doi:10.20396/revpibic.v0i0.id

XXVI Congresso de Iniciação Científica Unicamp

17 a 19 de outubro Campinas | Brasil

Implementation of Deutsch-Jozsa Quantum algorithm in functional programming language.

Lucas Lacava*, Mariana Godoy Vazquez.

Abstract

The goal of this research is to show the implementation of Deutsch-Jozsa quantum algorithm in functional programming language and its advantages comparing to Classical Computing. Are presented definitions and characteristics of the algorithm and the language Quipper, in which the implementation of the Deutsch-Jozsa quantum algorithm occurs.

Key words:

Quantum Computing, Functional Programming Language, Quipper.

Introduction

Physics and Quantum Computing are constantly evolving and have attracted interest from researchers in several fields. Therefore, it is important that programmers understand the logic of this new paradigm.

The study of this theme is justified by the obsolescence of Moore's Law that predicted 100% of processing capacity growth every 18 months while its cost would remain constant. According to Nielsen and Chuang (2010)[1], a possible explanation for the flaw in Moore's law is the paradigm shift, since Quantum Computation is based on quantum mechanics rather than Classical Physics, allowing the expansion of research.

The goal of this work is the study of Deutsch-Jozsa algorithm and practical demonstration of the implementation and its advantages.

Results and Discussion

According to Nielsen and Chuang (2010) the Deutsch-Jozsa algorithm describes a problem where it is necessary to define whether a function is constant or balanced, if it is constant, all results will be equal to 0 or equal to 1, if balanced, half of the results will be equal to 0 and the other half equal to 1.

In Classical Computing, at worst, it would be necessary $\left(\frac{2n}{2}\right) + 1$ tries to define the type of function, for example: if the situation studied has 3 bits, there are eight possible combinations: 000, 001, 010, 011, 100, 101, 110, 111.

Considering the number of bits, we have: $\left(\frac{2^3}{2}\right) + 1 = \left(\frac{8}{2}\right) + 1 = 5$. This way, it would take five parameters to be sure of the type of the function, but the great advantage in this algorithm is that the result is discovered with only one parameter. Image 1 exemplifies the circuit with the Deutsch-Jozsa algorithm, with the Hadamard port and the oracle function being applied to both qubits, and then the Hadamard port on the qubits with zero value and a measurement performed at each interval:

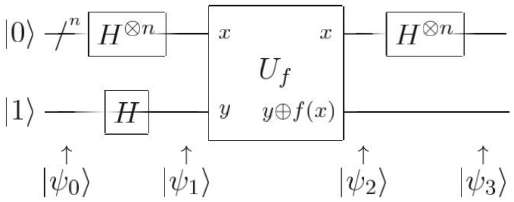

Image 1. Quantum circuit of Deutsch-Jozsa algorithm.

Posteriorly the algorithm was implemented in Quipper, a framework based on Haskell and focused on the development of applications for Quantum Computing.

As shown in Image 1, the Hadamard gate was applied in n qubits in state |0⟩ and a qubit in state |1⟩, the oracle function in both, and lastly, the Hadamard gate in the qubits in state |0⟩, in addition the measurements were performed between each application of quantum gate, and $|\psi_3\rangle$ is the result, defining whether the function is constant or balanced. The language also offers a pdf file that is exported with the diagram equivalent to what was programmed.

Briefly, the operation of the algorithm in its first phase realizes the import of the necessary libraries, creation of oracle function, initialization of the qubits, application of Hadamard in all the qubits, application of the oracle function in all the qubits, application of Hadamard in the set of qubits in state |0⟩, and finally, the measurement of the results.

In the second step a simulate function is created to perform the simulation that will return a Boolean value and operators such as *map*, *not*, and *and* are used. The next function, circuit, uses the result of the simulate function, if it is true, the function is constant, otherwise, it is balanced.

Conclusions

The implementation of Deutsch-Jozsa algorithm in Quipper, a framework based on Haskell, demonstrated how important is the use of functional languages for the advance of Quantum Computation, since performance and efficiency are superior in relation to the equivalent in Classical Computing.

[1] NIELSEN, Michael A.; CHUANG, Isaac L.. Quantum Computation and Quantum Information. Cambridge: Cambridge University Press, **2010**. 698 p.