# Program analysis for quantum computing

Jiawei Ren's Research Proposal

## Introduction

Quantum computing plays a critical role in exploiting the ability to more efficiently solve problems compared with using a classical computer. Thanks to quantum superiority, many famous quantum algorithms, such as Shor's Algorithm[1], Grover's Algorithm[2], etc. have been developed to reduce the time complexity from exponential to polynomial. Moreover, quantum machine learning algorithms[3] are also proposed to step machine learning from classical computer to the quantum world.

## Problem statement

Quantum programming has different programming logic compared to programming on classical computers, which may lead to programmers who are familiar with programming on classical computers make mistakes. Besides, it is hard to know the internal state of a quantum computer, because the quantum computer has entangled states and we cannot copy a variable as we did on classical computers[4]. Thus, it is important to explore approaches to analyze quantum programs so that the program executes correctly. An assertion is a useful tool for debugging and checking the correctness of a quantum program. The goal of my study will focus on developing an efficient assertion method.

## Preliminary Literature review

Currently, many researchers have made contributions to the run-time quantum assertion. One approach is based on the Quantum State Tomography(QST). Statistical tests on classical observations[5] were proposed by Huang and Martonosi. This method executes the program from the beginning to the place of assertion followed by measurement. It is beneficial for programmers to realize a bug the first-time running, but the approach needs to execute the program repeatedly if there are several

assertions injected, which is very time-consuming. Dynamic assertion circuits[6] were proposed by Liu et al. to improve the assertion scheme. In this approach, ancilla qubits were introduced and the method could detect in noisy scenarios.

Another direction is based on projection. Projection-based run-time assertion[7] was proposed by Li et al. to enhance the ability of quantum assertion. The advantage of the projection-based predicate is its strong expressive power and efficient run-time checking property. The method could be used in more general cases by considering three rank conditions and support multiple assertions without repeated executing the program. However, it still needs programmers to manually analyze and design the projection operator.

## Methodology

The first challenge is to choose a suitable representation for quantum computing run-time assertion. Currently, two candidates are QST and projection-based predicate. Projection-based predicate looks good at this stage, but there may be other representation methods that are more beneficial for the run-time assertion. This will require further study and investigation.

The second challenge is the limited resource of public accessible quantum computers. IMB Q is a well-known platform for the public to get in touch with quantum computing. However, it only offers maximum 16 qubits. Hence, we can only do small-scale experiments and hope for the success of the implementation of quantum computers.

Thirdly, the difficulty of debugging is the superposition and no-clone property of the quantum system. Besides, because of the physical limitation, it is easier to detect $|0\rangle$ and $|1\rangle$ compared to superposition. Hence, how to transform and restore the state is important. This can be turned into a matrix decomposition question for quantum

computing, which requires high time complexity.

Finally, the current versions of run-time assertion all need manual design, which requires the programmer to have good knowledge of quantum computing. However, software developing and testing are usually separated in software engineering. Automatic implementation of run-time assertion is an interesting target for the future. Thanks to previous researcher's works, many general functions are proposed for solving problems. Thus, finding a way to generalize the quantum assertion is the final goal.

## Schedule

| | |
|---|---|
| Months 1-6 | Summarize previous work and expand literature review to get the global knowledge of this topic.<br>Identify specific aim of the project. |
| Months 7-11 | Confirm my research target and plan.<br>Collect new methods which could improve current approach. |
| Months 12-16 | Design and do experiment to get data. |
| Months 17-27 | Write research paper draft.<br>Ask external reviewer to review draft and provide comments. |
| Months 28-33 | Rewrite research paper based on comments. |
| Months 33-36 | Submit phd paper to UTS graduate school and review. |

# Reference


[1] Peter W Shor. 1999. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM review 41, 2 (1999), 303–332.

[2] Lov K Grover. 1996. A fast quantum mechanical algorithm for database search. In Proceedings of the twenty-eighth annual ACM symposium on Theory of computing. 212–219.

[3] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. 2017. Quantum machine learning. Nature 549, 7671 (2017), 195–202.

[4] Andriy Miranskyy, Lei Zhang, and Javad Doliskani. 2020. Is Your Quantum Program Bug-Free? arXiv preprint arXiv:2001.10870 (2020).

[5] Yipeng Huang and Margaret Martonosi. Statistical assertions for vali- dating patterns and finding bugs in quantum programs. In Proceedings of the 46th International Symposium on Computer Architecture, pages 541–553. ACM, 2019.

[6] Ji Liu, Gregory T Byrd, and Huiyang Zhou. Quantum circuits for dynamic runtime assertions in quantum computation. In Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems, pages 1017–1030, 2020.

[7] Gushu Li, Li Zhou, Nengkun Yu, Yufei Ding, Mingsheng Ying, and Yuan Xie. 2019. Poq: Projection-based Runtime Assertions for Debugging on a Quantum Computer. arXiv preprint arXiv:1911.12855 (2019).