Review Article

A Review of Formal Methods in Quantum Circuit Verification

Arun Govindankutty[1]

1. North Dakota State University, United States

Quantum computing exploits the principles of quantum mechanics to perform computation by manipulating qubit states through sequences of quantum gates, known as quantum circuits. As quantum hardware continues to scale in both complexity and qubit count, conventional verification techniques quickly become computationally impractical, motivating the need for more advanced verification approaches. Formal methods, including theorem proving, model checking, and symbolic reasoning, offer systematic and mathematically rigorous techniques for verifying functional correctness, equivalence, and implementation, while enabling early detection of design errors. This review explores how formal methods are currently applied to the verification of quantum circuits, with a particular focus on barrier certificates, abstract interpretation, model checking, theorem proving, and emerging hybrid approaches. Examining the theoretical foundations and practical applications of these techniques, discussing their strengths, limitations, and comparative effectiveness through representative case studies, review hopes to provide a holistic understanding of the topic. Highlighting the open challenges and outlining promising directions for future research, the review aims to provide a roadmap towards more scalable and robust verification frameworks for quantum computing. For readers seeking deeper engagement, the review offers an extensive set of references to support further study and exploration.

Correspondence: papers@team.qeios.com — Qeios will forward to the authors

1. Introduction

The field of quantum computing has witnessed rapid and sustained progress in recent years [1][2][3][4][5][6]. Driven by major advances from industry, increasingly sophisticated quantum processors and improved error-correction strategies are beginning to make large-scale quantum computation

practical [7][8][9][10][11]. Google, IBM, and Amazon have substantially refined their quantum hardware platforms and experimentally validated the effectiveness of modern quantum error-correction codes [12][13][14]. Microsoft has demonstrated topological quantum computing architectures that promise inherently robust qubits and gate operations [15]. Together, these developments signal a decisive shift beyond the Noisy Intermediate-Scale Quantum (NISQ) era toward architectures capable of reliable, fault-tolerant quantum computation. The goal of fault-tolerant quantum computing is thus being realized.

In this rapidly evolving landscape, ensuring that quantum computers and their algorithmic implementations operate as intended has become a critical priority. Quantum circuits remain the fundamental abstraction for expressing quantum algorithms, making their formal verification a central and increasingly urgent challenge. However, verifying quantum systems is notoriously difficult [16], as simulating general quantum processes is widely regarded as intractable for classical computers. Unlike classical circuit validation, empirical testing alone cannot provide a sound or scalable verification strategy for quantum circuits. This necessitates the need for alternative, and rigorously grounded methods [17].

Formal verification provides a rigorous framework for validating complex systems by applying mathematical reasoning and proofs to ensure design correctness. Unlike traditional testing, it examines the full design space rather than sampled behaviours, enabling the detection of subtle corner-case bugs that may otherwise remain hidden. This systematic approach offers strong correctness guarantees and has become a standard methodology in domains where reliability is paramount, including software verification, hardware verification, and VLSI design [18][19][20]. Motivated by its success in classical settings, extensive research is now focused on extending formal methods to quantum circuits [16][17]. By leveraging these techniques, researchers aim to address challenges unique to quantum computation, such as superposition, entanglement, and noise. This enables high confidence in verification of quantum circuit behaviour and performance.

This work provides a concise review of formal methods used in quantum circuit verification, covering both their theoretical foundations and practical applications. From a quantum-centric perspective, it examines barrier certificates, abstract interpretation, model checking, theorem proving, and various hybrid approaches that combine these techniques. The review organizes these methods within a unified conceptual framework, evaluates their respective strengths and limitations, and highlights promising

research directions for overcoming the verification challenges posed by increasingly complex quantum systems.

2. Background

This section provides the essential background required to understand the fundamentals of quantum circuits and the formal verification techniques applied to them.

2.1. Quantum Circuits and Their Representation

Quantum circuits differ fundamentally from their classical counterparts due to the unique properties of quantum information. Unlike classical bits, which occupy definite states 0 or 1, quantum bits (qubits) can exist in superpositions of basis states and are represented by complex-valued probability amplitudes [21] [22]. A single-qubit state can be written as

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle,$$

where $|\psi\rangle$ denotes the quantum state, $|0\rangle$ and $|1\rangle$ are the computational basis vectors, and $\alpha, \beta \in \mathbb{C}$. The squared magnitudes $|\alpha|^2$ and $|\beta|^2$ give the corresponding measurement probabilities and satisfy the normalization condition

$$|\alpha|^2 + |\beta|^2 = 1.$$

The computational basis vectors are defined as

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Qubits may also exhibit entanglement, and superposition. They are non-classical correlations that cannot be reproduced by any classical system. These characteristics enable quantum algorithms to outperform classical ones for certain tasks, but they also introduce significant challenges for the verification of quantum circuits.

A quantum circuit is modelled as a sequence of quantum gates acting on a set of qubits. Each gate applies a unitary transformation to the underlying state vector, thereby preserving its norm. Common single-qubit gates include the Pauli operators (X, Y, Z), the Hadamard gate, and various phase gates, while multi-qubit gates such as the CNOT (controlled-NOT) gate generate entanglement between qubits [21]. Verification becomes difficult because the state space grows exponentially with the number of qubits: an

n-qubit system requires a Hilbert space of dimension 2^n, making classical simulation impractical for large-scale quantum circuits.

2.2. Formal Verification Principles

Formal verification, as introduced in Section 1, refers to mathematically proving or disproving the correctness of a system with respect to a formal specification. Unlike testing, which evaluates only a subset of behaviours, formal methods aim to reason about all possible behaviours of a system, providing stronger and more comprehensive correctness guarantees. Approaches using formal verification fall into three primary categories:

1. **Deductive verification**: Establishing correctness properties using mathematical proof systems,
2. **Model checking**: Automatically verifying finite-state systems against temporal-logic specifications, and
3. **Abstract interpretation**: Analysing program behaviour using sound semantic approximations.

In the quantum setting, formal verification must contend with challenges arising from superposition, entanglement, measurement, and decoherence. Classical verification techniques therefore require significant adaptation to reason soundly about these phenomena while remaining computationally tractable.

3. Barrier Certificates

This section introduces barrier certificates, a key technique employed in the formal verification of quantum circuits.

3.1. Foundations

Barrier certificates provide a powerful method for verifying the safety of dynamical systems by ensuring that no system trajectory can reach an undesirable state. Recently, this technique has been extended to quantum circuit verification [23][24][25][26][27].

Formally, a barrier certificate for a quantum system is a function $B : \mathbb{X} \to \mathbb{R}$, satisfying three key conditions:

1. **Initial state condition**: $B(x) \leq 0$ for all initial states $x \in \mathbb{X}_0$,
2. **Unsafe state condition**: $B(x) > 0$ for all unsafe states $x \in \mathbb{X}_u$,

3. **Decrement condition**: $B(x') - B(x) \leq 0$ for all $x \in \mathbb{X}$ and all $x' \in f(x)$.

Here, $\mathbb{X}$ denotes the state space, $\mathbb{X}_0 \subseteq \mathbb{X}$ is the set of initial states, $\mathbb{X}_u \subseteq \mathbb{X}$ is the set of unsafe states, and $f : \mathbb{X} \Rightarrow \mathbb{X}$, represents the set-valued transition map defining the system dynamics. These conditions guarantee that trajectories starting from any initial state never enter the unsafe region at any time step.

3.2. Scenario-Based Approach

A notable advancement in barrier certificate synthesis for quantum circuits is the scenario-based approach [23][28][29]. This method employs sampling to construct barrier certificates over both finite and infinite time horizons, while explicitly accounting for uncertainties in initial states and system dynamics. Such capabilities make it particularly suitable for noisy quantum systems, where exact characterization of all parameters is often infeasible.

The scenario approach transforms the verification problem into a convex optimization problem based on sampled constraints. This allows efficient computation of a barrier certificate that satisfies the safety conditions with high probability. This method offers several advantages for quantum circuit verification:

1. Supports continuous state spaces inherent to quantum systems,
2. Accommodates uncertain dynamics arising from quantum noise and device imperfections,
3. Provides probabilistic guarantees of correctness, and
4. Is applicable to both finite and infinite time horizons.

3.3. Applications and Case Studies

Barrier certificates have been successfully applied to the verification of a variety of quantum circuits, including Grover's search algorithm and other quantum oracles. Researchers have evaluated different classes of barrier certificates—such as polynomial, exponential, and rational functions—to identify the most effective choice for each application. Table 1 summarizes these barrier certificate types, highlighting their respective strengths, limitations, and typical use cases.

Barrier certificates have been successfully applied to verify various quantum circuits, including Grover's quantum search algorithm and other quantum oracles. Researchers have compared the performance of different types of barrier certificates (e.g., polynomial, exponential, and rational functions) to determine the most suitable certificate for each case.

Barrier Certificate Type	Strengths	Limitations	Ideal Use Cases
Polynomial	Efficient synthesis, good scalability	Limited expressiveness	Linear and mildly nonlinear systems
Exponential	Handles exponential dynamics	Numerical stability issues	Systems with exponential convergence
Rational	High expressiveness	Complex optimization	Highly nonlinear systems
Scenario-based	Handles uncertainty, probabilistic guarantees	Sampling may miss rare cases	Noisy quantum systems

Table 1. summarises the various barrier certificate types, their strengths, limitations, and use cases.

Case studies indicate that the selection of the barrier function strongly influences both the efficiency and effectiveness of verification. For many quantum circuits, polynomial barrier certificates provide a favourable balance between expressiveness and computational tractability. More complex barrier functions may be required for circuits exhibiting highly nonlinear dynamics.

4. Abstract Interpretation

This section provides an overview of how abstract interpretation is applied to the formal verification of quantum circuits.

4.1. Semantic Framework

Abstract interpretation offers a theoretical framework for approximating the semantics of computational systems, enabling static analysis of program properties [30][31][32][33][34][35]. Recently, this methodology has been extended to variational quantum circuits (VQCs) [36][37], which form the core of many quantum machine learning algorithms. Analogous to classical deep neural networks, VQCs are susceptible to adversarial inputs. Minor perturbations can lead to incorrect outputs or predictions.

[36] have developed a semantic framework grounded in abstract interpretation specifically for the verification of VQCs. Their approach explicitly addresses quantum-specific features, such as state normalization, which introduces inter-variable dependencies that challenge conventional verification techniques. This framework enables a formal formulation of the verification problem for VQCs and provides tools to analyse its computational complexity.

4.2. Interval-Based Reachability

A central technique in the abstract interpretation of quantum circuits is interval-based reachability analysis. This method computes over approximations of the reachable states at each circuit layer [36][38][39][40]. They propagate interval bounds through the quantum circuit to provide formal guarantees of its behaviour. However, quantum phenomena such as superposition and entanglement introduce inter-variable dependencies that complicate interval analysis.

To address these challenges, enhanced abstraction techniques have been developed to explicitly track dependencies between variables, though this comes at increased computational cost [41]. Achieving the right balance between precision and efficiency remains an active area of research in quantum abstract interpretation.

4.3. Verification of Robustness Properties

Abstract interpretation has proven particularly effective for verifying the robustness of variational quantum circuits against adversarial perturbations. By analysing how small variations in input states influence the final measurement probabilities, these techniques can provide formal robustness certificates analogous to those developed for classical neural networks [42][43][44][45][46][47]. The verification process generally involves:

1. Defining perturbation models for the input states,
2. Propagating these perturbations through the quantum circuit using abstract domains,
3. Computing bounds on the output probabilities, and
4. Ensuring that classification decisions remain stable within the specified perturbation range.

This methodology has been applied to standard verification benchmarks, demonstrating its potential for certifying the reliability of quantum machine learning models in safety-critical applications.

5. Other Formal Methods

This section highlights additional formal methods used in the verification of quantum circuits.

5.1. Model Checking

Model checking is a formal verification technique that exhaustively explores all possible states of a system [48]. From the perspective of quantum circuit verification, it is used to determine whether a system satisfies specified temporal logic properties. Significant research has been conducted in this area [49][50][51][52][53][54][55][56][57]. For quantum circuits, model checking typically involves:

1. Encoding the quantum circuit as a finite-state system,
2. Specifying desired properties using appropriate temporal logics, and
3. Automatically verifying these properties against the encoded model.

Early studies in this area verified quantum circuits by mapping them to quantum Markov chains, which were then subjected to model checking [58]. Other approaches have extended probabilistic model checking to account for the inherent stochastic nature of quantum measurements and decoherence [59].

Despite its theoretical strengths, model checking faces severe scalability challenges in quantum systems, as the state space grows exponentially with the number of qubits. To mitigate this state explosion problem, symbolic model checking techniques, employing binary decision diagrams (BDDs) and related compression methods have been developed [60][61][62][63][64][65][66], achieving varying degrees of success.

5.2. Theorem Proving

Theorem proving provides a deductive framework for verifying quantum circuits, employing formal logical systems to construct rigorous mathematical proofs of correctness. This approach has been realized in several proof assistants, including Coq, Isabelle/HOL, and Lean, often supplemented with specialized libraries for quantum computation [67][68][69][70][71]. The authors of [72][73] have developed a deductive approaches to quantum circuit verification leveraging SMT solvers. The Giallar tool exploits SMT solvers to verify quantum circuit compiler passes, ensuring that quantum semantics is preserved at each pass [74]. While theorem proving offers extremely high assurance, it demands substantial expertise and manual effort, which limits its suitability for rapid iteration in quantum circuit design workflows.

5.3. Runtime Verification

Runtime verification is an approach that monitors quantum circuit execution against specified properties. Although it does not provide full formal guarantees, runtime verification can detect property violations during testing or actual operation, making it particularly practical for near-term quantum applications [74][75][76][77][78][79][80][81][82].

The runtime verification process typically involves:

1. Instrumenting the quantum circuit with additional measurement operations,
2. Defining assertion-like properties for targeted circuit states, and
3. Performing statistical testing of these properties during execution.

Runtime verification is especially useful for validating specific executions on quantum hardware. They can serves as a practical complement to exhaustive formal methods that analyse circuit designs.

6. Applications and Case Studies

This section highlights applications and case studies demonstrating quantum circuit verification in practice.

6.1. Quantum Error Correction

Quantum error correction (QEC) is one of the most important applications of formal verification methods. Fault-tolerant quantum computation relies critically on the correct operation of QEC circuits [83][84][85]. Formal methods have been employed to verify various aspects of QEC implementations [86][87][88][89][90][91], including:

1. Correctness of stabilizer measurements,
2. Fault-tolerance thresholds, and
3. Implementation of logical operations.

Barrier certificates have been particularly effective in ensuring that erroneous states remain within correctable regions of the state space. Model checking has been applied to verify the sequential behaviour of QEC protocols under various fault models.

6.2. Compiler Verification

Quantum circuits are typically expressed in high-level programming languages and subsequently compiled into hardware-specific instructions This makes the verification of compilation correctness critical and essential [35][76][78][80][92][93][94][95]. The Giallar tool [74] employs SMT solvers to ensure that each compiler pass preserves the semantic integrity of the quantum circuit. This verification process typically involves:

1. Translating quantum circuits into logical formulae,
2. Checking equivalence between the original and compiled circuits, and
3. Verifying the correctness of optimization rules applied during compilation.

Compiler verification is particularly critical for high-stakes quantum applications, where even minor compilation errors could result in catastrophic failures.

6.3. Quantum Algorithms

Formal methods have been employed to verify the correctness of a range of quantum algorithms [96][97], including Grover's search algorithm [98], quantum phase estimation [99][100], and quantum approximate optimization algorithms (QAOA) [101]. Each algorithm poses distinct verification challenges:

1. **Grover's algorithm**: Verification requires proving convergence properties and establishing bounds on the success probability.
2. **Quantum phase estimation**: Verification involves ensuring precision guarantees under different noise models.
3. **QAOA**: Verification focuses on approximation ratios and convergence properties.

These verifications often combine multiple formal techniques, for example, employing barrier certificates to establish safety properties and theorem proving to ensure functional correctness. Table 2 summarizes key domains in quantum computing, the formal verification methods applied, associated challenges, and representative tools used in each case.

Application Domain	Primary Verification Methods	Key Challenges	Notable Tools
Quantum Error Correction	Barrier certificates, Model checking	Complexity of feedback control	QECVerifier, QuaVer
Quantum Compilers	Theorem proving, SMT solvers	Semantic preservation across layers	Giallar, Quartz
Quantum Algorithms	Abstract interpretation, Barrier certificates	Handling exponential state spaces	QVVerify, CertiQ
Quantum Hardware	Model checking, Runtime verification	Modeling physical imperfections	HQVer, PulseVerifier

Table 2. Formal Verification Applications in Quantum Computing

7. Challenges and Limitations

This section provides the author's critical assessment of the challenges and limitations inherent in current quantum circuit verification methodologies.

7.1. Scalability

A major challenge in quantum circuit verification is the exponential growth of the state space with increasing qubit counts. Although classical hardware performance continues to improve, the intrinsic complexity of representing quantum states exactly fundamentally limits the scalability of all verification methods. Current approaches attempt to address this challenge through:

1. Abstraction and approximation techniques that trade completeness for scalability,
2. Modular verification, which decomposes large circuits into smaller components, and
3. Symbolic methods that compactly represent sets of quantum states.

Despite these advances, verifying circuits with large numbers of qubits remains difficult, highlighting the need for further research into scalable verification methodologies.

7.2. Quantum Specific Features

Quantum-specific phenomena such as entanglement, superposition, and measurement introduce verification challenges with no classical counterparts. These effects generate complex correlations between qubits that are difficult to abstract or approximate without losing critical information. Measurement, in particular, is challenging because it collapses the quantum state and is inherently non-unitary, complicating continuous verification. Approaches to mitigate these challenges include:

1. Developing specialized abstract domains to capture quantum correlations,
2. Designing measurement-aware verification techniques, and
3. Employing relational logics to represent and reason about entanglement.

7.3. Tool Support

Although numerous tools have been developed for quantum circuit verification, the ecosystem remains less mature than that for classical software verification. Most existing tools require substantial expertise and are often specialized for particular verification methods or quantum programming languages. Enhancing tool support will require:

1. Standardizing interfaces between verification tools and quantum programming frameworks,
2. Developing user-friendly interfaces for specifying verification properties,
3. Establishing benchmark suites to evaluate and compare verification tools, and
4. Automating the selection of suitable verification methods for a given circuit.

8. Future Directions

Here, several potential frontiers and directions for future research in quantum circuit verification are outlined, as identified by the reviewer.

8.1. Hybrid Verification Methods

Future verification frameworks are expected to integrate multiple techniques into hybrid approaches that capitalize on the strengths of each method. For instance, abstract interpretation could rapidly identify potential problem regions, which are subsequently examined in detail using more precise methods such as theorem proving or model checking.

Promising hybrid combinations include:

1. Barrier certificates combined with abstract interpretation for safety verification,
2. Theorem proving integrated with model checking for simultaneous functional and temporal verification, and
3. Runtime verification augmented with formal methods to provide practical assurance.

8.2. Machine Learning Assisted Verification

Machine learning techniques present promising opportunities to enhance and support formal verification of quantum circuits. Potential applications include:

1. Learning barrier certificates directly from simulation data,
2. Predicting circuit components that are challenging to verify, thereby focusing verification efforts,
3. Guiding abstract interpretation with learned heuristics, and
4. Accelerating model checking through learned representations of the state space.

These approaches have the potential to substantially improve the scalability and automation of quantum circuit verification while preserving formal guarantees.

8.3. Verifying Fault Tolerant Quantum Computing

As quantum computing advances toward fault-tolerant operation, new verification challenges and opportunities arise. Future research directions include:

1. Verifying quantum error correction protocols under realistic noise models,
2. Developing verification techniques for distributed quantum systems,
3. Establishing certification frameworks for quantum hardware components, and
4. Creating standards for quantum software verification.

Progress in these areas will be crucial for building reliable and trustworthy quantum computing systems for critical applications.

9. Conclusion

Formal methods for quantum circuit verification have seen substantial progress in recent years, evolving from theoretical frameworks to practical tools applicable to real quantum circuits. This review has surveyed the landscape of these methods, including barrier certificates, abstract interpretation, model

checking, and theorem proving, each providing unique advantages suited to different verification scenarios.

Barrier certificates offer a robust approach for safety verification and, when integrated with scenario-based methods, can accommodate uncertainties in initial states and system dynamics. Abstract interpretation provides a semantic framework for analysing variational quantum circuits and verifying robustness properties. Model checking enables exhaustive verification of temporal properties, while theorem proving delivers the highest assurance through rigorous mathematical proofs.

Despite these advances, significant challenges persist, particularly in scaling verification to larger quantum systems and addressing quantum-specific features such as entanglement and measurement. Future research need to focus on developing hybrid approaches that integrate multiple verification techniques, leveraging machine learning to enhance scalability, and meeting the verification requirements of fault-tolerant quantum computation.

As quantum computing progresses toward practical applications, formal verification methods will become increasingly essential for ensuring the reliability and correctness of both quantum software and hardware. The advancement of robust verification tools and methodologies will be crucial for fostering trust in quantum computing systems and unlocking their full potential to tackle computationally challenging problems.

References

1. ^*Hassija V, Chamola V, Goyal A, Kanhere SS, Guizani N (2020). "Forthcoming applications of quantum com puting: Peeking into the future." IET Quantum Commun. **1**(2):35–41. doi:10.1049/iet-qtc.2020.0026.*
2. ^*Jose P, Hariharan S, Madhivanan V, N S, Krisnamoorthy M, Cherukuri AK (2024). "Enhanced QSVM with eli tist non-dominated sorting genetic optimisation algorithm for breast cancer diagnosis." IET Quantum Com mun. **5**(4):384–398. doi:10.1049/qtc2.12113.*
3. ^*Arun G, Mishra V (2014). "A Review on Quantum Computing and Communication." In: 2014 2nd Internatio nal Conference on Emerging Technology Trends in Electronics, Communication and Networking. pp. 1–5. d oi:10.1109/ET2ECN.2014.7044953.*
4. ^*Brooks M (2019). "Beyond Quantum Supremacy: The Hunt for Useful Quantum Computers." Nature. **574**(7 776):19–21. doi:10.1038/d41586-019-02936-3.*

5. ^Chen S (2019). "Quantum Computing Scientists: Give Them Lemons, They'll Make Lemonade." APS News. https://www.aps.org/apsnews/2019/05/quantum-computing-lemons-lemonade.
6. ^Preskill J (2018). "Quantum Computing in the NISQ Era and Beyond." Quantum. 2:79. doi:10.22331/q-2018-08-06-79.
7. ^Porter J (2021). "Google Wants to Build a Useful Quantum Computer by 2029." The Verge. https://www.theverge.com/2021/5/19/22443453/google-quantum-computer-2029-decade-commercial-useful-qubits-quantum-transistor.
8. ^Gambetta J (2022). "Expanding the IBM Quantum Roadmap to Anticipate the Future of Quantum-Centric Supercomputing." IBM Research. https://research.ibm.com/blog/ibm-quantum-roadmap-2025.
9. ^Krewell K, Tirias Research (2022). "The Next Generation of IBM Quantum Computers." Forbes. https://www.forbes.com/sites/tiriasresearch/2022/06/22/the-next-generation-of-ibm-quantum-computers/?sh=5083d463266f.
10. ^Gambetta J, Faro I, Wehden K (2021). "IBM's Roadmap for Building an Open Quantum Software Ecosystem." IBM Research. https://research.ibm.com/blog/quantum-development-roadmap.
11. ^Hsu J (2018). "CES 2018: Intel's 49-Qubit Chip Shoots for Quantum Supremacy." IEEE Spectrum. https://spectrum.ieee.org/intels-49qubit-chip-aims-for-quantum-supremacy.
12. ^Google Quantum AI and Collaborators, Acharya R, Abanin DA, Aghababaie-Beni L, Aleiner I, Andersen TI, Ansmann M, Arute F, Arya K, Asfaw A, Astrakhantsev N, Atalaya J, Babbush R, Bacon D, Ballard B, Bardin JC, Bausch J, Bengtsson A, Bilmes A, Blackwell S, Boixo S, Bortoli G, Bourassa A, Bovaird J, Brill L, Broughton M, Browne DA, Buchea B, Buckley BB, Buell DA, Burger T, Burkett B, Bushnell N, Cabrera A, Campero J, Chang H S, Chen Y, Chen Z, Chiaro B, Chik D, Chou C, Claes J, Cleland AY, Cogan J, Collins R, Conner P, Courtney W, Crook AL, Curtin B, Das S, Davies A, De Lorenzo L, Debroy DM, Demura S, Devoret M, Di Paolo A, Donohoe P, Drozdov I, Dunsworth A, Earle C, Edlich T, Eickbusch A, Elbag AM, Elzouka M, Erickson C, Faoro L, Farhi E, Ferreira VS, Burgos LF, Forati E, Fowler AG, Foxen B, Ganjam S, Garcia G, Gasca R, Genois É, Giang W, Gidney C, Gilboa D, Gosula R, Dau AG, Graumann D, Greene A, Gross JA, Habegger S, Hall J, Hamilton MC, Hansen M, Harrigan MP, Harrington SD, Heras FJH, Heslin S, Heu P, Higgott O, Hill G, Hilton J, Holland G, Hong S, Huang HY, Huff A, Huggins WJ, Ioffe LB, Isakov SV, Iveland J, Jeffrey E, Jiang Z, Jones C, Jordan S, Joshi C, Juhas P, Kafri D, Kang H, Karamlou AH, Kechedzhi K, Kelly J, Khaire T, Khattar T, Khezri M, Kim S, Klimov PV, Klots AR, Kobrin B, Kohli P, Korotkov AN, Kostritsa F, Kothari R, Kozlovskii B, Kreikebaum JM, Kurilovich VD, Lacroix N, Landhuis D, Lange-Dei T, Langley BW, Laptev P, Lau KM, Le Guevel L, Ledford J, Lee J, Lee K, Lensky YD, Leon S, Lester BJ, Li WY, Li Y, Lill AT, Liu W, Livingston WP, Locharla A, Lucero E, Lundahl D, Lunt A, Madhuk

*S, Malone FD, Maloney A, Mandrà S, Manyika J, Martin LS, Martin O, Martin S, Maxfield C, McClean JR, McE wen M, Meeks S, Megrant A, Mi X, Miao KC, Mieszala A, Molavi R, Molina S, Montazeri S, Morvan A, Movass agh R, Mruczkiewicz W, Naaman O, Neeley M, Neill C, Nersisyan A, Neven H, Newman M, Ng JH, Nguyen A, Nguyen M, Ni CH, Niu MY, O'Brien TE, Oliver WD, Opremcak A, Ottosson K, Petukhov A, Pizzuto A, Platt J, Po tter R, Pritchard O, Pryadko LP, Quintana C, Ramachandran G, Reagor MJ, Redding J, Rhodes DM, Roberts G, Rosenberg E, Rosenfeld E, Roushan P, Rubin NC, Saei N, Sank D, Sankaragomathi K, Satzinger KJ, Schurkus HF, Schuster C, Senior AW, Shearn MJ, Shorter A, Shutty N, Shvarts V, Singh S, Sivak V, Skruzny J, Small S, Sm elyanskiy V, Smith WC, Somma RD, Springer S, Sterling G, Strain D, Suchard J, Szasz A, Sztein A, Thor D, Torr es A, Torunbalci MM, Vaishnav A, Vargas J, Vdovichev S, Vidal G, Villalonga B, Heidweiller CV, Waltman S, W ang SX, Ware B, Weber K, Weidel T, White T, Wong K, Woo BWK, Xing C, Yao ZJ, Yeh P, Ying B, Yoo J, Yosri N, Y oung G, Zalcman A, Zhang Y, Zhu N, Zobrist N (2025). "Quantum Error Correction Below the Surface Code T hreshold." Nature. **638**(8052):920–926. doi:10.1038/s41586-024-08449-y.*

13. ^*Bravyi S, Cross AW, Gambetta JM, Maslov D, Rall P, Yoder TJ (2024). "High-Threshold and Low-Overhead F ault-Tolerant Quantum Memory." Nature. **627**(8005):778–782. doi:10.1038/s41586-024-07107-7.*

14. ^*Putterman H, Noh K, Hann CT, MacCabe GS, Aghaeimeibodi S, Patel RN, Lee M, Jones WM, Moradinejad H, Rodriguez R, Mahuli N, Rose J, Owens JC, Levine H, Rosenfeld E, Reinhold P, Moncelsi L, Alcid JA, Alidoust N, Arrangoiz-Arriola P, Barnett J, Bienias P, Carson HA, Chen C, Chen L, Chinkezian H, Chisholm EM, Chou MH, Clerk A, Clifford A, Cosmic R, Curiel AV, Davis E, DeLorenzo L, D'Ewart JM, Diky A, D'Souza N, Dumitrescu P T, Eisenmann S, Elkhouly E, Evenbly G, Fang MT, Fang Y, Fling MJ, Fon W, Garcia G, Gorshkov AV, Grant JA, Gr ay MJ, Grimberg S, Grimsmo AL, Haim A, Hand J, He Y, Hernandez M, Hover D, Hung JSC, Hunt M, Iverson J, J arrige I, Jaskula JC, Jiang L, Kalaee M, Karabalin R, Karalekas PJ, Keller AJ, Khalajhedayati A, Kubica A, Lee H, Leroux C, Lieu S, Ly V, Madrigal KV, Marcaud G, McCabe G, Miles C, Milsted A, Minguzzi J, Mishra A, Muk herjee B, Naghiloo M, Oblepias E, Ortuno G, Pagdilao J, Pancotti N, Panduro A, Paquette JP, Park M, Peairs G A, Perello D, Peterson EC, Ponte S, Preskill J, Qiao J, Refael G, Resnick R, Retzker A, Reyna OA, Runyan M, Rya n CA, Sahmoud A, Sanchez E, Sanil R, Sankar K, Sato Y, Scaffidi T, Siavoshi S, Sivarajah P, Skogland T, Su CJ, Swenson LJ, Teo SM, Tomada A, Torlai G, Wollack EA, Ye Y, Zerrudo JA, Zhang K, Brandão FGSL, Matheny M H, Painter O (2025). "Hardware-Efficient Quantum Error Correction Via Concatenated Bosonic Qubits." Nat ure. **638**(8052):927–934. doi:10.1038/s41586-025-08642-7.*

15. ^*Microsoft Azure Quantum, Aghaee M, Ramirez AA, Alam Z, Ali R, Andrzejczuk M, Antipov A, Astafev M, Ba rzegar A, Bauer B, Becker J, Bhaskar UK, Bocharov A, Boddapati S, Bohn D, Bommer J, Bourdet L, Bousquet A, Boutin S, Casparis L, Chapman BJ, Chatoor S, Christensen AW, Chua C, Codd P, Cole W, Cooper P, Corsetti*

*F, Cui A, Dalpasso P, Dehollain JP, de Lange G, de Moor M, Ekefjärd A, El Dandachi T, Saldaña JCE, Fallahi S, Galletti L, Gardner G, Govender D, Griggio F, Grigoryan R, Grijalva S, Gronin S, Gukelberger J, Hamdast M, Ha mze F, Hansen EB, Heedt S, Heidarnia Z, Zamorano J H, Ho S, Holgaard L, Hornibrook J, Indrapiromkul J, Ing erslev H, Ivancevic L, Jensen T, Jhoja J, Jones J, Kalashnikov KV, Kallaher R, Kalra R, Karimi F, Karzig T, King E, Kloster ME, Knapp C, Kocon D, Koski JV, Kostamo P, Kumar M, Laeven T, Larsen T, Lee J, Lee K, Leum G, Li K, Lindemann T, Looij M, Love J, Lucas M, Lutchyn R, Madsen MH, Madulid N, Malmros A, Manfra M, Mantr i D, Markussen SB, Martinez E, Mattila M, McNeil R, Mei AB, Mishmash RV, Mohandas G, Mollgaard C, Morg an T, Moussa G, Nayak C, Nielsen JH, Nielsen JM, Nielsen WHP, Nijholt B, Nystrom M, O'Farrell E, Ohki T, Ota ni K, Wütz BP, Pauka S, Petersson K, Petit L, Pikulin D, Prawiroatmodjo G, Preiss F, Morejon EP, Rajpalke M, Ranta C, Rasmussen K, Razmadze D, Reentila O, Reilly DJ, Ren Y, Reneris K, Rouse R, Sadovskyy I, Sainiemi L, Sanlorenzo I, Schmidgall E, Sfiligoj C, Shah MB, Simoes K, Singh S, Sinha S, Soerensen T, Sohr P, Stankevic T, Stek L, Stuppard E, Suominen H, Suter J, Teicher S, Thiyagarajah N, Tholapi R, Thomas M, Toomey E, Trac y J, Turley M, Upadhyay S, Urban I, Van Hoogdalem K, Van Woerkom DJ, Viazmitinov DV, Vogel D, Watson J, Webster A, Weston J, Winkler GW, Xu D, Yang CK, Yucelen E, Zeisel R, Zheng G, Zilke J (2025). "Interferometric Single-Shot Parity Measurement in InAs–Al Hybrid Devices." Nature. **638**(8051):651–655. doi:10.1049/iet-qtc.2020.0026.*

16. [a], [b] *Quist AJ, Mei J, Coopmans T, Laarman A (2025). "Advancing Quantum Computing with Formal Method s." In: Platzer A, Rozier KY, Pradella M, Rossi M editors. Formal Methods. Cham: Springer Nature Switzerlan d. pp. 420–446.*
17. [a], [b] *Chareton C, Bardin S, Lee D, Valiron B, Vilmart R, et al. (2022). "Formal Methods for Quantum Programs: A Survey." https://arxiv.org/abs/2109.06493.*
18. [^] *Hasan O, Tahar S (2015). "Formal Verification Methods." In: Encyclopedia of Information Science and Tech nology, Third Edition. IGI Global. pp. 7162–7170.*
19. [^] *Gupta A (1992). "Formal Hardware Verification Methods: A Survey." Formal Methods Syst Des. **1**:151–238.*
20. [^] *Schubert T, Kumar MVA, Seligman E (2015). Formal Verification. Elsevier. doi:10.1016/C2013-0-18672-2. ISB N 9780128007273.*
21. [a], [b] *Nielsen MA, Chuang IL (2011). Quantum Computation and Quantum Information: 10th Anniversary Edit ion. 10th ed. USA: Cambridge University Press. ISBN 1107002176.*
22. [^] *Bernhardt C (2020). Quantum Computing for Everyone. MIT Press.*
23. [a], [b] *Hu S, Lopata V, Soudjani S, Zuliani P (2025). "Verification of Quantum Circuits Through Barrier Certificat es Using a Scenario Approach." In: 2025 IEEE International Conference on Quantum Software (QSW). pp. 15*

1–161. doi:10.1109/QSW67625.2025.00027.

24. ^*Lewis M, Soudjani S, Zuliani P (2024). "Verification of Quantum Circuits Through Discrete-Time Barrier C ertificates." arXiv preprint arXiv:240807591.*
25. ^*Soudjani S. "Verification of Quantum Systems Using Barrier Certificates." Quantitative Evaluation of Syste ms. :346.*
26. ^*Lewis M, Zuliani P, Soudjani S (2023). "Verification of Quantum Systems Using Barrier Certificates." In: Int ernational Conference on Quantitative Evaluation of Systems. Springer. pp. 346–362.*
27. ^*Prajna S, Jadbabaie A, Pappas GJ (2007). "A Framework for Worst-Case and Stochastic Safety Verification Using Barrier Certificates." IEEE Trans Automat Contr. **52**(8):1415–1428. doi:10.1109/TAC.2007.902736.*
28. ^*Murali V, Trivedi A, Zamani M (2022). "A Scenario Approach for Synthesizing K-Inductive Barrier Certifica tes." IEEE Control Syst Lett. **6**:3247–3252.*
29. ^*Akella P, Ames AD (2022). "A Barrier-Based Scenario Approach to Verifying Safety-Critical Systems." IEEE Robot Autom Lett. **7**(4):11062–11069.*
30. ^*Yu N, Palsberg J (2021). "Quantum Abstract Interpretation." In: Proceedings of the 42nd ACM SIGPLAN Inte rnational Conference on Programming Language Design and Implementation. pp. 542–558.*
31. ^*Ying M, Zhang Z (2024). "Verification of Recursively Defined Quantum Circuits." arXiv preprint arXiv:2404 05934.*
32. ^*Rizvi SMA, Khalid U, Chatzinotas S, Duong TQ, Shin H (2025). "Controlled Quantum Semantic Communic ation for Industrial CPS Networks." IEEE Trans Netw Sci Eng.*
33. ^*Chen YF, Chung KM, Lengál O, Lin JA, Tsai WL, et al. (2025). "An Automata-Based Framework for Verificati on and Bug Hunting in Quantum Circuits." Commun ACM. **68**(6):85–93.*
34. ^*Li L, Zhu M, Cleaveland R, Nicolellis A, Lee Y, et al. (2022). "Qafny: A Quantum-Program Verifier." arXiv pre print arXiv:221106411.*
35. a, b*Shi Y, Tao R, Li X, Javadi-Abhari A, Cross AW, et al. (2019). "Certiq: A Mostly-Automated Verification of a R ealistic Quantum Compiler." arXiv preprint arXiv:190808963.*
36. a, b, c*Assolini N, Marzari L, Mastroeni I, di Pierro A (2025). "Formal Verification of Variational Quantum Circ uits." https://arxiv.org/abs/2507.10635.*
37. ^*Peham T, Burgholzer L, Wille R (2023). "Equivalence Checking of Parameterized Quantum Circuits: Verifyi ng the Compilation of Variational Quantum Algorithms." In: Proceedings of the 28th Asia and South Pacific Design Automation Conference. pp. 702–708.*

38. ^Ye K, Dai S, Guo B, Shen Y, Liu C, et al. (2024). "A Mutual-Influence-Aware Heuristic Method for Quantum C ircuit Mapping." IEEE Trans Comput.

39. ^Hung WNN, Song X, Yang G, Yang J, Perkowski M (2004). "Quantum Logic Synthesis by Symbolic Reachabi lity Analysis." In: Proceedings of the 41st Annual Design Automation Conference. pp. 838–841.

40. ^Li XW, Zhang XM, Cheng B, Yung MH (2024). "Reachability Deficit of Variational Grover Search." Phys Rev A. **109**(1):012414.

41. ^Hietala K, Rand R, Hung SH, Li L, Hicks M (2021). "Proving Quantum Programs Correct." In: Cohen L, Kalis zyk C editors. 12th International Conference on Interactive Theorem Proving (ITP 2021). Dagstuhl, German y: Schloss Dagstuhl – Leibniz-Zentrum für Informatik. pp. 21:1–21:19. doi:10.4230/LIPIcs.ITP.2021.21.

42. ^Lin Y, Guan J, Fang W, Ying M, Su Z (2024). "Veriqr: A Robustness Verification Tool for Quantum Machine L earning Models." arXiv preprint arXiv:240713533.

43. ^Guan J, Fang W, Ying M (2020). "Robustness Verification of Quantum Machine Learning." CoRR.

44. ^Guan J, Fang W, Ying M (2021). "Robustness Verification of Quantum Classifiers." In: International Confere nce on Computer Aided Verification. Springer. pp. 151–174.

45. ^Franco N, Wollschläger T, Gao N, Lorenz JM, Günnemann S (2022). "Quantum Robustness Verification: A H ybrid Quantum-Classical Neural Network Certification Algorithm." In: 2022 IEEE International Conference on Quantum Computing and Engineering (QCE). IEEE. pp. 142–153.

46. ^Gheorghiu A, Kashefi E, Wallden P (2015). "Robustness and Device Independence of Verifiable Blind Quant um Computing." New J Phys. **17**(8):083040.

47. ^Fey G, Drechsler R (2008). "A Basis for Formal Robustness Checking." In: 9th International Symposium on Quality Electronic Design (ISQED 2008). IEEE. pp. 784–789.

48. ^Gay SJ, Nagarajan R, Papanikolaou N (2008). "QMC: A Model Checker for Quantum Systems." In: Gupta A, Malik S editors. Computer Aided Verification. Berlin, Heidelberg: Springer Berlin Heidelberg. pp. 543–547.

49. ^Ying M (2021). "Model Checking for Verification of Quantum Circuits." In: International Symposium on For mal Methods. Springer. pp. 23–39.

50. ^Do CM, Ogata K (2024). "Symbolic Model Checking Quantum Circuits in Maude." PeerJ Comput Sci. **10**:e20 98.

51. ^Munir M, Gopikanna A, Fayyazi A, Pedram M, Nazarian S (2021). "QMC: A Formal Model Checking Verifica tion Framework for Superconducting Logic." In: Proceedings of the 2021 Great Lakes Symposium on VLSI. p p. 259–264.

52. ^Davidson TAS, Gay SJ, Mlnarik H, Nagarajan R, Papanikolaou N (2012). "Model Checking for Communicating Quantum Processes." Int J Unconv Comput. **8**(1):73–98.

53. ^Papanikolaou NK (2009). Model Checking Quantum Protocols. PhD thesis, University of Warwick.

54. ^Jeon S, Cho K, Kang CG, Lee J, Oh H, et al. (2024). "Quantum Probabilistic Model Checking for Time-Bounded Properties." Proc ACM Program Lang. **8**(OOPSLA2):557–587.

55. ^Turrini A (2022). "An Introduction to Quantum Model Checking." Appl Sci. **12**(4):2016.

56. ^Chen YF, Chung KM, Lengál O, Lin JA, Tsai WL (2023). "AutoQ: An Automata-Based Quantum Circuit Verifier." In: International Conference on Computer Aided Verification. Springer. pp. 139–153.

57. ^Anticoli L, Piazza C, Taglialegne L, Zuliani P (2016). "Towards Quantum Programs Verification: From Quipper Circuits to QPMC." In: International Conference on Reversible Computation. Springer. pp. 213–219.

58. ^Feng Y, Yu N, Ying M (2013). "Model Checking Quantum Markov Chains." J Comput Syst Sci. **79**(7):1181–1198.

59. ^Gay S, Nagarajan R, Papanikolaou N (2005). "Probabilistic Model–Checking of Quantum Protocols." https://arxiv.org/abs/quant-ph/0504007.

60. ^Seiter J, Soeken M, Wille R, Drechsler R (2012). "Property Checking of Quantum Circuits Using Quantum Multiple-Valued Decision Diagrams." In: International Workshop on Reversible Computation. Springer. pp. 183–196.

61. ^Wille R, Hillmich S, Burgholzer L (2022). "Decision Diagrams for Quantum Computing." In: Design Automation of Quantum Computers. Springer. pp. 1–23.

62. ^Wille R, Hillmich S, Burgholzer L (2022). "Tools for Quantum Computing Based on Decision Diagrams." ACM Trans Quantum Comput. **3**(3):1–17.

63. ^Yamashita S, Minato S, Miller DM (2008). "DDMF: An Efficient Decision Diagram Structure for Design Verification of Quantum Circuits Under a Practical Restriction." IEICE Trans Fundam Electron Commun Comput Sci. **91**(12):3793–3802.

64. ^Hong X, Huang WJ, Chien WC, Feng Y, Hsieh MH, et al. (2023). "Decision Diagrams for Symbolic Verification of Quantum Circuits." In: 2023 IEEE International Conference on Quantum Computing and Engineering (QCE). IEEE. pp. 970–977.

65. ^Wang SA, Lu CY, Tsai IM, Kuo SY (2008). "An XQDD-Based Verification Method for Quantum Circuits." IEICE Trans Fundam Electron Commun Comput Sci. **91**(2):584–594.

66. ^Wang Z, Cheng B, Yuan L, Ji Z (2025). "FeynmanDD: Quantum Circuit Analysis with Classical Decision Diagrams." In: International Conference on Computer Aided Verification. Springer. pp. 28–52.

67. ^Rand R, Paykin J, Zdancewic S (2018). "QWIRE Practice: Formal Verification of Quantum Circuits in Coq." arXiv preprint arXiv:180300699.

68. ^Amy M (2018). "Towards Large-Scale Functional Verification of Universal Quantum Circuits." arXiv preprint arXiv:180506908.

69. ^Kaivola R, Aagaard MD (2000). "Divider Circuit Verification with Model Checking and Theorem Proving." In: International Conference on Theorem Proving in Higher Order Logics. Springer. pp. 338–355.

70. ^Chen YF, Rümmer P, Tsai WL (2023). "A Theory of Cartesian Arrays (with Applications in Quantum Circuit Verification)." In: International Conference on Automated Deduction. Springer. pp. 170–189.

71. ^Mahadev U (2018). "Classical Verification of Quantum Computations." In: 2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS). IEEE. pp. 259–267.

72. ^Govindankutty A, Srinivasan SK, Mathure N (2023). "Rotational Abstractions for Verification of Quantum Fourier Transform Circuits." IET Quantum Commun. **4**(2):84–92. doi:10.1049/qtc2.12055.

73. ^Govindankutty A, Srinivasan SK (2025). "Superposition-Based Abstractions for Quantum Data Encoding Verification." IET Quantum Commun.

74. a, b, cTao R, Shi Y, Yao J, Li X, Javadi-Abhari A, et al. (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. pp. 641–656.

75. ^Chen YF, Chung KM, Lengál O, Lin JA, Tsai WL, et al. (2023). "An Automata-Based Framework for Verification and Bug Hunting in Quantum Circuits." Proc ACM Program Lang. **7**(PLDI):1218–1243.

76. a, bHietala K, Rand R, Hung SH, Wu X, Hicks M (2021). "A Verified Optimizer for Quantum Circuits." Proc ACM Program Lang. **5**(POPL):1–29.

77. ^Tang W, Tomesh T, Suchara M, Larson J, Martonosi M (2021). "Cutqc: Using Small Quantum Computers for Large Quantum Circuit Evaluations." In: Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems. pp. 473–486.

78. a, bBurgholzer L, Raymond R, Wille R (2020). "Verifying Results of the IBM Qiskit Quantum Circuit Compilation Flow." In: 2020 IEEE International Conference on Quantum Computing and Engineering (QCE). IEEE. pp. 356–365.

79. ^Villalonga B, Boixo S, Nelson B, Henze C, Rieffel E, et al. (2019). "A Flexible High-Performance Simulator for Verifying and Benchmarking Quantum Circuits Implemented on Real Hardware." npj Quantum Inf. **5**(1):86.

80. a, bWille R, Hillmich S, Burgholzer L (2020). "Efficient and Correct Compilation of Quantum Circuits." In: 2020 IEEE International Symposium on Circuits and Systems (ISCAS). IEEE. pp. 1–5.

81. ^Hurwitz L, Datta K, Kole A, Drechsler R (2024). "Is Simulation the Only Alternative for Effective Verificatio n of Dynamic Quantum Circuits?" In: International Conference on Reversible Computation. Springer. pp. 20 1–217.

82. ^Xu A, Molavi A, Pick L, Tannu S, Albarghouthi A (2023). "Synthesizing Quantum-Circuit Optimizers." Proc ACM Program Lang. **7**(PLDI):835–859.

83. ^Devitt SJ, Munro WJ, Nemoto K (2013). "Quantum Error Correction for Beginners." Rep Prog Phys. **76**(7):076 001.

84. ^Kitaev AY (1997). "Quantum Computations: Algorithms and Error Correction." Russ Math Surv. **52**(6):1191.

85. ^Knill E, Laflamme R (1997). "Theory of Quantum Error-Correcting Codes." Phys Rev A. **55**:900–911. doi:10. 1103/PhysRevA.55.900.

86. ^Huang Q, Zhou L, Fang W, Zhao M, Ying M (2025). "Efficient Formal Verification of Quantum Error Correct ing Programs." Proc ACM Program Lang. **9**(PLDI):1068–1093.

87. ^Poulin D (2005). "Stabilizer Formalism for Operator Quantum Error Correction." Phys Rev Lett. **95**(23):230 504.

88. ^Fang W, Ying M (2024). "Symbolic Execution for Quantum Error Correction Programs." Proc ACM Program Lang. **8**(PLDI):1040–1065.

89. ^Wu A, Li G, Zhang H, Guerreschi GG, Xie Y, et al. (2021). "QECV: Quantum Error Correction Verification." https://arxiv.org/abs/2111.13728.

90. ^Huang Q, Zhou L, Fang W, Zhao M, Ying M (2025). "Efficient Formal Verification of Quantum Error Correct ing Programs." Proc ACM Program Lang. **9**(PLDI):1068–1093. doi:10.1145/3729293.

91. ^Chen K, Liu Y, Fang W, Paykin J, Wu XC, et al. (2025). "Verifying Fault-Tolerance of Quantum Error Correcti on Codes." In: Piskac R, Rakamarić Z editors. Computer Aided Verification. Cham: Springer Nature Switzerla nd. pp. 3–27.

92. ^Chong FT, Franklin D, Martonosi M (2017). "Programming Languages and Compiler Design for Realistic Q uantum Hardware." Nature. **549**(7671):180–187. doi:10.1038/nature23459.

93. ^Salm M, Barzen J, Leymann F, Weder B, Wild K (2021). "Automating the Comparison of Quantum Compiler s for Quantum Circuits." In: Barzen J editor. Service-Oriented Computing. Cham: Springer International Pub lishing. pp. 64–80.

94. ^Nguyen T, Lyakh D, Pooser RC, Humble TS, Proctor T, et al. (2021). "Quantum Circuit Transformations with a Multi-Level Intermediate Representation Compiler." https://arxiv.org/abs/2112.10677.

95. ^Yan G, Wu W, Chen Y, Pan K, Lu X, et al. (2025). "Quantum Circuit Synthesis and Compilation Optimization: Overview and Prospects." https://arxiv.org/abs/2407.00736.

96. ^Liu J, Zhan B, Wang S, Ying S, Liu T, et al. (2019). "Formal Verification of Quantum Algorithms Using Quantum Hoare Logic." In: Dillig I, Tasiran S editors. Computer Aided Verification. Cham: Springer International Publishing. pp. 187–207.

97. ^Amy M, Lunderville J (2025). "Linear and Non-Linear Relational Analyses for Quantum Program Optimization." Proc ACM Program Lang. **9**(POPL):1072–1103. doi:10.1145/3704873.

98. ^Zuliani P (2007). "A Formal Derivation of Grover's Quantum Search Algorithm." In: First Joint IEEE/IFIP Symposium on Theoretical Aspects of Software Engineering (TASE '07). pp. 67–74. doi:10.1109/TASE.2007.3.

99. ^Witzel WM, Craft WD, Carr R, Kapur D (2023). "Verifying Quantum Phase Estimation Using an Expressive Theorem-Proving Assistant." Phys Rev A. **108**(5):052609. doi:10.1103/PhysRevA.108.052609.

100. ^Peng Y, Hietala K, Tao R, Li L, Rand R, Hicks M, Wu X (2023). "A Formally Certified End-to-End Implementation of Shor's Factorization Algorithm." Proc Natl Acad Sci USA. **120**(21):e2218775120. doi:10.1073/pnas.2218775120.

101. ^Alam MS, Wudarski FA, Reagor MJ, Sud J, Grabbe S, et al. (2022). "Practical Verification of Quantum Properties in Quantum-Approximate-Optimization Runs." Phys. Rev. Applied. **17**:024026. doi:10.1103/physrevapplied.17.024026.

Declarations

Funding: No specific funding was received for this work.

Potential competing interests: No potential competing interests to declare.