Automated Reasoning with Complex Weights to Simulate and Verify Noisy Quantum Circuits

Study reference number: 24-5100
Type of activity: Standard study

Project Summary

Objective

To advance significantly the state-of-the-art in automated reasoning exploring the introduction of complex numbers in model counting techniques. To prove the value of the idea in the context of the simulation and verification of noisy quantum circuits.

Target university partner competences

Automated Reasoning, Formal Methods, Theoretical Computer Science

ACT provided competences

Quantum Computing, Fundamental Physics

Keywords

Quantum Computing, Model Counting, Simulation, Encodings

Study Objective

This study will examine either the theoretical background of algorithms and encoding techniques to manage complex weights in existing model counting pipelines, or the development of new model counters that support complex weights by design. Based on the chosen path, a new model-counting-based pipeline to simulate and verify quantum circuits with noise will be implemented, utilizing either existing counters in conjunction with the new encoding, or using the model counter developed within this study. The performance of the pipeline shall be assessed by comparing it to other quantum circuit simulators that support noise (e.g., Qiskit) and by benchmarking it on a broader range of model counting instances. To support this endeavour, the participation in computational challenges (e.g., the corresponding track of the model counting competition) is encouraged.

Background and Study Motivation

Interest in gate-based quantum computing has grown steadily in recent years, largely due to advancements in qubit scaling and improvements in error correction. Companies like IBM and Atom

Computing have announced processors with over 1,000 qubits, marking important progress toward practical, large-scale quantum computers [1, 2]. While the increase in qubit count is notable, ongoing improvements in gate fidelity and coherence times are equally important for the realization of useful quantum systems.

In the space domain, researchers are exploring quantum computing for optimizing satellite communication, mission scheduling, and processing large volumes of remote sensing data. For example, quantum algorithms have been proposed for satellite constellation optimization [4] and quantum machine learning techniques for analysing space-based sensor data [5]. The European Space Agency has initiated projects to assess quantum computing's impact on Earth observation [6,7].

However, scalable quantum computing remains a distant goal, as current hardware is limited by noise, qubit connectivity, and the overhead of quantum error correction [8,9]. As a result, near-term exploration of quantum applications must rely heavily on classical simulators to assess the performance, robustness, and practical feasibility of quantum algorithms [10,11]. These simulators provide a crucial testing ground for studying quantum behaviour under realistic noise models and device constraints. Furthermore, once practical quantum computers become available, simulators will continue to play a vital role in the verification and debugging of quantum programs—ensuring correctness, identifying sources of error, and validating outputs that are otherwise intractable to compute classically [12].

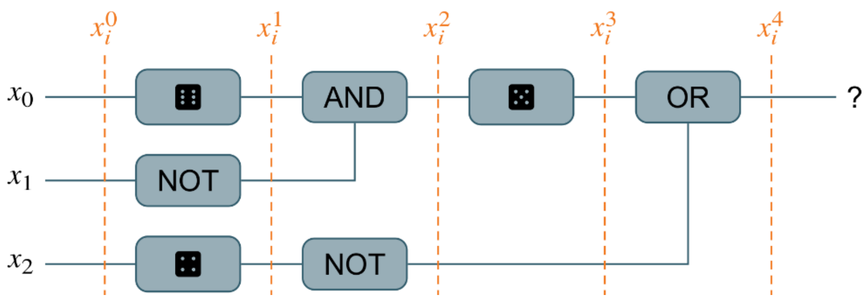

Figure 1: A probabilistic circuit with three inputs and one output. The gates with dice flip the input with a certain probability, the other are the usual logical gates. In orange the layers of the circuit are highlighted, which are referenced in the encoding in Figure 2.

Automated Reasoning emerged as a promising route in the quantum software development stack. For instance, SAT is used for quantum circuit mapping [20], MaxSAT is used to decode quantum colour codes [13], for quantum error correction decoders [14], and was deployed by ACT researchers as an intuitive interface to near-term quantum computers [15,16]. Model counting (#SAT) was identified as particularly promising for the simulation and verification of quantum circuits [17-19]. The connection between classical circuits and the satisfiability problem of propositional logic (SAT) is well-understood via the Cook–Levin theorem, which provides a fundamental corner stone of our understanding of computer science [21]. It is relatively easy to lift this connection to *probabilistic* circuits and (weighted) model counting (#SAT[ℝ]). Imagine for instance a circuit with 3 inputs, which are initially all set to false — see Figure 1, whereby "⚃", "⚄", and "⚅" gates negate the input with probability $^1/_4$, $^1/_5$, or $^1/_6$, respectively. In the context of a *strong simulation*, we are interested in the question: "What is the probability that the circuit outputs true?" We can answer this question using #SAT[$\mathbb{R}$] and the formula presented in Figure 2, which evaluates the circuit layer by layer and uses weighted variables b_1, b_2, and b_3 to emulate the random gates, e.g., the weight of b_2 is $w(b_2) = {}^{1}/_{4}$. The observation made in the cited articles is that this idea can be further lifted from probabilistic circuits to *quantum* circuits if the weights of the b-variables are allowed to be complex, i.e., if we consider #SAT[$\mathbb{Q}$].

$$
\begin{array}{lll}
 & \neg x_0^0 \wedge \neg x_1^0 \wedge \neg x_2^0 & \text{// input gates are false} \\
\wedge & (x_0^1 \leftrightarrow x_0^0 \oplus b_1) \wedge (x_1^1 \leftrightarrow \neg x_1^0) \wedge (x_2^1 \leftrightarrow x_2^0 \oplus b_2) & \text{// compute probabilities in layer 1} \\
\wedge & (x_0^2 \leftrightarrow x_0^1 \wedge x_1^1) \wedge (x_2^2 \leftrightarrow \neg x_2^1) & \text{// compute probabilities in layer 2} \\
\wedge & (x_0^3 \leftrightarrow x_0^2 \oplus b_3) \wedge (x_3^3 \leftrightarrow x_3^2) & \text{// compute probabilities in layer 3} \\
\wedge & (x_0^4 \leftrightarrow x_0^3 \vee x_2^3) & \text{// compute probabilities in layer 4} \\
\wedge & (x_0^4) & \text{// probability that true is output}
\end{array}
$$

Figure 2: A #SAT[$\mathbb{R}$] formula that simulates the circuit from Figure 1. For each layer of the circuit, there are variables describing the truth value of the various wires. Weighted auxiliary variables b_i are used to emulate the probabilities.

Unfortunately, while SAT, MaxSAT, and #SAT[$\mathbb{R}$] are well understood, #SAT[$\mathbb{Q}$] has received less attrition in the scientific literature so far. In particular, no model counter that supports complex weights by design is available. To utilize the connection between quantum circuits and #SAT[$\mathbb{Q}$] to enable strong simulations, one needs, thus, to either modify the circuit such that no complex numbers appear (e.g., change the gate basis), encode the resulting #SAT[$\mathbb{Q}$] into #SAT[$\mathbb{R}$], or develop a new model counter that natively supports complex weights. The central scientific question of this study is *which of these routes is the most promising one*.

Proposed Methodology

From the current state of research, three possible methodologies immediately emerge to get rid of the complex numbers in model-counting-based simulators:

1. Compile the given circuit into a gate basis that does not involve complex numbers (e.g., Toffoli+T), or introduce ancilla bits to simulate complex amplitudes using real ones.
2. Compile the obtained #SAT[$\mathbb{Q}$] formula into an equivalent #SAT[$\mathbb{R}$] formula from which the weighted count of the original formula can be derived (this compilation process could also produce multiple formulas, e.g., one for the real and one for the imaginary part of the solution).
3. Develop a model counter that supports complex weights, i.e., that solves #SAT[$\mathbb{Q}$]. Adaptations of existing tools are possible, but careful consideration on how these changes influence internal heuristics need to be carried out.

The ACT is in the process of developing a quantum circuit simulator in Rust, *qrab,* that is able to map quantum circuits to #SAT[$\mathbb{R}$] formulas using technique 1, or to #SAT[$\mathbb{Q}$] formulas in general. To compare the different strategies, applicants are invited to propose strategies to solve the produced #SAT[$\mathbb{Q}$] formulas using technique 2 or 3 (or both). For item 2, we have in mind a throughout theoretical study of the subject including algorithmic and complexity theoretic aspects, as well as clean and polished description of the relevant encodings. It is encouraged to provide multiple different encodings or strategies if applicable, and it is required that at least one translation from #SAT[ℚ] to #SAT[ℝ] is implemented. For item 3, we envision a highly efficient *library* to solve #SAT[ℚ] that can be incorporated into *qrab* (preferably via Rust bindings).

Alternative methodologies and different approaches are welcome as well, as long as they are supported by accompanying arguments and are based on tools from automated reasoning. For instance, a potential route could be the use of SAT or MaxSAT to improve the compilation process in technique 1.

As final goal, the study aims to establish an end-to-end pipeline that performs a strong simulation of a given (potentially noisy) quantum circuit using model counting. The inhouse solution *qrab* of the ACT will be used to parse the quantum circuit and to apply various noise models to them. It can also be used to transform the circuit into #SAT[ℚ] instances which then must be solved by the proposed solutions — either via an encoding to #SAT[ℝ] or using a newly developed model counter that support complex weights.

ACT Contribution

The project will be conducted in close scientific collaboration with ACT researchers. In particular, ACT researchers will provide technical expertise in quantum computing, quantum circuits, noise models, and fundamental physics. Scientists of ESA will also constantly incorporate progress made during the study into *qrab* to evaluate the approach. Benchmark formulas will regularly be produced and shared with the applicants to assess the chosen methodology. In case the participation in a computational challenge is desired (like the corresponding track of the model counting competition), the ACT will support the preparation of a submission.

Bibliography

[1] https://www.idtechex.com/en/research-article/how-will-gate-based-quantum-computers-scale-up/30918

[2] https://quantumzeitgeist.com/a-brief-look-at-qubit-growth-and-the-rise-of-quantum-computing/

[3] https://www.plainconcepts.com/quantum-computing-potential-challenges/

[4] *Vinicius Marchioli et al.:* **Scheduling of Satellite Constellation Operations in EO Missions Using Quantum Optimization.** Optimization, Learning Algorithms and Applications, 2024

[5] *Su Yeon Chang and Bertrand Le Saux:* **Quantum Machine Learning for Earth Observation Images.** NeurIPS 2nd WS on Quantum Tensor Networks in Machine Learning, 2022.

[6] https://eo4society.esa.int/projects/qc4eo-study/

[7] https://esastar-publication-ext.sso.esa.int/ESATenderActions/details/92141

[8] *John Preskill:* **Quantum Computing in the NISQ Era and Beyond.** Quantum, 2018

[9] *Earl T. Campbell, Barbara M. Terhal, and Christophe Vuillot:* **Roads Towards Fault-Tolerant Universal Quantum Computation.** Nature, 2017

[10] *Thomas Häner, Damian S Steiger, Krysta Svore, and Matthias Troyer:* **A Software methodology for compiling quantum programs.** Quantum Science and Technology, 2016

[11] *Edwin Pednault et al.:* **Breaking the 49-Qubit Barrier in the Simulation of Quantum Circuits.** arXiv:1710.05867

[12] *Hsin-Yuan Huang, Richard Kueng, and John Preskill:* **Predicting many properties of a quantum system from very few measurements.** Nature Physics, 2020

[13] *Lucas Berent et al.:* **Decoding Quantum Color Codes with MaxSAT.** Quantum 2023

[14] *Mohammadreza Noormandipour and Tobias Haug:* **MaxSAT Quantum Error Correction Decoders: a new formulation and benchmark study.** The Quantum Techniques in Machine Learning Conference 2024

[15] *Max Bannach, Jai Grover, and Markus Hecher:* **Strong Structural Bounds for MaxSAT: The Fine Details of Using Neuromorphic and Quantum Hardware Accelerators.** NASA Formal Methods 2025

[16] *Max Bannach:* **Ising Machines with Feasibility Guarantee.** SPAICE 2024

[17] *Jingyi Mei, Marcello M. Bonsangue, and Alfons Laarman:* **Simulating Quantum Circuits by Model Counting.** CAV 2024

[18] *Arend-Jan Quist, Jingyi Mei, Tim Coopmans, and Alfons Laarman:* **Advancing Quantum Computing with Formal Methods.** FM 2024

[19] *Jingyi Mei, Tim Coopmans, Marcello M. Bonsangue, and Alfons Laarman:* **Equivalence Checking of Quantum Circuits by Model Counting.** IJCAR 2024

[20] *Jiong Yang, Yaroslav A. Kharkov, Yunong Shi, Marijn J. H. Heule, and Bruno Dutertre:* **Quantum Circuit Mapping Based on Incremental and Parallel SAT Solving.** SAT 2024

[21] *Joseph Y. Halpern, Robert Harper, Neil Immerman, Phokion G. Kolaitis, Moshe Y. Vardi, and Victor Vianu:* **On the Unusual Effectiveness of Logic in Computer Science.** Bull. Symb. Log., 2001