu, Mingsheng Ying, Nengkun Yu, and Li Zhou. 2019. Relational Proofs for Quantum Programs (Extended Version). *CoRR* abs/1901.05184 (2019). arXiv:1901.05184 http://arxiv.org/abs/1901.05184

Gilles Barthe, Boris Köpf, Federico Olmedo, and Santiago Zanella Béguelin. 2012. Probabilistic relational reasoning for differential privacy. In *Proceedings of the 39th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2012, Philadelphia, Pennsylvania, USA, January 22-28, 2012*, John Field and Michael Hicks (Eds.). ACM, 97–110. https://doi.org/10.1145/2103656.2103670

Gilles Barthe, Boris Köpf, Federico Olmedo, and Santiago Zanella-Béguelin. 2013. Probabilistic Relational Reasoning for Differential Privacy. *ACM Trans. Program. Lang. Syst.* 35, 3, Article 9 (nov 2013), 49 pages. https://doi.org/10.1145/2492061

Charles H. Bennett, Gilles Brassard, Claude Crépeau, Richard Jozsa, Asher Peres, and William K. Wootters. 1993. Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels. *Phys. Rev. Lett.* 70 (Mar 1993), 1895–1899. Issue 13. https://doi.org/10.1103/PhysRevLett.70.1895

P. Oscar Boykin and Vwani Roychowdhury. 2003. Optimal encryption of quantum bits. *Phys. Rev. A* 67 (Apr 2003), 042317. Issue 4. https://doi.org/10.1103/PhysRevA.67.042317

Rohit Chadha, Paulo Mateus, and Amílcar Sernadas. 2006. Reasoning About Imperative Quantum Programs. *Electr. Notes Theor. Comput. Sci.* 158 (2006), 19–39. https://doi.org/10.1016/j.entcs.2006.04.003

Howard Dale, David Jennings, and Terry Rudolph. 2015. Provable quantum advantage in randomness processing. *Nature communications* 6 (2015), 8203.

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

Yuan Feng, Runyao Duan, Zheng-Feng Ji, and Mingsheng Ying. 2007. Proof rules for the correctness of quantum programs. *Theor. Comput. Sci.* 386, 1-2 (2007), 151–166. https://doi.org/10.1016/j.tcs.2007.06.011

Yuan Feng and Mingsheng Ying. 2015. Toward Automatic Verification of Quantum Cryptographic Protocols. In *26th International Conference on Concurrency Theory, CONCUR 2015, Madrid, Spain, September 1.4, 2015 (LIPIcs)*, Luca Aceto and David de Frutos-Escrig (Eds.), Vol. 42. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 441–455. https:

//doi.org/10.4230/LIPIcs.CONCUR.2015.441

Justin Hsu. 2017. Probabilistic Couplings for Probabilistic Reasoning. *CoRR* abs/1710.09951 (2017). arXiv:1710.09951 http://arxiv.org/abs/1710.09951

Shih-Han Hung, Kesha Hietala, Shaopeng Zhu, Mingsheng Ying, Michael Hicks, and Xiaodi Wu. 2018. Quantitative Robustness Analysis of Quantum Programs (Extended Version). *CoRR* abs/1811.03585 (2018). arXiv:1811.03585 http://arxiv.org/abs/1811.03585 To appear at POPL'19.

Yoshihiko Kakutani. 2009. A Logic for Formal Verification of Quantum Programs. In *Advances in Computer Science - ASIAN 2009. Information Security and Privacy, 13th Asian Computing Science Conference, Seoul, Korea, December 14-16, 2009. Proceedings (Lecture Notes in Computer Science)*, Anupam Datta (Ed.), Vol. 5913. Springer, 79–93. https://doi.org/10.1007/978-3-642-10622-4_7

MS Keane and George L O'Brien. 1994. A Bernoulli factory. *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 4, 2 (1994), 213–219.

Julia Kempe. 2003. Quantum random walks: an introductory overview. *Contemporary Physics* 44, 4 (2003), 307–327.

Takahiro Kubota, Yoshihiko Kakutani, Go Kato, Yasuhito Kawano, and Hideki Sakurada. 2013. Automated Verification of Equivalence on Quantum Cryptographic Protocols. In *5th International Symposium on Symbolic Computation in Software Science, SCSS 2013, Castle of Hagenberg, Austria (EPiC Series in Computing)*, Laura Kovács and Temur Kutsia (Eds.), Vol. 15. EasyChair, 64–69. http://www.easychair.org/publications/paper/143661

Burkhard Kümmerer and Kay Schwieger. 2016. Diagonal couplings of quantum Markov chains. *Infinite Dimensional Analysis, Quantum Probability and Related Topics* 19, 2 (2016), 1650012.

Yangjia Li and Dominique Unruh. 2019. Quantum Relational Hoare Logic with Expectations. *CoRR* abs/1903.08357 (2019). arXiv:1903.08357 http://arxiv.org/abs/1903.08357

Torgny Lindvall. 2002. *Lectures on the coupling method.* Courier Corporation.

Michele Mosca, Alain Tapp, and Ronald de Wolf. 2000. Private quantum channels and the cost of randomizing quantum information. *arXiv preprint quant-ph/0003101* (2000). https://arxiv.org/abs/quant-ph/0003101

Michael A Nielsen and Isaac Chuang. 2002. *Quantum computation and quantum information.* Cambridge University Press.

Peter Selinger. 2004a. A Brief Survey of Quantum Programming Languages. In *Functional and Logic Programming, 7th International Symposium, FLOPS 2004, Nara, Japan, April 7-9, 2004, Proceedings (Lecture Notes in Computer Science)*, Yukiyoshi Kameyama and Peter J. Stuckey (Eds.), Vol. 2998. Springer, 1–6. https://doi.org/10.1007/978-3-540-24754-8_1

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

Volker Strassen. 1965. The existence of probability measures with given marginals. *The Annals of Mathematical Statistics* (1965), 423–439. http://projecteuclid.org/euclid.aoms/1177700153

Hermann Thorisson. 2000. *Coupling, Stationarity, and Regeneration.* springer.

Dominique Unruh. 2019a. Quantum Hoare Logic with Ghost Variables. In *2019 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*. 1–13. https://doi.org/10.1109/LICS.2019.8785779

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

Salvador Elías Venegas-Andraca. 2012. Quantum walks: a comprehensive review. *Quantum Information Processing* 11, 5 (2012), 1015–1106.

Cédric Villani. 2008. *Optimal transport: Old and new.* springer.

Andreas Winter. 2016. Tight uniform continuity bounds for quantum entropies: conditional entropy, relative entropy distance and energy constraints. *Communications in Mathematical Physics* 347, 1 (2016), 291–313.

Mingsheng Ying. 2011. Floyd-Hoare logic for quantum programs. *ACM Trans. Program. Lang. Syst.* 33, 6 (2011), 19:1–19:49. https://doi.org/10.1145/2049706.2049708

Mingsheng Ying. 2016. *Foundations of Quantum Programming.* Morgan-Kaufmann.

Mingsheng Ying, Nengkun Yu, Yuan Feng, and Runyao Duan. 2013. Verification of quantum programs. *Sci. Comput. Program.* 78, 9 (2013), 1679–1700. https://doi.org/10.1016/j.scico.2013.03.016

Li Zhou, Shenggang Ying, Nengkun Yu, and Mingsheng Ying. 2019a. Strassen's theorem for quantum couplings. *Theoretical Computer Science* (2019). https://doi.org/10.1016/j.tcs.2019.08.026

Li Zhou, Nengkun Yu, and Mingsheng Ying. 2019b. An Applied Quantum Hoare Logic. In *Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI 2019)*. ACM, New York, NY, USA, 1149–1162. https://doi.org/10.1145/3314221.3314584