SAQR-QC: A Logic for Scalable but Approximate Quantitative Reasoning about Quantum Circuits

NENGKUN YU, Stony Brook University, USA
JENS PALSBERG, University of California at Los Angeles, USA
THOMAS REPS, University of Wisconsin-Madison, USA

Reasoning about quantum programs remains a fundamental challenge, regardless of the programming model or computational paradigm. Existing verification techniques are insufficient—even for quantum circuits, a deliberately restricted model that lacks classical control, but still underpins many current quantum algorithms. Many existing formal methods require exponential time and space to represent and manipulate (representations of) assertions and judgments, making them impractical for quantum circuits with many qubits. This paper presents SAQR-QC, a logic for **S**calable but **A**pproximate **Q**uantitative **R**easoning about **Q**uantum **C**ircuits. SAQR-QC has three characteristics: (i) some deliberate loss of precision is built into it; (ii) it has a mechanism to help the accumulated loss of precision during a sequence of reasoning steps remain small; and (iii) every reasoning step is local—involving just a small number of qubits—making reasoning scalable. We demonstrate the effectiveness of SAQR-QC via two case studies: the verification of GHZ circuits involving non-Clifford gates, and the analysis of quantum phase estimation—a core subroutine in Shor's factoring algorithm.

CCS Concepts: • **Theory of computation → Quantum computation theory**; **Logic and verification**.

Additional Key Words and Phrases: Verification, quantitative reasoning for quantum circuits

ACM Reference Format:
Nengkun Yu, Jens Palsberg, and Thomas Reps. 2026. SAQR-QC: A Logic for Scalable but Approximate Quantitative Reasoning about Quantum Circuits. *Proc. ACM Program. Lang.* 10, PLDI, Article 206 (June 2026), 33 pages. https://doi.org/10.1145/3808284

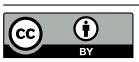
1 Introduction

Quantum computing leverages superposition and interference to achieve computational advantages over classical methods for specific tasks. For instance, Shor's algorithm [40] efficiently factors large integers using the Quantum Fourier Transform (QFT) and Quantum Phase Estimation (QPE) [26]. QPE is also a foundational component in quantum-simulation algorithms [6] and in quantum algorithms for solving linear-algebra problems [23].

Guaranteeing that quantum programs behave as intended is a fundamental challenge. This challenge has spurred significant research into verifying quantum programs using classical computers [5, 7, 14, 41, 42, 45, 48–50]. A landmark achievement is Ying's Quantum Hoare Logic (QHL) [46], which extends classical Hoare logic to reason about the correctness of quantum programs. QHL uses quantum predicates—semidefinite positive Hermitian operators—as pre- and post-conditions, with a quantum Hoare triple [18] defined as:

$$\{A\}\ \mathbf{C}\ \{B\} \quad \text{iff} \quad \text{for all input state } \rho, \mathrm{Tr}(A\rho) \le \mathrm{Tr}(B[\![\mathbf{C}]\!](\rho)). \tag{1}$$

Authors' Contact Information: Nengkun Yu, Stony Brook University, Stony Brook, USA, nengkun.yu@cs.stonybrook.edu; Jens Palsberg, University of California at Los Angeles, Los Angeles, USA, palsberg@ucla.edu; Thomas Reps, University of Wisconsin-Madison, Madison, USA, reps@cs.wisc.edu.



ACM 2475-1421/2026/6-ART206
https://doi.org/10.1145/3808284

Here, $\mathrm{Tr}(A\rho)$ represents the probability (or expected value) that the input state ρ satisfies the predicate A, while $\mathrm{Tr}\big(B\,[\![\mathbf{C}]\!](\rho)\big)$ represents the probability that the output state satisfies B after executing C. When $\{A\}\ \mathbf{C}\ \{B\}$ holds, the program C ensures that the probability of satisfying the postcondition B is at least as great as the probability of satisfying the precondition A.

Despite the development of QHL, scalable verification remains elusive, even for quantum circuits—a restricted model without classical control that encapsulates many current algorithms. Existing methods struggle with exponential complexity; for instance, while QHL is complete for unitary programs, its application requires computing $U_{\mathbf{C}}^{\dagger} B U_{\mathbf{C}}$ for a general n-qubit unitary $U_{\mathbf{C}}$, a task requiring time and space exponential in n. This intractability motivates our central question:

> *Can we establish a theoretical foundation for* ***scalable quantitative reasoning*** *about quantum programs that use many qubits?*

Here, "scalable" means that the size of a proof—the matrices involved and the logical derivation itself—grows polynomially in the number of qubits.

This paper answers this question affirmatively by introducing SAQR-QC (**S**calable but **A**pproximate **Q**uantitative **R**easoning for **Q**uantum **C**ircuits), a logic for reasoning about programs expressed as quantum circuits. SAQR-QC is designed with three principles: (i) it embraces a deliberate, controlled loss of precision to achieve scalability; (ii) it incorporates mechanisms to keep the accumulated imprecision small across reasoning steps; (iii) all reasoning is *local*, meaning each step involves only a constant number of qubits, independent of the total system size.

SAQR-QC is inspired by two prior approaches that sit at opposite ends of a spectrum. At one end, QHL—see Equation (1)—offers fully precise quantitative reasoning, but is computationally intractable for large systems. At the other end, Quantum Abstract Interpretation (QAI) [48] provides a scalable, qualitative framework. A QAI judgment,

$$\models^{\mathrm{QAI}} \{\mathcal{P}\}\ \mathbf{C}\ \{\mathcal{Q}\} \quad \text{iff} \quad \text{forall } \rho,\ \rho \vDash \mathcal{P} \text{ implies } [\![\mathbf{C}]\!](\rho) \vDash \mathcal{Q},$$

uses tuples of local projections $\mathcal{P}$ and $\mathcal{Q}$ as abstract states. QAI enables efficient, local reasoning but lacks the ability to reason about quantitative properties like success probabilities.

SAQR-QC integrates ideas from both to achieve scalable quantitative reasoning. We wish to stress that SAQR-QC is a *logic for manual proof construction*, analogous to how Hoare Logic, Separation Logic, and Linear Logic were first developed as conceptual frameworks to facilitate human-driven reasoning. Automation, while a vital future direction, is not the aim of this paper.

Each assertion that appears in a SAQR-QC judgment has a two-part structure, e.g., $\{\mathscr{A} \mid \mathcal{P}\}$, inspired by the spatial and pure assertions of separation logic [8, 19]. $\mathscr{A}$ is a tuple of *local observables*, each acting non-trivially on only a constant number of qubits (i.e., the number does not grow as the circuit size increases). These observables track quantitative information, such as success probabilities—similar to QHL—but restricted to local components. $\mathcal{P}$ is a QAI-style tuple of local projections, capturing qualitative spatial constraints on the state. In essence, $\{\mathscr{A} \mid \mathcal{P}\}$ gives us a way of specifying a collection of quantum states via a tuple of local observables and a tuple of local projections. SAQR-QC uses judgments of the form

$$\{\mathscr{A} \mid \mathcal{P}\}\ \mathbf{C}\ \{\mathscr{B} \mid \mathcal{Q}\}, \tag{2}$$

which asserts that for any input state ρ satisfying $\mathcal{P}$, (i) the output state $[\![\mathbf{C}]\!](\rho)$ satisfies $\mathcal{Q}$, and (ii) a certain quantitative relationship involving $\mathscr{A}$ and $\mathscr{B}$ holds. More precisely, judgment (2) means

$$\begin{aligned} \models^{\text{SAQR-QC}} \{\mathscr{A} \mid \mathcal{P}\}\ \mathbf{C}\ \{\mathscr{B} \mid \mathcal{Q}\} \quad \text{iff} \quad & \text{for all } \rho,\ \rho \vDash \mathcal{P} \text{ implies (i) } [\![\mathbf{C}]\!](\rho) \vDash \mathcal{Q}, \text{ and} \\ & \text{(ii) } \mathrm{Tr}(M_{\mathscr{A}}\rho) \le \mathrm{Tr}(M_{\mathscr{B}}[\![\mathbf{C}]\!](\rho)), \end{aligned} \tag{3}$$

where $M_{\mathscr{A}}$ and $M_{\mathscr{B}}$ are derived from $\mathscr{A}$ and $\mathscr{B}$, with the precise definition provided in §3. Here, $\mathrm{Tr}(M_{\mathscr{A}}\rho)$ denotes the total weight (or expectation value) with which the input state ρ satisfies the local observables specified by $\mathscr{A}$, while $\mathrm{Tr}(M_{\mathscr{B}}[\![\mathbf{C}]\!](\rho))$ denotes the corresponding weight for the output state $[\![\mathbf{C}]\!](\rho)$ with respect to $\mathscr{B}$. When $\{\mathscr{A} \mid \mathcal{P}\}\ \mathbf{C}\ \{\mathscr{B} \mid \mathcal{Q}\}$ holds, we restrict our attention to input states that satisfy $\mathcal{P}$—i.e., $\rho \models \mathcal{P}$—and on those states, the program $\mathbf{C}$ ensures that the output state $[\![\mathbf{C}]\!](\rho)$ satisfies $\mathcal{Q}$ and that the degree of satisfaction of $\mathscr{B}$ is at least that of $\mathscr{A}$.

This two-part structure is not merely stylistic; it is essential under the locality constraint. Without locality, the two components could be combined into a single QHL triple [50, Theorems 3.2 and 3.3]. However, with the restriction to local projections and observables, this combination is generally impossible, justifying the distinct roles of the QAI-like and QHL-like components. Their interaction—inspired by the reduced product in abstract interpretation [16, §10.1], as well as the use of abstract interpretation in logic-based tools [17]—helps to control the accumulation of imprecision.

Overview. The form of Equation (2) is motivated by the need for scalable reasoning. A logic for quantum computing must support sequences of reasoning steps. For instance, for sequential composition—i.e., for input state ρ_0, $[\![\mathbf{C}_1;\mathbf{C}_2]\!](\rho_0) = [\![\mathbf{C}_2]\!]([\![\mathbf{C}_1]\!](\rho_0))$—we aim to derive a judgment $\{\mathscr{A}\}\ \mathbf{C}_1;\mathbf{C}_2\ \{\mathscr{C}\}$ from (derivations of) judgments of the form $\{\mathscr{A}\}\ \mathbf{C}_1\ \{\mathscr{B}\}$ and $\{\mathscr{B}\}\ \mathbf{C}_2\ \{\mathscr{C}\}$. To ensure that we can bound the sizes of $\mathscr{A}$, $\mathscr{B}$, and $\mathscr{C}$ (where $\mathscr{B}$ describes the set of intermediate states $\{[\![\mathbf{C}_1]\!](\rho_0) \mid \rho_0 \models \mathscr{A}\}$, and $\mathscr{C}$ describes $\{[\![\mathbf{C}_2]\!](\rho) \mid \rho \models \mathscr{B}\}$), we must use a less-general language than that permitted in QHL: we require the logic to use constraints that have two properties: (i) they are efficiently representable, and (ii) satisfaction ($\rho \models \varphi$) is efficiently computable.

Let $\mathscr{A} := (A_{s_1}, \cdots, A_{s_m})$ and $\mathscr{B} := (B_{s_1}, \cdots, B_{s_m})$ be tuples of local observables, where each A_{s_i} and B_{s_i} acts only on a subset of qubits $s_i \subseteq \{1, 2, \cdots, n\}$. The tuple $(s_1, \cdots, s_m)$ is fixed in advance, and typically each s_i is chosen independently of the total number of qubits n. This choice defines a class of observables efficiently represented by structures, such as vectors and matrices, whose size grows at most polynomially in the number of qubits: reasoning about a sub-circuit that acts only on qubits 1 and 2 need only concern those qubits, not the full state space.

In §3, Theorem 3.1 shows that merely restricting QHL to use local observables yields overly imprecise reasoning, which motivates combining local observables $\mathscr{A}$ with the local projections $\mathcal{P}$ from QAI—giving rise to predicates of the form $\{\mathscr{A} \mid \mathcal{P}\}$ and to triples as in Equation (2).

A primary use case of SAQR-QC is forward reasoning: given pre-state assertion $\{\mathscr{A} \mid \mathcal{P}\}$ and circuit $\mathbf{C}$, derive a post-state assertion $\{\mathscr{B} \mid \mathcal{Q}\}$. Exploiting the compositional nature of quantum circuits, and noting that any circuit can be decomposed into two-qubit unitary operations, it suffices for SAQR-QC to handle the case where $\mathbf{C}$ is a two-qubit unitary U. In this setting, the derivation of $\mathscr{B}$ from a given $\mathscr{A}$ reduces to updating only those local observables in $\mathscr{A}$ that are relevant to the two qubits on which U acts. $\mathcal{Q}$ is derived from $\mathcal{P}$ by the local-reasoning rules of QAI. In principle, $\mathscr{B}$ can also be influenced by $\mathcal{P}$; such "cross-talk" can be seen in our example Theorem 3.1. The resulting post-condition may not be the strongest possible; however, it is designed to be sufficiently informative to extract useful quantitative properties, such as the success probability of a computation in the BQP model.[1]

Limitations. A fundamental limitation of SAQR-QC concerns expressivity. Its assertion language is based on tuples of local observables and tuples of local projections, making it well suited to reasoning about properties that can be decomposed into local components. Consequently, global properties that inherently depend on non-local correlations may not be directly expressible.

Compared with QHL, which allows arbitrary Hermitian operators as predicates, SAQR-QC is strictly less expressive: QHL can specify global properties of quantum states, including those

[1]BQP is the class of decision problems solvable by a quantum computer in polynomial time with bounded error, analogous to the class P for classical deterministic computation.

involving entanglement across an unbounded number of qubits, whereas SAQR-QC is restricted to properties that can be captured via fixed collections of local observables and local projections. This restriction is essential for scalability, but comes at the cost of expressivity.

On the other hand, SAQR-QC is more expressive than existing instances of QAI, which rely on coarse abstract domains. In particular, SAQR-QC can capture quantitative relationships among multiple observables and track their evolution through circuit transformations, enabling reasoning about properties such as success probabilities in BQP computations. Such properties are typically beyond the reach of standard QAI domains, which focus on qualitative properties.

More broadly, the expressivity of SAQR-QC is closely related to the question of how global quantum properties can be reconstructed from local information. This connection suggests that techniques from abstract interpretation—particularly results on the inherent limits of abstract domains and their reduced products—may offer useful perspectives. At the same time, the quantum setting involves fundamentally different mathematical structures, and it is not yet clear to what extent these classical insights carry over. We view this as an important issue that warrants further investigation in the development of SAQR-QC. We discuss how to address "failures" (i.e., from overly loose approximations) in §4.2.

As with any manual logic, constructing a SAQR-QC proof requires insight, particularly in choosing the appropriate local projections and observables. The logic provides the framework, but the user must supply the clever insights—the "eureka" steps—to define the predicates that make the verification possible. The case studies in §4 and §5 exemplify this principle, demonstrating how specific, carefully chosen predicates yield sharp results for non-trivial quantum circuits.

Contributions. This paper lays a foundation for scalable reasoning about quantum computations.

- We present SAQR-QC, which integrates quantitative, QHL-like reasoning with the scalable, local reasoning of QAI. The design of SAQR-QC ensures that all reasoning steps are local, leading to proofs whose size is polynomial in the number of qubits and gates.
- We provide a formal foundation for SAQR-QC, defining its assertions and proof rules.
- We demonstrate SAQR-QC's utility and expressiveness through two detailed case studies:
  - *Verification of GHZ circuits involving non-Clifford gates*: We derive a judgment that characterizes the output state precisely in every aspect except a relative phase factor.
  - *Analysis of QPE*: We derive a judgment showing that for any constant k, the QPE algorithm provides the best estimate of the last k bits of the phase, with probability at least $4/\pi^2$. A key step is a novel, lossless local-reasoning method for the QFT using QAI. To the best of our knowledge, no prior approach has achieved such a result with a proof that scales polynomially with the system size.
- We clarify the relationship between SAQR-QC and prior work, particularly Zhou et al. [50], and discuss the principles for constructing effective predicates for use in our framework.

Organization. §2 presents background material about quantum computing, as well as the basics of QAI. §3 defines local observables as predicates, and formalizes the judgments of SAQR-QC, which integrate local observables and QAI. We employ SAQR-QC to reason about the general GHZ circuit (§4) and quantum phase estimation (§5). §6 discusses related work. §7 concludes the paper and outlines future work. Some proofs and derivations are available in Appendices A-G.

2 Background & Notation

To make the paper self-contained, this section briefly reviews basic quantum computing concepts and QAI (following [31, 46, 48]); some readers may wish to skip directly to §3.

Preliminaries. We write $[n] = \{1, \ldots, n\}$, $\setminus$ for set difference, and $|s|$ for the cardinality of a set s. We assume familiarity with Dirac notation, $|\cdot\rangle$, and standard linear-algebra concepts, including Hilbert spaces, tensor products, orthonormal bases, and inner/outer products.

Linear *operators* on d-dimensional complex vector spaces are represented by $d \times d$ matrices $\mathbb{C}^{d\times d}$. The identity is I, and the conjugate transpose of A is $A^\dagger = (A^T)^*$. An operator is *Hermitian* if $A = A^\dagger$ and *positive semi-definite* if all eigenvalues are nonnegative; its trace is $\mathrm{Tr}(A) = \sum_i A_{ii}$.

The *Löwner order* on Hermitian matrices, $A \le B$ if $B - A$ is positive semidefinite, is fundamental for comparing states and operators in quantum mechanics.

Quantum States. A *quantum state* describes the state of a quantum system. A single qubit *pure* state $|\psi\rangle$ lies in a two-dimensional Hilbert space as a superposition of $|0\rangle$ and $|1\rangle$. An n-qubit system resides in a 2^n-dimensional space, allowing complex superpositions and entanglement. *Mixed states*, represented by a density matrix ρ, generalize pure states to probabilistic mixtures.

Reduced Density Matrices. Reduced density matrices are central to analyzing multipartite quantum systems, as many properties depend solely on subsystem reductions. In quantum computation, for example, the success probability of algorithms like HHL [23] depends only on the reduced density matrix of the "signal" qubit. Measuring this qubit, success (outcome $|1\rangle$) depends only on its reduced state, independent of global entanglement or structure.

Let $\mathbb{C}^{d_1}$ and $\mathbb{C}^{d_2}$ be the Hilbert spaces of two quantum systems. The composite system lives in $\mathbb{C}^{d_1} \otimes \mathbb{C}^{d_2}$, and analyzing subsystems uses the *partial trace*. The partial trace over $\mathbb{C}^{d_1}$, $\mathrm{Tr}_1(\cdot)$, maps operators on $\mathbb{C}^{d_1} \otimes \mathbb{C}^{d_2}$ to $\mathbb{C}^{d_2}$: $\mathrm{Tr}_1\big(|\varphi_1\rangle\langle\psi_1| \otimes |\varphi_2\rangle\langle\psi_2|\big) = \langle\psi_1|\varphi_1\rangle\ |\varphi_2\rangle\langle\psi_2|$, for all $|\varphi_1\rangle, |\psi_1\rangle \in \mathbb{C}^{d_1}$ and $|\varphi_2\rangle, |\psi_2\rangle \in \mathbb{C}^{d_2}$, extended linearly. Similarly, $\mathrm{Tr}_2(\cdot)$ traces out $\mathbb{C}^{d_2}$.

For an n-qubit system and $s \subseteq [n]$, the reduced density matrix is $\rho_s = \mathrm{Tr}_{[n]\setminus s}(\rho)$, with $\mathrm{Tr}_{[n]\setminus s}$ tracing out all qubits not in s. The partial trace preserves positive semi-definiteness [31].

Unitary Operations. Unitary operations, represented by matrices U with $U^\dagger U = I$, preserve the norm of quantum states and are fundamental for manipulating them and implementing algorithms. They act on pure states as $|\psi\rangle \mapsto U|\psi\rangle$ and on density operators as $\rho \mapsto U\rho U^\dagger$. Commonly used single-qubit operators include the Pauli gates I, X, Y, and Z; the Hadamard gate H; the T gate; the family of gates $\{ R_m \mid m \in \mathbb{N} \}$. Commonly used two-qubit gates include the SWAP operation SWAP and the controlled-NOT operation CNOT.

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

$$T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix} \qquad R_m = \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^m} \end{pmatrix} \qquad SWAP = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Observables. Quantum observables represent measurable quantities, such as position, energy, or spin, and are modeled by *Hermitian* operators O that satisfy $O^\dagger = O$, which ensures real-valued measurement outcomes. In contexts like QHL, observables are often restricted to $0 \le O \le I$ in the *Löwner order*, meaning both O and $I - O$ are positive semidefinite. These observables serve as predicates or quantum effects, capturing partial truth values in program verification.

Quantum Circuits and Semantics. We consider quantum programs, represented as circuits on n qubits, composed of p unitary gates $U_{f_1}, \ldots, U_{f_p}$, where each U_{f_ℓ} acts on a subset of qubits $f_\ell \subseteq [n]$. The program starts in $|0\rangle^{\otimes n} = |0^n\rangle$, and its semantics is given by $U_{f_p} \cdots U_{f_1} |0^n\rangle$.

Each gate is lifted to an n-qubit unitary by tensoring with identities. For a single-qubit gate U acting on qubit i, $U \otimes I_{[n]\setminus\{i\}} := (\otimes_{k>i} I) \otimes U \otimes (\otimes_{0<j<i} I)$, and similarly for two-qubit gates U_{f_ℓ} on $f_\ell = \{i, j\}$, $U \otimes I_{[n]\setminus\{i,j\}}$, with placement determined by the qubit indices. For clarity, we describe a program's semantics assuming 2-qubit gates, although all results extend to gates acting on up to m qubits for any constant m.

Definition 2.1 (Syntax). The syntax of quantum programs is given by

$$\mathbf{C} ::= \mathbf{skip} \mid \bar{q} := U[\bar{q}] \mid \mathbf{C}_1; \mathbf{C}_2$$

We write $[\![\mathbf{C}]\!]$ to denote the semantics of a quantum program $\mathbf{C}$. If $\mathbf{C}$ represents a unitary transformation U_C, then for any input density matrix ρ, its semantics is given by $[\![\mathbf{C}]\!](\rho) := U_C \rho U_C^\dagger$. We also define the dual action on observable A as: $[\![\mathbf{C}]\!]^*(A) := U_C^\dagger A U_C$. We define $\boldsymbol{\mathcal{U}} := \lambda x. UxU^\dagger$ to denote the quantum operation on density matrices that corresponds to the unitary matrix U (which operates on quantum states). That is, $\boldsymbol{\mathcal{U}}\rho = \lambda x. UxU^\dagger \rho = U\rho U^\dagger$.

Projections. An orthogonal projection satisfies $P = P^\dagger = P^2$, a stronger condition than the classical $P = P^2$. We simply call such matrices *projections*. Each projection P corresponds to a subspace $S_P = \{v \mid Pv = v\}$, and we use "projection" and "subspace" interchangeably. Viewing projections as subspaces induces a partial order: $P \subseteq Q$ iff $S_P \subseteq S_Q$. For example, $|00\rangle\langle 00| + |11\rangle\langle 11|$ is a rank-2 projection onto a 2D subspace of $\mathbb{C}^4$.

Projections are positive semidefinite. The support of a positive semidefinite matrix A, $\mathrm{supp}(A)$, is the span of eigenvectors with nonzero eigenvalues. A density matrix ρ *satisfies* P, written $\rho \vDash P$, if $\mathrm{supp}(\rho) \subseteq P$, equivalently $P\rho = \rho$ [10].

Lemmas. Our development of SAQR-QC relies on three fundamental operations on operators:

- **Löwner order** of operators (denoted $A \leq B$),
- **Partial trace and trace operators** (denoted Tr_s and Tr, where Tr_s traces out subsystem s),
- **Expansion** of an operator via tensor product (denoted $A_s \otimes I_{[n]\setminus s}$, where the operator A_s acts on subsystem s and is expanded to the full system).

The following lemmas summarize key algebraic relationships among these operations, forming the basis for the correctness of our SAQR-QC framework. Unless stated otherwise, let $s \subseteq [n]$; P be a projection on an n-qubit system; A, B be positive semidefinite matrices; E a matrix; and ρ a quantum state. The proof of Lemma 2.1 is given in §A. The other lemmas can be proven using the definition of trace, partial trace, and support.

The relevance of these lemmas to SAQR-QC is that (i) a positive semidefinite matrix A serves as a predicate on a density matrix ρ via the expectation value $\mathrm{Tr}(A\rho)$; and (ii) a subclass of such predicates with good locality properties are ones defined by a positive semidefinite matrix A_s that only acts on a qubit set $s \subseteq [n]$, producing the expectation value $\mathrm{Tr}\left((A_s \otimes I_{[n]\setminus s})\rho\right) = \mathrm{Tr}\left(A_s \rho_s\right)$.

Lemma 2.1. *Let ρ be the density matrix of an n-qubit system, and let $s \subseteq [n]$. Then for any observable A_s acting on subsystem s,* $\mathrm{Tr}\left((A_s \otimes I_{[n]\setminus s})\rho\right) = \mathrm{Tr}\left(A_s \rho_s\right)$.

Lemma 2.2. *For two square matrices B and E of the same size,* $\mathrm{Tr}(BE) = \mathrm{Tr}(EB)$.

Lemma 2.3. *For $A, B \geq 0$,* $\mathrm{Tr}(AB) \geq 0$.

Lemma 2.4. *If $A \leq B$, then (i)* $\mathrm{Tr}(A\rho) \leq \mathrm{Tr}(B\rho)$ *for any density operator ρ, and in particular* $\mathrm{Tr}(A) \leq \mathrm{Tr}(B)$*; (ii) $PAP^\dagger \leq PBP^\dagger$ for any operator P.*

Lemma 2.5. *For a density matrix ρ and a projection P, we have*

$$\mathrm{Tr}(P\rho) = 1 \quad \Leftrightarrow \quad \mathrm{supp}(\rho) \subseteq P \quad \Leftrightarrow \quad \rho \vDash P.$$

Qualitative Predicates for Local Reasoning: Quantum Abstract Interpretation. The qualitative predicates used in QAI express logical properties of quantum states, e.g., whether a subsystem lies in a subspace. Represented as tuples of local projectors, they support scalable local reasoning, inferring global behavior from partial views. QAI [48] propagates these predicates through circuits using small-subsystem projectors, avoiding exponential full-state analysis.

This section reviews local projective predicates, formalizes their semantics, and describes their transformation under unitaries. Soundness is established via support-based semantics and partial trace, forming the basis for abstract reasoning about quantum programs. "Locality" means that we work with a tuple of sets $(s_1, \ldots, s_m)$, where each $s_i \subseteq [n]$ is a small subset of bounded size.

Definition 2.1 ([48]). *A tuple $(P_{s_1}, \cdots, P_{s_m})$ is called a* projective predicate *if each P_{s_i} is a projection, i.e., $P_{s_i}^2 = P_{s_i}$. We use $\mathcal{P}$ (or $\mathcal{Q}, \mathcal{R}$) to denote projective predicates. In particular, we write $\mathcal{I} := (I_{s_1}, \cdots, I_{s_m})$ to represent the identity predicate.*

Remark. Projective predicates are also referred to as *abstract states*.

Definition 2.2 ([48]). *A state ρ satisfies a projective predicate $\mathcal{P} = (P_{s_1}, \cdots, P_{s_m})$, denoted by $\rho \models^{QAI} \mathcal{P}$, if for all $1 \le i \le m$, $P_{s_i}\rho_{s_i} = \rho_{s_i}$, i.e., $\rho_{s_i} \models P_{s_i}$. Equivalently, $\rho \models \gamma(\mathcal{P})$, where*

$$\gamma(\mathcal{P}) := \bigcap_i P_{s_i} \otimes I_{[n]\setminus s_i}.$$

Given quantum circuit $\mathbf{C}$ and state $\rho \models \mathcal{P}$, QAI [48] constructs a predicate $\mathcal{Q}$ such that the post-state $[\![\mathbf{C}]\!](\rho)$ satisfies $\mathcal{Q}$, denoted by $\models^{QAI} \{\mathcal{P}\}\mathbf{C}\{\mathcal{Q}\}$. The idea—encapsulated in the following theorem—is to perform partial concretization rather than complete concretization.

Theorem 2.1 ([48]). *Let U_F be a unitary gate applied to the qubit set $s(F)$, and let $\mathcal{P} = (P_{s_1}, \cdots, P_{s_m})$ be a projective predicate. For each s_i, define*

$$R_i = \bigcap_{s_j \subseteq s_i \cup s(F)} P_{s_j} \otimes I_{s_i \cup s(F)\setminus s_j}, \quad Q_{s_i} = \mathrm{supp}\left(\mathrm{Tr}_{s_i \cup s(F)\setminus s_i}\left(U_F R_i U_F^\dagger\right)\right), \quad U^\sharp(\mathcal{P}) = (Q_{s_1}, \cdots, Q_{s_m}).$$

Then $\rho \models^{QAI} \mathcal{P} \implies U_F \rho U_F^\dagger \models^{QAI} U^\sharp(\mathcal{P})$.

Theorem 2.2 ([48]). *Let $P = \mathrm{span}\{|a_1 a_2 \cdots a_n\rangle, |b_1 b_2 \cdots b_n\rangle\}$, where the product states $|a_i\rangle$ and $|b_i\rangle$ are not proportional for every $i \in [n]$. Then $P = \gamma(\mathcal{P})$, where $\mathcal{P} = (P_{1,2}, \ldots, P_{n-1,n})$ and each $P_{i,i+1} = \mathrm{span}\{|a_i a_{i+1}\rangle, |b_i b_{i+1}\rangle\}$.*

3 Correctness Formulas and the Logical System SAQR-QC

In this section, we introduce a class of predicates that can be used for quantitative local reasoning (§3.1). We then sketch a strawman approach that incorporates these predicates in a logic similar to QHL (§3.2). Via a simple example, we demonstrate that this strawman approach can lead to a significant loss of precision in reasoning. To address this limitation, we integrate the QAI technique into the framework, leading to the formal definition of the judgments used in SAQR-QC (§3.3), a theorem about how SAQR-QC judgments relate to judgments in QAI (§3.4), a systematic presentation of the inference rules of SAQR-QC (§3.5), and an approximation strategy that addresses how to work with the one inference rule of SAQR-QC that poses a challenge to scalability (§3.6).

3.1 Quantitative Local Reasoning via Generalized Predicates

Reasoning about full quantum states quickly becomes infeasible due to the exponential growth of the state space with the number of qubits. To address this issue, we want an approach akin to QAI [48], which uses tuples of projections as predicates that capture properties of small subsystems. However, QAI itself is unsatisfactory because QAI can only perform *qualitative* local reasoning.

To enable *quantitative* local reasoning, we focus on *reduced density matrices*, which capture the behavior of local subsystems. Inspired by Quantum Hoare Logic (QHL)—where a global positive semidefinite matrix A serves as a *predicate* describing properties of a state ρ via the expectation value $\mathrm{Tr}(A\rho)$—we introduce the notion of *local observables* to reason about subsystem states ρ_s.

Definition 3.1. *A* local observable *over a qubit set $s \subseteq [n]$ is a positive semidefinite operator A_s satisfying $0 \le A_s \le I_s$, acting nontrivially only on the qubits in s.*

According to Lemma 2.1, $\mathrm{Tr}(A_s\rho_s) = \mathrm{Tr}(A\rho)$ with $A := A_s \otimes I_{[n]\setminus s}$. Local observables thus form a natural subclass of predicates in QHL, and we use them to monitor reduced density matrices via

the inner product $\mathrm{Tr}(\cdot)$, in direct analogy with QHL; here, "monitoring ρ_s" means computing the expectation value of local observable A_s with respect to ρ_s—i.e., computing $\mathrm{Tr}(A_s\rho_s)$.

To capture richer information, we track not a single ρ_s but a tuple of reduced density matrices $(\rho_{s_1}, \ldots, \rho_{s_m})$, where each $s_i \subseteq [n]$ is a small subset of bounded size (i.e., in the same spirit as QAI). This representation (i) provides significantly deeper insight into the system's local structure, and (ii) remains tractable, because it requires only *linear* resources, i.e., scales linearly with m.

The idea is to define a quantum predicate for each s_i, represented by an observable A_{s_i}, focusing solely on the state of the quantum registers in s_i.

Definition 3.2. *A* predicate *over an n-qubit space and $S = (s_1, \ldots, s_m)$ is a tuple of local observables $(A_{s_1}, \ldots, A_{s_m})$. with $0 \le A_{s_i} \le I_{s_i}$, each acting non-trivially only on s_i. Predicates are denoted by $\mathscr{A}$ (or $\mathscr{B}, \mathscr{D}$); m is the predicate's* size. *The domain of $(A_{s_1}, \ldots, A_{s_m})$, denoted by* $\mathrm{dom}(A_{s_1}, \ldots, A_{s_m})$, *is $(s_1, \ldots, s_m)$.*

Each local observable acts nontrivially on just a few qubits, and captures the expectation value of a measurable quantity, such as the success probability, fidelity, or entanglement, providing a direct link between semantics and operational outcomes. By tracing their evolution under unitaries and partial traces, we obtain a *scalable, compositional* framework for quantum program verification, supporting precise analysis without reconstructing the full state.

These generalized predicates track quantitative aspects of quantum programs—such as probabilities and expectation values—by monitoring reduced density matrices over small, constant-size subsystems. This abstraction underpins scalable, compositional reasoning about quantum computations via low-dimensional summaries.

Definition 3.3. *For a predicate $\mathscr{A} = (A_{s_1}, \cdots, A_{s_m})$, its matrix representation $M_{\mathscr{A}}$ is defined as:*

$$M_{\mathscr{A}} = \sum_{i=1}^{m} A_{s_i} \otimes I_{[n]\setminus s_i}. \tag{4}$$

This definition induces a function $\mathrm{Tr}(M_{\mathscr{A}}\rho)$ on a state ρ. By Lemma 2.1, the value of $\mathrm{Tr}(M_{\mathscr{A}}\rho)$ depends only on the reduced states ρ_{s_i} and is equal to the sum of the expectations obtained by measuring A_{s_i} on each ρ_{s_i}:

$$\mathrm{Tr}(M_{\mathscr{A}}\rho) = \mathrm{Tr}\Big(\sum_{i=1}^{m}(A_{s_i} \otimes I_{[n]\setminus s_i})\rho\Big) = \sum_{i=1}^{m}\mathrm{Tr}[(A_{s_i} \otimes I_{[n]\setminus s_i})\rho] = \sum_{i=1}^{m}\mathrm{Tr}(A_{s_i}\rho_{s_i}). \tag{5}$$

Moreover, matrix $M_{\mathscr{A}}$ in Equation (4) implicitly specifies a set of quantum states in terms of how their expectations of local observables relate to a given threshold r: $\{\rho \mid \mathrm{Tr}(M_{\mathscr{A}}\rho) \ge r\}$, for $r \ge 0$.

To reason about program behavior, we additionally require a notion of predicate transformation that captures how such predicates evolve under quantum operations. In SAQR-QC, this notion is realized via linear predicate transformers, which map pre-state predicates to post-state predicates. Operationally, these transformers describe how the local observables—and hence the corresponding linear functions—are updated by a given circuit element.

This separation between representation (predicates) and transformation (predicate transformers) enables scalable, compositional quantitative reasoning.

We are also interested in the order relation between predicates.

Definition 3.4. *Given predicates $(A_{s_1}, \cdots, A_{s_m})$ and $(A'_{s_1}, \cdots, A'_{s_m})$ over the same domains, define*

$$(A_{s_1}, \cdots, A_{s_m}) \sqsubseteq (A'_{s_1}, \cdots, A'_{s_m}) \quad \text{if and only if} \quad A_{s_i} \le A'_{s_i} \text{ for all } i.$$

Lemma 3.1. *The matrix representation of predicates is monotonic with respect to this ordering—i.e.,*

$$\mathscr{A} \sqsubseteq \mathscr{B} \quad \Rightarrow \quad M_{\mathscr{A}} \le M_{\mathscr{B}}.$$

3.2 A First Attempt at Defining Judgments

This section presents an initial attempt for *quantitative local reasoning* in quantum programs using tuples of local observables.

In QHL [46], the unitary rule gives the *weakest precondition* for a postcondition: $\{U_{\mathbf{C}}^{\dagger} B U_{\mathbf{C}}\}\ \mathbf{C}\ \{B\}$. To ensure that postcondition B is efficiently representable and physically relevant, we restrict B to the kind of predicate introduced in Definition 3.3: we consider a tuple of local observables, where locality indicates that each observable acts nontrivially on only a constant number of qubits. Let $S = (s_1, \ldots, s_m)$, with $s_i \subseteq [n]$, denote the tuples of local qubits that are of interest, and let $\mathscr{B} = (B_{s_1}, \ldots, B_{s_m})$ with $0 \leq B_{s_i} \leq I_{s_i}$; then $M_{\mathscr{B}} = \sum_i B_{s_i} \otimes I_{[n]\setminus s_i}$. The question now is "What is the analogue of $U_{\mathbf{C}}^{\dagger} B U_{\mathbf{C}}$?"

Definition 3.5. *Given a fixed domain $S = (s_1, \ldots, s_m)$ and a quantum program $\mathbf{C} = \lambda\rho.\, U\rho U^{\dagger}$ on a density matrix, we say that $\mathscr{A} = (A_{s_1}, \ldots, A_{s_m})$ is a* local precondition *of $\mathscr{B} = (B_{s_1}, \ldots, B_{s_m})$ if*

$$\models \{M_{\mathscr{A}}\}\ \mathbf{C}\ \{M_{\mathscr{B}}\},$$

where $M_{\mathscr{A}} = \sum_i A_{s_i} \otimes I_{[n]\setminus s_i}$ and $M_{\mathscr{B}} = \sum_i B_{s_i} \otimes I_{[n]\setminus s_i}$, and the judgment means

$$\forall \rho, \quad \mathrm{Tr}(M_{\mathscr{A}}\rho) \leq \mathrm{Tr}(M_{\mathscr{B}}[\![\mathbf{C}]\!](\rho)).$$

We could use a QHL-like proof rule for a unitary, namely

$$\frac{M_{\mathscr{A}} \leq U^{\dagger} M_{\mathscr{B}} U}{\{M_{\mathscr{A}}\}\ \mathbf{C}\ \{M_{\mathscr{B}}\}} \tag{6}$$

To see that $M_{\mathscr{A}} \leq U^{\dagger} M_{\mathscr{B}} U$ leads to a valid QHL judgement—without requiring that the observables are bounded above by the identity—we observe that

$$\mathrm{Tr}\left(M_{\mathscr{A}}\rho\right) \leq \mathrm{Tr}\left(U^{\dagger} M_{\mathscr{B}} U \rho\right) = \mathrm{Tr}\left(M_{\mathscr{B}} U \rho U^{\dagger}\right) = \mathrm{Tr}\left(M_{\mathscr{B}}\,[\![\mathbf{C}]\!](\rho)\right), \tag{7}$$

where the inequality follows from Lemma 2.4 applied to the premise of Rule (6), and the first equality uses Lemma 2.2. Definition 3.5 tells us that the conclusion of Rule (6) holds.

This definition naturally gives rise to a correctness judgment of the form $\{\mathscr{A}\}\ \mathbf{C}\ \{\mathscr{B}\}$ for quantum circuits. In addition to enabling *backward reasoning*, Rule (6) with Definition 3.5 also supports *forward reasoning* by changing the premise of Rule (6) to "$U M_{\mathscr{A}} U^{\dagger} \leq M_{\mathscr{B}}$."

Unfortunately, it is not possible, in general, to use either version of Rule (6) *algorithmically* to compute the weakest precondition and the strongest postcondition, respectively. The issue is one of *expressibility*: can the answer be decomposed according to the chosen scheme $S = (s_1, \ldots, s_m)$? In general, this *structural constraint* of our candidate logic presents an obstacle. For instance, even when U_F is a two-qubit unitary, the transformed observable $U_F^{\dagger}\left(\sum_i B_{s_i} \otimes I_{[n]\setminus s_i}\right) U_F$ may not admit a decomposition of the form $\sum_i A_{s_i} \otimes I_{[n]\setminus s_i}$ for any choice of local observables A_{s_i}. That is, the following equality may have *no solution* for local predicates $\mathscr{A}$, for some given $\mathscr{B}$.

$$M_{\mathscr{A}} = U_F^{\dagger} M_{\mathscr{B}} U_F \tag{8}$$

Example 3.1. To understand this limitation concretely, consider a circuit in which a CNOT gate is applied to the input state $|00\rangle$, as shown in Figure 1. The output state is $|00\rangle$. Let us reason about the circuit $\mathbf{C}$ using Equation (8), where we choose $(s_1, s_2) = (\{q_1\}, \{q_2\})$. Let the postcondition be $\mathscr{B} = (B_{s_1}, B_{s_2}) = (|0\rangle\langle 0|, |0\rangle\langle 0|)$, which has the matrix representation $M_{\mathscr{B}} = |0\rangle\langle 0| \otimes I + I \otimes |0\rangle\langle 0|$.

> ***Observation**: The following calculation will demonstrate that, in local reasoning, it is not always possible to express the weakest precondition.*

We will show that there is no A_1, A_2 such that

$$U_{\mathrm{C}}^{\dagger}\left(\sum_i B_{s_i}\otimes I_{[n]\setminus s_i}\right)U_{\mathrm{C}} = |0\rangle\langle 0|\otimes I + |0\rangle\langle 0|\otimes|0\rangle\langle 0| + |1\rangle\langle 1|\otimes|1\rangle\langle 1| = A_1\otimes I + I\otimes A_2$$

We see that the left-hand side is orthogonal to $|1\rangle\langle 1|\otimes|0\rangle\langle 0|$. Therefore, we know that

$$\begin{aligned}
&\mathrm{Tr}[(|1\rangle\langle 1|\otimes|0\rangle\langle 0|)(A_1\otimes I + I\otimes A_2)] = 0\\
\Rightarrow\ &\mathrm{Tr}[(|1\rangle\langle 1|A_1)\otimes|0\rangle\langle 0|] + \mathrm{Tr}[|1\rangle\langle 1|\otimes(|0\rangle\langle 0|A_2)] = 0\\
\Rightarrow\ &\mathrm{Tr}(|1\rangle\langle 1|A_1) = 0 \quad\text{and}\quad \mathrm{Tr}(|0\rangle\langle 0|A_2) = 0 \qquad \text{(Lemma 2.3)}\\
\Rightarrow\ &A_1 = \lambda_1|0\rangle\langle 0|, \quad A_2 = \lambda_2|1\rangle\langle 1|\\
\Rightarrow\ &A_1\otimes I + I\otimes A_2
\end{aligned}$$

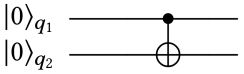


Fig. 1. CNOT circuit C.

Moreover, we can show that the QHL-like proof rule (6) is sometimes unable to prove a simple assertion. For instance, for the input $|00\rangle$, the *CNOT* gate has no effect; therefore, the output state is also $|00\rangle$. Consequently, the following assertion holds:

$$\{(|0\rangle\langle 0|, |0\rangle\langle 0|)\}\ \mathbf{C}\ \{(|0\rangle\langle 0|, |0\rangle\langle 0|)\}. \tag{9}$$

To establish that Rule (6) cannot prove that Assertion (9) holds, we compute

$$\begin{aligned}
&\{(|0\rangle\langle 0|, |0\rangle\langle 0|)\}\ \mathbf{C}\ \{(|0\rangle\langle 0|, |0\rangle\langle 0|)\}\\
\Longleftrightarrow\ &CNOT(|0\rangle\langle 0|\otimes I + I\otimes|0\rangle\langle 0|)CNOT^{\dagger} \le |0\rangle\langle 0|\otimes I + I\otimes|0\rangle\langle 0|\\
\Longleftrightarrow\ &|0\rangle\langle 0|\otimes I + |00\rangle\langle 00| + |11\rangle\langle 11| \le |0\rangle\langle 0|\otimes I + I\otimes|0\rangle\langle 0|\\
\Longleftrightarrow\ &|00\rangle\langle 00| + |11\rangle\langle 11| \le I\otimes|0\rangle\langle 0|
\end{aligned}$$

Because the final inequality fails, in our strawman logic based on tuples of local observables and Rule (6), Assertion (9) is unprovable; i.e.,

$$\nvdash \{(|0\rangle\langle 0|, |0\rangle\langle 0|)\}\ \mathbf{C}\ \{(|0\rangle\langle 0|, |0\rangle\langle 0|)\}.$$

What this example shows is that without additional constraints that restrict the input, the candidate logic lacks precision. As we show in Theorem 3.2 (see §3.3), by imposing the constraint $\mathcal{P} := (|0\rangle\langle 0|, |0\rangle\langle 0|)$ on the input, SAQR-QC can establish the desired property that input state $|00\rangle$ yields output state $|00\rangle$.

3.3 Quantitative Judgments and Validity

Beyond the expressivity limitations noted in §3.2, the predicates used in §3.2 cannot even capture the singleton state $|0\rangle^{\otimes n}$, revealing a mismatch with our goals: Definition 3.5 demands validity for all states satisfying the precondition, whereas quantum programs typically concern the specific input $|0\rangle^{\otimes n}$. To address both issues, we adopt a more expressive logic that still supports quantitative local reasoning, leveraging quantum abstract interpretation [48] as a foundation.

Definition 3.6. *A judgment is a triple of the form $\{\mathscr{A} \mid \mathcal{P}\}\ \mathbf{C}\ \{\mathscr{B} \mid \mathcal{Q}\}$ for program $\mathbf{C}$, general predicates $\mathscr{A}$ and $\mathscr{B}$, and projective predicates $\mathcal{P}$ and $\mathcal{Q}$.*

For simplicity, we assume that $\mathscr{A}$, $\mathscr{B}$, $\mathcal{P}$, and $\mathcal{Q}$ share the same domain.

Definition 3.7 (Validity). *The judgment $\{\mathscr{A} \mid \mathcal{P}\}\ \mathbf{C}\ \{\mathscr{B} \mid \mathcal{Q}\}$ in Definition 3.6 holds if*

$$\forall \rho \vDash^{QAI} \mathcal{P},\ [\![\mathbf{C}]\!](\rho) \vDash^{QAI} \mathcal{Q},\ \text{and}\ \mathrm{Tr}(M_{\mathscr{A}}\rho) \le \mathrm{Tr}(M_{\mathscr{B}}[\![\mathbf{C}]\!](\rho)). \tag{10}$$

where $M_{\mathscr{A}}$ and $M_{\mathscr{B}}$ are the matrix representations of $\mathscr{A}$ and $\mathscr{B}$, as defined in Definition 3.3.

Consider the special case $\mathcal{P} = \mathcal{Q} = \mathcal{I}$ with $\mathcal{I} := (I_{s_1}, \cdots, I_{s_m})$, we will write

$$\{\mathscr{A} \mid \mathcal{I}\}\ \mathbf{C}\ \{\mathscr{B} \mid \mathcal{I}\} = \{\mathscr{A}\}\ \mathbf{C}\ \{\mathscr{B}\}$$

in which case this definition simplifies to Definition 3.5.

As we know from Equation (5), $\mathrm{Tr}(M_{\mathscr{A}}\rho) = \sum_{i=1}^{m} \mathrm{Tr}\left(A_{s_i}\rho_{s_i}\right)$. In other words, $\mathrm{Tr}(M_{\mathscr{A}}\rho)$ only depends on the reduced density matrices of ρ with respect to the systems $\{ s_i \}$. Intuitively, this judgment tracks a linear function of the tuple of reduced density matrices to enable quantitative reasoning. Moreover, $\rho_{s_i} \vDash P_{s_i}$ implies that $\rho_{s_i} = P_{s_i}\rho_{s_i}P_{s_i}$. Due to Lemma 2.2, we have

$$\sum_{i=1}^{m} \mathrm{Tr}(A_{s_i}\rho_{s_i}) = \sum_{i=1}^{m} \mathrm{Tr}(A_{s_i}P_{s_i}\rho_{s_i}P_{s_i}) = \sum_{i=1}^{m} \mathrm{Tr}(P_{s_i}A_{s_i}P_{s_i}\rho_{s_i}),$$

We observe that $\mathrm{Tr}(M_{\mathscr{A}}\rho) = \mathrm{Tr}(M_{\mathscr{A}'}\rho)$ with $\mathscr{A}' = (P_{s_1}A_{s_1}P_{s_1}, \ldots, P_{s_m}A_{s_m}P_{s_m})$. Then

$$\{\mathscr{A}' \mid \mathcal{P}\}\ \mathbf{skip}\ \{\mathscr{A} \mid \mathcal{P}\} \quad \text{and} \quad \{\mathscr{A} \mid \mathcal{P}\}\ \mathbf{skip}\ \{\mathscr{A}' \mid \mathcal{P}\}.$$

Thus, the qualitative insights from quantum abstract interpretation refine the reasoning process by yielding "stronger" postconditions and "weaker" preconditions.[2]

Example 3.2. Consider again the example from Theorem 3.1 in which a CNOT gate is applied to the input state $|00\rangle$ (Figure 1). The output state is $|00\rangle$. Again, we choose $(s_1, s_2) = (\{q_1\}, \{q_2\})$, but now our two-part assertions are able to prove the desired property, now stated as follows:

$$\vdash \{(|0\rangle\langle 0|, |0\rangle\langle 0|)|(|0\rangle\langle 0|, |0\rangle\langle 0|)\}\ \mathbf{C}\ \{(|0\rangle\langle 0|, |0\rangle\langle 0|)|(|0\rangle\langle 0|, |0\rangle\langle 0|)\}.$$

3.4 Reduction

We now consider the relationship between the correctness $\models^{\mathrm{QAI}}$ in QAI and $\models$ in Definition 3.7.

Theorem 3.1 (Reduction Principle). *Consider the behavior of a quantum circuit with respect to any input state. For any projective predicates $\mathcal{P} = (P_{s_1}, \cdots, P_{s_m})$ and $\mathcal{Q} = (Q_{s_1}, \cdots, Q_{s_m})$, $\mathcal{P}$ and $\mathcal{Q}$ can be regarded as the observables $\mathscr{P} := (P_{s_1}, \cdots, P_{s_m})$ and $\mathscr{Q} = (Q_{s_1}, \cdots, Q_{s_m})$, respectively. Then we have the property*

$$\textit{If} \models \{\mathscr{P}\}\ \mathbf{C}\ \{\mathscr{Q}\}\ \textit{in the sense of Definition 3.7, then} \models^{\mathrm{QAI}} \{\mathcal{P}\}\ \mathbf{C}\ \{\mathcal{Q}\}.$$

We defer the proof to §B.

3.5 Logical System with Soundness

The inference rules for program constructs in SAQR-QC are presented in Figure 2. The proof of the following theorem is deferred to §C.

Theorem 3.3 (Soundness). *The proof system in Figure 2 is sound. That is, for quantum program* $\mathbf{C}$, $\vdash \{\mathscr{A} \mid \mathcal{P}\}\ \mathbf{C}\ \{\mathscr{B} \mid \mathcal{Q}\}$ implies $\models \{\mathscr{A} \mid \mathcal{P}\}\ \mathbf{C}\ \{\mathscr{B} \mid \mathcal{Q}\}$.

3.6 Unit-2: Scalable Application of the Unit-1 Rule

Among the proof rules in Figure 2, only the rule Unit-1 poses a significant challenge to scalability. Specifically, the condition

$$\gamma(\mathcal{P})M_{\mathscr{A}}\gamma(\mathcal{P}) \le \gamma(\mathcal{P})[\![\mathbf{C}]\!]^*(M_{\mathscr{B}})\gamma(\mathcal{P}) \tag{11}$$

requires computing the projector $\gamma(\mathcal{P})$, which becomes intractable for systems with many qubits. That is, given a postcondition $\mathscr{B}$ and a set of projective predicates $\mathcal{P}$, we are not aware of any scalable method for synthesizing a precondition $\mathscr{A}$ that satisfies this inequality.

To address this issue, we introduce a *compositional approximation* strategy by replacing the Unit-1 rule with Unit-2. The key idea behind Unit-2 is to construct inequalities over high-dimensional

[2]Here, "stronger" and "weaker" simply refer to the relation $\mathrm{Tr}(A_{s_i}) \ge \mathrm{Tr}\left((P_{s_i}P_{s_i})A_{s_i}\right) \ge \mathrm{Tr}\left(P_{s_i}A_{s_i}P_{s_i}\right)$, as established by Lemma 2.2. They do **not** indicate any ordering relationship between $\mathscr{A}$ and $\mathscr{A}'$.

$$\textsc{Skip}\quad \frac{}{\{\mathscr{A} \mid \mathcal{P}\}\ \mathbf{Skip}\ \{\mathscr{A} \mid \mathcal{P}\}}$$

$$\textsc{Unit-1}\quad \frac{\gamma(\mathcal{P}) M_{\mathscr{A}} \gamma(\mathcal{P}) \le \gamma(\mathcal{P}) U_F^\dagger M_{\mathscr{B}} U_F \gamma(\mathcal{P})}{\{\mathscr{A} \mid \mathcal{P}\}\ \bar{q} := U_F\,[\bar{q}]\ \{\mathscr{B} \mid U_F^\sharp(\mathcal{P})\}}$$

$$\textsc{Unit-2}\quad \frac{\{\mathscr{A}_i \mid \mathcal{P}_i\}\ \bar{q} := U_F\,[\bar{q}]\ \{\mathscr{B}_i \mid U_F^\sharp(\mathcal{P}_i)\}}{\{\oplus_i \mathscr{A}_i \mid \mathcal{P}\}\ \bar{q} := U_F\,[\bar{q}]\ \{\oplus_i \mathscr{B}_i \mid \oplus_i U_F^\sharp(\mathcal{P}_i)\}}$$

$$\textsc{Seq}\quad \frac{\{\mathscr{A} \mid \mathcal{P}\}\ \mathbf{C}_1\ \{\mathscr{D} \mid \mathcal{R}\} \qquad \{\mathscr{D} \mid \mathcal{R}\}\ \mathbf{C}_2\ \{\mathscr{B} \mid \mathcal{Q}\}}{\{\mathscr{A} \mid \mathcal{P}\}\ \mathbf{C}_1; \mathbf{C}_2\ \{\mathscr{B} \mid \mathcal{Q}\}}$$

$$\textsc{Con}\quad \frac{\{\mathscr{A} \mid \mathcal{P}\}\ \mathbf{C}\ \{\mathscr{B} \mid \mathcal{Q}\},\ \ \mathscr{D} \sqsubseteq \mathscr{A},\ \mathscr{B} \sqsubseteq \mathscr{E},\ \mathcal{R} \sqsubseteq \mathcal{P},\ \mathcal{Q} \sqsubseteq \mathcal{T}}{\{\mathscr{D} \mid \mathcal{R}\}\ \mathbf{C}\ \{\mathscr{E} \mid \mathcal{T}\}}$$

Fig. 2. Inference rules for program constructs in SAQR-QC. We can use the proof rules for both forward reasoning or backward reasoning. The Skip, Seq, and Con rules are standard, and operate in conjunction with the ordering on predicates. However, a direct application of the Unit-1 rule would break scalability, because it requires computing and applying $\gamma(\mathcal{P})$, i.e., the full concretization of $\mathcal{P}$. Unit-2 builds on Unit-1 to enable scalable reasoning by partitioning the index set $\{1, \ldots, m\}$ into disjoint subsets, each corresponding to a group of local predicates to which Unit-1 is applied. For tuples of matrices $\mathscr{X}_1, \ldots, \mathscr{X}_k$, where $\mathscr{X}_i = (X_{i,1}, \ldots, X_{i,m_i})$, we define their concatenation by $\bigoplus_{i=1}^k \mathscr{X}_i := (X_{1,1}, \ldots, X_{1,m_1}, \ldots, X_{k,1}, \ldots, X_{k,m_k})$, which produces a single tuple containing all matrices in order; this notation applies equally to $\mathscr{A}_i$, $\mathscr{B}_i$, and $\mathcal{P}_i$. See §3.6 for details.

systems by composing inequalities over smaller subsystems. These approximations trade some precision for tractability, but preserve overall soundness and enable reasoning about large quantum programs.

We first partition the index set $\{s_1, \ldots, s_m\}$ into disjoint subsets $T_1, \ldots, T_k$, and consider the following Unit-1-like inequalities for each T_i:

$$\forall j.\quad \gamma(\mathcal{P}) \left(\sum_{i \in T_j} A_{s_i} \otimes I_{[n]\setminus s_i} \right) \gamma(\mathcal{P}) \le \gamma(\mathcal{P}) U^\dagger \left(\sum_{i \in T_j} B_{s_i} \otimes I_{[n]\setminus s_i} \right) U \gamma(\mathcal{P}). \tag{12}$$

Even if each T_i is small, computing $\gamma(\mathcal{P})$ remains challenging. The only aspect that does not fit the Unit-2 rule is the role of $\gamma(\mathcal{P})$, which will be discussed next.

Warm-up: Ignoring $\boldsymbol{\gamma(\mathcal{P})}$. As a first step, for the sake of scalability we can use the following version of the Unit-1 rule, which omits the occurrences of the projection $\gamma(\mathcal{P})$:

$$\textsc{Unit-1'}\quad \frac{M_{\mathscr{A}} \le U_F^\dagger M_{\mathscr{B}} U_F}{\{\mathscr{A} \mid \mathcal{P}\}\ \bar{q} := U_F\,[\bar{q}]\ \{\mathscr{B} \mid U_F^\sharp(\mathcal{P})\}} \tag{13}$$

Although rule (13) is imprecise (see §3.2), it may still yield useful results, as we will see in §4.1. The rule (13) is sound by the unitary rule of QHL [46], and it implies the Unit-1 rule via Lemma 2.4.

We make use of Unit-1' by structuring a proof into arguments that involve only small matrices, following the pattern on the left-hand side of Equation (14), which, by Equation (14), establishes the premise of the Unit-1' rule.

$$\forall j.\quad \sum_{i \in T_j} A_{s_i} \otimes I_{[n]\setminus s_i} \le U_F^\dagger \left(\sum_{i \in T_j} B_{s_i} \otimes I_{[n]\setminus s_i} \right) U_F \quad \Longrightarrow \quad M_{\mathscr{A}} \le U_F^\dagger M_{\mathscr{B}} U_F. \tag{14}$$

The forward form of this inequality is:

$$\forall j. \quad U_F \left(\sum_{i \in T_j} A_{s_i} \otimes I_{[n]\setminus s_i} \right) U_F^\dagger \leq \sum_{i \in T_j} B_{s_i} \otimes I_{[n]\setminus s_i}. \tag{15}$$

Because each inequality involves a small subsystem, constraints can be efficiently discharged using symbolic solvers or semidefinite programming, depending on the structure of A_{s_i} and B_{s_i}.

A More Precise Approach: Guarded Inequalities. The warm-up approach may lose critical precision due to ignoring $\gamma(\mathcal{P})$. To refine this, we incorporate the projection information in a compositional way. Let $\mathcal{P} = \{P_{s_1}, \ldots, P_{s_m}\}$, and again partition the index set into disjoint subsets $T_1, \ldots, T_k$. For each T_j, we define a local over-approximation of the global projector:

$$\gamma(\mathcal{P}) \subseteq P_j := \bigcap_{i \in T_j} P_{s_i} \otimes I_{[n]\setminus s_i}. \tag{16}$$

We can write guarded versions of the local inequalities:

$$\forall j. \quad P_j \left(\sum_{i \in T_j} A_{s_i} \otimes I_{[n]\setminus s_i} \right) P_j \leq P_j U_F^\dagger \left(\sum_{i \in T_j} B_{s_i} \otimes I_{[n]\setminus s_i} \right) U_F P_j. \tag{17}$$

These inequalities are more precise than those in (14), because they account for the structure imposed by $\mathcal{P}$, while still avoiding the need to compute the full projector $\gamma(\mathcal{P})$.

Summary of Reasoning Tools. We have practical tools for scalable reasoning under the Unit-2 rule:

- **Unguarded inequalities:** Equations (14) and (15) enable fast reasoning, but may be imprecise.
- **Guarded inequalities:** Equation (17) preserves more of $\gamma(\mathcal{P})$'s structure while remaining tractable.

Finally, in §D, we prove that the guarded inequalities imply the condition stated as Equation (11), thereby validating the correctness of our compositional approximation framework.

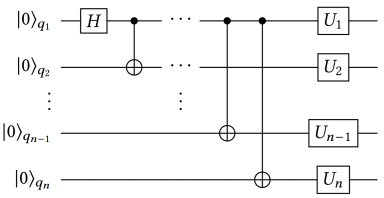


Fig. 3. GHZ circuit with unitaries U_i.

Remark: The methodology presented in this section is general. For a given circuit, useful predicates can typically be constructed straightforwardly because each gate in a quantum circuit is represented by a local unitary operator that typically acts on at most two qubits. In each reasoning step, we analyze a single constant-qubit unitary—typically a one- or two-qubit gate—and extract the predicate B_{s_i} over its qubits, leaving the remainder untouched. This approach is sound because the unitary does not affect the reduced density matrices of qubits outside its support. The convenience and effectiveness of this approach are illustrated in the examples throughout the paper.

4 Quantitative Reasoning about a Generalized GHZ Circuit

In this section, we consider a generalized GHZ circuit in which half the gates are arbitrary single-qubit unitaries, producing a highly entangled, densely parameterized state beyond the reach of classical simulation methods built on the Gottesman–Knill theorem [1]. The scalable techniques illustrated herein perform compositional and projection-free reasoning over subsystems, thereby supporting tractable approximation of the circuit's complex output state.

After presenting SAQR-QC in action on the generalized GHZ circuit, we discuss approaches to handling imprecision at the end of §4.2.

4.1 Reasoning about the GHZ Circuit (Figure 3)

The analysis first applies the "warm-up" method from §3.6 (Equation (14)). The second phase of the analysis leverages QAI, employing Theorem 2.2 to confine the output state within a two-dimensional subspace. This critical dimensionality reduction facilitates precise amplitude approximations within our quantitative-reasoning framework. Strikingly, the resulting characterization matches the exact quantum state up to a phase, underscoring that with the right predicates SAQR-QC can be used to obtain quite precise results, even though SAQR-QC has restricted power because of the concern that proofs be scalable—i.e., both the matrices involved in the proof and the logical derivation grow only polynomially in the number of qubits.

We select the domain $(\{1,2\},\{2,3\},\cdots,\{n-1,n\})$ for reasoning about Figure 3.

STEP 1. We first select the precondition to be $\{\mathscr{A} \mid \mathcal{P}\}$

$$\mathscr{A} = (A_{1,2}, A_{2,3}, \cdots, A_{n-1,n}) = (|++\rangle\langle++|, |++\rangle\langle++|, \cdots, |++\rangle\langle++|)$$
$$\mathcal{P} = (P_{1,2}, P_{2,3}, \cdots, P_{n-1,n}) = (|00\rangle\langle00|, |00\rangle\langle00|, \cdots, |00\rangle\langle00|).$$

One can verify that the initial state $|0\cdots0\rangle\langle0\cdots0| \vDash \mathcal{P}$. We now use our proof rules to compute a postcondition for the GHZ circuit, given the precondition $\{\mathscr{A} \mid \mathcal{P}\}$.

After the first H gate, we use inequality (14) to derive the following:

$$\begin{aligned}
&\sum_i A_{s_i} \otimes I_{[n]\setminus\{i,i+1\}} \le H^\dagger \sum_i B_{i,i+1} \otimes I_{[n]\setminus\{i,i+1\}} H \\
\iff &\sum_i HA_{i,i+1} \otimes I_{[n]\setminus\{i,i+1\}} H^\dagger \le \sum_i B_{i,i+1} \otimes I_{[n]\setminus\{i,i+1\}} \\
\iff &HA_{1,2}H^\dagger \otimes I_{[n]\setminus\{1,2\}} + \sum_{i>1} A_{i,i+1} \otimes HI_{[n]\setminus\{i,i+1\}} H^\dagger \le \sum_i B_{i,i+1} \otimes I_{[n]\setminus\{i,i+1\}} \\
\iff &HA_{1,2}H^\dagger \otimes I_{[n]\setminus\{1,2\}} + \sum_{i>1} A_{i,i+1} \otimes I_{[n]\setminus\{i,i+1\}} \le \sum_i B_{i,i+1} \otimes I_{[n]\setminus\{i,i+1\}}.
\end{aligned}$$

The final inequality above matches inequality (15) on a term-by-term basis. We can satisfy inequality (15) when each of the following single-term inequalities hold:

$$HA_{1,2}H^\dagger \otimes I_{[n]\setminus\{1,2\}} \le B_{1,2} \otimes I_{[n]\setminus\{1,2\}} \qquad A_{i,i+1} \otimes I_{[n]\setminus\{i,i+1\}} \le B_{i,i+1} \otimes I_{[n]\setminus\{i,i+1\}} \text{ for } i > 1.$$

Each inequality can be realized as an equality, yielding the postcondition $\{\mathscr{B} \mid \mathcal{Q}\}$,

$$\mathscr{B} = (B_{1,2}, B_{2,3}, \ldots, B_{n-1,n}) = (HA_{1,2}H^\dagger, A_{2,3}, \ldots, A_{n-1,n}) = (|0+\rangle\langle0+|, |++\rangle\langle++|, \ldots, |++\rangle\langle++|)$$
$$\mathcal{Q} = (Q_{1,2}, Q_{2,3}, \ldots, Q_{n-1,n}) = (HP_{1,2}H^\dagger, P_{2,3}, \ldots, P_{n-1,n}) = (|+0\rangle\langle+0|, |00\rangle\langle00|, \ldots, |00\rangle\langle00|).$$

After the first CNOT gate has been applied to qubits q_1q_2, we use the Unit Rule to obtain a postcondition $\{\mathscr{C} \mid \mathcal{R}\} := \{(C_{1,2}, C_{2,3}, C_{3,4}\cdots, C_{n-1,n}) \mid (R_{1,2}, R_{2,3}, R_{3,4}\cdots, R_{n-1,n})\}$. We use QAI to compute $R_{1,2} = |00\rangle\langle00| + |11\rangle\langle11|$, $R_{2,3} = |00\rangle\langle00| + |10\rangle\langle10|$, and $R_{i,i+1} = |00\rangle\langle00|$ for $i > 2$.

What is left to determine are suitable values for $C_{1,2}$ and $C_{2,3}$. By the Unit-2 rule, partition $\mathscr{C} = (C_{1,2}, C_{2,3}, C_{3,4}, \ldots, C_{n-1,n})$ into $(C_{1,2}, C_{2,3}), (C_{3,4}), \ldots, (C_{n-1,n})$. That is,

$$\begin{aligned}
&\mathrm{CNOT}_{1,2}(I_{[n]\setminus\{i,i+1\}} \otimes B_{i,i+1})\mathrm{CNOT}_{1,2}^\dagger \le I_{[n]\setminus\{i,i+1\}} \otimes C_{i,i+1} \quad \forall\, i > 2, \\
&\mathrm{CNOT}_{1,2}(B_{1,2} \otimes I_3 + I_1 \otimes B_{2,3})\mathrm{CNOT}_{1,2}^\dagger \le C_{1,2} \otimes I_3 + I_1 \otimes C_{2,3}
\end{aligned}$$

We can choose $C_{i,i+1}$ to be $|++\rangle\langle++|$ for $i > 2$, and derive the following:

$$\begin{aligned}
&\mathrm{CNOT}_{1,2}(B_{1,2}\otimes I_3 + I_1\otimes B_{2,3})\mathrm{CNOT}_{1,2}^\dagger \le C_{1,2}\otimes I_3 + I_1\otimes C_{2,3}\\
\Longleftrightarrow&\mathrm{CNOT}_{1,2}B_{1,2}\mathrm{CNOT}_{1,2}^\dagger\otimes I_3 + \mathrm{CNOT}_{1,2}I_1\otimes B_{2,3}\mathrm{CNOT}_{1,2}^\dagger \le C_{1,2}\otimes I_3 + I_1\otimes C_{2,3}\\
\Longleftrightarrow&\mathrm{CNOT}_{1,2}|0+\rangle\langle 0+|\mathrm{CNOT}_{1,2}^\dagger\otimes I_3 + \mathrm{CNOT}_{1,2}I_1\otimes |++\rangle\langle++|\mathrm{CNOT}_{1,2}^\dagger \le C_{1,2}\otimes I_3 + I_1\otimes C_{2,3}\\
\Longleftrightarrow&|0+\rangle\langle 0+|\otimes I_3 + \mathrm{CNOT}_{1,2}(|0\rangle\langle 0|+|1\rangle\langle 1|)\otimes |++\rangle\langle++|\mathrm{CNOT}_{1,2}^\dagger \le C_{1,2}\otimes I_3 + I_1\otimes C_{2,3}\\
\Longleftrightarrow&|0+\rangle\langle 0+|\otimes I_3 + |0\rangle\langle 0|\otimes |++\rangle\langle++| + \mathrm{CNOT}_{1,2}|1\rangle\langle 1|\otimes |++\rangle\langle++|\mathrm{CNOT}_{1,2}^\dagger \le C_{1,2}\otimes I_3 + I_1\otimes C_{2,3}\\
\Longleftrightarrow&|0+\rangle\langle 0+|\otimes I_3 + |0\rangle\langle 0|\otimes |++\rangle\langle++| + \mathrm{CNOT}_{1,2}|1\rangle\langle 1|\otimes |+\rangle\langle+|\mathrm{CNOT}_{1,2}^\dagger\otimes |+\rangle\langle+| \le C_{1,2}\otimes I_3 + I_1\otimes C_{2,3}\\
\Longleftrightarrow&|0+\rangle\langle 0+|\otimes I_3 + I_1\otimes |++\rangle\langle++| \le C_{1,2}\otimes I_3 + I_1\otimes C_{2,3}
\end{aligned}$$

where, in the third-to-last and last steps, we use the following facts:

$$\mathrm{CNOT}_{1,2}|0\rangle|+\rangle = |0\rangle|+\rangle,\quad \mathrm{CNOT}_{1,2}|1\rangle|+\rangle = |1\rangle X|+\rangle = |1\rangle|+\rangle \tag{18}$$

Therefore, we find that the post-state $\mathscr{C}$ predicate is

$$(C_{1,2}, C_{2,3}, C_{3,4}\cdots, C_{n-1,n}) = (|0+\rangle\langle 0+|, |++\rangle\langle++|, \cdots, |++\rangle\langle++|). \tag{19}$$

(coinciding with the pre-state predicate $\mathscr{B} = (B_{1,2}, B_{2,3}, \ldots, B_{n-1,n})$), and the post-state predicate is

$$\mathcal{R} = (R_{1,2}, R_{2,3}, \cdots, R_{n-1,n}) = (|00\rangle\langle 00| + |11\rangle\langle 11|, |00\rangle\langle 00| + |10\rangle\langle 10|, |00\rangle\langle 00|, \cdots, |00\rangle\langle 00|).$$

Equation (19) illustrates an advantage of our choice of predicates. Because of properties such as those given in Equation (18), Equation (15) remains invariant under the application of CNOT gates. This invariance allows us to derive the strongest postcondition, while preserving the local structure of the matrix representation of predicates. As a result, we were able to make choices that made the inequalities that we worked with tight (or saturated, i.e., satified as equalities), making it easier to determine the postcondition. The right-hand side of Equation (19) continues to serve as the predicate of local observables, as reasoning continues about the remaining CNOT gates.

After applying the CNOT gate on q_1 and q_r, the postcondition can be chosen as $(\mathscr{D} \mid \mathcal{S})$, where

$$\begin{aligned}
\mathscr{D} &= (|0+\rangle\langle 0+|, |++\rangle\langle++|, \cdots, |++\rangle\langle++|)\\
\mathcal{S} &= (|00\rangle\langle 00| + |11\rangle\langle 11|, \cdots, |00\rangle\langle 00| + |11\rangle\langle 11|, |00\rangle\langle 00| + |10\rangle\langle 10|, |00\rangle\langle 00|, \cdots, |00\rangle\langle 00|).
\end{aligned}$$

At the right end of the circuit, after the application of $U_1\otimes U_2\otimes\cdots\otimes U_n$—where each U_i is a single-qubit unitary—the locality structure of the predicates remains unchanged. Therefore, we can choose the postcondition to be $(\mathscr{F} \mid \mathcal{T})$, where

$$\begin{aligned}
\mathscr{F} &= (\beta_1\otimes\beta_2, \cdots, \beta_{n-1}\otimes\beta_n)\\
\mathcal{T} &= (\psi_1\otimes\psi_2 + \phi_1\otimes\phi_2, \cdots, \psi_{n-1}\otimes\psi_n + \phi_{n-1}\otimes\phi_n)
\end{aligned}$$

with $\beta_i = |\beta_i\rangle\langle\beta_i|$, $\psi_i = |\psi_i\rangle\langle\psi_i|$, $\phi_i = |\phi_i\rangle\langle\phi_i|$, and

$$|\beta_1\rangle = U_1|0\rangle \quad |\beta_i\rangle = U_i|+\rangle \quad \forall i > 1 \qquad\qquad |\psi_i\rangle = U_i|0\rangle \quad |\phi_i\rangle = U_i|1\rangle \quad \forall i \ge 1.$$

Let the output state be $\rho = |\Psi\rangle\langle\Psi|$. By Definition 3.7, our proof of $\mathscr{F}$ implies that

$$\frac{n-1}{4} = \sum_{i=1}^{n-1}\mathrm{Tr}(|00\rangle\langle 00||++\rangle\langle++|) \le \sum_{i=1}^{n-1}\mathrm{Tr}[\rho_{i,i+1}(\beta_i\otimes\beta_{i+1})]. \tag{20}$$

STEP 2. This step mirrors Step 1, starting from the precondition $((|--\rangle\langle--|, |--\rangle\langle--|, \ldots, |--\rangle\langle--|) \mid \mathcal{P})$. The postcondition of the full circuit can then be computed as

$$\Big((\delta_1\otimes\delta_2, \ldots, \delta_{n-1}\otimes\delta_n)\ \Big|\ \mathcal{T}\Big), \quad \text{with} \quad \delta_i = |\delta_i\rangle\langle\delta_i|,\ \ |\delta_1\rangle = U_1|1\rangle,\ \ |\delta_i\rangle = U_i|-\rangle\ \ \forall i > 1$$

These conditions imply that

$$\frac{n-1}{4} \leq \sum_{i=1}^{n-1} \mathrm{Tr}[\rho_{i,i+1}(\delta_i \otimes \delta_{i+1})]. \tag{21}$$

STEP 3 (QAI influence on QHL). According to Theorem 2.2, Q implies that there exist complex numbers a and b such that $|a|^2 + |b|^2 = 1$, and the output state of the GHZ circuit is

$$|\Psi\rangle = a\,|\psi_1 \cdots \psi_n\rangle + b\,|\phi_1 \cdots \phi_n\rangle\,.$$

According to $\langle\psi_i|\phi_i\rangle = 0$, we have $\rho_{i,i+1} = |a|^2\psi_i \otimes \psi_{i+1} + |b|^2\phi_i \otimes \phi_{i+1}$. Then

$$\mathrm{Tr}[(\psi_1 \otimes \psi_2)(\beta_1 \otimes \beta_2)] = \frac{1}{2},\ \mathrm{Tr}[(\psi_i \otimes \psi_{i+1})(\beta_i \otimes \beta_{i+1})] = \frac{1}{4}\ \ \forall i > 1$$

$$\mathrm{Tr}[(\phi_1 \otimes \phi_2)(\beta_1 \otimes \beta_2)] = 0,\ \mathrm{Tr}[(\phi_i \otimes \phi_{i+1})(\beta_i \otimes \beta_{i+1})] = \frac{1}{4}\ \ \forall i > 1$$

$$\mathrm{Tr}[(\psi_1 \otimes \psi_2)(\delta_1 \otimes \delta_2)] = 0,\ \mathrm{Tr}[(\psi_i \otimes \psi_{i+1})(\delta_i \otimes \delta_{i+1})] = \frac{1}{4}\ \ \forall i > 1$$

$$\mathrm{Tr}[(\phi_1 \otimes \phi_2)(\delta_1 \otimes \delta_2)] = \frac{1}{2},\ \mathrm{Tr}[(\phi_i \otimes \phi_{i+1})(\delta_i \otimes \delta_{i+1})] = \frac{1}{4}\ \ \forall i > 1.$$

Equations (20) and (21) imply

$$\frac{n-1}{4} \leq |a|^2/2 + \sum_{i=2}^{n-1} \frac{|a|^2 + |b|^2}{4} = |a|^2/2 + \frac{n-2}{4} \Longrightarrow \frac{1}{2} \leq |a|^2$$

$$\frac{n-1}{4} \leq |b|^2/2 + \sum_{i=2}^{n-1} \frac{|a|^2 + |b|^2}{4} = |b|^2/2 + + \frac{n-2}{4} \Longrightarrow \frac{1}{2} \leq |b|^2$$

Together with $|a|^2 + |b|^2 = 1$, we have $|a|^2 = |b|^2 = \frac{1}{2}$. In other words, there exists θ such that

$$|\Psi\rangle = \frac{1}{\sqrt{2}}(|\psi_1 \cdots \psi_n\rangle + e^{i\theta}\,|\phi_1 \cdots \phi_n\rangle). \tag{22}$$

Equation (22) represents a closed-form expression for the circuit's output with a single unknown real parameter—a quantitative result unachievable by qualitative methods like QAI alone.

Moreover, the reasoning process scales with the number of qubits in the circuit: during the forward-reasoning process described above, for the reasoning steps carried out for each gate, the total size of the matrices that represent local observables and local projections in the pre- and post-conditions is always linear in the number of qubits. There are $O(n)$ gates; hence, the total amount of space needed to write down the SAQR-QC proof is $O(n^2)$.

4.2 Precision, Imprecision, and Principles for Selection of Local Observables

Inherent Imprecision: SAQR-QC characterizes the joint behavior of local reduced density matrices in a multi-qubit system, but it cannot capture global correlations that depend on relative phases. Because local reduced matrices cannot encode relative-phase information, any reasoning based solely on them necessarily cannot distinguish states that differ only in relative phase. For example, in the GHZ circuit, even if $U_i = I$, the states $\frac{1}{\sqrt{2}}(|0\cdots0\rangle \pm |1\cdots1\rangle)$ have identical k-qubit reduced density matrices for all $k < n$, namely $\rho_k = \frac{1}{2}(|0\cdots0\rangle\langle0\cdots0| + |1\cdots1\rangle\langle1\cdots1|)$, and are thus indistinguishable by SAQR-QC, even though they differ in their *relative-phase information θ in Equation* (22). In other words, while the local properties of each subsystem are identical, the relative-phase information that determines interference and correlations across all qubits is lost. As

discussed in the Introduction, the QAI-like and QHL-like components in SAQR-QC function as two components of a reduced product, in the sense of abstract interpretation. Consequently, SAQR-QC is inherently unable to express relative-phase information.

Principles for Selection of Local Observables: An important aspect in applying SAQR-QC is the systematic selection of local observables and the corresponding "local domains" relevant to proving a given property of a circuit. For instance, the domain can be chosen according to the circuit structure: as we show in §E, selecting the sets $\{(1,2),(1,3),(1,4),\dots\}$ as the abstract domain for the GHZ circuit of Figure 3 yields the same precision as Equation (22).

Given a global property of a quantum circuit, incorporating it into our framework requires constructing a tuple of local observables that can capture or closely approximate this property. This challenge touches on a fundamental theme in physics—the relationship between local and global properties—which has been extensively studied and for which many powerful techniques exist. Our approach can also draw inspiration from programming logic, such as best-effort decision procedures in separation logic [8, 12] and shape-analysis abstractions [21, 29, 37], which rely on local information to establish global properties and can guide the principled selection of appropriate local observables.

From these insights, we can state some guiding principles for selecting effective local domains based on circuit structure and entanglement patterns: (i) Balance precision and tractability while aligning with the properties of interest; in particular, approximate global properties using suitable local observables. (ii) Account for the circuit structure: for circuits consisting of relatively few two-qubit gates, consider all two-qubit pairs appearing in the gate pattern. (iii) Refine the domain when it is insufficiently precise for the reasoning task. We hope that future work will be able to refine and sharpen these principles.

Mitigation: For a given circuit, each proof in SAQR-QC generates an inequality involving the tuple of selected reduced density matrices. For example, in the derivation given in §4.1, Step 1 derives Equation (20), which only gives $|a|^2 > \frac{1}{2}$ and is insufficient to obtain Equation (22). By creating a second SAQR-QC proof with respect to a different precondition, and combining the two inequalities with the QAI component of the postcondition allowed us to derive $|a|^2 = \frac{1}{2}$. When reasoning with only one predicate, some properties may be lost or become indistinguishable due to the limited information it captures. By combining two or more predicates, we can cross-validate and recover information that would otherwise be missed, improving precision and reducing the "failure" cases that arise from single-predicate reasoning. Nevertheless, there exist properties that cannot be captured solely by tuples of local reduced density matrices; in such cases, SAQR-QC alone is insufficient, and integrating it with some non-local assertions as preconditions—such as efficiently representable non-local stabilizers compatible with local reasoning—may be necessary.

5 Quantum Phase Estimation

Quantum Phase Estimation (QPE) is a fundamental quantum algorithm that, given a unitary U and an eigenstate $|\psi\rangle$, estimates the eigenphase ϕ, where $U|\psi\rangle = e^{2\pi i\phi}|\psi\rangle$. The algorithm uses a register of ancilla qubits (whose size determines precision), applies controlled-U operations, and then takes an inverse QFT to concentrate amplitude on the binary fraction closest to ϕ. The success probability of QPE is the probability that the measurement outcome is the best approximation to the true eigenphase representable in the ancilla register. For a single run, this probability is bounded below by $4/\pi^2 \approx 0.405$, and it increases with additional qubits or repeated executions, ensuring a high likelihood of correct best phase estimation. We apply SAQR-QC to analyze the QPE algorithm, whose intricate entanglement and delicate success probabilities have historically resisted scalable formal verification. The presentation has two stages: first, we establish the correctness of the Quantum Fourier Transform (QFT) via QAI; second, we leverage this foundation to analyze

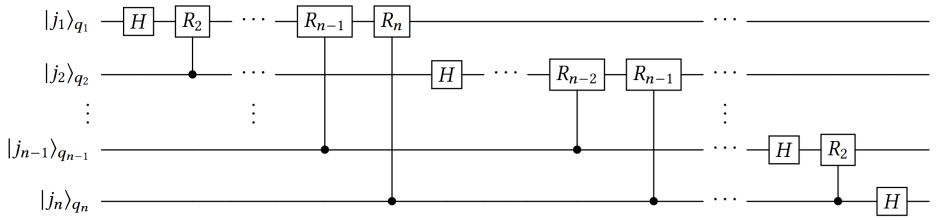


Fig. 4. Quantum Fourier Transform: swap gates that reverse the qubit order at the circuit's end are omitted.

the full QPE circuit using SAQR-QC, deriving a lower bound on its success probability. To the best of our knowledge, this argument constitutes the first scalable formal verification of QPE and QFT, offering a compositional, tractable methodology for reasoning about both correctness and probabilistic performance in key quantum algorithms.

5.1 Quantum Fourier Transform (QFT)

The QFT—the quantum analogue of the discrete Fourier transform (see Figure 4)—forms the computational core of landmark algorithms such as Shor's factoring and QPE. We introduce a lossless local-reasoning framework for the QFT grounded in QAI. Our method derives an abstract output state whose concretization exactly coincides with the true QFT output for all computational-basis inputs. Moreover, SAQR-QC provides proofs that are polynomial in the size of the program while achieving exact semantic correspondence with the true QFT output—and, by extension, for QPE.

The circuit uses Hadamard and R_m gates (see §2); for $m > 2$, R_m lies outside the Clifford group. We use $\psi = |\psi\rangle\langle\psi|$ for pure state $|\psi\rangle$,$0.x_1x_2\cdots x_n = \sum_{i=1}^{n} \frac{x_i}{2^i}$, and the following notation:

$$|\psi_x\rangle := \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i 0.x}\,|1\rangle). \tag{23}$$

We choose the domain $(\{1\},\{2\},\ldots,\{n\})$. For the input state $|j_1\rangle_{q_1}\otimes|j_2\rangle_{q_2}\otimes\cdots\otimes|j_{n-1}\rangle_{q_{n-1}}\otimes|j_n\rangle_{q_n}$, we set the precondition to be $\mathcal{P} = (P_1, P_2, \cdots, P_n) = (|j_1\rangle\langle j_1|, |j_2\rangle\langle j_2|, \cdots, |j_n\rangle\langle j_n|)$. In §F, we prove the following property using QAI:

$$\models^{QAI} \{\mathcal{P}\}\ QFT\ \{(\psi_{j_n}, \psi_{j_{n-1}j_n}, \cdots, \psi_{j_2\cdots j_n}, \psi_{j_1\cdots j_n})\} \tag{24}$$

The postcondition derived from QAI is an abstract state represented as a tuple of density matrices corresponding to pure quantum states. By applying the concretization function from Definition 2.2, we infer that the concrete state lies in the subspace

$$\psi_{j_n}\otimes\psi_{j_{n-1}j_n}\otimes\cdots\otimes\psi_{j_2\cdots j_n}\otimes\psi_{j_1\cdots j_n}.$$

Because the space so defined is a 1-dimensional subspace, it follows that the density matrix of the output state must exactly equal the pure-state projection onto this vector—that is, $\psi_{j_n}\otimes\psi_{j_{n-1}j_n}\otimes\cdots\otimes\psi_{j_2\cdots j_n}\otimes\psi_{j_1\cdots j_n}$.

During the forward-reasoning process described above, for the reasoning steps carried out for each gate, the total size of the matrices that represent local observables and local projections in the pre- and post-conditions is always linear in the number of qubits. There are $O(n^2)$ gates; hence, the total amount of space needed to write down the proof is $O(n^3)$.

5.2 Quantum Phase Estimation (QPE)

In this section, we present a quantitative local analysis of the QPE algorithm by combining both backward and forward reasoning techniques. We begin by decomposing the QPE algorithm into

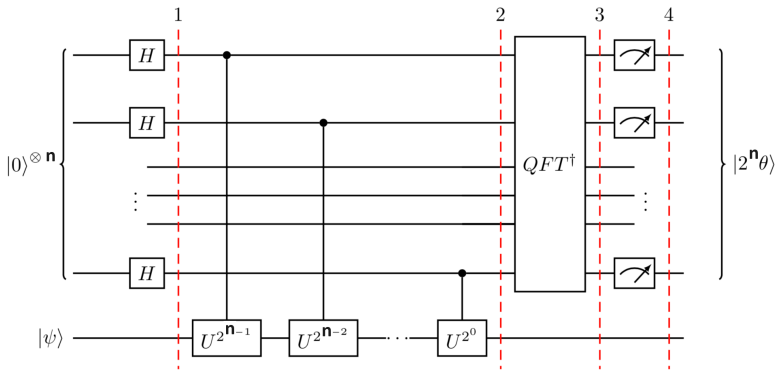


Fig. 5. Quantum Phase Estimation: $U|\psi\rangle = e^{i\theta}|\psi\rangle$. We only consider the circuit without the measurements.

three constituent circuits and apply SAQR-QC to reason about each component individually—see Figure 5. For an unknown phase θ and a given constant k, SAQR-QC can formally establish that—with probability at least $\frac{4}{\pi^2}$—the QPE algorithm produces an output bitstring whose last k bits constitute the optimal k-bit approximation to the least-significant k bits of any n-bit binary representation/approximation of θ.

We denote by $\mathbf{C_1}$ the segment of the program that precedes the application of the inverse Quantum Fourier Transform, $\mathcal{QFT}^{\dagger}$. The $\mathcal{QFT}^{\dagger}$ operation can be decomposed into two parts: the initial sequence of swap gates that reverses the order of the qubits, and the subsequent controlled rotation gates that implement the core of the inverse Fourier transform. We use $\mathbf{C_2}$ to refer specifically to the subcircuit following the swap gates within $\mathcal{QFT}^{\dagger}$.

Our analysis focuses on the least significant k qubits, and proceeds in three steps. We will rely on the notation introduced in Equation (23), and apply the QAI-based reasoning framework introduced in §3.6 to verify the behavior of $\mathbf{C_1}$.

Step 1: Reasoning about $\mathbf{C_2}$. Let U be the circuit corresponding to $\mathbf{C_2}$. For $\mathbf{C_2}$, we choose the domain of the predicate as the first k qubits of the first n qubits, together with the last m qubits—the qubits that U acts upon. In §G.1, we use backward reasoning to show

$$\{\mathscr{A} \mid \mathcal{I}\}\ \mathbf{C_2}\ \{(|j_{n-k+1}\cdots j_n\rangle\langle j_{n-k+1}\cdots j_n|) \mid \mathcal{I}\}, \tag{25}$$

where $\mathscr{A} = (|\tau\rangle\langle\tau| \otimes |\psi\rangle\langle\psi|)$, $|\tau\rangle = |\psi_{j_n}\rangle \otimes \cdots \otimes |\psi_{j_{n-k+1}\cdots j_n}\rangle$, and $M_{\mathscr{A}} = I_{1,\cdots,n-k} \otimes |\tau\rangle\langle\tau| \otimes |\psi\rangle\langle\psi|$.

Remark: Matrix multiplication is efficient for constant dimension. Here, the matrices are $s^k \times 2^k$, which is a constant independent of n.

Step 2: Reasoning about SWAP gates. The precondition of Equation (25), $\{\mathscr{A} \mid \mathcal{I}\}$, becomes the postconditon of the SWAP gates. Because the action of the SWAP gates only changes the last k qubits into the first k qubits, we can write the precondition of the SWAP gates as $\{\mathscr{A}' \mid \mathcal{I}\}$ with $\mathscr{A}' = (|\tau\rangle\langle\tau| \otimes |\psi\rangle\langle\psi|)$ on the first k qubits and the last m qubits. These arguments prove

$$\{\mathscr{A}' \mid \mathcal{I}\}\ \mathbf{S}\ \{\mathscr{A} \mid \mathcal{I}\} \tag{26}$$

where we use $\mathbf{S}$ to denote the SWAP gates.[3]

[3]We change the abstract domain for simplicity of presentation. This change does not affect our statement's correctness because the SWAP gates' action is clear. If we want to fix the abstract domain, we can consider $(s_1, \cdots, s_m)$ with s_1 being the last k qubits together with the last m qubits; s_2 being the result of applying the first SWAP gate on s_1; $\cdots$; s_m being the result of applying the last SWAP gate on s_{m-1}.

Step 3: Reasoning about $\mathbf{C_1}$. For $\mathbf{C_1}$, we select the predicate domain to include the first k qubits of the initial n and the last m qubits where U acts. In §G.2, we prove the triple

$$\{r|0\cdots0\rangle\langle 0\cdots0|\otimes|\psi\rangle\langle\psi|) \mid (|0\cdots0\rangle\langle 0\cdots0|\otimes|\psi\rangle\langle\psi|\}\ \mathbf{C_1}\ \{\mathscr{A}' \mid \mathcal{P}\} \tag{27}$$

where

$$\begin{aligned} r &= \Pi_{t=1}^{k}\cos^2[(2^{n-t}\theta - 0.j_{n-t+1}\cdots j_n)\pi] = \frac{\sin^2(2^n\theta\pi)}{4^k\sin^2[(2^{n-k}\theta - 0.j_{n-k+1}\cdots j_n)\pi]} \\ \mathcal{P} &= (|\omega\rangle\langle\omega|\otimes|\psi\rangle\langle\psi|) \\ |\omega\rangle &= \frac{1}{2^{k/2}}(|0\rangle + e^{2\pi i 2^{n-1}\theta}|1\rangle)\otimes\cdots\otimes(|0\rangle + e^{2\pi i 2^{n-k}\theta}|1\rangle). \end{aligned}$$

Let $U|\psi\rangle = e^{i\theta}|\psi\rangle$, and $\theta = \frac{a}{2^n} + \epsilon$ with $-\frac{1}{2^{n+1}} \le \epsilon \le \frac{1}{2^{n+1}}$ and a is an integer with binary representation $a_1a_2\cdots a_n$. For $j_{n-k}\cdots j_n = a_{n-k}\cdots a_n$, we have

$$r = \frac{\sin^2(2^n\theta\pi)}{4^k\sin^2[(2^{n-k}\theta - 0.j_{n-k+1}\cdots j_n)\pi]} = \frac{\sin^2(2^n\epsilon\pi)}{4^k\sin^2(2^{n-k}\epsilon\pi)} \ge \frac{|2\cdot 2^n\epsilon|^2}{4^k\cdot|2^{n-k}\epsilon\cdot\pi|^2} \ge \frac{4}{\pi^2}.$$

Together with the Seq Rule and the Con Rule, the proved result can be interpreted as follows: for any constant k, the last k bits of the output will—with probability at least $\frac{4}{\pi^2}$—match the best k-bit binary approximation to the least significant k bits of the phase θ, provided that the first k input qubits are initialized to $|0,\ldots,0\rangle$, regardless of the state of the remaining $n-k$ qubits.

6 Related Work

Combining Logic and Abstract Interpretation. Combining Hoare logic with abstract interpretation is a well-established strategy in classical program analysis. Many software model-checking tools integrate logic-driven state-space exploration with dataflow information obtained via abstract interpretation, often computed as a preliminary pass. A prominent example is the SeaHorn solver for Constrained Horn Clauses, whose Crab component provides abstract-interpretation capabilities [17, 22]. This integration illustrates how abstract interpretation can complement logical reasoning, resulting in verification that is both scalable and precise. Our work extends this principle to the quantum domain, where the combination supports reasoning about program semantics that would be difficult to achieve with either technique in isolation.

QHL. There are a variety of quantum Hoare logics. A comparative study in [34] examined three representative logics [13, 25, 46]. Quantum Hoare logics can be broadly categorized into expectation-based and satisfaction-based approaches. Following the seminal work of D'Hondt and Panangaden [18], expectation-based approaches [7, 20, 30, 46] use positive operators as assertions for quantum states, defining the expectation that a state ρ satisfies an assertion M as $\mathrm{Tr}(M\rho)$. In contrast, satisfaction-based logics—exemplified by Zhou et al. [50] and indepedently by Unruh [44, 45]—treat subspaces of the Hilbert space as assertions, providing a direct semantic correspondence between program correctness and the geometry of quantum states. Here, a state ρ satisfies an assertion P if the support of ρ is included in P. In all of these works, predicates M or P are represented as $2^n\times 2^n$ matrices, which precludes efficient computation and limits scalability for larger quantum systems.

Quantum Separation Logic. Quantum reasoning frameworks have made significant progress along multiple fronts. Quantum separation logic, introduced in [49], extends the principles of Bunched Implications [32] to the quantum setting, enabling local reasoning about quantum programs. The framework in [27] further incorporates hybrid state spaces by supporting both classical variables and the dynamic allocation and deallocation of quantum qubits. However, in both approaches, the separating conjunction is interpreted strictly as a tensor product—corresponding to quantum independence—which substantially limits the expressiveness and applicability of the logic. In

addition, because predicates are restricted to projection operators, these systems support only qualitative rather than quantitative reasoning about quantum states and their interactions.

QAI. [48] presented an approach to quantum abstract interpretation for reasoning about quantum circuits, using the satisfaction-based approach. Other works, such as [9], investigate the abstract interpretation of quantum programming using variants of the Gottesman-Knill theorem.

View abstraction. In classical program verification, *view abstraction* [4] bears a strong resemblance to QAI: both track sets of small-size abstractions of a system. This idea of focusing on compact representations was already used in shape-analysis frameworks [29, 37], highlighting a conceptual connection between SAQR-QC and classical shape analysis. A key difference is that view abstraction targets parameterized model checking, while SAQR-QC currently applies to a given circuit with a fixed number of qubits. Nevertheless, this connection suggests an exciting avenue for future work: if SAQR-QC can be mechanized, it may be possible to extend it to parameterized quantum circuits [3]. One could adopt ideas from view abstraction, such as dynamically detecting cut-off points beyond which the state-space search need not continue, or employing the heuristic that when views of size k fail, switch to views of size $k + 1$.

Symbolic abstraction, strongest consequence, and weakest sufficient condition. The inexpressibility issues discussed in §3.2 are a manifestation of the constraints that one faces when working with an "impoverished" logic (or logic fragment). These issues have been studied in the context of abstract interpretation as what is (now) called the *symbolic-abstraction* problem [35, 43], and phrased in purely logical terms as the *strongest-consequence* problem [36, §5], as follows:

Given formula $\varphi \in \mathcal{L}$, and another logic $\mathcal{L}'$, find the strongest formula $\psi \in \mathcal{L}'$ such that $\varphi \vDash \psi$.

The strongest-consequence problem naturally arises in approximate forwards reasoning, to over-approximate a postcondition. The discussion in §3.2 concerned backwards reasoning for which one faces the dual problem, the *weakest sufficient-condition* problem:

Given formula $\varphi \in \mathcal{L}$, and a different logic $\mathcal{L}'$, find the weakest formula $\chi \in \mathcal{L}'$ such that $\chi \vDash \varphi$.

As we saw in §3.2, the strongest consequence or weakest sufficient condition may not always be expressible in $\mathcal{L}'$, in which case one has to fall back on finding what one hopes is a suitably strong consequence or a suitably weak sufficient condition, respectively. Scherpelz et al. [39] presented a best-effort method for computing sufficient conditions as part of an algorithm for creating abstract transformers for use with parameterized predicate abstraction [15]. Their method performs weakest liberal precondition (WLP) of a post-state predicate with respect to a concrete transformer τ, and then uses heuristics to identify combinations of pre-state predicates that entail the WLP value.

Reasoning about Shor's factoring. The QFT circuit contains many non-Clifford gates, so variants of the Gottesman-Knill theorem do not directly apply. Shor's algorithm has been formally verified in [20, 33]. The former uses an expectation-based approach with classical variables, while the latter employs the Coq proof assistant and the Small Quantum Intermediate Representation (sQIR) [24]. Since Coq operates symbolically, the reasoning about quantum phase estimation in [33] is symbolic.

Symbolic verification. There has been work to extend symbolic-verification techniques to the quantum domain [2, 14], using logic- and automata-based techniques developed for symbolic verification of classical programs to analyze the correctness of quantum programs. In contrast, SAQR-QC is not symbolic in nature. Reasoning steps in SAQR-QC can involve matrix multiplications and other mathematical operations on specific values of specific sizes.

Packing of variables in abstract domains. The motivation for using local observables and local projections defined with respect to a tuple $(s_1, \cdots, s_m)$ of sets of qubit indexes is to make reasoning

scalable: each reasoning step involves only a small number of qubits. This idea is similar to the idea of "packing" variables in numeric domains, as used in Astrée [11]: a program's numeric variables are placed in sets ("packs") so that each abstract transformer of a numeric abstract domain can operate on a single pack at a time. As with our qubit sets, each variable can be in multiple packs.

7 Conclusion and Future work

This paper introduces a framework for local reasoning about quantum circuits that is both scalable and addresses quantitative properties. Our logic, SAQR-QC, combines QHL, which tracks quantitative properties of quantum states, with QAI, which provides scalable reasoning about qualitative properties. Unlike a prior scalable framework that is restricted to Clifford circuits [1], SAQR-QC supports non-Clifford gates, giving it much broader applicability. By tracking tuples of reduced density matrices via tuples of local observables, SAQR-QC fills a long-standing gap in methods for reasoning about quantum programs, enabling scalable and quantitative verification of quantum circuits on classical computers. We apply SAQR-QC to GHZ circuits with non-Clifford unitaries, successfully characterizing the output state up to a relative phase. We demonstrate its use on QFT, and derive a lower bound on the success probability of QPE. Avenues for future research include

(1) **Other applications.** Apply SAQR-QC to other circuits, such as W-state circuits and the Bernstein–Vazirani (BV) algorithm, as well as other important classes of quantum algorithms, including parameterized quantum circuits. SAQR-QC is well-suited to quantum circuits composed of unitary gates acting on a constant number of qubits. The W-state circuit appears to be a particularly suitable target. A key advantage of the W-state circuit is that the collection of all single-qubit reduced density matrices contains sufficient information to uniquely identify the state within the continuous family of W-type states [38, 47]. The BV circuit requires a model describing the oracle, which remains manageable, but could introduce additional complexity.
(2) **Quantum programs with measurements and classical control.** Develop the theoretical and technical foundations for handling quantum programs that include measurements, branching, classical control, and loops. Currently, SAQR-QC applies to measurement-free circuits, rather than a full programming language with these features, and while we anticipate such extensions are feasible, they introduce additional challenges. In particular, in the presence of loops, verification typically relies on suitable loop invariants, which must be both sufficiently strong to establish the desired property and expressible within the assertion language of SAQR-QC. This requirement may rule out certain properties that could otherwise be handled in richer logical frameworks.
(3) **Mechanized proof search.** We face several challenges if we are to automate reasoning with SAQR-QC. On the theoretical side, a key open problem is how to systematically construct suitable local observables from desired global properties. On the practical side, our approach relies on semidefinite programming (SDP) solvers that can produce corresponding preimages of the optimization outputs, which can serve as candidate predicates or local observables. Given a collection of such candidates, a further challenge is to select high-quality ones effectively.
(4) **Integration with symbolic methods.** Our approach leverages numerical analysis to enable scalable reasoning about general quantum circuits, even in the absence of known algebraic structure. This situation is complementary to existing mechanized proof frameworks (e.g., VyZX [28] and VOQC [24]), which rely primarily on symbolic verification. By generating candidate predicates and invariants numerically, our method opens a pathway for integrating numerical and symbolic approaches, expanding the class of quantum programs amenable to mechanized reasoning.

Acknowledgments

Supported, in part, by NSF grant OMA-2016245 and DARPA grant HR00112520301.

References

[1] Scott Aaronson and Daniel Gottesman. 2004. Improved simulation of stabilizer circuits. *Physical Review A* 70, 5 (2004), 052328. doi:10.1103/PhysRevA.70.052328

[2] Parosh Aziz Abdulla. 2025. A Symbolic Approach to Verifying Quantum Systems. *Commun. ACM* 68, 6 (June 2025), 84. doi:10.1145/3725725

[3] Parosh Aziz Abdulla, Yu-Fang Chen, Michal Hečko, Lukáš Holík, Ondřej Lengál, Jyun-Ao Lin, and Ramanathan S. Thinniyam. 2026. Parameterized Verification of Quantum Circuits. *Proc. ACM Program. Lang.* 10, POPL, Article 70 (Jan. 2026), 30 pages. doi:10.1145/3776712

[4] Parosh Aziz Abdulla, Frédéric Haziza, and Lukás Holík. 2013. All for the Price of Few. In *Verification, Model Checking, and Abstract Interpretation (VMCAI 2013) (Lecture Notes in Computer Science, Vol. 7737)*. Springer, 476–495. doi:10.1007/978-3-642-35873-9_28

[5] Matthew Amy and Joseph Lunderville. 2025. Linear and Non-linear Relational Analyses for Quantum Program Optimization. *Proc. ACM Program. Lang.* 9, POPL, Article 37 (Jan. 2025), 32 pages. doi:10.1145/3704873

[6] Alán Aspuru-Guzik, Anthony D. Dutoi, Peter J. Love, and Martin Head-Gordon. 2005. Simulated Quantum Computation of Molecular Energies. *Science* 309, 5741 (2005), 1704–1707. doi:10.1126/science.1113479

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

[8] Josh Berdine, Cristiano Calcagno, and Peter W. O'Hearn. 2005. Smallfoot: Modular Automatic Assertion Checking with Separation Logic. In *International Symposium on Formal Methods for Components and Objects (FMCO 2005) (Lecture Notes in Computer Science, Vol. 4111)*. Springer, 115–137. doi:10.1007/11804192_6

[9] Benjamin Bichsel, Anouk Paradis, Maximilian Baader, and Martin Vechev. 2023. Abstraqt: Analysis of Quantum Circuits via Abstract Stabilizer Simulation. *Quantum* 7 (Nov. 2023), 1185. doi:10.22331/q-2023-11-20-1185

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

[11] Bruno Blanchet, Patrick Cousot, Radhia Cousot, Jérôme Feret, Laurent Mauborgne, Antoine Miné, David Monniaux, and Xavier Rival. 2003. A static analyzer for large safety-critical software. In *Proceedings of the ACM SIGPLAN 2003 Conference on Programming Language Design and Implementation 2003, San Diego, California, USA, June 9-11, 2003*, Ron Cytron and Rajiv Gupta (Eds.). ACM, 196–207. doi:10.1145/781131.781153

[12] Cristiano Calcagno, Dino Distefano, Peter O'Hearn, and Hongseok Yang. 2009. Compositional Shape Analysis by Means of Bi-Abduction. In *Proceedings of the 36th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL)*. ACM, 289–300. doi:10.1145/2049697.2049700

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

[14] Yu-Fang Chen, Kai-Min Chung, Ondřej Lengál, Jyun-Ao Lin, Wei-Lun Tsai, and Di-De Yen. 2025. An Automata-Based Framework for Verification and Bug Hunting in Quantum Circuits. *CACM* 68, 6 (2025), 85–93. doi:10.1145/3591270

[15] Patrick Cousot. 2003. Verification by Abstract Interpretation. In *Verification: Theory and Practice, Essays Dedicated to Zohar Manna on the Occasion of His 64th Birthday (Lecture Notes in Computer Science, Vol. 2772)*, Nachum Dershowitz (Ed.). Springer, 243–268. doi:10.1007/978-3-540-39910-0_11

[16] Patrick Cousot and Radhia Cousot. 1979. Systematic Design of Program Analysis Frameworks. In *Conference Record of the Sixth Annual ACM Symposium on Principles of Programming Languages, San Antonio, Texas, USA, January 1979*, Alfred V. Aho, Stephen N. Zilles, and Barry K. Rosen (Eds.). ACM Press, 269–282. doi:10.1145/567752.567778

[17] Crab 2023. Crab: A C++ Library for Building Program Static Analyses. https://github.com/seahorn/crab?tab=readme-ov-file. GitHub repository, accessed: 2023-10-26.

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

[19] Dino Distefano, Peter W. O'Hearn, and Hongseok Yang. 2006. A Local Shape Analysis Based on Separation Logic. In *Tools and Algorithms for the Construction and Analysis of Systems, 12th International Conference, TACAS 2006 Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2006, Vienna, Austria, March 25 - April 2, 2006, Proceedings (Lecture Notes in Computer Science, Vol. 3920)*, Holger Hermanns and Jens Palsberg (Eds.). Springer, 287–302. doi:10.1007/11691372_19

[20] Yuan Feng and Mingsheng Ying. 2021. Quantum Hoare Logic with Classical Variables. *ACM Transactions on Quantum Computing* 2, 4, Article 16 (Dec. 2021), 43 pages. doi:10.1145/3456877

[21] Denis Gopan, Frank DiMaio, Nurit Dor, Thomas W. Reps, and Mooly Sagiv. 2004. Numeric Domains with Summarized Dimensions. In *Tools and Algorithms for the Construction and Analysis of Systems (TACAS) (Lecture Notes in Computer Science, Vol. 2988)*. Springer, 512–529. doi:10.1007/978-3-540-24730-2_38

[22] Arie Gurfinkel and Jorge A. Navas. 2021. Abstract Interpretation of LLVM with a Region-Based Memory Model. In *Verified Software: Theories, Tools, and Experiments (VSTTE) (Lecture Notes in Computer Science, Vol. 12900)*. Springer,

Berlin, Heidelberg, 122–144. doi:10.1007/978-3-030-95561-8_8

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

[24] Kesha Hietala, Robert Rand, Liyi Li, Shih-Han Hung, Xiaodi Wu, and Michael Hicks. 2023. A Verified Optimizer for Quantum Circuits. *ACM Trans. Program. Lang. Syst.* 45, 3, Article 18 (Sept. 2023), 35 pages. doi:10.1145/3604630

[25] Yoshihiko Kakutani. 2009. A Logic for Formal Verification of Quantum Programs. In *Advances in Computer Science - ASIAN 2009. Information Security and Privacy*, Anupam Datta (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 79–93. doi:10.1007/978-3-642-10622-4_7

[26] A. Yu. Kitaev. 1995. Quantum measurements and the Abelian Stabilizer Problem. *arXiv:quant-ph/9511026* (1995), 22 pages.

[27] Xuan-Bach Le, Shang-Wei Lin, Jun Sun, and David Sanan. 2022. A Quantum Interpretation of Separating Conjunction for Local Reasoning of Quantum Programs Based on Separation Logic. *Proc. ACM Program. Lang.* 6, POPL, Article 36 (jan 2022), 27 pages. doi:10.1145/3498697

[28] Adrian Lehmann, Ben Caldwell, Bhakti Shah, William Spencer, and Robert Rand. 2026. VyZX: Formal Verification of a Graphical Quantum Language. *ACM Trans. Program. Lang. Syst.* (2026). doi:10.1145/3807780

[29] Tal Lev-Ami, Thomas W. Reps, Mooly Sagiv, and Reinhard Wilhelm. 2000. Putting Static Analysis to Work for Verification: A Case Study. In *International Symposium on Software Testing and Analysis (ISSTA)*. ACM, 26–38. doi:10.1145/347324.348031

[30] Yangjia Li and Dominique Unruh. 2021. Quantum Relational Hoare Logic with Expectations. In *Proceedings of the 48th International Colloquium on Automata, Languages, and Programming*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 136:1–136:20. doi:10.4230/LIPICS.ICALP.2021.136

[31] Michael A. Nielsen and Isaac L. Chuang. 2011. *Quantum Computation and Quantum Information: 10th Anniversary Edition* (10th ed.). Cambridge University Press, New York, NY, USA.

[32] Peter W. O'Hearn and David J. Pym. 1999. The Logic of Bunched Implications. *Bulletin of Symbolic Logic* 5, 2 (1999), 215–244. doi:10.2307/421090

[33] Yuxiang Peng, Kesha Hietala, Runzhou Tao, Liyi Li, Robert Rand, Michael Hicks, and Xiaodi Wu. 2023. A formally certified end-to-end implementation of Shor's factorization algorithm. *Proceedings of the National Academy of Sciences* 120, 21 (2023), e2218775120. arXiv:https://www.pnas.org/doi/pdf/10.1073/pnas.2218775120 doi:10.1073/pnas.2218775120

[34] Robert Rand. 2019. Verification logics for quantum programs. *arXiv preprint arXiv:1904.04304* (2019), 42 pages.

[35] Thomas W. Reps, Shmuel Sagiv, and Greta Yorsh. 2004. Symbolic Implementation of the Best Transformer. In *Verification, Model Checking, and Abstract Interpretation, 5th International Conference, VMCAI 2004, Venice, Italy, January 11-13, 2004, Proceedings (Lecture Notes in Computer Science, Vol. 2937)*, Bernhard Steffen and Giorgio Levi (Eds.). Springer, 252–266. doi:10.1007/978-3-540-24622-0_21

[36] Thomas W. Reps and Aditya V. Thakur. 2016. Automating Abstract Interpretation. In *Verification, Model Checking, and Abstract Interpretation - 17th International Conference, VMCAI 2016, St. Petersburg, FL, USA, January 17-19, 2016. Proceedings (Lecture Notes in Computer Science, Vol. 9583)*, Barbara Jobstmann and K. Rustan M. Leino (Eds.). Springer, 3–40. doi:10.1007/978-3-662-49122-5_1

[37] Mooly Sagiv, Thomas W. Reps, and Reinhard Wilhelm. 2002. Parametric Shape Analysis via 3-Valued Logic. *ACM Transactions on Programming Languages and Systems* 24, 3 (2002), 217–298. doi:10.1145/514188.514190

[38] Adam Sawicki, Michael Walter, and Marek Kuś. 2013. When is a pure state of three qubits determined by its single-particle reduced density matrices? *Journal of Physics A: Mathematical and Theoretical* 46, 055304 (2013), 16 pages. doi:10.1088/1751-8113/46/5/055304

[39] Erika Rice Scherpelz, Sorin Lerner, and Craig Chambers. 2007. Automatic inference of optimizer flow functions from semantic meanings. In *Proceedings of the ACM SIGPLAN 2007 Conference on Programming Language Design and Implementation, San Diego, California, USA, June 10-13, 2007*, Jeanne Ferrante and Kathryn S. McKinley (Eds.). ACM, 135–145. doi:10.1145/1250734.1250750

[40] P. Shor. 1997. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM J. Comput.* 26, 5 (1997), 1484–1509. doi:10.1137/S0097539795293172

[41] Runzhou Tao, Yunong Shi, Jianan Yao, John Hui, Frederic T. Chong, and Ronghui Gu. 2021. Gleipnir: Toward Practical Error Analysis for Quantum Programs. In *Proceedings of the 42nd ACM SIGPLAN International Conference on Programming Language Design and Implementation* (Virtual, Canada) *(PLDI 2021)*. Association for Computing Machinery, New York, NY, USA, 48–64. doi:10.1145/3453483.3454029

[42] Runzhou Tao, Yunong Shi, Jianan Yao, Xupeng Li, Ali Javadi-Abhari, Andrew W. Cross, Frederic T. Chong, and Ronghui Gu. 2022. Giallar: Push-button verification for the Qiskit Quantum compiler. In *Proceedings of the 43rd ACM SIGPLAN International Conference on Programming Language Design and Implementation* (San Diego, CA, USA) *(PLDI 2022)*. Association for Computing Machinery, New York, NY, USA, 641–656. doi:10.1145/3519939.3523431

[43] Aditya V. Thakur and Thomas W. Reps. 2012. A Method for Symbolic Computation of Abstract Operations. In *Computer Aided Verification - 24th International Conference, CAV 2012, Berkeley, CA, USA, July 7-13, 2012 Proceedings (Lecture Notes in Computer Science, Vol. 7358)*, P. Madhusudan and Sanjit A. Seshia (Eds.). Springer, 174–192. doi:10.1007/978-3-642-31424-7_17

[44] Dominique Unruh. 2019. Quantum Hoare Logic with Ghost Variables. In *34th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2019, Vancouver, BC, Canada, June 24-27, 2019*. IEEE, 1–13. doi:10.1109/LICS.2019.8785779

[45] Dominique Unruh. 2019. Quantum relational Hoare logic. *Proceedings of the ACM on Programming Languages* 3, POPL (2019), 1–31. doi:10.1145/3290346

[46] Mingsheng Ying. 2011. Floyd–Hoare logic for quantum programs. *ACM Transactions on Programming Languages and Systems (TOPLAS)* 33, 6 (2011), 19:1–19:49. doi:10.1145/2049706.2049708

[47] Nengkun Yu. 2013. Multi-partite W-type state is determined by its single-particle reduced density matrices. *Physical Review A* 87, 5 (2013), 052310. doi:10.1103/PhysRevA.87.052310

[48] Nengkun Yu and Jens Palsberg. 2021. Quantum Abstract Interpretation. In *Proceedings of the 42th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI 2021)*. Association for Computing Machinery, New York, NY, USA, 542–558. doi:10.1145/3453483.3454061

[49] Li Zhou, Gilles Barthe, Justin Hsu, Mingsheng Ying, and Nengkun Yu. 2021. A Quantum Interpretation of Bunched Logic for Quantum Separation Logic. In *2021 36th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*. IEEE, 1–14. doi:10.1109/LICS52264.2021.9470673

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