A Practical Quantum Hoare Logic

Mingsheng Ying

Centre for Quantum Software and Information
University of Technology Sydney

Outline

Outline

Quantum programming platforms

- Qiskit @ IBM
- Q# @ Microsoft

[1] B. Heim, M. Soeken, S. Marshall, C. Granade, M. Roetteler, A. Geller, S. Troyer and K. Svore, Quantum programming languages, *Nature Reviews Physics* 2020.

Quantum programming platforms

- Qiskit @ IBM
- Cirq @ Google

Q# @ Microsoft

Braket @ AWS

[1] B. Heim, M. Soeken, S. Marshall, C. Granade, M. Roetteler, A. Geller, S. Troyer and K. Svore, Quantum programming languages, *Nature Reviews Physics* 2020.

How to verify quantum programs?

Quantum programming platforms

- Qiskit @ IBM
- Cirq @ Google
- TKET @ Quantinuum

Q# @ Microsoft

Braket @ AWS

[1] B. Heim, M. Soeken, S. Marshall, C. Granade, M. Roetteler, A. Geller, S. Troyer and K. Svore, Quantum programming languages, *Nature Reviews Physics* 2020.

How to verify quantum programs?

Quantum programming platforms

- Qiskit @ IBM
- Cirq @ Google
- TKET @ Quantinuum
- .....

Q# @ Microsoft

Braket @ AWS

[1] B. Heim, M. Soeken, S. Marshall, C. Granade, M. Roetteler, A. Geller, S. Troyer and K. Svore, Quantum programming languages, *Nature Reviews Physics* 2020.

How to verify quantum programs?

Quantum programming platforms

- Qiskit @ IBM
- Cirq @ Google
- TKET @ Quantinuum
- .....

Q# @ Microsoft

Braket @ AWS

[1] B. Heim, M. Soeken, S. Marshall, C. Granade, M. Roetteler, A. Geller, S. Troyer and K. Svore, Quantum programming languages, *Nature Reviews Physics* 2020.

How to verify quantum programs?

- How can we develop a Hoare-style logic for quantum programs?

Outline

Programming Language **qWhile**

$$
\begin{aligned}
S ::= \ & \textbf{skip} \mid q := |0\rangle \\
& \mid S_1; S_2 \\
& \mid \bar{q} := U[\bar{q}] \\
& \mid \textbf{if } (\square m \cdot M[\bar{q}] = m \to S_m) \textbf{ fi} \\
& \mid \textbf{while } M[\bar{q}] = 1 \textbf{ do } S \textbf{ od}
\end{aligned}
$$

Example: Quantum Walk

- Quantum walk on an n-circle with an absorbing boundary at position 1.

Example: Quantum Walk

- Quantum walk on an n-circle with an absorbing boundary at position 1.
- $\mathcal{H}_c$ —— 2-dimensional Hilbert space with basis states $|L\rangle$ and $|R\rangle$, indicating directions.

Example: Quantum Walk

- Quantum walk on an n-circle with an absorbing boundary at position 1.
- $\mathcal{H}_c$ —— 2-dimensional Hilbert space with basis states $|L\rangle$ and $|R\rangle$, indicating directions.
- $\mathcal{H}_p$ —— n-dimensional Hilbert space withl basis states $|0\rangle, |1\rangle, ..., |n-1\rangle$, denoting positions.

Example: Quantum Walk

- Quantum walk on an n-circle with an absorbing boundary at position 1.
- $\mathcal{H}_c$ —— 2-dimensional Hilbert space with basis states $|L\rangle$ and $|R\rangle$, indicating directions.
- $\mathcal{H}_p$ —— n-dimensional Hilbert space withl basis states $|0\rangle, |1\rangle, ..., |n-1\rangle$, denoting positions.
- Quantum walk —— composite system of a coin and a walker moving on these positions.

Example: Quantum Walk

- Quantum walk on an n-circle with an absorbing boundary at position 1.
- $\mathcal{H}_c$ —— 2-dimensional Hilbert space with basis states $|L\rangle$ and $|R\rangle$, indicating directions.
- $\mathcal{H}_p$ —— n-dimensional Hilbert space withl basis states $|0\rangle, |1\rangle, ..., |n-1\rangle$, denoting positions.
- Quantum walk —— composite system of a coin and a walker moving on these positions.
- State space —— $\mathcal{H} = \mathcal{H}_c \otimes \mathcal{H}_p$.

Example: Quantum Walk

- Initial state —— $|L\rangle|0\rangle$.

Example: Quantum Walk

- Initial state —— $|L\rangle|0\rangle$.
- Each step of the walk:

Example: Quantum Walk

- Initial state —— $|L\rangle|0\rangle$.
- Each step of the walk:
  1. Measure to see whether it is at position 1 (absorbing boundary). If "yes", then terminate; otherwise, continue:

$$M_{yes} = |1\rangle\langle 1|,\ M_{no} = I_p - M_{yes} = \sum_{i\neq 1} |i\rangle\langle i|$$

Example: Quantum Walk

- Initial state —— $|L\rangle|0\rangle$.
- Each step of the walk:
  1. Measure to see whether it is at position 1 (absorbing boundary). If "yes", then terminate; otherwise, continue:

$$M_{yes} = |1\rangle\langle 1|,\ M_{no} = I_p - M_{yes} = \sum_{i\neq 1} |i\rangle\langle i|$$

  2. Coin-tossing:

$$\text{Hadamard operator } H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

Example: Quantum Walk

- Initial state —— $|L\rangle|0\rangle$.
- Each step of the walk:
  1. Measure to see whether it is at position 1 (absorbing boundary). If "yes", then terminate; otherwise, continue:

$$M_{yes} = |1\rangle\langle 1|,\ M_{no} = I_p - M_{yes} = \sum_{i\neq 1} |i\rangle\langle i|$$

  2. Coin-tossing:

$$\text{Hadamard operator } H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

  3. Shift operator:

$$S = \sum_{i=0}^{n-1} |L\rangle\langle L| \otimes |i \ominus 1\rangle\langle i| + \sum_{i=0}^{n-1} |R\rangle\langle R| \otimes |i \oplus 1\rangle\langle i|.$$

Example: Quantum Walk

- Difference between quantum walk and classical random walk:

Example: Quantum Walk

- Difference between quantum walk and classical random walk:
  - Coin (or direction) variable c can be in a superposition:

$$|+\rangle = \frac{1}{\sqrt{2}}(|L\rangle + |R\rangle)$$

Example: Quantum Walk

- Difference between quantum walk and classical random walk:
  - Coin (or direction) variable c can be in a superposition:

$$|+\rangle = \frac{1}{\sqrt{2}}(|L\rangle + |R\rangle)$$

  - Walker moves left and right "simultaneously":

$$\frac{1}{\sqrt{2}}(|L\rangle + |R\rangle)|i\rangle \rightarrow \frac{1}{\sqrt{2}}(|L\rangle|i \ominus 1\rangle + |R\rangle|i \oplus 1\rangle).$$

Example: Quantum Walk

- Difference between quantum walk and classical random walk:
  - Coin (or direction) variable c can be in a superposition:

$$|+\rangle = \frac{1}{\sqrt{2}}(|L\rangle + |R\rangle)$$

  - Walker moves left and right "simultaneously":

$$\frac{1}{\sqrt{2}}(|L\rangle + |R\rangle)|i\rangle \to \frac{1}{\sqrt{2}}(|L\rangle|i \ominus 1\rangle + |R\rangle|i \oplus 1\rangle).$$

- Quantum walk as quantum program:

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

Operational Semantics

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

$(Ini)\ \langle q := |0\rangle, \rho \rangle \to \langle \downarrow, \rho_0^q \rangle \qquad (\rho_0^q = \sum_n |0\rangle_q \langle n|\rho|n\rangle_q \langle 0|)$

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

$(Seq)\ \dfrac{\langle S_1, \rho \rangle \to \langle S_1', \rho' \rangle}{\langle S_1; S_2, \rho \rangle \to \langle S_1'; S_2, \rho' \rangle} \qquad (\downarrow; S_2 = S_2)$

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

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

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

Denotational Semantics

Semantic function of quantum program S:

$$[\![S]\!] : \mathcal{D}(\mathcal{H}_{\text{all}}) \to \mathcal{D}(\mathcal{H}_{\text{all}})$$

$$[\![S]\!](\rho) = \sum\{|\rho' : \langle S, \rho\rangle \to^* \langle\downarrow, \rho'\rangle|\} \text{ for any input } \rho$$

Quantum Predicates

- A *quantum predicate* is a Hermitian operator (observable) P such that $0 \sqsubseteq P \sqsubseteq I$.

[2] E. D'Hondt and P. Panangaden, Quantum weakest preconditions, *Math. Struct. in Comput. Sci.* 2006.

Quantum Predicates

- A *quantum predicate* is a Hermitian operator (observable) P such that $0 \sqsubseteq P \sqsubseteq I$.
- Quantum predicates = Effects (*quantum foundations literature*).

[2] E. D'Hondt and P. Panangaden, Quantum weakest preconditions, *Math. Struct. in Comput. Sci.* 2006.

Quantum Hoare Triples

Quantum Predicates

- A *quantum predicate* is a Hermitian operator (observable) P such that $0 \sqsubseteq P \sqsubseteq I$.
- Quantum predicates = Effects (*quantum foundations literature*).

[2] E. D'Hondt and P. Panangaden, Quantum weakest preconditions, *Math. Struct. in Comput. Sci.* 2006.

Quantum Hoare Triples

- A *correctness formula* is a statement of the form:

$$\{P\}S\{Q\}$$

Quantum Predicates

- A *quantum predicate* is a Hermitian operator (observable) P such that $0 \sqsubseteq P \sqsubseteq I$.
- Quantum predicates = Effects (*quantum foundations literature*).

[2] E. D'Hondt and P. Panangaden, Quantum weakest preconditions, *Math. Struct. in Comput. Sci.* 2006.

Quantum Hoare Triples

- A *correctness formula* is a statement of the form:

$$\{P\}S\{Q\}$$

  - S is a quantum program

Quantum Predicates

- A *quantum predicate* is a Hermitian operator (observable) P such that $0 \sqsubseteq P \sqsubseteq I$.
- Quantum predicates = Effects (*quantum foundations literature*).

[2] E. D'Hondt and P. Panangaden, Quantum weakest preconditions, *Math. Struct. in Comput. Sci.* 2006.

Quantum Hoare Triples

- A *correctness formula* is a statement of the form:

$$\{P\}S\{Q\}$$

  - S is a quantum program
  - Precondition P and postcondition Q are quantum predicates.

Total and Partial Correctness

1. $\{P\}S\{Q\}$ is true in the sense of *total correctness*:

$$\models_{\text{tot}} \{P\}S\{Q\}$$

if for all inputs ρ:

$$tr(P\rho) \leq tr(Q[\![S]\!](\rho))$$

Total and Partial Correctness

1. $\{P\}S\{Q\}$ is true in the sense of *total correctness*:

$$\models_{\text{tot}} \{P\}S\{Q\}$$

if for all inputs ρ:

$$tr(P\rho) \leq tr(Q[\![S]\!](\rho))$$

2. $\{P\}S\{Q\}$ is true in the sense of *partial correctness*:

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

if for all inputs ρ:

$$tr(P\rho) \leq tr(Q[\![S]\!](\rho)) + [tr(\rho) - tr([\![S]\!](\rho))]$$

Proof System for Partial Correctness

(Axiom-Sk) $$\{P\}\mathbf{Skip}\{P\}$$

(Axiom-Ini) $$\left\{\sum_n |n\rangle_q\langle 0|P|0\rangle_q\langle n|\right\} q := |0\rangle\{P\}$$

(Axiom-Uni) $$\{U^\dagger P U\}\bar{q} := U[\bar{q}]\{P\}$$

(Rule-Seq) $$\frac{\{P\}S_1\{Q\} \quad \{Q\}S_2\{R\}}{\{P\}S_1;S_2\{R\}}$$

(Rule-IF) $$\frac{\{P_m\}S_m\{Q\} \text{ for all } m}{\{\sum_m M_m^\dagger P_m M_m\}\mathbf{if}\ (\square m \cdot M[\bar{q}] = m \to S_m)\ \mathbf{fi}\{Q\}}$$

(Rule-LP) $$\frac{\{Q\}S\{M_0^\dagger P M_0 + M_1^\dagger Q M_1\}}{\{M_0^\dagger P M_0 + M_1^\dagger Q M_1\}\mathbf{while}\ M[\bar{q}] = 1\ \mathbf{do}\ S\{P\}}$$

(Rule-Ord) $$\frac{P \sqsubseteq P' \quad \{P'\}S\{Q'\} \quad Q' \sqsubseteq Q}{\{P\}S\{Q\}}$$

Soundness and Completeness

Theorem

For any quantum program S and quantum predicates P, Q,

$$\models_{\text{par}} \{P\}S\{Q\} \text{ if and only if } \vdash_{PD} \{P\}S\{Q\}.$$

[3] M. S. Ying, Floyd-Hoare logic for quantum programs, *TOPLAS* 2011.

Soundness and Completeness

Theorem

For any quantum program S and quantum predicates P, Q,

$$\models_{\text{par}} \{P\}S\{Q\} \text{ if and only if } \vdash_{PD} \{P\}S\{Q\}.$$

[3] M. S. Ying, Floyd-Hoare logic for quantum programs, *TOPLAS* 2011.

- Proof system for total correctness (omitted)

QHL Theorem Provers

Soundness and Completeness

Theorem

For any quantum program S and quantum predicates P, Q,

$$\models_{\text{par}} \{P\}S\{Q\} \text{ if and only if } \vdash_{PD} \{P\}S\{Q\}.$$

[3] M. S. Ying, Floyd-Hoare logic for quantum programs, *TOPLAS* 2011.

- Proof system for total correctness (omitted)

QHL Theorem Provers

- Isabelle/HOL —- J. Y. Liu et al., *CAV'19*

Soundness and Completeness

Theorem

For any quantum program S and quantum predicates P, Q,

$$\models_{\text{par}} \{P\}S\{Q\} \text{ if and only if } \vdash_{PD} \{P\}S\{Q\}.$$

[3] M. S. Ying, Floyd-Hoare logic for quantum programs, *TOPLAS* 2011.

- Proof system for total correctness (omitted)

QHL Theorem Provers

- Isabelle/HOL —- J. Y. Liu et al., *CAV'19*
- CoqQ —- L. Zhou et al., *POPL'24*

Outline

Limitations of QHL

- No classical variables

Limitations of QHL

- No classical variables
- Curse of dimensionality: a quantum predicate for n qubits is a $2^n \times 2^n$ matrix —— a bottleneck for scalable verification.

Limitations of QHL

- No classical variables
- Curse of dimensionality: a quantum predicate for n qubits is a $2^n \times 2^n$ matrix —— a bottleneck for scalable verification.
- Assertion language?

Limitations of QHL

- No classical variables
- Curse of dimensionality: a quantum predicate for n qubits is a $2^n \times 2^n$ matrix —— a bottleneck for scalable verification.
- Assertion language?
  - An assertion for a classical program is a predicate —— Boolean-valued function, over the state space.

Limitations of QHL

- No classical variables
- Curse of dimensionality: a quantum predicate for n qubits is a $2^n \times 2^n$ matrix —— a bottleneck for scalable verification.
- Assertion language?
  - An assertion for a classical program is a predicate —— Boolean-valued function, over the state space.
  - It can be represented by a first-order logical formula —— constructed from atomic formulas using connectives and quantifiers.

Limitations of QHL

- No classical variables
- Curse of dimensionality: a quantum predicate for n qubits is a $2^n \times 2^n$ matrix —— a bottleneck for scalable verification.
- Assertion language?
  - An assertion for a classical program is a predicate —— Boolean-valued function, over the state space.
  - It can be represented by a first-order logical formula —— constructed from atomic formulas using connectives and quantifiers.
  - Often much more economic than as a Boolean-valued function over the entire state space.

Limitations of QHL

- No classical variables
- Curse of dimensionality: a quantum predicate for n qubits is a $2^n \times 2^n$ matrix —— a bottleneck for scalable verification.
- Assertion language?
  - An assertion for a classical program is a predicate —— Boolean-valued function, over the state space.
  - It can be represented by a first-order logical formula —— constructed from atomic formulas using connectives and quantifiers.
  - Often much more economic than as a Boolean-valued function over the entire state space.
  - How to define an assertion language for quantum programs?

Enhanced Expressivity of Programming Language

- Classical variables

Enhanced Expressivity of Programming Language

- Classical variables
- Quantum arrays

Enhanced Expressivity of Programming Language

- Classical variables
- Quantum arrays
- Parameterized quantum gates

Enhanced Expressivity of Programming Language

- Classical variables
- Quantum arrays
- Parameterized quantum gates
- Syntax of **qWhile**$^+$:

$$
\begin{aligned}
P ::= \ & \textbf{skip} \mid x := e \mid q := |0\rangle \\
& \mid U(t_1, ..., t_m)[q_1, ..., q_n] \\
& \mid x := M[q_1, ..., q_n] \\
& \mid P_1; P_2 \mid \textbf{if } b \textbf{ then } P_1 \textbf{ else } P_0 \mid \textbf{while } b \textbf{ do } P
\end{aligned}
$$

Intuitive Correctness Specifications

- Syntax of quantum predicates:

$$A ::= K(\bar{t})[\bar{q}] \mid \neg A \mid A_1 \otimes A_2 \mid F(\bar{t})[\bar{q}](\{A_i\}).$$

Intuitive Correctness Specifications

- Syntax of quantum predicates:

$$A ::= K(\bar{t})[\bar{q}] \mid \neg A \mid A_1 \otimes A_2 \mid F(\bar{t})[\bar{q}](\{A_i\}).$$

- Hoare triples:

$$\{\varphi, A\}\ P\ \{\psi, B\}$$

Intuitive Correctness Specifications

- Syntax of quantum predicates:

$$A ::= K(\bar{t})[\bar{q}] \mid \neg A \mid A_1 \otimes A_2 \mid F(\bar{t})[\bar{q}](\{A_i\}).$$

- Hoare triples:

$$\{\varphi, A\}\ P\ \{\psi, B\}$$

  - φ, ψ are first-order logical formulas;

Intuitive Correctness Specifications

- Syntax of quantum predicates:

$$A ::= K(\bar{t})[\bar{q}] \mid \neg A \mid A_1 \otimes A_2 \mid F(\bar{t})[\bar{q}](\{A_i\}).$$

- Hoare triples:

$$\{\varphi, A\}\ P\ \{\psi, B\}$$

  - φ, ψ are first-order logical formulas;
  - A, B are quantum predicates parameterized by classical variables

Simplified proof system

- A new idea in formulating proof rule for quantum measurements.

Simplified proof system

- A new idea in formulating proof rule for quantum measurements.
- The proof system can be conveniently combined with classical first-order logic.

Proof System

$$(\textit{Axiom-Ski}) \quad \{\varphi, A\}\ \textbf{skip}\ \{\varphi, A\}$$

$$(\textit{Axiom-Ass}) \quad \{\varphi[e/x], A[e/x]\}\ x := e\ \{\varphi, A\}$$

$$(\textit{Axiom-Init}) \quad \{\varphi, F_B[q](A)\}\ q := |0\rangle\ \{\varphi, A\}$$

$$(\textit{Axiom-Uni}) \quad \{\varphi, F_U(\bar{t})[\bar{q}](A)\}\ U(\bar{t})[\bar{q}]\ \{\varphi, A\}$$

$$(\textit{Axiom-Meas}) \quad \frac{y \notin \textit{free}(\varphi) \cup \textit{cv}(A) \cup \{x\}}{\{\varphi[y/x], F_M(y)[\bar{q}](A[y/x])\}\ x := M[\bar{q}]\ \{\varphi \wedge x = y, A\}}$$

$$(\textit{Rule-Seq}) \quad \frac{\{\varphi, A\}\ P_1\ \{\psi, B\} \qquad \{\psi, B\}\ P_2\ \{\theta, , C\}}{\{\varphi, A\}\ P_1; P_2\ \{\theta, C\}}$$

$$(\textit{Rule-Cond}) \quad \frac{\{\varphi \wedge b, A\}\ P_1\ \{\psi, B\} \qquad \{\varphi \wedge \neg b, A\}\ P_0\ \{\psi, B\}}{\{\varphi, A\}\ \textbf{if}\ b\ \textbf{then}\ P_1\ \textbf{else}\ P_0\ \{\psi, B\}}$$

Proof System (Continued)

$$(\textit{Rule-Loop-par}) \quad \frac{\{\varphi \wedge b, A\}\ P\ \{\varphi, A\}}{\{\varphi, A\}\ \textbf{while}\ b\ \textbf{do}\ P\ \{\varphi \wedge \neg b, A\}}$$

$$(\textit{Rule-Conseq}) \quad \frac{(\varphi', A') \models (\varphi, A) \quad \{\varphi, A\}\ P\ \{\psi, B\} \quad (\psi, B) \models (\psi', B')}{\{\varphi', A'\}\ P\ \{\psi', B'\}}$$

$$(\textit{Rule-Accum1}) \quad \frac{\begin{array}{c}\{\varphi, A_i\}\ P\ \{\psi_i, B\} \text{ for every } i = 1, ..., n \\ (\forall i_1, i_2)\ \left(i_1 \neq i_2 \rightarrow \neg(\psi_{i_1} \wedge \psi_{i_2})\right)\end{array}}{\left\{\varphi, \frac{1}{n}\sum_{i=1}^{n} A_i\right\}\ P\ \left\{\bigvee_{i=1}^{n} \psi_i, \frac{1}{n} B\right\}}$$

$$(\textit{Rule-Accum2}) \quad \frac{\begin{array}{c}\{\varphi, A_i\}\ P\ \{\psi, B_i\} \text{ for every } i \\ 0 \leq p_i \text{ for every } i \qquad \sum_i p_i \leq 1\end{array}}{\{\varphi, \sum_i p_i A_i\}\ P\ \{\psi, \sum_i p_i B_i\}}$$

[4] M. S. Ying, A practical quantum Hoare logical with classical variables, I, *arXiv* 2412.09869.

- Proof system for total correctness (omitted)

Outline

Research Topics

- (Relative) Completeness?

Research Topics

- (Relative) Completeness?
- Applications?

Research Topics

- (Relative) Completeness?
- Applications?
- Theorem Provers in Lean, Rocq, Isabelle/HOL?

Research Topics

- (Relative) Completeness?
- Applications?
- Theorem Provers in Lean, Rocq, Isabelle/HOL?
- Extend to parallel and distributed quantum programs?

Related Work

- S. -H. Hung, K. Hietala, et al., Quantitative robustness analysis of quantum programs, *POPL* 2018.

Related Work

- S. -H. Hung, K. Hietala, et al., Quantitative robustness analysis of quantum programs, *POPL* 2018.
- D. Unruh, Quantum Hoare logic with ghost variables, *LICS* 2019.

Related Work

- S. -H. Hung, K. Hietala, et al., Quantitative robustness analysis of quantum programs, *POPL* 2018.
- D. Unruh, Quantum Hoare logic with ghost variables, *LICS* 2019.
- D. Unruh, Quantum relational Hoare logic, *POPL* 2019.

Related Work

- S. -H. Hung, K. Hietala, et al., Quantitative robustness analysis of quantum programs, *POPL* 2018.
- D. Unruh, Quantum Hoare logic with ghost variables, *LICS* 2019.
- D. Unruh, Quantum relational Hoare logic, *POPL* 2019.
- L. Zhou, et al., An applied quantum Hoare logic, *PLDI* 2019.

Related Work

- S. -H. Hung, K. Hietala, et al., Quantitative robustness analysis of quantum programs, *POPL* 2018.
- D. Unruh, Quantum Hoare logic with ghost variables, *LICS* 2019.
- D. Unruh, Quantum relational Hoare logic, *POPL* 2019.
- L. Zhou, et al., An applied quantum Hoare logic, *PLDI* 2019.
- G. Barthe, et al., Relational proofs for quantum programs, *POPL* 2020.

Related Work

- S. -H. Hung, K. Hietala, et al., Quantitative robustness analysis of quantum programs, *POPL* 2018.
- D. Unruh, Quantum Hoare logic with ghost variables, *LICS* 2019.
- D. Unruh, Quantum relational Hoare logic, *POPL* 2019.
- L. Zhou, et al., An applied quantum Hoare logic, *PLDI* 2019.
- G. Barthe, et al., Relational proofs for quantum programs, *POPL* 2020.
- R. Z. Tao, Y. N. Shi, et al., Gleipnir: Toward practical error analysis for quantum programs, *PLDI* 2021.

Related Work

- S. -H. Hung, K. Hietala, et al., Quantitative robustness analysis of quantum programs, *POPL* 2018.
- D. Unruh, Quantum Hoare logic with ghost variables, *LICS* 2019.
- D. Unruh, Quantum relational Hoare logic, *POPL* 2019.
- L. Zhou, et al., An applied quantum Hoare logic, *PLDI* 2019.
- G. Barthe, et al., Relational proofs for quantum programs, *POPL* 2020.
- R. Z. Tao, Y. N. Shi, et al., Gleipnir: Toward practical error analysis for quantum programs, *PLDI* 2021.
- C. Chareton, et al., An automated deductive verification framework for circuit-building quantum programs, *ESOP* 2021.

Related Work

- S. -H. Hung, K. Hietala, et al., Quantitative robustness analysis of quantum programs, *POPL* 2018.
- D. Unruh, Quantum Hoare logic with ghost variables, *LICS* 2019.
- D. Unruh, Quantum relational Hoare logic, *POPL* 2019.
- L. Zhou, et al., An applied quantum Hoare logic, *PLDI* 2019.
- G. Barthe, et al., Relational proofs for quantum programs, *POPL* 2020.
- R. Z. Tao, Y. N. Shi, et al., Gleipnir: Toward practical error analysis for quantum programs, *PLDI* 2021.
- C. Chareton, et al., An automated deductive verification framework for circuit-building quantum programs, *ESOP* 2021.
- L. Zhou, et al., A quantum interpretation of bunched logic for quantum separation logic, *LICS* 2021

Related Work

- S. -H. Hung, K. Hietala, et al., Quantitative robustness analysis of quantum programs, *POPL* 2018.
- D. Unruh, Quantum Hoare logic with ghost variables, *LICS* 2019.
- D. Unruh, Quantum relational Hoare logic, *POPL* 2019.
- L. Zhou, et al., An applied quantum Hoare logic, *PLDI* 2019.
- G. Barthe, et al., Relational proofs for quantum programs, *POPL* 2020.
- R. Z. Tao, Y. N. Shi, et al., Gleipnir: Toward practical error analysis for quantum programs, *PLDI* 2021.
- C. Chareton, et al., An automated deductive verification framework for circuit-building quantum programs, *ESOP* 2021.
- L. Zhou, et al., A quantum interpretation of bunched logic for quantum separation logic, *LICS* 2021
- Y. Feng and M. S. Ying, Quantum Hoare logic with classical variables, *TQC* 2021.

Related Work

- P. Yan, H. R. Jiang and N. K. Yu, On incorrectness logic for quantum programs, *OOPSLA* 2022.

Related Work

- P. Yan, H. R. Jiang and N. K. Yu, On incorrectness logic for quantum programs, *OOPSLA* 2022.
- X. -B. Le, S. -. Lin, et al., A quantum interpretation of separating conjunction for local reasoning of quantum programs based on separation logic, *POPL* 2022.

Related Work

- P. Yan, H. R. Jiang and N. K. Yu, On incorrectness logic for quantum programs, *OOPSLA* 2022.
- X. -B. Le, S. -. Lin, et al., A quantum interpretation of separating conjunction for local reasoning of quantum programs based on separation logic, *POPL* 2022.
- J. Y. Liu, et al., Quantum weakest preconditions for reasoning about expected runtimes of quantum programs, *JACM* 2025.

Related Work

- P. Yan, H. R. Jiang and N. K. Yu, On incorrectness logic for quantum programs, *OOPSLA* 2022.
- X. -B. Le, S. -. Lin, et al., A quantum interpretation of separating conjunction for local reasoning of quantum programs based on separation logic, *POPL* 2022.
- J. Y. Liu, et al., Quantum weakest preconditions for reasoning about expected runtimes of quantum programs, *JACM* 2025.

- Apologies to the many others not mentioned!

Thanks!