A Survey of Quantum Programming Languages: History, Methods, and Tools
Donald A. Sofge
Proceedings of the Second International Conference on Quantum, Nano, and Micro Technologies
2008
Dylan Miracle
ICS 698-02
Spring 2021
Feb 17, 2021
Dr. Jigang Liu

A Survey of Quantum Programming Languages: History, Methods, and Tools

Dylan Miracle
Department of Computer Science
Metropolitan State University
St. Paul, Minnesota, USA
dylan.miracle@my.metrostate.edu

March 2, 2021

1 Background

This paper is old in the world of quantum computing, but it gives a look at the early days of quantum language development. Theoretical work has been done on quantum information since Von Neuman in the 1930's, but actual computational frameworks did not begin developing unit the work of Feynman in 1982 and Deutsch in 1985. Deutsch proposed a quantum Turing machine, a model that is used to show an example of a quantum algorithm that has substantial speedup over its classical counterpart.

2 Main idea/conclusion

Quantum programming languages, like classical languages, can be imperative or functional. Various approaches have been made to developing quantum algorithms including the Quantum Turing Machine, Linear logic machines, quantum random access machine (QRAM). The gate model of quantum algorithms builds on QRAM fundamentals and is the most developed.

3 Support facts/algorithms/methods

Origins of quantum computing date to ideas of Feynman in 1982 who proposed the quantum computer as a means of simulating other quantum systems such as molecules. Classical simulations of quantum systems require exponential resources. In the example of a molecule this can be understood because as we add more atoms we have to track the interactions between all the atoms in the molecule. As the number of atoms increase the number of pairwise interaction increases exponentially.

Several important quantum algorithms have been developed using the gate model of quantum computing. These include the Deutsch-Jozsa algorithm which is an example of a toy problem that a quantum computer can solve better than a classical computer.

4 Arguments/disagreements/concerns

Complexity classes in quantum mechanical problems are thought to be different than those of classical computing, however this is not proven. All this work could end up without appreciable gains for most problems, and the technological leads that the digital computer has means that we would be better off using a classical computer for most problems if we cannot reduce complexity using a quantum computer.

While a taxonomy separating functional and imperative languages is useful as an analogy to digital computing, there is probably a more logical split quantum languages. One taxonomy that would be interesting is the levels of abstraction available to different languages. For example QASM is a language that only allows direct control of quantum gates, while qiskit gives access to applications in different industries, apis that connect to hardware and built in simulators. A taxonomy that takes into account the depth or levels of abstraction available for different languages would be useful.

5 Interesting findings

Early languages focused on the Quantum Turing Machine. This idea was coined by Deutsch and extended to study complexity theory of quantum computing. This was however not a very useful machine for implementing quantum algorithms and has been supplanted by the gate model of quantum computing for algorithm design.

The authors propose a taxonomy of quantum programming languages: imperative, functional, other quantum programming language paradigms. This is useful to a classical programmer as we are already familiar with the imperative/ functional divide.

6 Quotations

Quantum programming languages may be taxonomically divided into (A) imperative quantum programming languages, (B) functional quantum programming languages, and (C)others (may include mathematical formalisms not intended for computer execution).

The difficulties in formulating useful, effective, and in some sense universally capable quantum programming languages arise from several root causes. First, quantum mechanics itself (and by extension quantum information theory) is incomplete. Specifically missing is a theory of measurement. Quantum theory

is quite successful in describing the evolution of quantum states, and even in predicting probabilistic outcomes after measurements have been made, but the process of state collapse is (with a few exceptional cases) not covered. So issues such as decoherence, diffusion, entanglement between particles (or entangled state, of whatever physical instantiation), and communication (including teleportation) are not well defined from a quantum information (and by extension quantum computation) perspective. Work with semantic formalisms and linear logic attempt to redress this by providing a firmer basis in a more complete logic consistent with quantum mechanics.