


Quantum Circuit Verification – A Potential Roadmap

Parosh Aziz Abdulla 

Uppsala University, Sweden
Mälardalen University, Sweden

Yu-Fang Chen

Academia Sinica, Taipei, Taiwan

Michal Hečko

Brno University of Technology, Czech Republic

Lukáš Holík


Brno University of Technology, Czech Republic

Ondřej Lengál 

Brno University of Technology, Czech Republic

Jyun-Ao Lin

National Taipei University of Technology, Taiwan

Ramanathan Thinniyam Srinivasan 

Uppsala University, Sweden

Abstract

Quantum technologies are progressing at an extraordinary pace and are poised to transform numerous sectors both nationally and globally. Among them, quantum computing stands out for its potential to revolutionize areas such as cryptography, optimization, and the simulation of quantum systems, offering dramatic speed-ups for specific classes of problems.

As quantum devices evolve and become increasingly pervasive, guaranteeing their correctness is of paramount importance. This necessitates the development of rigorous methods and tools to analyze and verify their behavior. However, the construction of such verification frameworks presents fundamental challenges. Quantum phenomena such as superposition and entanglement give rise to computational behaviors that differ profoundly from those of classical systems, leading to inherently probabilistic models and exponentially large state spaces, even for relatively small programs.

Addressing these challenges requires building on the extensive expertise of the formal methods community in classical program verification, while incorporating recent advances and collaborative efforts in quantum systems. An interesting challenge for the verification community is to design and implement novel verification frameworks that transfer the key strengths of classical verification, such as expressive specification, precise error detection, automation, and scalability, to the quantum domain. We expect that the results of this research will play a crucial role in enabling the dependable deployment of quantum technologies across a wide range of future applications.

2012 ACM Subject Classification Theory of computation → Program verification

Keywords and phrases Quantum Circuits, Quantum Computing, Program Verification, Automata, Model Checking

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2025.1

Category Invited Talk



© Parosh Aziz Abdulla, Yu-Fang Chen, Michal Hečko, Lukáš Holík, Ondřej Lengál, Jyun-Ao Lin, and Ramanathan Thinniyam Srinivasan;

licensed under Creative Commons License CC-BY 4.0

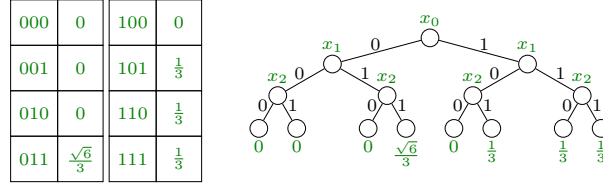
45th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2025).

Editors: C. Aiswarya, Ruta Mehta, and Subhajit Roy; Article No. 1; pp. 1:1–1:8



Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



■ **Figure 1** A quantum state with three qubits and its tree representation. The system can be in any of the eight basis states, with the corresponding probability amplitudes indicated in the table.

1 Background

Classical computing has undergone remarkable advances over the past decades and today forms the backbone of virtually all modern technological infrastructure. By contrast, *quantum computing* introduces a fundamentally different paradigm for information processing, one that is deeply rooted in the principles of quantum mechanics. This new model holds the potential to solve computational problems that are intractable by classical means.

The essential difference between classical and quantum computation lies in how information is represented and manipulated. Classical computers use *bits*, which can exist only in one of two binary states, 0 or 1. Quantum computers, on the other hand, operate on *quantum bits* (*qubits*), which obey the principles of *superposition* and *entanglement*. Superposition allows a qubit to exist in a combination of states simultaneously, enabling parallel exploration of many computational possibilities. Entanglement introduces correlations between qubits such that the state of one qubit can instantaneously influence the state of another, regardless of physical separation. Together, these features enable the simultaneous representation and manipulation of an exponentially large number of states and allow the encoding of complex correlations that classical systems cannot efficiently replicate.

Figure 1 illustrates a quantum state composed of three qubits.¹ In a classical setting, the system would be restricted to a single *basis state* among the eight possible configurations 000, 001, ..., 111. In contrast, a quantum system may exist in a superposition of all eight basis states simultaneously, each associated with a given *probability amplitude*.

A quantum state can therefore be viewed as a distribution over basis states, where each state is assigned a complex amplitude. This distribution can be represented as a tree: each path from the root to a leaf corresponds to a basis state, and each leaf stores the associated amplitude. Unlike classical probabilistic systems, where probabilities are real and sum to one, quantum amplitudes are complex numbers whose squared absolute values sum to one.²

A major milestone in the field is the achievement of *quantum supremacy*, the point at which a quantum computer can solve a problem beyond the reach of any classical supercomputer. Achieving quantum supremacy would yield exponential performance improvements on certain computational tasks. Although practical quantum computing remains in its early stages, applications are advancing rapidly. In the near term, *hybrid systems* that integrate classical and quantum processors are expected to become increasingly common. In the longer term, quantum supremacy has the potential to transform entire industries by enabling solutions to previously intractable problems.

¹ Throughout this document, we use simplified examples to highlight the verification challenges arising in quantum circuits. Detailed technical presentations can be found, for example, in [20].

² An *amplitude* generalizes the notion of probability. The squared magnitude of a complex amplitude corresponds to the probability of observing a basis state. The use of complex numbers allows for interference effects, where contributions with opposite phases cancel each other, enabling phenomena such as “negative probabilities” after squaring.

Promising application domains for quantum technologies include medical diagnostics, cryptography, secure communication infrastructures, and autonomous systems, among others. These areas require extremely high reliability and cannot tolerate critical system failures, making rigorous certification and assurance processes indispensable. Ensuring the dependability of quantum systems is therefore a central requirement for their widespread adoption.

However, verifying quantum systems is inherently challenging. Their probabilistic nature, combined with the exponential growth of state spaces as the number of qubits increases, creates profound obstacles for formal reasoning. Addressing these challenges demands new theoretical foundations and algorithmic techniques.

In our research, we are currently focusing on the *formal verification of quantum circuits*. Our approach is twofold: (i) to develop precise computational models that faithfully represent quantum state spaces and (ii) to design algorithmic techniques for verifying the behavior of quantum circuits [10]. In parallel with these theoretical efforts, we are building verification tools and evaluate them extensively through experiments on a diverse set of applications and case studies.

2 Purpose and Goals

It is essential to develop methods, algorithms, and tools for the automated verification of quantum systems, with particular focus on the unique challenges arising from their fundamentally different behavior. We envision significant added value in bridging two complementary areas of computer science, especially when established techniques from a mature field are adapted to address complex problems in an emerging domain. Such work embodies such interdisciplinary integration by applying well-developed methods from logic, automata theory, and symbolic verification to the problem of ensuring the correctness of quantum programs.

A particularly promising direction is the adaptation of techniques from automata theory, a foundational discipline in formal verification, to the verification of quantum systems. It would be interesting to use automata-based reasoning with symbolic representations tailored for the automated analysis of quantum behavior. The goal is to generalize core concepts from classical verification, such as state-space exploration and symbolic reasoning, into the quantum realm, thereby enabling the adaptation of robust classical paradigms to quantum computing, where formal assurances are essential.

We believe that such an approach also opens new avenues for research in automata theory itself. The mathematical structures encountered in quantum systems introduce both novel challenges and opportunities, extending the expressive and analytical power of automata-based techniques and fostering new theoretical developments. It establishes a conceptual bridge between quantum program verification and automata theory, fostering potential for new theoretical developments and extending the expressive and analytical power of automata-based techniques into the context of quantum computation. To achieve these goals, we have identified four specific objectives.

- We need to develop *symbolic representations*, based on *automata*, that serve as a theoretical and algorithmic foundation for efficiently modeling the state spaces of quantum circuits. These representations are grounded in the idea of modeling quantum states as binary trees, where each path from the root to a leaf corresponds to a computational basis state (see Figure 1). We would like to design new algorithms and establish complexity results for fundamental operations – such as language inclusion, successor computation, minimization, (bi)simulation, as well as predecessor and successor state analysis – which are vital for performing symbolic verification of quantum systems.

- We need to establish a framework for *algorithmic verification* of quantum systems. This framework will be based on a regular model checking approach, employing automata as symbolic representations of quantum state spaces. A central challenge in this effort is parameterized verification, where the goal is to prove correctness regardless of the number of input qubits or computational steps. This requires the development of new techniques for reasoning about infinite families of quantum states and operations, which is a significant departure from traditional finite-state verification methods.
- It is important to identify *key areas* of quantum computing where verification is critically needed. These include quantum measurement and circuit equivalence checking. By instantiating our methods in these application areas, we can try to demonstrate both the generality and the practical value of our verification techniques.
- To increase impact, we need to undertake a focused *implementation effort*. Our own existing tool, AutoQ [9, 10], is a case in point. It serves as a basis for implementing the above mentioned algorithms. We need to evaluate such tools extensively using a variety of representative applications and case studies, thereby validating the effectiveness and scalability of our approach.

We believe these objectives will help make significant contributions to the field of quantum computing by providing robust methods for ensuring the correctness and reliability of quantum systems.

3 Current Limitations and Potential Solutions

Research on the algorithmic verification of quantum systems is still in its infancy. Only recently have the first foundational contributions appeared, including initial work employing SAT solvers for quantum verification [15]. A number of recent studies have further advanced the field by introducing approaches for symbolic verification of quantum circuits [10, 2, 9].

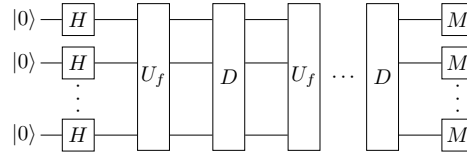
By contrast, the algorithmic verification of classical systems, particularly through *model checking* [12], has been a mature and active research area for nearly four decades. This line of work has yielded a substantial and well-established body of theory and practice, and continues to evolve rapidly, with results regularly published in leading venues such as POPL, CAV, PLDI, MICRO, and ASPLOS.

While extending verification techniques to quantum systems is a natural and necessary progression, it introduces a range of new and technically demanding challenges, as discussed in Section 1 and Section 2. The following sections review the current state of the art, outline its limitations, and summarize key results in both quantum and classical program verification that inform future research directions.

3.1 Symbolic Encodings

Symbolic state-space representations have been among the most powerful techniques in classical program verification over the past three decades. By representing program semantics symbolically, using logical formulas and decision structures, verification tools can efficiently reason about large or infinite state spaces and check correctness properties. Binary Decision Diagrams (BDDs) have been instrumental in hardware and protocol verification, while automata over words and trees have enabled the verification of parameterized systems and programs with dynamic data structures.

Despite their success in classical settings, symbolic techniques have so far seen limited application in quantum verification. The mathematical foundations of quantum computation – superposition, probabilistic measurement, and non-local entanglement – pose fundamental



■ **Figure 2** The classical Grover's search algorithm.

obstacles for classical symbolic reasoning. Nevertheless, suitably adapted symbolic methods are expected to play a crucial role in the scalability and precision of quantum program verification.

Recent work has introduced symbolic representations for sets of quantum states based on extended classes of tree automata [10, 9]. These automata generalize classical tree automata by allowing transitions labeled with sets of choices and enabling synchronization between subtrees within an accepted tree. Importantly, they retain key theoretical properties such as closure under union and intersection, and decidability of language emptiness and inclusion. Building on this foundation, new fully automated symbolic verification algorithms for quantum circuits have been proposed, supporting quantum gate operations with at most quadratic complexity – an improvement over earlier tree-automata-based methods with exponential worst-case complexity.

In the classical setting, numerous symbolic encodings have been proposed for infinite-state systems, including those based on automata, well-quasi-ordered constraint systems, and linear arithmetic. These techniques have been successfully applied to communication protocols, distributed algorithms, and real-time systems. Advances in automata-based verification include new simulation relations for tree automata and efficient algorithms for their computation [5], as well as the first notions of bisimulation for tree automata [11, 6], where the classical Paige–Tarjan algorithm was extended from word automata to tree automata. Subsequent work introduced additional bisimulation relations whose computation reduces to checking bisimulation over word automata [7].

Despite these advances, several limitations remain in the context of quantum verification.

- With the exception of [14] and a handful of more recent studies [10, 9], there is currently little work on symbolic encodings specifically tailored to the automated verification of quantum circuits. Existing approaches, such as quantum Hoare logic [17, 21], define predicates as mappings from mixed states to real values in $[0, 1]$, representing the probability that a state satisfies a given condition. These logics are inherently deductive and interactive, requiring significant manual reasoning, and are therefore unsuitable for automated verification.

A fundamentally different direction is a *set-based* approach, where predicates map quantum states to $\{0, 1\}$. Automata can then serve as compact representations of such predicates: a tree representing a quantum state is accepted if and only if the predicate evaluates to 1. This formulation facilitates the construction of efficient and fully automated verification algorithms, offering much greater scalability in practical applications.

- No algorithms currently exist for minimization, simulation, or bisimulation over automata models specifically designed for quantum verification. Developing such algorithms remains a central open problem and requires adapting classical automata-theoretic techniques to account for the unique algebraic structure and non-local behavior of quantum systems.

3.2 Algorithmic Verification

Algorithmic verification techniques, most prominently *model checking*, have been among the most influential approaches for verifying classical systems over the past three decades. They provide fully automated methods for determining whether a system satisfies a given specification and, unlike testing, can guarantee the absence of errors. Extending such techniques to quantum systems introduces fundamental challenges.

First, quantum systems naturally involve infinite state spaces, even when composed of a finite number of qubits. Second, they are inherently parameterized, often in multiple dimensions. For example, the circuit shown in Figure 2 implements Grover’s search algorithm, which operates on an n -bit Boolean function and achieves a quadratic speed-up over classical search, requiring $O(\sqrt{N})$ iterations for $N = 2^n$. Both the number of input qubits to the circuit and the number of computational stages (the number of gates) are unbounded, and correctness must hold across this entire parameter space.

Third, quantum systems exhibit *globally entangled transitions*: applying a gate to a single qubit can affect an exponential number of classical basis states. For instance, in Figure 1, negating qubit x_1 swaps multiple leaf positions simultaneously, demonstrating the non-local nature of quantum transformations.

Recent work [10, 9] has introduced verification tools for quantum circuits that are fully automated, support expressive property specification, and generate informative bug traces. These tools use a variant of tree automata [16] to represent sets of quantum states and to perform symbolic execution of quantum gates. Their improved scalability compared to traditional approaches is largely due to *level synchronization*, which constrains quantum operations to qubits located at the same tree level (Section 1).

In the classical domain, significant progress has been achieved in the verification of infinite-state programs, including systems operating under weak memory models [4, 3, 8]. Parameterized verification has been extensively studied and provides methods for proving correctness independently of the number of system components [12]. One particularly successful paradigm is *Regular Model Checking* [1], which represents sets of states using automata and transition relations using transducers. Reachability is determined by iteratively applying the transducer to the initial states and checking for intersection with the set of target states.

In a similar manner to above, there are several limitations in the current state of the art for quantum verification.

- Approaches such as SYMQV [13] use symbolic execution [18] and SMT solving to verify input-output relationships but face scalability limitations. The SMT-based approach of [15] improves encoding size but remains restricted. Other methods, such as quantum abstract interpretation [22, 19], offer automated analysis but rely on over-approximation, potentially reducing precision in bug detection.
- No approaches currently support regular model checking or parameterized verification for quantum systems. While some recent work [10] proposes algorithms for specific classes of parameterized quantum circuits, a general solution is still lacking. Bridging this gap requires the design and integration of model checking algorithms with abstraction techniques tailored to the mathematical and operational characteristics of quantum computation.

4 Conclusions and Outlook

Quantum circuit verification stands at a unique intersection of disciplines, bridging automata theory, quantum physics, and software engineering. This confluence not only enriches the methodological foundations of the field but also opens promising avenues for future research. A natural question is whether automata-based techniques, which have proven powerful for modeling and verifying quantum circuits, can be extended to other facets of quantum software engineering. For instance, can similar methods be adapted to reason about the more abstract and expressive constructs found in quantum programming languages?

Beyond detecting errors, the overarching goal of this line of work is to build confidence in quantum computing systems as they mature towards solving problems beyond the reach of classical computation. The challenges are substantial, stemming from the inherent complexity of quantum mechanics and the intricacies of quantum software, but so are the potential benefits. As quantum technologies advance, the demand for rigorous and scalable verification techniques will only intensify.

By harnessing the expressive power of automata theory and the precision of symbolic reasoning, we can make significant strides toward ensuring the reliability and correctness of quantum software. Such advances will be crucial for enabling the widespread adoption of quantum computing and realizing its transformative potential across science and technology.

References

- 1 Parosh Aziz Abdulla. Regular model checking. *STTT*, 14(2):109–118, 2012. doi:10.1007/s10009-011-0216-8.
- 2 Parosh Aziz Abdulla. A symbolic approach to verifying quantum systems, 2025. To appear. doi:10.1145/3725725.
- 3 Parosh Aziz Abdulla, Mohamed Faouzi Atig, Ahmed Bouajjani, and Tuan Phong Ngo. The benefits of duality in verifying concurrent programs under TSO. In *CONCUR*, volume 59 of *LIPICs*, pages 5:1–5:15. Schloss Dagstuhl, 2016. doi:10.4230/LIPICS.CONCUR.2016.5.
- 4 Parosh Aziz Abdulla, Mohamed Faouzi Atig, Yu-Fang Chen, Carl Leonardsson, and Ahmed Rezine. Counter-example guided fence insertion under tso. In *TACAS*, volume 7214 of *LNCS*, pages 204–219. Springer, 2012. doi:10.1007/978-3-642-28756-5_15.
- 5 Parosh Aziz Abdulla, Ahmed Bouajjani, Lukás Holík, Lisa Kaati, and Tomás Vojnar. Computing simulations over tree automata. In C. R. Ramakrishnan and Jakob Rehof, editors, *Tools and Algorithms for the Construction and Analysis of Systems, 14th International Conference, TACAS 2008, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2008, Budapest, Hungary, March 29-April 6, 2008. Proceedings*, volume 4963 of *Lecture Notes in Computer Science*, pages 93–108. Springer, 2008. doi:10.1007/978-3-540-78800-3_8.
- 6 Parosh Aziz Abdulla, Ahmed Bouajjani, Lukás Holík, Lisa Kaati, and Tomás Vojnar. Composed bisimulation for tree automata. *Int. J. Found. Comput. Sci.*, 20(4):685–700, 2009. doi:10.1142/S0129054109006814.
- 7 Parosh Aziz Abdulla, Ahmed Bouajjani, Lukás Holík, Lisa Kaati, and Tomás Vojnar. Composed bisimulation for tree automata. In *CIAA08, 13th International Conference on the Implementation and Applications of Automata*, 2008.
- 8 Parosh Aziz Abdulla, Karlis Cerans, Bengt Jonsson, and Yih-Kuen Tsay. General decidability theorems for infinite-state systems. In *Proceedings, 11th Annual IEEE Symposium on Logic in Computer Science, New Brunswick, New Jersey, USA, July 27-30, 1996*, pages 313–321. IEEE Computer Society, 1996. doi:10.1109/LICS.1996.561359.

- 9 Parosh Aziz Abdulla, Yo-Ga Chen, Yu-Fang Chen, Kai-Min Chung, Lukás Holík, Ondrej Lengál, Jyun-Ao Lin, Fang-Yi Lo, Wei-Lun Tsai, and Di-De Yen. An automata-based framework for quantum circuit verification. 2025.
- 10 Parosh Aziz Abdulla, Yo-Ga Chen, Yu-Fang Chen, Lukás Holík, Ondrej Lengál, Jyun-Ao Lin, Fang-Yi Lo, and Wei-Lun Tsai. Verifying quantum circuits with level-synchronized tree automata. *Proc. ACM Program. Lang.*, 9(POPL):923–953, 2025. doi:10.1145/3704868.
- 11 Parosh Aziz Abdulla, Johanna Höfberg, and Lisa Kaati. Bisimulation minimization of tree automata. *Int. J. Found. Comput. Sci.*, 18(4):699–713, 2007. doi:10.1142/S0129054107004929.
- 12 Parosh Aziz Abdulla, A. Prasad Sistla, and Muralidhar Talupur. Model checking parameterized systems. In Edmund M. Clarke, Thomas A. Henzinger, Helmut Veith, and Roderick Bloem, editors, *Handbook of Model Checking.*, pages 685–725. Springer, 2018. doi:10.1007/978-3-319-10575-8_21.
- 13 Fabian Bauer-Marquart, Stefan Leue, and Christian Schilling. symqv: Automated symbolic verification of quantum programs. In Marsha Chechik, Joost-Pieter Katoen, and Martin Leucker, editors, *Formal Methods - 25th International Symposium, FM 2023, Lübeck, Germany, March 6-10, 2023, Proceedings*, volume 14000 of *Lecture Notes in Computer Science*, pages 181–198. Springer, 2023. doi:10.1007/978-3-031-27481-7_12.
- 14 Yu-Fang Chen, Kai-Min Chung, Ondrej Lengál, Jyun-Ao Lin, Wei-Lun Tsai, and Di-De Yen. An automata-based framework for verification and bug hunting in quantum circuits. *Proc. ACM Program. Lang.*, 7(PLDI):1218–1243, 2023. doi:10.1145/3591270.
- 15 Yu-Fang Chen, Philipp Rümmer, and Wei-Lun Tsai. A theory of cartesian arrays (with applications in quantum circuit verification). In Brigitte Pientka and Cesare Tinelli, editors, *Automated Deduction - CADE 29 - 29th International Conference on Automated Deduction, Rome, Italy, July 1-4, 2023, Proceedings*, volume 14132 of *Lecture Notes in Computer Science*, pages 170–189. Springer, 2023. doi:10.1007/978-3-031-38499-8_10.
- 16 Hubert Comon, Max Dauchet, Rémi Gilleron, Florent Jacquemard, Denis Lugiez, Christof Löding, Sophie Tison, and Marc Tommasi. *Tree Automata Techniques and Applications*. 2008. URL: <https://inria.hal.science/hal-03367725>.
- 17 Ellie D’Hondt and Prakash Panangaden. Quantum weakest preconditions. *Mathematical Structures in Computer Science*, 16(3):429–451, 2006. doi:10.1017/S0960129506005251.
- 18 James C. King. Symbolic execution and program testing. *Commun. ACM*, 19(7), 1976. doi:10.1145/360248.360252.
- 19 Simon Perdrix. Quantum entanglement analysis based on abstract interpretation. In *International Static Analysis Symposium*, pages 270–282. Springer, 2008. doi:10.1007/978-3-540-69166-2_18.
- 20 Noson S. Yanofsky and Mirco A. Mannucci. *Quantum Computing for Computer Scientists*. Cambridge University Press, USA, 1 edition, 2008.
- 21 Mingsheng Ying. Floyd-Hoare logic for quantum programs. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 33(6):1–49, 2012. doi:10.1145/2049706.2049708.
- 22 Nengkun Yu and Jens Palsberg. Quantum abstract interpretation. In *Proceedings of the 42nd ACM SIGPLAN International Conference on Programming Language Design and Implementation*, pages 542–558, 2021. doi:10.1145/3453483.3454061.