Identifying and Detecting Bugs in Quantum Programs

趙, 鵬展

出版情報：Kyushu University, 2023, 博士（工学）, 課程博士
バージョン：
権利関係：

（様式 2）

氏　　名　：趙　鵬展

論 文 名　：Identifying and Detecting Bugs in Quantum Programs
（量子プログラムのバグの特定と検出）

区　　分　：甲

論　文　内　容　の　要　旨

Quantum programming involves designing and constructing executable quantum programs to achieve specific computational results. Several quantum programming languages, such as Scaffold, Quipper, Qiskit, Q#, and Cirq, enable researchers and developers to implement and experiment with quantum computing techniques efficiently. However, ensuring the correctness of quantum programs is challenging due to unique features like superposition, entanglement, and no-cloning, which make bug detection difficult.

While some approaches for debugging and testing quantum software have been proposed, the research in identifying and detecting bugs in quantum programs is still in its early stages. Existing research has shortcomings in the following areas:

1. Redundancy in Bug Patterns: Many publicly reported bugs follow similar patterns, necessitating proper bug pattern classification metrics to assist developers in understanding and avoiding these bugs effectively.
2. Lack of Suitable Bug Benchmark Suites: The absence of comprehensive bug benchmark suites restricts the systematic evaluation of debugging and testing methods for quantum programs, hindering the progress of quantum software development.
3. Challenges of Dynamic Techniques and Classic Static Analysis: Dynamic techniques relying on the execution of quantum programs can be cumbersome and expensive, while classic static analysis tools designed for traditional programs are unsuitable for analyzing quantum programs due to their unique mechanisms.

To address these challenges and improve the quality assurance of quantum software, this thesis primarily focuses on identifying and detecting bugs in quantum programs. The work is divided into three phases:

1. Identification and Categorization of Bug Patterns: The thesis identifies and categorizes bug patterns in the Qiskit quantum programming language. This provides researchers and programmers with a clear understanding of the types of bugs that can occur in quantum programs and how to detect them. The bug patterns cover quantum-related constructs, focusing on symptoms, root causes, cures, and preventions.
2. Bugs4Q Benchmark Suite: As an initial step towards evaluating debugging and testing tools for quantum software, the thesis introduces Bugs4Q. This benchmark suite consists of forty-two real Qiskit bugs obtained from popular platforms like GitHub, StackOverflow, and Stack Exchange. Bugs4Q includes a bug database and test cases to reproduce faulty behaviors. Researchers have access to the actual bugs and corresponding fixes for further study. The Bugs4Q framework modularizes bugs and offers a user-friendly interface for easy extension. The buggy programs are also evaluated with two test case generation tools, demonstrating the benchmark's effectiveness in evaluating quantum program testing techniques.
3. QChecker: To efficiently detect bugs in quantum programs, the thesis presents QChecker, a static analysis tool specifically designed for Qiskit. QChecker comprises two modules: one for extracting

program information based on the abstract syntax tree (AST) and another for detecting bugs based on patterns. The performance of QChecker is evaluated using the Bugs4Q benchmark suite, demonstrating its ability to detect various bugs in quantum programs effectively.

In summary, this thesis fills the gaps in identifying and detecting bugs in quantum programs by: 1) Providing bug patterns as a foundation for further research in debugging and testing quantum programs. 2) Introducing Bugs4Q, a comprehensive benchmark suite that offers a comprehensive view of quantum bugs and facilitates the evaluation of quantum debugging and testing techniques. 3) Developing QChecker, a static analysis tool that efficiently detects bugs in quantum programs.