A Quantum Interpretation of Bunched Logic & Quantum Separation Logic

Li Zhou*, Gilles Barthe*†, Justin Hsu‡, Mingsheng Ying§¶ Nengkun Yu§
*Max Planck Institute for Security and Privacy, †IMDEA Software Institute, ‡University of Wisconsin–Madison,
§University of Technology Sydney, ¶Institute of Software, Chinese Academy of Sciences and Tsinghua University

***Abstract*—We propose a model of the substructural logic of Bunched Implications (BI) that is suitable for reasoning about quantum states. In our model, the separating conjunction of BI describes separable quantum states. We develop a program logic where pre- and post-conditions are BI formulas describing quantum states—the program logic can be seen as a counterpart of separation logic for imperative quantum programs. We exercise the logic for proving the security of quantum one-time pad and secret sharing, and we show how the program logic can be used to discover a flaw in Google Cirq's tutorial on the Variational Quantum Algorithm (VQA).**

I. Introduction

The logic of Bunched Implications (BI) of O'Hearn and Pym [1]–[3] is a substructural logic that features resource-aware connectives. One such connective is $*$, known as separating conjunction: informally, an assertion $\phi * \psi$ holds with respect to a resource R if the resource R can be split into resources R' and R'' such that ϕ holds with respect to R' and ψ holds with respect to R''. This interpretation is particularly well suited for reasoning about programs in settings where computations can have interfering effects. In particular, BI has found success as an assertion language for Separation Logic [4]–[6], a program logic for reasoning about programs with mutable state, and Concurrent Separation Logic [7], [8], a program logic for reasoning about shared-memory concurrent processes.

Recent works seek to extend the separation logic framework beyond memory-manipulating programs by considering other notions of resources and other models of computation. Broadly speaking, separation logics are a good fit whenever programs manipulate resources in a *local* fashion: that is, there is a natural notion of two resources being *separate*, and a program can operate on the first resource without affecting the second. This idea underlies recent separation logics for probabilistic programs, where separation is probabilistic independence [9].

Quantum computation is another domain where the ideas of separation logic seem relevant. Recent work [10], [11] suggests that reasoning about resources (in particular, entanglement – a resource unique in the quantum world) can bring similar benefits to quantum computing and communications. Motivated by this broad perspective, we propose a quantum model of BI and develop a novel separation logic for quantum programs. Our development is guided by concrete examples of quantum algorithms and security protocols.

978-1-6654-4895-6/21/$31.00 ©2021 IEEE

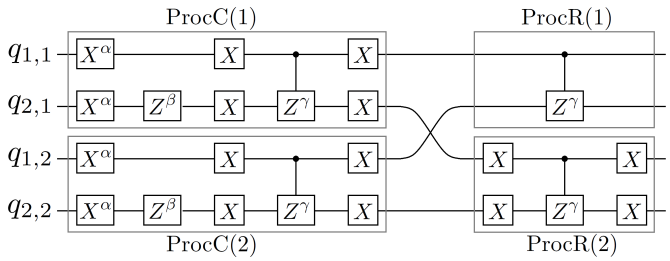

Fig. 1: VQA(2) with parameters taken in Sec. V-C.

Motivating Local Reasoning for Quantum Programs: Quantum Machine Learning [12], [13] and VQAs (Variational Quantum Algorithms) [14], [15] are new classes of quantum algorithms that emerged in recent years as a leading application of quantum computing. These algorithms solve problems by training parameterized quantum circuits. The trained circuits are usually very large in terms of both their size and the required quantum resources, i.e., the number of involved quantum bits (qubits). This makes them particularly challenging to verify with existing techniques such as quantum Hoare logic [16], [17] and verification based on a density matrix representation of quantum states [18]–[20], since the dimension of the matrices used to represent assertions increases exponentially w.r.t. the number of qubits. Fortunately, these algorithms can benefit from local reasoning, since each of their operations is performed locally on a small number of qubits. Consider for instance the quantum circuit shown in Figure 1, that implements a VQA circuit with 2×2 grid qubits. Instead of reasoning about the circuit as a whole, we would like to reason about sub-circuits $\mathrm{ProcC}(1), \mathrm{ProcC}(2), \mathrm{Proc(R)}(1), \mathrm{ProcR}(2)$ separately, and then combine the results to establish the correctness of the whole program. This is precisely the kind of reasoning enabled by Quantum Separation Logic (QSL for short).

Technical Challenges and Contributions: QSL will be developed by first developing a model of BI, where formulas describe quantum states and then building a separation logic using these assertions as pre- and post-conditions, introducing proof rules to reason about quantum programs.

BI and Its Quantum Interpretation: To characterize the properties of quantum systems, we first identify a quantum interpretation of BI appropriate for our target applications. We choose to interpret our separating conjunction $*$ as separability

of quantum states. Roughly speaking, $\phi_1 * \phi_2$ holds in a quantum state ρ if ρ can be factored into two quantum states ρ_1 and ρ_2 over disjoint registers satisfying ϕ_1 and ϕ_2, respectively.

Proof System for Program Logic: next, we define a program logic for a quantum **while**-language [16] (for simplicity, we do not consider classical variables). Our language follows the "classical control and quantum data" paradigm. We develop a set of proof rules that are effective for verifying quantum programs over a large set of qubits. Our proof system has several novel ingredients:

1) **Modification** on BI formulas. The basic rule for assignments in classical program logics is defined using the syntactic notion of substitution. Due to the non-cloning law of quantum information, the role of assignments has to be played by initialization $q := |0\rangle$ and unitary transformations $\overline{q} := U[\overline{q}]$, and inference rules for these operations involve a quantum operation (e.g., [16], [17]). Unfortunately, the rules for initialization and unitary transformations are not simple adaptations of the rule for assignment, because a quantum generalization of substitution is not straightforward. For atomic predicates, substitutions are not always defined. For composite formulas, a straightforward definition of substitution is too weak for applications. We overcome this hurdle by introducing a *modification* operation for atomic formulas (see Definition 9), which is essentially a quantum version of substitution. Extending this operation to composite formulas requires some care (see Definition 10).
2) **Frame rule**: The frame rule is one of the most characteristic structural rules in separation logic. QSL also enjoys a frame rule FRAME that is similar in spirit to frame rules from standard separation logics, but our new interpretation of separating conjunction means that the meaning of the rule is different. Furthermore, the frame rule can be generalized slightly: even if the standard side condition for frame rules does not hold, the frame rule still applies if the post-condition is a *supported* assertion—a concept first proposed by Reynolds [21] in the context of standard separation logic. This extra bit of freedom seems to be particular to the quantum setting, and we crucially use this feature when using the frame rule to establish uniformity. The soundness proof of our quantum frame rule requires a nontrivial calculation based on *purification*, a fundamental technique used in quantum information for transforming mixed states to pure states by introducing reference systems [22].
3) Reasoning about **entangled predicates**:[1] The structural rules FRAME and CONST enable us to lift local reasoning to global correctness of quantum algorithms *only* when no entanglement occur in the pre- and post-conditions. However, entangled predicates play an essential role in revealing the non-local (global) properties of a composite quantum system; for instance, some entangled predicates are used when reasoning about the (in)correctness of VQA (see Sec. V-C). With the help of auxiliary variables, we set up a new rule UNCR which enables us to prove the correctness of large quantum algorithms with respect to entangled pre- and post-conditions. Intuitively, when the program (as the principal system) combined with auxiliary variables (as ancillary systems), modification can be used to create (mathematically rather than physically) entanglement and rule UNCR is used to preserve correctness under the modification on the auxiliary variables in the pre- and post-conditions (but not in the program). The key idea behind was first proposed in [23] for reasoning about parallel quantum programs; UNCR is its generalization tailored for our purpose.

[1]Here, entangled predicates refer to the projections that cannot be factored as a product of projections of its local constituents.

QSL: Proving global correctness via local reasoning. As motivated, our logic is designed for scalable verification of large-scale quantum programs: once genuinely quantum properties of small subprograms are verified, QSL can quickly and efficiently incorporate them into larger quantum programs. While existing tools [18], [20], [24], [25] are suitable for reasoning about relatively small quantum algorithms with good algebraic structure, QSL provides a way to verify larger quantum programs and protocols containing them.

Applications: To demonstrate the breadth of the application range of our logic QSL, we present several case studies from two very different areas:

- Our first example given in Section V is formal verification of Variational Quantum Algorithm (VQA) [14], [15] for finding the ground state energy of a quantum system, which has potential applications in quantum chemistry for designing new materials and drugs. A typical VQA can be split into different subprograms that are suited to local reasoning. Then the frame rules together with UNCR are used to derive global correctness with entangled pre- and post-conditions. *In particular, an analysis based on QSL reveals that the VQA presented in the tutorial of Google's Cirq [26] is incorrect.*
- In Section VI, we use QSL to verify the security of quantum one-time pad (QOTP) [27], [28] and quantum secret sharing (QSS) [29], [30]. Unlike previous work, the QSL verification of QOTP and QSS is *scalable*: increasing the number of registers that algorithms employ does not complicate the verification. In particular, rule FRAME with the supported assertion (SP) enables us to avoid the very complicated mathematical calculations used in earlier verifications of QOTP [31].

II. PRELIMINARIES

For the convenience of the reader, we briefly review basic notions of quantum information and programming as well as the logic of bunched implication.

A. Basics of Quantum Information

The *state space* of a quantum system is a Hilbert space $\mathcal{H}$, which is essentially a vector space with inner product in the

finite-dimensional case. A *pure state* of the system is a unit column vector $|\psi\rangle \in \mathcal{H}$. For example, the state space of a quantum bit (aka qubit) is a two-dimensional Hilbert space with basis states $|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, and any pure state of a qubit can be described in the form $\alpha|0\rangle + \beta|1\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$ satisfying normalization condition $|\alpha|^2 + |\beta|^2 = 1$. When the state is not completely known but could be in one of some pure states $|\psi_i\rangle$ with respective probabilities p_i, we call $\{(p_i, |\psi_i\rangle)\}$ an *ensemble* of pure states or a *mixed state*, and the system is fully described by the *density operator* $\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i|$. For example, the completely mixed state of a qubit can be seen as ensemble $\{(0.5, |0\rangle), (0.5, |1\rangle)\}$ (i.e. the state is either $|0\rangle$ or $|1\rangle$ with the same probability 0.5) or density matrix $\frac{1}{2}(|0\rangle\langle 0| + |1\rangle\langle 1|) = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}$.

The evolution of a quantum system is modelled by a *unitary operator* U; i.e. a complex matrix with $UU^\dagger = U^\dagger U = I$ where I is the identity operator and $\dagger$ stands for conjugate transpose. In quantum computing, operators are often called *quantum gates*. For example, the Hadamard gate $H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$ maps $|0\rangle, |1\rangle$ to their superpositions $|\pm\rangle = \frac{1}{\sqrt{2}}(|0\rangle \pm |1\rangle)$.

Unlike a classical system which can be observed directly without changing its state, we need to perform a quantum measurement to extract information from a quantum state which inevitably leads to state collapse. Formally, a *projective quantum measurement* consists of a set of *projections*, i.e., self-adjoint and idempotent linear operators,[2] $M_0, M_1, \ldots, M_n$. When such a measurement is applied to a quantum state ρ, we obtain one of the classical outcome $i \in \{0, 1, \ldots, n\}$ with probability $p_i = \mathrm{tr}(M_i \rho)$, and the post-measurement state of the system is then $\frac{M_i \rho M_i}{p_i}$.

We use variables $p, q, r, \ldots$ to denote quantum systems. Operations in quantum computing are often performed on a composite system consisting of multiple qubits. To indicate which system a state describes or an operation acts on, we use subscripts; for example, $\mathcal{H}_p$ is the state space of system p, $|0\rangle_p$ is the pure state $|0\rangle$ of the system p and $|1\rangle_q\langle 1|$ is the density matrix of the system q. The composite system is described by the tensor product of its subsystems; for example, a composite system pq has the state space $\mathcal{H}_p \otimes \mathcal{H}_q$, and $|0\rangle_p \otimes |1\rangle_q$ (or, $|0\rangle_p |1\rangle_q$ for short) is a pure state in which subsystem p is in state $|0\rangle$ and subsystem q is in state $|1\rangle$. Due to the superposition principle, there exist states like $|\Phi\rangle_{pq} = \frac{1}{\sqrt{2}}(|0\rangle_p |0\rangle_q + |1\rangle_p |1\rangle_q)$ that cannot be written in the simple tensor form $|\phi\rangle_p |\psi\rangle_q$, which are called *entangled states*. These states play a crucial role in applications of quantum computation and quantum communication.

The state of a composite system fully determines the state of each subsystem. Formally, given composite system pq in state ρ, subsystem q is then in state $\mathrm{tr}_p(\rho)$, where the partial trace $\mathrm{tr}_p(\cdot)$ over p is a mapping from operators on $\mathcal{H}_p \otimes \mathcal{H}_q$ to operators on $\mathcal{H}_q$ defined by:

$$\mathrm{tr}_p(|\phi_p\rangle_p\langle\psi_p| \otimes |\phi_q\rangle_q\langle\psi_q|) = \langle\psi_p|\phi_p\rangle \cdot |\phi_q\rangle_q\langle\psi_q|$$

for all $|\phi_p\rangle, |\psi_p\rangle \in \mathcal{H}_p$ and $|\phi_q\rangle, |\psi_q\rangle \in \mathcal{H}_q$ together with linearity. The state $\mathrm{tr}_q(\rho)$ of subsystem q can be defined symmetrically. We often use the notations $\rho|_p \triangleq \mathrm{tr}_q(\rho)$ and $\rho|_q \triangleq \mathrm{tr}_p(\rho)$ in order to explicitly indicate that $\rho|_p$ and $\rho|_q$ are states of p, q, respectively.

Summary of Notations. Let $\mathbf{V}$ be the set of all quantum variables. A *quantum register* is a list of *distinct* variables $\overline{q} = q_1, \ldots, q_n$. Each quantum variable q has a type $\mathcal{H}_q$, which is the state Hilbert space of quantum system denoted by q. For a set of quantum variables $S = \{q_1, \ldots, q_n\} \subseteq \mathbf{V}$ (or a quantum register $\overline{q} = q_1, \ldots, q_n$), we fix following notations:

- $\mathcal{H}_S = \bigotimes_{i=1}^n \mathcal{H}_{q_i}$: the Hilbert space of S.
- $\dim(S)$: the dimension of $\mathcal{H}_S$.
- $\mathcal{D}(S)$: the set of all (mixed) quantum states (i.e. density matrices) of S. In particular, for any $\rho \in \mathcal{D}(S)$, its *domain* is defined as $\mathsf{dom}(\rho) \triangleq S$; we write $\mathcal{D} \triangleq \bigcup_{S \subseteq \mathbf{V}} \mathcal{D}(S)$ for the set of all states.
- $\mathcal{P}(S)$: the set of projections on $\mathcal{H}_S$. In particular, for any $P \in \mathcal{P}(S)$, its domain is defined as $\mathsf{free}(P) \triangleq S$. Since there is a one-to-one correspondence between projections and closed subspaces, we sometimes call closed subspaces of $\mathcal{H}_S$ projections. We write $\mathcal{P} \triangleq \bigcup_{S \subseteq \mathbf{V}} \mathcal{P}(S)$ for the set of all projections.
- $\rho|_S \triangleq \mathrm{tr}_{\mathsf{dom}(\rho) \setminus S}(\rho)$: the *restriction* of state ρ on S, defined as a reduced density operator over $S \cap \mathsf{dom}(\rho)$.

B. Quantum Programs: Syntax and Semantics

For simplicity of presentation, we consider the quantum **while**-language [16] which does not include classical variables.

Definition 1 (Syntax [16])**.** *The quantum **while**-programs are defined by the grammar:*

$$\begin{aligned} \mathbf{C} ::= {} & \mathbf{skip} \mid \mathbf{C}_1; \mathbf{C}_2 \mid q := |0\rangle \mid \overline{q} := U[\overline{q}] \\ & \mid \mathbf{if}\ (\square m \cdot M[\overline{q}] = m \to \mathbf{C}_m)\ \mathbf{fi} \\ & \mid \mathbf{while}\ M[\overline{q}] = 1\ \mathbf{do}\ \mathbf{C}\ \mathbf{od} \end{aligned}$$

The program constructs defined above are explained as follows. First, $q := |0\rangle$ initializes the quantum variable q in a basis state $|0\rangle$, and $\overline{q} := U[\overline{q}]$ applies a unitary transformation U to a sequence $\overline{q}$ of quantum variables. The case statement $\mathbf{if} \cdots \mathbf{fi}$ performs the projective measurement $M = \{M_m\}$ on $\overline{q}$, and then chooses a subprogram $\mathbf{C}_m$ to execute according to measurement outcome m. In the loop $\mathbf{while} \cdots \mathbf{od}$, the projective measurement $M = \{M_0, M_1\}$ in the guard has only two possible outcomes $0, 1$: if the outcome is 0 the loop terminates, and if the outcome is 1 it executes the loop body $\mathbf{C}$ and enters the loop again. For simplicity of presentation, we will use the following abbreviation: $\mathbf{for}\ i = 1, \ldots, N\ \mathbf{do}\ \mathbf{C}_i\ \mathbf{od} \triangleq \mathbf{C}_1; \ldots; \mathbf{C}_N$.

[2] That is, $P : \mathcal{H} \to \mathcal{H}$ is a projection over $\mathcal{H}$ iff $P = P^\dagger = P^2$.

For each program $\mathbf{C}$, we write $\mathsf{var}(\mathbf{C})$ for the set of all quantum variables in $\mathbf{C}$. If $\mathbf{V} \supseteq \mathsf{var}(\mathbf{C})$ is a set of quantum variables, and $\rho \in \mathcal{D}(\mathbf{V})$, then $\langle \mathbf{C}, \rho \rangle$ is called a configuration (of domain $\mathbf{V}$).

Definition 2 (Operational Semantics [16])**.** *The operational semantics of quantum programs is defined as a transition relation $\rightarrow$ by the following transition rules:*

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

$$\text{(UT)}\ \langle \overline{q} := U[\overline{q}], \rho \rangle \rightarrow \langle \mathbf{E}, U\rho U^\dagger \rangle$$

$$\text{(SC)}\ \frac{\langle \mathbf{C}_1, \rho \rangle \rightarrow \langle \mathbf{C}_1', \rho' \rangle}{\langle \mathbf{C}_1; \mathbf{C}_2, \rho \rangle \rightarrow \langle \mathbf{C}_1'; \mathbf{C}_2, \rho' \rangle}$$

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

$$\text{(L0)}\ \langle \mathbf{while}\ M[\overline{q}] = 1\ \mathbf{do}\ \mathbf{C}\ \mathbf{od}, \rho \rangle \rightarrow \langle \mathbf{E}, M_0 \rho M_0^\dagger \rangle$$

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

$\mathbf{E}$ is the empty program. In (In), $\rho_0^q = \sum_n |0\rangle_q \langle n|\rho|n\rangle_q \langle 0|$. In (SC), we use the convention $\mathbf{E}; \mathbf{C}_2 = \mathbf{C}_2$. In (IF), m ranges over every possible outcome of measurement $M = \{M_m\}$.

Transitions in rules (IF), (L0) and (L1) are essentially probabilistic; but we adopt a convention from [32] to present them as non-probabilistic transitions. For example, for each m, the transition in (IF) happens with probability $p_m = \mathrm{tr}(M_m^\dagger M_m \rho)$ and the program state ρ is changed to $\rho_m = M_m \rho M_m^\dagger / p_m$. We can combine probability p_m and density operator ρ_m into a partial density operator $M_m \rho M_m^\dagger = p_m \rho_m$. This convention significantly simplifies the presentation.

Definition 3 (Denotational Semantics [16])**.** *Let $\mathbf{V}$ be a set of variables. Then for any quantum program $\mathbf{C}$ with $\mathsf{var}(\mathbf{C}) \subseteq \mathbf{V}$, its semantic function of domain $\mathbf{V}$ is the mapping $[\![\mathbf{C}]\!]_{\mathbf{V}} : \mathcal{D}(\mathbf{V}) \rightarrow \mathcal{D}(\mathbf{V})$ defined by $[\![\mathbf{C}]\!]_{\mathbf{V}}(\rho) = \sum \{\!|\rho' : \langle \mathbf{C}, \rho \rangle \rightarrow^* \langle \mathbf{E}, \rho' \rangle|\!\}$ for every $\rho \in \mathcal{D}(\mathbf{V})$, where $\rightarrow^*$ is the reflexive and transitive closure of $\rightarrow$, and $\{\!|\cdot|\!\}$ denotes a multi-set.*

Note that auxiliary variables in $\mathbf{V} \setminus \mathsf{var}(\mathbf{C})$ are allowed in the above definition of semantic function $[\![\mathbf{C}]\!]_{\mathbf{V}}$. The following proposition shows that the denotational semantics of a program $\mathbf{C}$ is independent of these auxiliary variables.

Proposition 1 (Proposition 3.3.5 in [33])**.** *For any program $\mathbf{C}$ and any set $\mathbf{V} \supseteq \mathsf{var}(\mathbf{C})$ of variables, the semantic function of domain $\mathbf{V}$ is a cylindric extension of the semantic function of domain $\mathsf{var}(\mathbf{C})$: $[\![\mathbf{C}]\!]_{\mathbf{V}} = [\![\mathbf{C}]\!]_{\mathsf{var}(\mathbf{C})} \otimes \mathcal{I}_{\mathbf{V} \setminus \mathsf{var}(\mathbf{C})}$, where $\mathcal{I}_{\mathbf{V} \setminus \mathsf{var}(\mathbf{C})}$ is the identity quantum operation in $\mathcal{H}_{\mathbf{V} \setminus \mathsf{var}(\mathbf{C})}$.*

C. Brief review of BI-Logic

Next, we briefly review the logic of Bunched Implications (BI) [1], [2]. BI is a sub-structural logic with the following syntax:

$$\phi, \psi ::= p \in \mathcal{AP} \mid \top \mid \bot \mid \phi \wedge \psi \mid \phi \vee \psi \mid \phi \rightarrow \psi \mid \phi * \psi \mid \phi \mathrel{-\!\!*} \psi$$

where p ranges over a set $\mathcal{AP}$ of atomic propositions. Besides standard propositional logic, BI contains a substructural fragment – the separating conjunction $*$ and separating implication $\mathrel{-\!\!*}$ ("magic wand"). A distinction between $*$ and $\wedge$ is that $*$ is not idempotent, i.e., $P * P \neq P$. For example, in the standard heap model of separation logic, the separating conjunction $P * Q$ is true of a heap if it can be split into two heaplets, one of which makes P true and the other of which makes Q true. The implication $\mathrel{-\!\!*}$ is adjoint to $*$. For example, $P \mathrel{-\!\!*} Q$ holds in some heap if adding a separate heap satisfying P leads to a combined heap satisfying Q.

The most general semantics of BI is given in terms of a kind of Kripke structures, called BI frames. Standard BI frame is based on a pre-ordered commutative monoid:

Definition 4 (BI frame [1], [2])**.** *A BI frame is a tuple $\mathcal{X} = (X, \circ, \preceq, e)$, where X is a set equipped with a preorder $\preceq$, and $\circ : X \times X \rightarrow X$ is a partial binary operation with an unit element e and satisfying the following conditions:*

1) *(Unit Existence) for all x, $x = x \circ e = e \circ x$;*
2) *(Commutativity) $x \circ y = y \circ x$;*
3) *(Associativity) $x \circ (y \circ z) = (x \circ y) \circ z$;*
4) *(Compatible with $\preceq$) $x \preceq x'$ and $y \preceq y'$ and both $x \circ x'$ and $y \circ y'$ are defined, then $x \circ x' \preceq y \circ y'$.*

Above, equalities state that either both sides are defined and equal, or both sides are undefined.

Intuitively, if we choose the collections of resources as possible worlds, then $\circ$ can be interpreted as a commutative combination of resources. The identity e is an empty resource or lack of resource, and combining any resource x and empty resource e yields x itself. Based on the combination, a preorder is defined: if x is a combination of resources y and z, it should be "larger" than y since it contains y.

The semantics of formulas depends on the semantics of atomic propositions. A *valuation* is a mapping $\mathcal{V} : \mathcal{AP} \rightarrow \wp(X)$ where $\wp$ represents the power set, and it is *monotonic* if $x \in \mathcal{V}(p)$ and $y \succeq x$ implies $y \in \mathcal{V}(p)$. A BI frame $\mathcal{X}$ together with a monotonic valuation $\mathcal{V}$ gives a BI model $\mathcal{M}$.

Definition 5 (Satisfaction in BI models [1], [2])**.** *Given a BI formula ϕ and a BI model $\mathcal{M} = (X, \circ, \preceq, e, \mathcal{V})$. For each $x \in X$, the relation $x \models \phi$ is defined by induction on ϕ:*

$$\begin{aligned}
&x \models_{\mathcal{M}} p \text{ iff } x \in \mathcal{V}(p)\\
&x \models_{\mathcal{M}} \top : \textit{ always} \qquad x \models_{\mathcal{M}} \bot : \textit{ never}\\
&x \models_{\mathcal{M}} \phi_1 \wedge \phi_2 \ \text{ iff } \ x \models_{\mathcal{M}} \phi_1 \textit{ and } x \models_{\mathcal{M}} \phi_2\\
&x \models_{\mathcal{M}} \phi_1 \vee \phi_2 \ \text{ iff } \ x \models_{\mathcal{M}} \phi_1 \textit{ or } x \models_{\mathcal{M}} \phi_2\\
&x \models_{\mathcal{M}} \phi_1 \rightarrow \phi_2 \ \text{ iff } \ \forall x' \succeq x,\ x' \models_{\mathcal{M}} \phi_1 \textit{ implies } x' \models_{\mathcal{M}} \phi_2\\
&x \models_{\mathcal{M}} \phi_1 * \phi_2 \ \text{ iff } \ \exists y, z \textit{ s.t. } y \circ z \textit{ is defined and } x \succeq y \circ z,\\
&\qquad\qquad y \models_{\mathcal{M}} \phi_1 \textit{ and } z \models_{\mathcal{M}} \phi_2\\
&x \models_{\mathcal{M}} \phi_1 \mathrel{-\!\!*} \phi_2 \ \text{ iff } \ \forall y \textit{ s.t. } x \circ y \textit{ is defined},\\
&\qquad\qquad y \models_{\mathcal{M}} \phi_1 \textit{ implies } x \circ y \models_{\mathcal{M}} \phi_2.
\end{aligned}$$

Following [2] (see also [11]), a sound and complete Hilbert-style proof system of BI is presented in the our extended version [34].

III. Quantum Interpretation of BI Logic

Now, we are ready to present our quantum model of BI, using the resource semantics of BI. After defining the model, we introduce some atomic propositions. To lay the groundwork for the separation logic, we explore a technical property called *restriction*—which will be important for the frame rule—and we define a *modification* operation, an analog of substitution that we will use for reasoning about initialization and unitary transformations.

A. BI Frame of Quantum States

The basic idea of our model is to consider quantum states over specific registers as resources. Then, the separating conjunction is introduced to model independent combinations of spatially separate quantum resources (quantum states over disjoint registers). Formally, we define:

Definition 6. *The partial binary functions $\circ : \mathcal{D} \times \mathcal{D} \to \mathcal{D}$ on quantum states is defined by:*

$$\rho_1 \circ \rho_2 \triangleq \begin{cases} \rho_1 \otimes \rho_2 & : \textit{if } \mathsf{dom}(\rho_1) \cap \mathsf{dom}(\rho_2) = \emptyset \\ \textit{undefined} & : \textit{otherwise.} \end{cases}$$

Essentially, $\circ$ takes the tensor product of two quantum states with disjoint domains. Note that in our setting, the tensor product $\otimes$ is commutative since every quantum state $\rho \in \mathcal{D}$ is tagged with its domain. For example, $|1\rangle_p\langle 1| \otimes |0\rangle_q\langle 0| = |0\rangle_q\langle 0| \otimes |1\rangle_p\langle 1|$ denote the same state in pq. For the partial order over quantum states, we take the following:

Definition 7. *Let $\preceq$ be the partial order over $\mathcal{D}$: $\rho \preceq \rho'$ iff $\mathsf{dom}(\rho) \subseteq \mathsf{dom}(\rho')$ and $\rho = \rho'|_{\mathsf{dom}(\rho)}$.*

Intuitively, $\rho \preceq \rho'$ means that ρ describes a subsystem of ρ'; more precisely, if we discard the subsystem $\mathsf{dom}(\rho')\backslash\mathsf{dom}(\rho)$ of ρ', then the remaining subsystem is in state ρ. Combining all of the ingredients defined, we have:

Proposition 2. *$(\mathcal{D}, \circ, \preceq, 1)$ forms a BI frame, where scalar number 1 is understood as the state over the empty register.*

B. Atomic Propositions about Quantum States

To complete our description of the quantum BI logic, we introduce three atomic propositions and interpret them in quantum states. In general, we have a great deal of freedom in selecting these atomic propositions; the only requirement is that their interpretation must be monotone with respect to the pre-order $\preceq$. It worth pointing out that adding new atomic propositions requires extending the definition of modification (see Sec. III-D); fortunately, the most frequently used ones like uniformity/distribution, projections, and observables, have already been studied. Our atomic propositions are fairly general, but motivated by applications of our separation logic.

Propositions denoting free variables. We first introduce a set of atomic propositions $\mathbf{D}[S]$ for each variable set $S \subseteq \mathbf{V}$ with domain defined by $\mathsf{free}(\mathbf{D}[S]) \triangleq S$, and interpret it as the state with domain at least S:

$$[\![\mathbf{D}[S]]\!] \triangleq \{\rho \in \mathcal{D} : S \subseteq \mathsf{dom}(\rho)\}. \tag{1}$$

Propositions for qualitative analysis. For qualitative analysis of quantum programs, we often use projection operators as atomic propositions [17], [31], [35]–[37]. For a projection $P \in \mathcal{P}$ as an atomic proposition, its semantics $[\![P]\!]$ is defined as the following set of quantum states:

$$[\![P]\!] \triangleq \left\{\rho \in \mathcal{D} : \mathsf{free}(P) \subseteq \mathsf{dom}(\rho) \text{ and } \mathrm{supp}\big(\rho|_{\mathsf{free}(P)}\big) \subseteq P\right\} \tag{2}$$

where the *support* of a state $\rho \in \mathcal{D}$ is the (topological) closure of the subspace spanned by its eigenvectors with nonzero eigenvalues, or equivalently, $\mathrm{supp}(\rho) = \{|\phi\rangle \in \mathcal{H}_{\mathsf{dom}(\rho)} : \langle\phi|\rho|\phi\rangle = 0\}^{\perp}$.[3] Let us carefully explain the definition of $[\![P]\!]$. In the case that ρ has the same domain of P, it is natural to define $\rho \in [\![P]\!]$ if its support $\mathrm{supp}(\rho)$ lies in P, or equivalently, ρ is invariant under projection operator P. In the case where $\mathsf{dom}(\rho)$ and $\mathsf{free}(P)$ are not the same, in order to make $[\![P]\!]$ upward-closed (i.e., monotonic): $\rho \in [\![P]\!]$ and $\rho \preceq \rho'$ imply $\rho' \in [\![P]\!]$, it is appropriate to require that $\rho \in [\![P]\!]$ iff (i) $\mathsf{dom}(\rho) \supseteq \mathsf{free}(P)$; and (ii) the restricted state of ρ on $\mathsf{free}(P)$ is in $[\![P]\!]$.

Atomic propositions expressing uniformity in quantum security. As is well-known, probabilistic uniformity is a basic property in verification of security protocols. To describe uniformity in quantum protocols, we introduce an atomic proposition $\mathbf{U}[S]$ for each $S \subseteq \mathbf{V}$ denoting finite-dimensional quantum systems [38]. Its domain is $\mathsf{free}(\mathbf{U}[S]) \triangleq S$. The semantics of $\mathbf{U}[S]$ is defined as the following set of quantum states:

$$[\![\mathbf{U}[S]]\!] \triangleq \left\{\rho \in \mathcal{D} : S \subseteq \mathsf{dom}(\rho) \text{ and } \rho|_S = \frac{I_S}{\dim(S)}\right\}, \tag{3}$$

where I_S is the identity density on the quantum system over registers S. The intuition behind defining equation (3) is quite simple: for a state ρ in $[\![\mathbf{U}[S]]\!]$ such that $S \subseteq \mathsf{dom}(\rho)$, its restriction on S should be the completely mixed state, $\frac{I_S}{\dim(S)}$, which means "uniformly distributed" over all orthonormal bases of the system denoted by S.

Axiom schema for atomic formulas. With the interpretation of atomic propositions, we have:

Proposition 3.

1) *For all $S \subseteq \mathbf{V}$ and identity operator I_S over $\mathcal{H}_S$, we have: $\models \mathbf{D}[S] \leftrightarrow I_S$.*
2) *For all $P, Q \in \mathcal{P}$ with disjoint domains, we have: $\models P \wedge Q \leftrightarrow (P \otimes Q)$;*
3) *If $S_1 \subseteq S_2$, then $\models \mathbf{U}[S_2] \to \mathbf{U}[S_1]$.*
4) *If $S_1 \cap S_2 = \emptyset$, then $\models (\mathbf{U}[S_1] * \mathbf{U}[S_2]) \leftrightarrow \mathbf{U}[S_1 \cup S_2]$.*

Note that $\otimes$ is not *a connective in BI: instead, it stands for the mathematical tensor product. Thus, $P \otimes Q$ is a projection and can be considered as atomic formula.*

C. Restriction Property

After choosing (the interpretation of) atomic propositions in the quantum frame $(\mathcal{D}, \circ, \preceq, 1)$, the semantics of all BI formulas can be defined using Definitions 5. As is well-known,

[3] $^{\perp}$ stands for ortho-complement.

the frame rule plays an essential role in separation logic, and in turn it heavily relies on the restriction property that satisfaction only depends on the free variables appearing in a BI formula ϕ. The restriction property was also identified and generalized in prior work on probabilistic separation logic [9]. However, the restriction property:

$$\rho \models \phi \Rightarrow \rho|_{\mathsf{free}(\phi)} \models \phi$$

where $\mathsf{free}(\phi)$ stands for the free variables occurring in ϕ, does not hold for our quantum setting, even for the ordinary implication $\phi = \phi_1 \to \phi_2$ (see Definition 5 for its semantics). Essentially, the validity of the restriction property in the probabilistic setting can be attributed to a fundamental fact in probability theory—the existence of extensions.[4] Unfortunately, this does not always hold for quantum systems. Indeed, it is violated by the well-known phenomenon of "Monogamy" – one of the most fundamental properties of entanglement.[5]

Since we wish to have a frame rule in QSL, we need to recover the restriction property to a certain extent. While not all formulas satisfy this property, we can identify a subset of them that do satisfy it.

Definition 8. *The formulas generated by following grammar are denoted by* Res.

$$\phi, \psi ::= p \in \mathcal{AP} \mid \top \mid \bot \mid \phi \wedge \psi \mid \phi \vee \psi \mid \phi * \psi$$

Proposition 4. *Any formula $\phi \in \mathrm{Res}$ is restrictive; that is, for any $\rho \models \phi$, $\rho|_{\mathsf{free}(\phi)} \models \phi$.*

The above simple treatment of restriction property is sufficient for the purpose of this paper. A more intrinsic way for recovering this property in the quantum setting will be discussed in Section VII-A.

D. Quantum Modification of BI Formulas

In classical program logic, substitution is used in the inference rule about assignment statements. In the quantum setting, due to no-cloning of quantum data, the role of assignment is played by two basic constructs: unitary transformation and initialization. We conclude this section by defining a technique of modifying BI formulas, which we will need reasoning about these operations.

Definition 9 (Modification of atomic propositions)**.** *Let $\mathbf{C}$ be a unitary transformation $\overline{q} := U[\overline{q}]$ or an initialization $q := |0\rangle$. For any $p \in \mathcal{AP}$, we write $p[\mathbf{C}]$ for the $\mathbf{C}$-modification of p. For the three classes of atomic propositions defined in Sec. III-B, $p[\mathbf{C}]$ is defined as follows:*

1) *For an atomic proposition $\mathbf{D}[S]$ defined in Eq. (1), $\mathbf{D}[S][\mathbf{C}] \triangleq \mathbf{D}[S]$;*
2) *For an atomic proposition $P \in \mathcal{P}$ as a projection defined in Eq. (2),*

$$P[\overline{q} := U[\overline{q}]] \triangleq \begin{cases} P_{U[\overline{q}]} & \text{if } \overline{q} \subseteq \mathsf{free}(P); \\ P & \overline{q} \cap \mathsf{free}(P) = \emptyset; \\ \text{undefined} & \text{otherwise;} \end{cases}$$

$$P[q := |0\rangle] \triangleq \begin{cases} \mathbf{D}[q] \wedge \lceil P \rceil_q & \text{if } q \in \mathsf{free}(P); \\ P & \text{otherwise;} \end{cases}$$

where projections $P_{U[\overline{q}]}$ and $\lceil P \rceil_q$ are given as follows:

$$P_{U[\overline{q}]} = (U^{\overline{q}\dagger} \otimes I_{\mathsf{free}(P)\backslash\overline{q}}) P (U^{\overline{q}} \otimes I_{\mathsf{free}(P)\backslash\overline{q}}),$$

and $\lceil P \rceil_q = \bigsqcup\{$ closed subspaces $T : |0\rangle_q\langle 0| \otimes T \subseteq P\} \in \mathcal{P}(\mathsf{free}(P)\backslash q)$. Here, $\sqcup$ is the disjunction of projections in quantum logic, that is, for projections P, Q with the same domain, $P \sqcup Q = \overline{\mathrm{span}(P \cup Q)}$ with "$\overline{\cdot}$" standing for (topological) closure.
3) *For any atomic proposition $\mathbf{U}[S] \in \mathcal{U}$ for uniformity defined in Eq.(3),*
   a) *If $\overline{q} \subseteq S$ or $\overline{q} \cap S = \emptyset$, then $\mathbf{U}[S][\overline{q} := U[\overline{q}]] \triangleq \mathbf{U}[S]$; otherwise, $\mathbf{U}[S][\overline{q} := U[\overline{q}]]$ is undefined;*
   b) *If $q \notin S$, then $\mathbf{U}[S][q := |0\rangle] \triangleq \mathbf{U}[S]$; otherwise, $\mathbf{U}[S][q := |0\rangle]$ is undefined.*

The modification of some atomic propositions/BI formulas may not exist; we write $\phi[\mathbf{C}]\downarrow$ whenever $\phi[\mathbf{C}]$ is defined. The notion of modification can be easily extended to all BI formulae:

Definition 10 (Modification of BI formulas)**.** *Let $\mathbf{C}$ be unitary transformation $\overline{q} := U[\overline{q}]$ or initialization $q := |0\rangle$. The modification $\phi[\mathbf{C}]$ of BI formula ϕ is defined by induction on the structure of ϕ:*

1) *if $\phi \equiv \top$ or $\bot$, then $\phi[\mathbf{C}] \triangleq \phi$;*
2) *if $\phi \equiv p \in \mathcal{AP}$, then $\phi[\mathbf{C}]$ is defined according to Definition 9;*
3) *if $\phi \equiv \phi_1 \triangle \phi_2$ where $\triangle \in \{\wedge, \vee\}$ and $\phi_1[\mathbf{C}]\downarrow$ and $\phi_2[\mathbf{C}]\downarrow$, then $\phi[\mathbf{C}] \triangleq \phi_1[\mathbf{C}] \triangle \phi_2[\mathbf{C}]$;*
4) *if $\phi \equiv \phi_1 * \phi_2$, $\phi_i[\mathbf{C}]\downarrow$ and $\overline{q} \subseteq \mathsf{free}(\phi_i)$ or $\overline{q} \cap \mathsf{free}(\phi_i) = \emptyset$ for $i = 1, 2$, then*
   a) *if $\mathbf{C} \equiv \overline{q} := U[\overline{q}]$, then $\phi[\mathbf{C}] \triangleq \phi_1[\mathbf{C}] * \phi_2[\mathbf{C}]$;*
   b) *if $\mathbf{C} \equiv q := |0\rangle$, then*
      - *if $q \notin \mathsf{free}(\phi_1) \cup \mathsf{free}(\phi_2)$, $\phi[\mathbf{C}] \triangleq \phi_1[\mathbf{C}] * \phi_2[\mathbf{C}]$;*
      - *if only one of $q \in \mathsf{free}(\phi_1)$, $q \in \mathsf{free}(\phi_2)$ is satisfied, then $\phi[\mathbf{C}] \triangleq (\phi_1[\mathbf{C}] \wedge \phi_2[\mathbf{C}]) \wedge (\mathbf{D}[\mathsf{free}(\phi_1)\backslash q] * \mathbf{D}[\mathsf{free}(\phi_2)\backslash q])$;*

   The reason for the complexity of this case will be seen in the program logic; roughly speaking, initialization on q is special because it can introduce independence: it makes q independent from all variables.
5) *otherwise, $\phi[\mathbf{C}]$ is undefined.*

A close connection between the semantics of a BI formula ϕ and its modification $\phi[\mathbf{C}]$ is shown in the following:

Proposition 5. *Let $\mathbf{C}$ be unitary transformation $\overline{q} := U[\overline{q}]$ or initialization $q := |0\rangle$, and ϕ be any BI formula. If its modification $\phi[\mathbf{C}]$ is defined, then:*

[4]For two joint-distributions μ_{AB} and μ_{BC} over sets A, B and B, C respectively, if they are consistent on B (with the same marginal on B) then there exists joint-distribution μ_{ABC} over A, B, C which takes μ_{AB} and μ_{BC} as marginals.

[5]If two qubits A and B are maximally correlated, then they cannot be correlated at all with a third qubit C; more precisely, if A and B are in a maximally entangled state, then A and C cannot be in any entangled state.

1) ϕ *and* $\phi[\mathbf{C}]$ *have the same domain:* $\mathsf{free}(\phi) = \mathsf{free}(\phi[\mathbf{C}])$*;*
2) *for all* $\rho \in \mathcal{D}(\mathsf{free}(\phi) \cup \mathsf{var}(\mathbf{C}))$*, if* $\rho \models \phi[\mathbf{C}]$*, then* $[\![\mathbf{C}]\!](\rho) \models \phi$*.*

IV. Separation Logic for Quantum Programs

Now we are ready to present our separation logic for quantum programs, using quantum BI formulas as the assertion language.

A. Judgments and Validity

Let us first define judgments (correctness formulas) in quantum separation logic. A judgment is a Hoare triple of the form $\{\phi\}\mathbf{C}\{\psi\}$ with both precondition ϕ and postcondition ψ being *restrictive* BI formulas (cf. Definition 8).

Definition 11 (Validity)**.** *Let* $\mathbf{V}$ *be a set of quantum variables with* $\mathsf{free}(\phi), \mathsf{free}(\psi), \mathsf{var}(\mathbf{C}) \subseteq \mathbf{V}$*. Then a correctness formula* $\{\phi\}\mathbf{C}\{\psi\}$ *is true in the sense of partial correctness with respect to* $\mathbf{V}$*, written* $\mathbf{V} \models \{\phi\}\mathbf{C}\{\psi\}$*, if we have:*

$$\forall \rho \in \mathcal{D}(\mathbf{V}), \quad \rho \models \phi \Rightarrow [\![\mathbf{C}]\!]_{\mathbf{V}}(\rho) \models \psi.$$

Here, satisfaction relation $\rho \models \phi$ *and* $[\![\mathbf{C}]\!]_{\mathbf{V}}(\rho) \models \psi$ *are defined according to the quantum interpretation of BI logic given in Section III.*

The following theorem indicates that satisfaction does not depends on auxiliary variables.

Theorem 1. *For any two sets* $\mathbf{V}$ *and* $\mathbf{V}'$ *containing all free variables of* ϕ, ψ *and* $\mathbf{C}$*,*

$$\mathbf{V} \models \{\phi\}\mathbf{C}\{\psi\} \text{ if and only if } \mathbf{V}' \models \{\phi\}\mathbf{C}\{\psi\}.$$

As a consequence, we can drop $\mathbf{V}$ from $\mathbf{V} \models \{\phi\}\mathbf{C}\{\psi\}$ and simply write $\models \{\phi\}\mathbf{C}\{\psi\}$.

In the remainder of this section, we gradually develop the proof system for our quantum separation logic. For better readability, this proof system is organised as several sets of inference rules.

B. Inference Rules for Program Constructs

The first set of our inference rules are designed for reasoning about basic quantum program constructs and displayed in Fig. 2. Some of them deserve careful explanations:

- **Rules** Init **and** Unit: With the definition of modification of BI formulas and Proposition 5 in mind, the rules Init and Unit are similar to the (backwards) inference rule $\{\phi[e/x]\}x := e\{\phi\}$ for assignment in classical program logics.

- **Rules** RIf **and** RLoop: These two rules use the separating conjunction to perform reasoning about different execution paths. Note that condition $\phi \in \mathrm{CM}$ is imposed in the premises of the rules RIf **and** RLoop.

The set CM of assertions is formally defined as follows:

Definition 12. *A formula* ϕ *is closed under mixtures (CM), written* $\phi \in \mathrm{CM}$*, if for any* ρ, ρ'*, whenever* $\mathsf{dom}(\rho) = \mathsf{dom}(\rho')$*,* $\rho \models \phi$ *and* $\rho' \models \phi$*, we have:* $\forall\, \lambda \in [0,1],\ \lambda\rho + (1-\lambda)\rho' \models \phi$.

$$\textsc{Skip}\ \frac{}{\{\phi\}\mathbf{skip}\{\phi\}} \qquad \textsc{Init}\ \frac{\phi[q := |0\rangle]\downarrow}{\{\phi[q := |0\rangle]\}\, q := |0\rangle\{\phi\}}$$

$$\textsc{Unit}\ \frac{\phi[\overline{q} := U[\overline{q}]]\downarrow}{\{\phi[\overline{q} := U[\overline{q}]]\}\overline{q} := U\,[\overline{q}]\,\{\phi\}}$$

$$\textsc{Seq}\ \frac{\{\phi\}\mathbf{C}_1\{\psi\} \qquad \{\psi\}\mathbf{C}_2\{\mu\}}{\{\phi\}\mathbf{C}_1; \mathbf{C}_2\{\mu\}}$$

$$\textsc{RIf}\ \frac{\{\phi * M_m\}\,\mathbf{C}_m\{\psi\} \text{ for all } m \quad \psi \in \mathrm{CM}}{\{\phi * \mathbf{D}(\overline{q})\}\mathbf{if}\ (\square m \cdot M[\overline{q}] = m \to \mathbf{C}_m)\ \mathbf{fi}\{\psi\}}$$

$$\textsc{RLoop}\ \frac{\{\phi * M_1\}\mathbf{C}\{\phi * \mathbf{D}(\overline{q})\} \quad \phi \in \mathrm{CM}}{\{\phi * \mathbf{D}(\overline{q})\}\mathbf{while}\ M[\overline{q}] = 1\ \mathbf{do}\ \mathbf{C}\ \mathbf{od}\{\phi \wedge M_0\}}$$

Fig. 2: Inference Rules for Program Constructs. In Init and Unit, $\downarrow$ means the existence of modification. In RIf and RLoop, M_0, M_1, M_m in assertions are regarded as projective predicates acting on $\overline{q}$.

Example 1. *For two projections* $P_0 = |0\rangle\langle 0|$ *and* $P_1 = |1\rangle\langle 1|$*,* $P_0 \wedge P_1$ *is* CM*, but* $P_0 \vee P_1$ *is not* CM *(both states* $|0\rangle\langle 0|$ *and* $|1\rangle\langle 1|$ *satisfies* $P_0 \vee P_1$*, but their affine combination* $\frac{I}{2} = \frac{|0\rangle\langle 0| + |1\rangle\langle 1|}{2}$ *does not satisfy* P_0 *nor* P_1 *and thus does not satisfy* $P_0 \vee P_1$*).*

To see why the condition $\phi \in \mathrm{CM}$ necessary, we note that a quantum program can be executed in different paths with non-zero probabilities, and its semantic function maps the input to a weighted summation of the outputs from different execution paths. The condition $\phi \in \mathrm{CM}$ is introduced so that satisfaction relation is preserved under affine combination. The following proposition identifies a class of formulas closed under mixture.

Proposition 6. *The formulas generated by following grammar are CM:*

$$\phi, \psi ::= p \in \mathcal{AP} \mid \top \mid \bot \mid \phi \wedge \psi \mid \mathbf{U}[S] * \phi$$

We need to pay special attention on the application of separating conjunctions $*$ in RIf and RLoop. Since the quantum measurement in the guards of **if**-statements and **while** loops may change the quantum state, we hereby consider a special kind of inputs that satisfying $\phi * I_{\overline{q}}$. Thus the subsystem being measured is uncorrelated to the part of the state described by ϕ, which ensures that the post-measurement state still satisfies ϕ. In RLoop, although $\phi * M_0$ is satisfied for each path, it does not belong to CM in general. Thus, only a weaker postcondition $\phi \wedge M_0 \in \mathrm{CM}$ can be achieved.

C. Structural rules

The second set of rules consists of the structural rules, presented in Fig. 3. The rules Conj and Disj are similar to their counterparts in classical program logics. To explain the other rules, let us fist define the global implication:

$$\textsc{Conseq}\quad \frac{\phi \overset{\text{G}}{\to} \phi' \quad \{\phi'\}\mathbf{C}\{\psi'\} \quad \psi' \overset{\text{G}}{\to} \psi}{\{\phi\}\mathbf{C}\{\psi\}}$$

$$\textsc{Conj}\quad \frac{\{\phi_1\}\mathbf{C}\{\psi_1\} \quad \{\phi_2\}\mathbf{C}\{\psi_2\}}{\{\phi_1 \wedge \phi_2\}\mathbf{C}\{\psi_1 \wedge \psi_2\}}$$

$$\textsc{Disj}\quad \frac{\{\phi_1\}\mathbf{C}\{\psi_1\} \quad \{\phi_2\}\mathbf{C}\{\psi_2\}}{\{\phi_1 \vee \phi_2\}\mathbf{C}\{\psi_1 \vee \psi_2\}}$$

$$\textsc{Const}\quad \frac{\{\phi\}\mathbf{C}\{\psi\} \quad \mathsf{free}(\mu) \cap \mathsf{var}(\mathbf{C}) = \emptyset}{\{\phi \wedge \mu\}\mathbf{C}\{\psi \wedge \mu\}}$$

$$\textsc{Frame}\quad \frac{\{\phi\}\mathbf{C}\{\psi\} \quad \mathsf{free}(\mu) \cap \mathsf{var}(\mathbf{C}) = \emptyset \quad \mathsf{free}(\psi) \cup \mathsf{var}(\mathbf{C}) \subseteq \mathsf{free}(\phi) \text{ or } \psi \in \mathrm{SP}}{\{\phi * \mu\}\mathbf{C}\{\psi * \mu\}}$$

Fig. 3: Structural Rules. Since $\overset{\text{G}}{\to}$ is strictly weaker than $\to$, CONSEQ is stronger than ordinary weak rule.

Definition 13 (Global implication)**.** *For any BI formulas ϕ, ψ, the global implication $\phi \overset{\text{G}}{\to} \psi$ is defined as the abbreviation of* $\mathbf{D}[\mathsf{free}(\phi) \cup \mathsf{free}(\psi)] \to (\phi \to \psi)$.

Trivially, $\overset{\text{G}}{\to}$ is strictly weaker than $\to$. The difference is that, $\phi \overset{\text{G}}{\to} \psi$ is already enough to ensure that for any state ρ with $\mathsf{dom}(\rho) \supseteq \mathsf{free}(\phi) \cup \mathsf{free}(\psi)$, $\rho \models \phi$ implies $\rho \models \psi$. For example, we have following proposition:

Proposition 7. *For all $\phi \in \mathrm{Res}$ and $S \subseteq \mathbf{V}$, it holds that* $\models \phi \overset{\text{G}}{\leftrightarrow} \phi \wedge \mathbf{D}[S]$.

Now we are ready to carefully examine the remaining rules in Fig. 3.

• **Rules** CONSEQ: This rule is also similar to its counterpart in classical program logics, but there is a subtle difference between them. Since only global states (i.e. the states whose domain contains all free variables appearing in the assertions and programs) are considered in defining the validity of the Hoare triple, we use $\overset{\text{G}}{\to}$ in the premise of the CONSEQ rule for comparing assertions. It is easy to see that the rule is also sound when using $\to$, but the CONSEQ rule with $\overset{\text{G}}{\to}$ is stronger.

• **Rules** CONST: This rule states that if any variable appearing in program $\mathbf{C}$ is not free in μ, then μ is preserved and thus can be conjoined to the pre- and post-conditions. The principle behind is that μ is restrictive, i.e., the satisfaction of μ depends only on the reduced state over subsystem $\mathsf{free}(\mu)$, which trivially remains unchanged after executing $\mathbf{C}$. An interesting application of this rule is proving product predicates from local reasoning using Proposition 3.

• **Rules** FRAME: The condition $\mathsf{free}(\mu) \cap \mathsf{var}(\mathbf{C}) = \emptyset$ in the premise ensures that μ can be conjoined with the pre- and post-conditions. The condition $\mathsf{free}(\psi) \cup \mathsf{var}(\mathbf{C}) \subseteq \mathsf{free}(\phi)$ guarantees that, if the input satisfies $\phi * \mu$, which asserts that subsystems $\mathsf{free}(\phi)$ and $\mathsf{free}(\mu)$ are uncorrelated, then after executing $\mathbf{C}$, these two subsystems are still independent since $\mathsf{var}(\mathbf{C}) \subseteq \mathsf{free}(\phi)$, and furthermore, by the downward closed property of independence[6], subsystems $\mathsf{free}(\psi)$ and $\mathsf{free}(\mu)$ are uncorrelated as $\mathsf{free}(\psi) \subseteq \mathsf{free}(\phi)$. It is particularly interesting to note that the latter condition can be altered by $\psi \in \mathrm{SP}$ defined in the following:

Definition 14 (Supported Assertion, c.f. [21])**.** *A formula ψ is called supported, written $\psi \in \mathrm{SP}$, if $[\![\psi]\!]$ is nonempty then it has a least element, or equivalently, there exists a $S \subseteq \mathbf{V}$ such that 1. at most one $\rho \in \mathcal{D}(S)$ satisfies ψ and 2. if $\sigma \models \psi$, $\sigma \succeq \rho$.*

Trivially, any uniformity proposition $\mathbf{U}[S]$ and any atomic proposition defined by a projection of rank 1 are in SP; more examples of SP are given in the extended version [34]. The frame rule with SP condition is nontrivial and it will be very useful in our later case studies on verification of quantum information-theoretic security; indeed, this application uncovered the condition $\psi \in \mathrm{SP}$. Note that under this condition, the frame rule is sound even without any restriction on $\mathsf{free}(\psi)$, $\mathsf{free}(\phi)$ and $\mathsf{var}(\mathbf{C})$. This seems counter-intuitive; but in fact, the premise $\{\phi\}\mathbf{C}\{\psi\}$ is much stronger than it looks at first sight, given that the postcondition $\psi \in \mathrm{SP}$. If the input satisfies precondition ϕ, then an execution of $\mathbf{C}$ is almost equivalent to first erasing any information on subsystem $\mathsf{free}(\psi)$ (of course, it is now uncorrelated with the rest part of the whole system), and then regenerating the singleton that satisfies the postcondition ψ.

D. Reasoning about Entangled Predicates

Many quantum algorithms are designed following the same pattern: start from a large entangled state, and then operate on various subsystems. Inevitably, entanglements often appear in the preconditions and/or postconditions of Hoare triples appropriate for specifying the correctness of these algorithms. But the frame rule itself is not strong enough to verify them. To see this more clearly, let us consider the following simple example:

Example 2. *Let $|\Phi^{\pm}\rangle = \frac{1}{\sqrt{2}}(|00\rangle \pm |11\rangle)$ be two Bell states (entanglement). Define projections $\Phi^{\pm} = |\Phi^{\pm}\rangle\langle\Phi^{\pm}|$ and let S be the phase gate. The program $\mathbf{C} \equiv S[q_1]; S[q_2]$ transforms one Bell states to the other; that is, both $\{\Phi^+\}\mathbf{C}\{\Phi^-\}$ and $\{\Phi^-\}\mathbf{C}\{\Phi^+\}$ are true. However, they cannot be proved by using* FRAME *or* CONST *to lift local correctness of $S[q_1]$ and $S[q_2]$ to global predicates $\Phi^{\pm}$, since $\Phi^{\pm}$ cannot be written in the form of $\Phi^{\pm} \not\equiv \phi_{q_1} * \psi_{q_2}$ or $\Phi^{\pm} \not\equiv \phi_{q_1} \wedge \psi_{q_2}$.*

Fortunately, our frame rule can be combined with a technique for reasoning about entangled predicates proposed in [23] to handle this problem. Originally, this technique was introduced for parallel quantum programs. Here, we need to reformulate it in a way convenient for our purpose. A combination of this technique with the frame rule can significantly broaden the range of applications of our quantum separation logic. To this end, we need to generalise Definition 9 from

[6]Roughly speaking, if subsystems S_1 and S_2 are independent, then subsystems S_1' and S_2' are also independent if $S_1' \subseteq S_1$ and $S_2' \subseteq S_2$.

$$\textsc{UnCR}\quad \frac{\{\phi\}\mathbf{C}\{\psi\}\quad \overline{q}\cap \mathsf{var}(\mathbf{C})=\emptyset\quad \phi[\mathcal{E}[\overline{q}]]\downarrow\ \psi[\mathcal{E}[\overline{q}]]\downarrow}{\{\phi[\mathcal{E}[\overline{q}]]\}\mathbf{C}\{\psi[\mathcal{E}[\overline{q}]]\}}$$

Fig. 4: Proof rule for dealing with entangled predicates. $\downarrow$ means the existence of modification.

modification by a unitary transformation, and initialization to modification by a general quantum operation.[7]

Definition 15 ($\mathcal{E}$-Modification)**.** *Let $\mathcal{E}$ be quantum operation on $\overline{q}$. The $\mathcal{E}$-Modification $\phi[\mathcal{E}[\overline{q}]]$ acting on register $\overline{q}$ of a BI formula ϕ is defined inductively:*

1) *(Atomic Proposition) For any $P\in\mathcal{P}$, we have:*[8]
   a) *if $\overline{q}\subseteq \mathsf{free}\,(P)$,*
   $$P[\mathcal{E}[\overline{q}]]\triangleq \left(\left(\mathcal{E}^*_{\overline{q}}\otimes \mathcal{I}_{\mathsf{free}(P)\setminus\overline{q}}\right)(P^{\perp})\right)^{\perp};$$
   b) *if $\overline{q}\cap \mathsf{free}\,(P)=\emptyset$, $P[\mathcal{E}[\overline{q}]]\triangleq P$;*
   c) *otherwise, $P[\mathcal{E}[\overline{q}]]$ is undefined;*
2) *(Composite) Write $\phi[\mathcal{E}[\overline{q}]]\downarrow$ if $\phi[\mathcal{E}[\overline{q}]]$ is defined.*
   a) *if $\phi\equiv\top$ or $\perp$, then $\phi[\mathcal{E}[\overline{q}]]\triangleq\phi$;*
   b) *if $\phi\equiv p\in\mathcal{AP}$, then $\phi[\mathcal{E}[\overline{q}]]$ is defined by Clause* (1)*;*
   c) *if $\phi\equiv\phi_1\ \triangle\ \phi_2$ where $\triangle\in\{\wedge,\vee\}$ and both $\phi_1[\mathcal{E}[\overline{q}]]\downarrow$ and $\phi_2[\mathcal{E}[\overline{q}]]\downarrow$, then $\phi[\mathcal{E}[\overline{q}]]\triangleq\phi_1[\mathcal{E}[\overline{q}]]\ \triangle\ \phi_2[\mathcal{E}[\overline{q}]]$*
   d) *otherwise, $\phi[\mathcal{E}[\overline{q}]]$ is undefined.*

Now we can introduce a new inference rule UNCR (stands for "uncorrelated") in Fig. 4. This rule plays an essential role in the verification of VQA (see Section V). We divide VQA into several pieces and reason locally, but the global predicate we desired is an entangled predicate that cannot be constructed using FRAME. UNCR is the bridge for structural reasoning from local to global predicates. In addition, a formal verification of Example 2 using UNCR can be found in the extended version [34].

Auxiliary variables are the key to using UNCR. In comparison, Unruh [38] also employs the ghost variables. We would like to point out that the uses of auxiliary variables are essentially different: ghost variables in [38] are interpreted by existential quantifiers and used for dealing with mixed states/distributions; in contrast, auxiliary variables in our QSL can be regarded as *actual* variables since we define the resource semantics of BI and provide Theorem 1, so they can be introduced freely and removed if they are used separately from the prime system and do not appear in pre and post-conditions. Furthermore, UNCR is strictly more powerful than TRANSMUTE in [38] when dealing with entangled predicates; in fact, [23] shows the completeness of this idea if predicates are projections.

[7]Quantum operation is used to describe the evolution of a (open) quantum system and can be characterized by an superoperator $\mathcal{E}$, namely a completely-positive and trace-non-increasing linear map from $\mathcal{D}$ to $\mathcal{D}$. For every superoperator $\mathcal{E}$, there exists a set of Kraus operators $\{E_i\}_i$ (linear operators that satisfy completeness condition $\sum_i E_i^\dagger E_i = I$) such that $\mathcal{E}(\rho)=\sum_i E_i\rho E_i^\dagger$ for any input ρ.

[8]Here $\perp$ stands for the ortho-complement, for not only the projections but Hermitian operators, in the sense that $A^\perp=\mathrm{span}\{|\psi\rangle\in\mathcal{H}_{\mathsf{free}(A)}: A|\psi\rangle=0\}$. $\mathcal{E}^*$ is dual of $\mathcal{E}$; in detail, $\mathcal{E}^*(A)=\sum_i E_i^\dagger A E_i$ if $\mathcal{E}$ has the operator-sum representation $\mathcal{E}(\rho)=\sum_i E_i\rho E_i^\dagger$.

E. Soundness

To conclude this section, we show that quantum separation logic QSL consisting of all the proof rules listed in Figure 2–4 are sound. The detailed proof can be found in [34].

Theorem 2 (Soundness of QSL)**.** *A program $\mathbf{C}$ is* almost surely terminating *if for all inputs ρ, $\mathrm{tr}([\![\mathbf{C}]\!](\rho))=\mathrm{tr}(\rho)$. If $\mathbf{C}$ is a most surely terminating program, then $\vdash\{\phi\}\mathbf{C}\{\psi\}$ implies $\models\{\phi\}\mathbf{C}\{\psi\}$.*

V. LOCAL REASONING: ANALYSIS OF VARIATIONAL QUANTUM ALGORITHMS

From now on we present a couple of examples to demonstrate applicability of our quantum separation logic. Variational quantum algorithms (VQA) are a class of hybrid quantum/classical algorithms solving a fundamental problem in quantum chemistry – determine the ground state of a quantum system [14], [15]. It has been identified as one of the first practical applications of near-term Noisy Intermediate Scale Quantum (NISQ) computers [39], and thus were chosen as an example in the tutorials of several quantum programming platforms including Google's Cirq [26]. Surprisingly, using the inference rules presented in the last section, we are able to show that the implementation of VQA in the tutorial of Cirq is actually incorrect; that is, the approximation of ground energy computed by the quantum circuit given there is sometimes far from the real one.

A. Variational Quantum Algorithm (VQA)

A typical VQA uses a hybrid computing system consisting of a QPU (quantum processing unit) and CPU to find a good approximation of the ground energy and ground state of a given Hamiltonian of the form:

$$H=\sum_{i,\alpha}h^i_\alpha\sigma^i_\alpha+\sum_{i,j,\alpha,\beta}h^{ij}_{\alpha\beta}\sigma^i_\alpha\sigma^j_\beta+\cdots$$

where h's are real numbers, and superscripts $i,j,\ldots$ identify the subsystem and subscripts $\alpha,\beta,\ldots\in\{x,y,z\}$ indicate the appropriate Pauli operators σ. The algorithm can be described in four steps:

1) Define a set of ansatz states $|f(\boldsymbol{\theta})\rangle$, which are characterized by parameters $\boldsymbol{\theta}=(\theta_1,\theta_2,\ldots,\theta_n)$ and can be efficiently prepared by a quantum circuit $\mathbf{C}(\boldsymbol{\theta})$. The goal of the algorithm is to find the optimal parameters $\boldsymbol{\theta}_{\min}$ which minimize the energy $\langle f(\boldsymbol{\theta})|H|f(\boldsymbol{\theta})\rangle$. Then $\langle f(\boldsymbol{\theta}_{\min})|H|f(\boldsymbol{\theta}_{\min})\rangle$ and $|f(\boldsymbol{\theta}_{\min})\rangle$ can be set as an approximation of the ground energy and ground state, respectively.
2) Use the QPU to execute the quantum computation represented as quantum circuit $\mathbf{C}(\boldsymbol{\theta})$ in order to generate state $|f(\boldsymbol{\theta})\rangle$ and compute the expectations of $\sigma^i_\alpha,\ \sigma^i_\alpha\sigma^j_\beta,\ \ldots$ in all the terms of H;

3) Use the CPU to sum up the expectations of all the terms of H with the weights h's and thus evaluate $\langle f(\boldsymbol{\theta})|H|f(\boldsymbol{\theta})\rangle$;
4) Feed $\langle f(\boldsymbol{\theta})|H|f(\boldsymbol{\theta})\rangle$ to an classical minimization algorithm. If the optimization is not completed, prepare the parameters $\boldsymbol{\theta}$ for the next round and go to step (2); otherwise, terminate and return $\boldsymbol{\theta}$ as output.

B. VQA in the Tutorial of Cirq

The VQA presented in the tutorial of Google's Cirq [9] deals with a 2D $+/-$ Ising model of size $N \times N$ with objective Hamiltonian (observable)

$$H = \sum_{(i,j)} h_{ij} Z_{ij} + \sum_{(i,j;i',j')\in S} J_{ij;i'j'} Z_{ij} Z_{i'j'},$$

where each index pair (i, j) is associated with a vertex in a the $N \times N$ grid, S is the set of all neighboring vertices in the grid, and all h_{ij} and $J_{ij;i'j'}$ are either $+1$ or -1. The algorithm for preparing the ansatz state with real parameters (α, β, γ) given in the tutorial of Cirq can be rewritten in the quantum-**while** language with $N \times N$ grid of qubits as follows:

$$\begin{aligned}\mathrm{VQA}(N) \equiv\ & \textbf{for } j = 1, \ldots, N \textbf{ do } \mathrm{ProcC}(j) \textbf{ od};\\ & \textbf{for } i = 1, \ldots, N \textbf{ do } \mathrm{ProcR}(i) \textbf{ od}.\end{aligned}$$

Here, subprogram $\mathrm{ProcC}(j)$ acts on the jth column of qubits and $\mathrm{ProcR}(i)$ acts on the ith row of qubits; each of them is a sequential composition of unitary transformations (see the extended version [34] for detailed subprograms).

C. Specifying and Proving Incorrectness in Quantum Separation Logic

As pointed out at the beginning of this section, we can use our quantum separation logic to show that algorithm $\mathrm{VQA}(N)$ is indeed incorrect. Let us first describe its incorrectness in our logical language. Suppose the Hamiltonian H has eigenvalues $E_0, E_1, ...$ ranged in increasing order, with corresponding eigenspaces (projections) $Q_0, Q_1 ....$ If for each $i \leq n$, we can find a precondition $P_i \in \mathcal{P}$ such that $\models \{P_i\}\mathrm{VQA}(N)\{1 - \sum_{k=0}^{i} Q_i\}$ $(i = 0, 1, ..., n)$, then by showing that $|0\rangle$ (the initial state of quantum circuit) is close to P_i; that is, $\langle 0|P_i|0\rangle \geq \delta_i$, we can conclude that the approximate ground energy computed by $\mathrm{VQA}(N)$ is at least:

$$E_0 + \sum_{i=1}^{n} (E_{i+1} - E_i)\delta_i. \tag{4}$$

Therefore, whenever the quantity in (4) is far away from the real ground energy E_0, then $\mathrm{VQA}(N)$ is incorrect.

To illustrate our idea more explicitly, let us consider the simplest case of 2×2 grid ($N = 2$) with parameters:

$$h = \begin{bmatrix} -1 & -1 \\ 1 & 1 \end{bmatrix}, \quad Jc = \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \quad Jr = \begin{bmatrix} -1 & 1 \end{bmatrix}$$

and $J_{ij;(i+1)j} = Jr_{ij}$ and $J_{ij;i(j+1)} = Jc_{ij}$; see Fig. 1 for its circuit model. The eigenvalues of the Hamiltonian H in this case are $E_0, \ldots, E_5 = -6, -4, -2, 0, 2, 4$ with corresponding eigenspaces $Q_0, Q_1, \ldots, Q_5$, respectively. Using QSL, we are able to prove: $\vdash \{P_i\}\mathrm{VQA}(N)\left\{1 - \sum_{k=0}^{i} Q_i\right\}$ for $i = 0, 1$ where

$$\langle 0|P_0|0\rangle = 1 - \frac{1}{16}\sin(\alpha\pi)^4 \geq \frac{15}{16}$$
$$\langle 0|P_1|0\rangle = 1 - \frac{1}{32}(7 + \cos(2\alpha\pi))\sin^2(\alpha\pi) \geq \frac{13}{16},$$

by first reasoning about each subprogram $\mathrm{ProcC}(1)$, $\mathrm{ProcC}(2)$, $\mathrm{Proc(R)}(1)$, $\mathrm{ProcR}(2)$ and then using CONST and UNCR to lift these local reasoning to global correctness above (details can be found in [34]). Then it follows from (4) that the approximate ground energy of VQA is at least -2.5, which is much higher than the real ground energy $E_0 = -6$.

Our quantum separation logic can also apply to higher dimensional versions of this program. In general, since the number of qubits in each subprogram of VQA is $\frac{1}{N}$ of that of the entire system, there is no extra cost for local reasoning no matter how large N is. Besides revealing the incorrectness of ground energy, we can prove that parameters β, γ are helpless for finding the ground energy in the sense that the expectation of measurement outcome[10] is independent of β, γ.

VI. Scalable Reasoning: Verification of Security

A major distinction between classical and quantum information can be stated as the no-cloning theorem that it is impossible to create an identical copy of an arbitrary unknown quantum state. Exploiting this fundamental property among others, many quantum cryptographic protocols with information-theoretical security have been proposed, including quantum key distribution, quantum one-time pad [27], [28] and quantum secret sharing [29], [30].

In this section, we show how quantum separation logic developed in this paper can be used to verify the security of quantum one-time pad and quantum secret sharing. In particular, such verification is scalable in the sense that only a constant computational resource is required in the verification as the length of protocols and the involved qubits increase.

Uniformity is essential in proving the information-theoretical security of many quantum cryptographic protocols. For convenience, let us first present a useful rule:

$$\textsc{FrameU} \quad \frac{\{\top\}\mathbf{C}\{\mathbf{U}[S_1]\} \quad S_2 \cap (\mathsf{var}(\mathbf{C}) \cup S_1) = \emptyset}{\{\mathbf{U}[S_2]\}\mathbf{C}\{\mathbf{U}[S_1 \cup S_2]\}}. \tag{5}$$

This rule is derived by instantiating $\phi \equiv \top$, $\psi \equiv \mathbf{U}[S_1]$ and $\mu \equiv \mathbf{U}[S_2]$ in the frame rule FRAME and using axiom scheme (see Proposition 3 (4)).

A. Security of Quantum One-Time Pad

Let us first verify the security of quantum one-time pad (QOTP) [27], [28], one of the basic quantum encryption schemes in quantum cryptography. Similar to the classical one-time pad, a one-time pre-shared secret key is employed to encrypt and decrypt the quantum data.

[9] https://quantumai.google/cirq/tutorials/variational_algorithm

[10] The QPU executes $\mathrm{VQA}(N)$ and then measures each qubit in computational basis and feed the outcome to CPU.

1) Single-Qubit Case: To warm up, we consider the simplest case for protecting one-qubit data. The QOTP scheme consists of three parts: key generation $\mathbf{KeyGen}$, encryption $\mathbf{Enc}$ and decryption $\mathbf{Dec}$, which can be written as programs:

$$\begin{aligned}\mathbf{KeyGen}[a,b] \equiv\ & a := |0\rangle; b := |0\rangle;\ a := H[a]; b := H[b];\\ & \textbf{if } \mathcal{M}[a,b] = 00 \to \textbf{skip} \square\ 01 \to \textbf{skip}\\ & \quad \square\ 10 \to \textbf{skip} \square\ 11 \to \textbf{skip}\quad \textbf{fi}\\ \mathbf{Enc}[a,b,q] \equiv\ & \textbf{if } \mathcal{M}[a,b] = 00 \to \textbf{skip}\quad \square\ 01 \to q = Z[q]\\ & \square\ 10 \to q = X[q] \square\ 11 \to q = Z[q]; q = X[q]\ \textbf{fi}\\ \mathrm{QOTP}[a,b,q] \equiv\ & \mathbf{KeyGen}[a,b]; \mathbf{Enc}[a,b,q]\end{aligned}$$

Here, registers a and b are used as the secret key, and measurement $\mathcal{M}$ consisting of operators $M_{ij} = |ij\rangle_{ab}\langle ij|$ for $i,j = 0,1$ is introduced to generate and detect the value of secret key, which returns a two-bit classical outcome with a certain probability. Register q is the input quantum data which we want to protect. H is the Hadamard gate and X, Z are Pauli gates as usual.

Security of QOTP for the single-qubit case can be specified as the following uniformity:

$$\vdash \{\top\}\mathrm{QOTP}[a,b,q]\{\mathbf{U}[q]\}. \tag{6}$$

This fact has been formally verified using quantum Hoare logic with ghost variables [38] and relational quantum Hoare logic in [31].

2) Multi-Qubit Case – Scaling Up: Now we show how can the verification for single-qubit be easily scaled up to the multi-qubit case using the frame rule in our quantum separation logic. The protocol for protecting n-qubit data stored in register $\overline{q} = q_1, \ldots, q_n$ can be written as:

$$\mathrm{QOTP}(n) \equiv \textbf{for } i = 1, \ldots, n \textbf{ do } \mathrm{QOTP}[a_i, b_i, q_i] \textbf{ od}$$

where $a_1, b_1, \ldots, a_n, b_n$ are secret key of size $2n$. Its security can be stated as the following uniformity:

$$\vdash \{\top\}\mathrm{QOTP}(n)\{\mathbf{U}[q_1, \ldots, q_n]\}, \tag{7}$$

which shows that, no matter what is the plain text initialised on $\overline{q}$, after encryption, the cipher text is always uniform and the eavesdropper cannot release any useful information. This judgment is proved as follows. First, it follows from (6) that

$$\vdash \{\top\}\mathrm{QOTP}[a_i, b_i, q_i]\{\mathbf{U}[q_i]\} \quad (i = 1, ..., n).$$

Using FRAMEU we obtain for all $i = 1, ..., n$:

$$\vdash \{\mathbf{U}[q_1, \ldots, q_{i-1}]\}\mathrm{QOTP}[a_i, b_i, q_i]\{\mathbf{U}[q_1, \ldots, q_i]\}$$

Then (7) is derived by repeatedly using rule SEQ.

3) Discussion: A comparison between the security verification of QOTP in quantum Hoare logic [31], [38] and in quantum separation logic presented above is interesting. Only the single-qubit case was considered in [38]. A crucial step in the verification for the multi-qubit case given in [31] is based on a complicated transformation of quantum predicates, which cannot be proved by the logic itself, but is derived from a mathematical result proved by quite involved calculations in the previous literature [28]. In contrast, the verification in quantum separation logic avoids such complicated calculations by using the frame rule FRAMEU.

B. Security of Quantum Secret Sharing

Now we turn to verify the security of another quantum cryptographic protocol: quantum secret sharing. Similar to classical secret sharing [40], [41], quantum secret sharing addresses the problem of how to distribute a secret amongst a group of participants so that the secret can be reconstructed by a sufficient number of participants while any individual has no information about it [29], [30]. For concreteness, let us focus on a typical scheme.

1) Quantum $(2,3)$ Threshold Scheme: The $(2,3)$ threshold scheme for sharing a single secret *qutrit* p (a 3-dimensional quantum state) takes p as the input and outputs three qutrits p', q', r' so that each of them has no information about the input secret while any two of them can recover the input. Formally, it can be written as the following program:

$$\mathbf{Enc}[p,q,r] \equiv q := |0\rangle;\ r := |0\rangle;\ p,q,r := U_{\mathrm{enc}}[p,q,r]$$

where unitary transformation U_{enc} maps $|i\rangle|0\rangle|0\rangle$ to $|e_i\rangle$ for $i = 0,1,2$, where $|e_i\rangle$ are three orthonormal states:

$$|e_i\rangle = \frac{1}{\sqrt{3}}\sum_{k=0}^{2}|k\rangle|k \oplus_3 i\rangle|k \oplus_3 2i\rangle$$

where $\oplus_3$ stands for the addition modulo 3. For secretly sharing information of multiple qutrits $\overline{p} = p_1, \ldots, p_n$, this scheme can simply be generalised to:

$$\mathrm{QSS}(n) \equiv \textbf{for } i = 1, \ldots, n \textbf{ do } \mathbf{Enc}[p_i, q_i, r_i] \textbf{ od}.$$

2) Security as Uniformity: Quantum secret sharing is designed against both dishonest agents and eavesdroppers [29], [30], [42]. Let us first consider the case without any eavesdropper during transmission. In this case, the security of $\mathrm{QSS}(n)$ can be specified as the following judgment:

$$\vdash \{\top\}\mathrm{QSS}(n)\{\mathbf{U}[q_1, \ldots, q_n]\}. \tag{8}$$

The above judgment can be easily proved in our quantum separation logic. First, using rules UNIT, INIT and SEQ directly we obtain:

$$\vdash \{\top\}\mathbf{Enc}[p,q,r]\{P_S[p,q,r]\}, \tag{9}$$

where projection $P_S = |e_0\rangle\langle e_0| + |e_1\rangle\langle e_1| + |e_2\rangle\langle e_2|$. It is easy to check that $\models P_S[p,q,r] \to (\mathbf{U}[p] \wedge \mathbf{U}[q] \wedge \mathbf{U}[r])$. Based on this we can conclude: $\vdash \{\top\}\mathbf{Enc}[p,q,r]\{\mathbf{U}[\alpha]\}$ for $\alpha \in \{p,q,r\}$. This proves the security for the case of a single qutrit. To generalise it to the case of multiple qutrits, we can use FRAMEU to derive: $\vdash \{\mathbf{U}[q_1, \ldots, q_{i-1}]\}\mathbf{Enc}[p_i, q_i, r_i]\{\mathbf{U}[q_1, \ldots, q_i]\}$ from $\vdash \{\top\}\mathbf{Enc}[p_i, q_i, r_i]\{\mathbf{U}[q_i]\}$. Then by setting formulas $\phi_i = \mathbf{U}[q_1, \ldots, q_{i-1}]$ and $\phi_1 = \top$, we have $\vdash \{\phi_i\}\mathbf{Enc}[p_i, q_i, r_i]\{\phi_{i+1}\}$ for all $1 \leq i \leq n$, and (8) is obtained by repeatedly using rule SEQ.

VII. DISCUSSION AND RELATED WORK

In this section, we briefly discuss an issue about restriction property left open in Subsection III-C as well as some previous work on verification of quantum programs.

A. Restriction property and BI with domain

Our quantum interpretation of standard BI logic is sufficient for the applications discussed in this paper. However, it has a drawback: the restriction property does not hold for all BI formulas, and thus the assertions in our QSL (Quantum Separation Logic) are confined in a special class of BI formulas (see Def. 8), which do not include implication and separating implication. One possible solution to this issue is to redefine the BI logic so that the restriction property becomes intrinsic – similar to the monotonicity. We can introduce a notion of *domain* into BI: the domain $\mathsf{dom}(x)$ of a *state* x is the set of variables specified by the state. Then a basic idea in classical separation logic [5]–[8], called the domain assumption for stack, can be adopted in defining satisfaction relation: $x \models \phi$ is defined only when $\mathsf{dom}(x) \supseteq \mathsf{free}(\phi)$, where $\mathsf{free}(\phi)$ is the set of free variables in a BI-*formula* ϕ. The domain assumption guarantees that the restriction property is true even when the extension of joint quantum states does not exist (see Sec. III-C). In this way, BI is upgraded to BID (BI with domain), and all BID formulas can be safely used as assertions in QSL. See our extended version [34] for details of this approach.

B. Related work

Quantum programming has become an active research field in recent years after two decades of development [43]. Various analysis, verification, testing and debugging methodologies and techniques for quantum programs have been developed [33], [35], [37], [44]–[53]. In particular, several quantum program logics have been established, including quantum Hoare logic [16], [17], [38] for verifying correctness of one quantum program, and relational quantum Hoare logic [31], [36], [54] for verifying equivalence of two quantum programs. Furthermore, the need of quantum separation logic is also motivated in quantum Hoare type theory [55], [56]; the authors define predicates from [38] to characterize local properties by introducing ghost variables.

The frame rule plays a key role in our QSL. We should mention that a frame rule was also introduced in relational quantum Hoare logic [31], [36], [54]. But it was defined using the ordinary conjunction $\wedge$ and thus is similar to our CONST. The frame rule in QSL is given using the separating conjunction $*$. Of course, the intuitions behind them are the same—an assertion is preserved by a program if it is independent of the program.

Partition of programs/computations is a basic approach to use separation logic; routed quantum circuits (RQC) [57] explores a similar idea in a different line of research, with even more refined structures of partitions, i.e., direct sums of Hilbert spaces. It worth exploring if our QSL can be extended to general RQC dealing with partitions of Hilbert spaces.

The target application of our QSL is verification of large-scale quantum programs, where the size of the representation of assertions and the complexity of the involved calculations can increase exponentially w.r.t the number of qubits. Two different approaches to this issue were proposed in [20] and [25]. They have achieved success, in particular for those large-scale quantum programs with a good algebraic structure that can be inductively defined. It seems that sometimes our QSL can be used in combination with them; for example, some larger VQAs (Variational Quantum Algorithms) can be divided into several blocks, each of which has a good algebraic structure and thus can be verified using the tools developed in [20], [25]. Then our QSL can be employed to lift these local reasoning to the global correctness of VQAs.

VIII. CONCLUSION

In this paper, we have developed a quantum separation logic QSL that enables local reasoning for scalable verification of quantum programs written in a simple quantum programming language, namely the quantum extension of **while**-language. The applicability of QSL has been demonstrated in the formal verification and analysis of several practical quantum algorithms and cryptographic protocols, including a VQA (Variational Quantum Algorithm), quantum one-time pad, and quantum secret sharing.

There are several interesting topics for future research:

(1) We would like to explore more applications of our logic QSL in the verification of those algorithms identified as practical applications of near-term Noisy Intermediate Scale Quantum (NISQ) computers [39]; for example, quantum machine learning from quantum data. We will also try to apply QSL in the security analysis of more quantum cryptographic protocols rather than those considered in this paper, in particular QKD (Quantum Key Distribution).

(2) Currently, QSL can only be used to quantum **while**-programs without indexed variables, like arrays. However, indexed variables has already been frequently used in writing large quantum algorithms. We would like to extend our logic for a more sophisticated quantum program language with indexing. Extending the language with quantum control [58]–[63] and exploring how local reasoning works in these constructs is also a valuable future direction.

(3) Resource theory has been emerging as a subarea of quantum information theory in recent years. Roughly speaking, it aims at understanding how the resources with quantum advantage in computing and communication can be generated and transformed (e.g. only using LOCC (local operations and classical communication)) [64]–[66]. As briefly mentioned in the Introduction, some connections between resource theory [10], [67] and the resource semantics of BI were already noticed in [11]. We would like to see how quantum separation logic can be used to reason about these quantum resources.

ACKNOWLEDGMENT

We thank the anonymous reviewers for their thoughtful comments and feedback. We would like to thank Riling Li for valuable discussions. This work was partially supported by the NSF (#2023222 and #1943130), Facebook, the National Key R&D Program of China (#2018YFA0306701), NSFC (#61832015), ARC Discovery Program (#DP210102449) and ARC DECRA (#DE180100156).

REFERENCES

[1] P. W. O'Hearn and D. J. Pym, "The logic of bunched implications," *The Bulletin of Symbolic Logic*, vol. 5, no. 2, pp. 215–244, 1999. [Online]. Available: https://doi.org/10.2307/421090

[2] D. J. Pym, *The semantics and proof theory of the logic of bunched implications*, ser. Applied Logic Series. Kluwer Academic Publishers, 2002, vol. 26.

[3] D. J. Pym, P. W. O'Hearn, and H. Yang, "Possible worlds and resources: the semantics of BI," *Theoretical Computer Science*, vol. 315, no. 1, pp. 257 – 305, 2004, mathematical Foundations of Programming Semantics. [Online]. Available: https://doi.org/10.1016/j.tcs.2003.11.020

[4] J. Reynolds, "Separation logic: a logic for shared mutable data structures," in *Proceedings 17th Annual IEEE Symposium on Logic in Computer Science*, 2002, pp. 55–74. [Online]. Available: https://doi.org/10.1109/LICS.2002.1029817

[5] P. O'Hearn, J. Reynolds, and H. Yang, "Local reasoning about programs that alter data structures," in *Computer Science Logic*, L. Fribourg, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 1–19. [Online]. Available: https://doi.org/10.1007/3-540-44802-0_1

[6] S. S. Ishtiaq and P. W. O'Hearn, "BI as an assertion language for mutable data structures," in *Proceedings of the 28th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, ser. POPL '01. New York, NY, USA: ACM, 2001, pp. 14–26. [Online]. Available: http://doi.acm.org/10.1145/360204.375719

[7] P. W. O'Hearn, "Resources, concurrency, and local reasoning," *Theoretical Computer Science*, vol. 375, no. 1, pp. 271 – 307, 2007, festschrift for John C. Reynolds's 70th birthday. [Online]. Available: https://doi.org/10.1016/j.tcs.2006.12.035

[8] S. Brookes, "A semantics for concurrent separation logic," *Theoretical Computer Science*, vol. 375, no. 1, pp. 227 – 270, 2007, festschrift for John C. Reynolds's 70th birthday. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0304397506009248

[9] G. Barthe, J. Hsu, and K. Liao, "A probabilistic separation logic," *Proc. ACM Program. Lang.*, vol. 4, no. POPL, Dec. 2019. [Online]. Available: https://doi.org/10.1145/3371123

[10] B. Coecke, T. Fritz, and R. W. Spekkens, "A mathematical theory of resources," *Inf. Comput.*, vol. 250, pp. 59–86, 2016. [Online]. Available: https://doi.org/10.1016/j.ic.2016.02.008

[11] S. R. Docherty, "Bunched logics: a uniform approach," Ph.D. dissertation, UCL (University College London), 2019. [Online]. Available: https://discovery.ucl.ac.uk/id/eprint/10073115/

[12] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, "Quantum machine learning," *Nature*, vol. 549, no. 7671, pp. 195–202, Sep 2017. [Online]. Available: https://doi.org/10.1038/nature23474

[13] M. Broughton, G. Verdon, T. McCourt, A. J. Martinez, J. H. Yoo, S. V. Isakov, P. Massey, M. Y. Niu, R. Halavati, E. Peters, M. Leib, A. Skolik, M. Streif, D. V. Dollen, J. R. McClean, S. Boixo, D. Bacon, A. K. Ho, H. Neven, and M. Mohseni, "TensorFlow Quantum: A software framework for quantum machine learning," 2020. [Online]. Available: https://arxiv.org/abs/2003.02989

[14] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'Brien, "A variational eigenvalue solver on a photonic quantum processor," *Nature Communications*, vol. 5, no. 1, p. 4213, Jul 2014. [Online]. Available: https://doi.org/10.1038/ncomms5213

[15] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik, "The theory of variational hybrid quantum-classical algorithms," *New Journal of Physics*, vol. 18, no. 2, p. 023023, feb 2016. [Online]. Available: https://doi.org/10.1088/1367-2630/18/2/023023

[16] M. Ying, "Floyd–Hoare logic for quantum programs," *ACM Transactions on Programming Languages and Systems (TOPLAS)*, vol. 33, no. 6, pp. 19:1–19:49, 2011. [Online]. Available: https://doi.org/10.1145/2049706.2049708

[17] L. Zhou, N. Yu, and M. Ying, "An applied quantum Hoare logic," in *Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation*, ser. PLDI 2019. New York, NY, USA: Association for Computing Machinery, 2019, p. 1149–1162. [Online]. Available: https://doi.org/10.1145/3314221.3314584

[18] S. Z. Robert Rand, Jennifer Paykin, "QWIRE practice: Formal verification of quantum circuits in Coq," in *14th International Conference on Quantum Physics and Logic 2017*, ser. QPL '17, may 2017. [Online]. Available: https://qpl.science.ru.nl/papers/QPL_2017_paper_45.pdf

[19] K. Hietala, R. Rand, S.-H. Hung, X. Wu, and M. Hicks, "A verified optimizer for quantum circuits," *Proc. ACM Program. Lang.*, vol. 5, no. POPL, Jan. 2021. [Online]. Available: https://doi.org/10.1145/3434318

[20] K. Hietala, R. Rand, S.-H. Hung, L. Li, and M. Hicks, "Proving quantum programs correct," 2020. [Online]. Available: https://arxiv.org/abs/2010.01240

[21] J. C. Reynolds, "An introduction to separation logic (preliminary draft)," *Course notes, October*, 2008. [Online]. Available: http://www.cs.cmu.edu/afs/cs.cmu.edu/user/jcr/www/copenhagen08.pdf

[22] M. A. Nielsen and I. Chuang, *Quantum computation and quantum information*. Cambridge University Press, 2002.

[23] M. Ying, L. Zhou, and Y. Li, "Reasoning about parallel quantum programs," 2018. [Online]. Available: https://arxiv.org/abs/1810.11334

[24] J. Liu, B. Zhan, S. Wang, S. Ying, T. Liu, Y. Li, M. Ying, and N. Zhan, "Formal verification of quantum algorithms using quantum Hoare logic," in *Computer Aided Verification*, I. Dillig and S. Tasiran, Eds. Cham: Springer International Publishing, 2019, pp. 187–207. [Online]. Available: https://doi.org/10.1007/978-3-030-25543-5_12

[25] A. Bordg, H. Lachnitt, and Y. He, "Certified quantum computation in Isabelle/HOL," *Journal of Automated Reasoning*, Dec 2020. [Online]. Available: https://doi.org/10.1007/s10817-020-09584-7

[26] The Cirq Developers, "quantumlib/Cirq: A Python framework for creating, editing, and invoking noisy intermediate scale quantum (NISQ) circuits," 2018. [Online]. Available: https://github.com/quantumlib/Cirq

[27] P. O. Boykin and V. Roychowdhury, "Optimal encryption of quantum bits," *Phys. Rev. A*, vol. 67, p. 042317, Apr 2003. [Online]. Available: https://doi.org/10.1103/PhysRevA.67.042317

[28] M. Mosca, A. Tapp, and R. de Wolf, "Private quantum channels and the cost of randomizing quantum information," *arXiv preprint quant-ph/0003101*, 2000. [Online]. Available: https://arxiv.org/abs/quant-ph/0003101

[29] R. Cleve, D. Gottesman, and H.-K. Lo, "How to share a quantum secret," *Phys. Rev. Lett.*, vol. 83, pp. 648–651, Jul 1999. [Online]. Available: https://doi.org/10.1103/PhysRevLett.83.648

[30] M. Hillery, V. Bužek, and A. Berthiaume, "Quantum secret sharing," *Phys. Rev. A*, vol. 59, pp. 1829–1834, Mar 1999. [Online]. Available: https://doi.org/10.1103/PhysRevA.59.1829

[31] G. Barthe, J. Hsu, M. Ying, N. Yu, and L. Zhou, "Relational proofs for quantum programs," *Proc. ACM Program. Lang.*, vol. 4, no. POPL, Dec. 2019. [Online]. Available: https://doi.org/10.1145/3371089

[32] P. Selinger, "Towards a quantum programming language," *Mathematical Structures in Computer Science*, vol. 14, no. 4, pp. 527–586, 2004. [Online]. Available: https://doi.org/10.1017/S0960129504004256

[33] M. Ying, *Foundations of Quantum Programming*. Morgan Kaufmann, 2016.

[34] L. Zhou, G. Barthe, J. Hsu, M. Ying, and N. Yu, "A quantum interpretation of bunched logic for quantum separation logic," 2021. [Online]. Available: https://arxiv.org/abs/2102.00329

[35] M. Ying, R. Duan, Y. Feng, and Z. Ji, "Predicate transformer semantics of quantum programs," *Semantic Techniques in Quantum Computation*, no. 8, pp. 311–360, 2010.

[36] D. Unruh, "Quantum relational Hoare logic," *Proc. ACM Program. Lang.*, vol. 3, no. POPL, Jan. 2019. [Online]. Available: https://doi.org/10.1145/3290346

[37] N. Yu and J. Palsberg, "Quantum abstract interpretation," in *Proceedings of the 42th ACM SIGPLAN Conference on Programming Language Design and Implementation*, ser. PLDI 2021. New York, NY, USA: Association for Computing Machinery, 2021.

[38] D. Unruh, "Quantum Hoare logic with ghost variables," in *2019 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, 2019, pp. 1–13. [Online]. Available: https://doi.org/10.1109/LICS.2019.8785779

[39] J. Preskill, "Quantum computing in the NISQ era and beyond," *Quantum*, vol. 2, p. 79, Aug. 2018. [Online]. Available: https://doi.org/10.22331/q-2018-08-06-79

[40] G. R. Blakley, "Safeguarding cryptographic keys," in *Managing Requirements Knowledge, International Workshop on*. Los Alamitos, CA, USA: IEEE Computer Society, jun 1979, p. 313. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/AFIPS.1979.98

[41] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, p. 612–613, Nov. 1979. [Online]. Available: https://doi.org/10.1145/359168.359176

[42] A. Karlsson, M. Koashi, and N. Imoto, "Quantum entanglement for secret sharing and secret splitting," *Phys. Rev. A*, vol. 59, pp. 162–168, Jan 1999. [Online]. Available: https://doi.org/10.1103/PhysRevA.59.162

[43] B. Heim, M. Soeken, S. Marshall, C. Granade, M. Roetteler, A. Geller, M. Troyer, and K. Svore, "Quantum programming languages," *Nature Reviews Physics*, vol. 2, no. 12, pp. 709–722, Dec 2020. [Online]. Available: https://doi.org/10.1038/s42254-020-00245-7

[44] D. Akatov, "The logic of quantum program verification," Master's thesis, Oxford University Computing Laboratory, 2005. [Online]. Available: http://www.academia.edu/download/7563948/thesis-1.1.ps

[45] E. D'hondt and P. Panangaden, "Quantum weakest preconditions," *Mathematical Structures in Computer Science*, vol. 16, no. 3, pp. 429–451, 2006. [Online]. Available: https://doi.org/10.1017/S0960129506005251

[46] A. Baltag and S. Smets, "The logic of quantum programs," in *Proceedings of the 2nd International Workshop on Quantum Programming Languages (QPL 2004)*, P. Selinger, Ed., 2004, pp. 39–56. [Online]. Available: https://www.mathstat.dal.ca/~selinger/qpl2004/PDFS/04Baltag-Smets.pdf

[47] ——, "LQP: the dynamic logic of quantum information," *Mathematical Structures in Computer Science*, vol. 16, no. 3, pp. 491–525, 2006. [Online]. Available: https://doi.org/10.1017/S0960129506005299

[48] O. Brunet and P. Jorrand, "Dynamic quantum logic for quantum programs," *International Journal of Quantum Information*, vol. 02, no. 01, pp. 45–54, 2004. [Online]. Available: https://doi.org/10.1142/S0219749904000067

[49] R. Chadha, P. Mateus, and A. Sernadas, "Reasoning about imperative quantum programs," *Electronic Notes in Theoretical Computer Science*, vol. 158, pp. 19–39, 2006. [Online]. Available: https://doi.org/10.1016/j.entcs.2006.04.003

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

[51] R. Rand, "Verification logics for quantum programs," 2016. [Online]. Available: http://www.cs.umd.edu/~rrand/wpe.pdf

[52] Y. Feng, R. Duan, Z. Ji, and M. Ying, "Proof rules for the correctness of quantum programs," *Theoretical Computer Science*, vol. 386, no. 1-2, pp. 151–166, 2007. [Online]. Available: https://doi.org/10.1016/j.tcs.2007.06.011

[53] N. Yu, "Quantum temporal logic," 2019. [Online]. Available: https://arxiv.org/abs/1908.00158

[54] Y. Li and D. Unruh, "Quantum relational Hoare logic with expectations," 2019.

[55] K. Singhal and J. Reppy, "Quantum Hoare type theory: Extended abstract," in *17th International Conference on Quantum Physics and Logic 2020*, ser. QPL '20, may 2020. [Online]. Available: http://ks.cs.uchicago.edu/publication/qhtt/

[56] K. Singhal, "Quantum Hoare type theory," 2020. [Online]. Available: https://arxiv.org/abs/2012.02154

[57] A. Vanrietvelde, H. Kristjánsson, and J. Barrett, "Routed quantum circuits," 2020. [Online]. Available: https://arxiv.org/abs/2011.08120

[58] T. Altenkirch and J. Grattage, "A functional quantum programming language," in *Proceedings of the 20th Annual IEEE Symposium on Logic in Computer Science (LICS' 05)*. IEEE, 2005, pp. 249–258. [Online]. Available: https://doi.org/10.1109/LICS.2005.1

[59] M. Ying, N. Yu, and Y. Feng, "Defining quantum control flow," 2012. [Online]. Available: https://arxiv.org/abs/1209.4379

[60] ——, "Alternation in quantum programming: From superposition of data to superposition of programs," 2014. [Online]. Available: https://arxiv.org/abs/1402.5172

[61] C. Bădescu and P. Panangaden, "Quantum alternation: Prospects and problems," in Proceedings of the 12th International Workshop on *Quantum Physics and Logic,* Oxford, U.K., July 15-17, 2015, ser. Electronic Proceedings in Theoretical Computer Science, C. Heunen, P. Selinger, and J. Vicary, Eds., vol. 195. Open Publishing Association, 2015, pp. 33–42. [Online]. Available: https://doi.org/10.4204/EPTCS.195.3

[62] Z. Gavorová, M. Seidel, and Y. Touati, "Topological obstructions to implementing controlled unknown unitaries," 2020. [Online]. Available: https://arxiv.org/abs/2011.10031

[63] P. Andrés-Martínez and C. Heunen, "Weakly measured while loops: peeking at quantum states," 2021. [Online]. Available: https://arxiv.org/abs/2009.08832

[64] M. Horodecki and J. Oppenheim, "(Quantumness in the context of) Resource theories," *International Journal of Modern Physics B*, vol. 27, no. 01n03, p. 1345019, 2013. [Online]. Available: https://doi.org/10.1142/S0217979213450197

[65] M. B. Plenio and S. S. Virmani, *An Introduction to Entanglement Theory*. Cham: Springer International Publishing, 2014, pp. 173–209. [Online]. Available: https://doi.org/10.1007/978-3-319-04063-9_8

[66] V. Veitch, S. A. H. Mousavian, D. Gottesman, and J. Emerson, "The resource theory of stabilizer quantum computation," *New Journal of Physics*, vol. 16, no. 1, p. 013009, jan 2014. [Online]. Available: https://doi.org/10.1088/1367-2630/16/1/013009

[67] T. Fritz, "Resource convertibility and ordered commutative monoids," *Mathematical Structures in Computer Science*, vol. 27, no. 6, p. 850–938, 2017. [Online]. Available: https://doi.org/10.1017/S0960129515000444