Related and Future Work

1 Related Works

1.1 Verification of Quantum Programs

Several approaches have been employed for formally verifying quantum programs. Unlike our research, which focuses on verifying an execution model of quantum programs, these methods are used to validate a particular property of a specific quantum program.

1.1.1 QWIRE: A Core Language for Quantum Circuits

$\mathcal{Q}$WIRE is a language for defining quantum circuits and an interface for their manipulation within any classical host language. The authors proved that every well-typed static circuits written in $\mathcal{Q}$WIRE preserves "$\mathrm{tr}(\rho) = 1$" property of density matrices.

1.2 Formal Verification of Quantum Algorithms Using Quantum Hoare Logic

In this study, the authors formalized the theory of quantum Hoare logic to reason about quantum programs. As an application, they verified the correctness of Grover's search algorithm.

1.3 Testing

Several pieces of research on testing quantum computing platforms such as Qiskit, including tests on physical semantic errors, have been conducted.

1.3.1 QDiff: Differential testing of Quantum Software Stacks

Qdiff performs differential testing across different backends and optimization levels of quantum computing platforms. The authors employed K-S testing and cross entropy to measure distances between statistical distributions.

1.3.2 MorphQ: Metamorphic Testing of the Qiskit Quantum Computing Platform

MorphQ conducts metamorphic testing on a novel set of metamorphic transformations. The authors use K-S testing to calculate a p-value of distribution equivalence.

2 Future Work

2.1 Optimization: Avoiding Unnecessary

2.1.1 Avoiding Unnecessarily Splitting World

The division each world into two doubles the cost of subsequent operations. If through analysis of a quantum circuit reveals that some qubits remain unaltered throught the circuit ater measurement, it becomes unnecessary to preserve the measurement result in a classical register, as the qubit itself retains the information of the measurement output. By eliminating the classical bit stroing the measurement output information, the need for dividing the world is also removed as the classical states are not impacted by the measurement. Hence, it suffices to simply add the two resultant density matrices, each multiplied by its measurmeent probability. (From Classical to Quantum Shannon Theory (mark M. Wilde): page 159)

2.1.2 Representation of Matrix as a Two-Dimensional Array

As previously noted, utilizing a function to represent a matrix is mathematically elegant and simplifies the proof of physical consistency. However, this design choice may not offer optimal computational performance. If we establish physical consistency while representing the matrix as a two-dimensional array, the resulting OCaml code will likely demonstrate superior performance.

2.2 Support for OpenQASM3.0

OpenQASM3.0 supports more expresesive and complex dynamic quantum-classical high-level interactions. Supports for OpenQASM3.0 is in its infancy and the implementation is expected to change significantly. (quoted from the official document) Hence, a physically-consistent execution for OpenQASM3.0 will prove beneficial for developing support for OpenQASM3.0.