Quantum Predicative Programming

Anya Tafliovich

University of Toronto, Toronto ON M5S 3G4, Canada
anya@cs.toronto.edu

1 Introduction

Quantum computation and quantum information is the study of information processing and communication accomplished with quantum mechanical systems. In recent years the field has grown immensely. Scientists from various fields of computer science have discovered that thinking physically about computation yields new and exciting results in computation and communication. There has been extensive research in the areas of quantum algorithms, quantum communication and information, quantum cryptography, quantum error-correction, measurement-based quantum computation, theoretical quantum optics, and more. Experimental quantum information and communication has also been a fruitful field.

The subject of our work is quantum programming — developing programs intended for execution on a quantum computer. We assume a model of a quantum computer proposed by Knill [1]: a classical computer with access to a quantum device that is capable of storing quantum bits, called qubits, performing certain operations and measurements on these qubits, and reporting the results of the measurements.

Traditionally, quantum computation is presented in terms of quantum circuits. Recently, there has been an attempt to depart from this convention for the same reason that classical computation is generally not presented in terms of classical circuits. As we develop more complex quantum algorithms, we will need ways to express higher-level concepts with control structures in a readable fashion.

We look at programming in the context of formal methods of program development, or programming methodology. Our theory of quantum programming is based on probabilistic predicative programming, a recent generalisation of the well-established predicative programming [2, 3], which we deem to be the simplest and the most elegant programming theory known today. It supports the style of program development in which each programming step is proven correct as it is made. We inherit the advantages of the theory, such as its generality, simple treatment of recursive programs, and time and space complexity. Our theory of quantum programming provides tools to write both classical and quantum specifications, develop quantum programs that implement these specifications, and reason about their comparative time and space complexity all in the same framework.

Since the work of Ömer [4] in 2000, several attempts have been made to formalize analysis and development of quantum algorithms and quantum communication protocols. For a review of related work, the reader is referred to [5].

2 Quantum Predicative Programming

In this section, we demonstrate our approach by developing the solution to the Deutsch-Jozsa's problem ([6]), an example of the broad class of quantum algorithms that are based on the quantum Fourier transform. Readers unfamiliar with probabilistic predicative programming and/or basics of quantum computing are referred to [5] for introduction.

In quantum predicative programming we define a state of an n-qubit system as a function $\psi : 0,..2^n \to \mathbb{C}$, such that $\sum x : 0,..2^n \cdot |\psi x|^2 = 1$. The unitary transformations that describe the evolution of a n-qubit quantum system are operations U defined on the system, such that $U^\dagger U = I^n$, where I^n is the identity operation and $U^\dagger$ denotes the adjoint of U. The simplest and the most commonly used measurement in the computational basis is defined by

$$\textbf{measure}\ \psi\, r == \ |\psi r'|^2 \times (\psi' = |\mathbf{r'}\rangle) \times (\sigma' = \sigma)$$

The most general quantum measurement is defined in [5].

The task in the Deutsch-Jozsa's problem is as follows: given a function $f : 0,..2^n \to 0,1$, such that f is either constant or balanced, determine which case it is. Without any restrictions on the number of calls to f, we can write the specification (let us call it S) as follows:

$$(f\ \textit{is constant} \vee f\ \textit{is balanced}) \implies\ b' = f\ \textit{is constant}$$

where b is a boolean variable and the informally stated properties of f are defined formally as follows:

$$f\ \textit{is constant} ==\ \forall i : 0,..2^n \cdot fi = f0$$

$$f\ \textit{is balanced} ==\ \left|\sum i : 0,..2^n \cdot (-1)^{fi}\right| = 0$$

It is easy to show that

$$(f\ \textit{is constant} \vee f\ \textit{is balanced})$$
$$\implies (f\ \textit{is constant} ==\ \forall (i : 0,..2^{n-1} + 1) \cdot fi = f0)$$

That is, more than half of the values need to be equal to *f0*.

In our setting, we need to implement the specification R defined as follows:

$$b' ==\ \forall i : (0,..2^{n-1} + 1) \cdot fi = f0$$

The idea for a quantum solution is to create a suitable superposition for state ψ, so that a measurement of ψ produces 0 if and only if f is constant, so that:

$$S \Longleftarrow\ Q;\ b := (r = 0) \qquad \text{, where}$$
$$Q ==\ f\ \textit{is constant} \vee f\ \textit{is balanced} \Rightarrow f\ \textit{is constant} = (r' = 0)$$

To implement Q we notice that:

$$\begin{aligned} f \textit{ is constant} &= \left(\left|\sum x \cdot (-1)^{fx}/2^n\right| = 1\right) \\ f \textit{ is balanced} &= \left(\left|\sum x \cdot (-1)^{fx}/2^n\right| = 0\right) \end{aligned}$$

We can show that if $f \textit{ is constant} \vee f \textit{ is balanced}$, variables x, y, and z are from the domain $0, ..2^n$, and $\mathbf{x} \cdot \mathbf{z}$ is the dot product of $\mathbf{x}$ and $\mathbf{z}$, then:

$$\begin{aligned} & f \textit{ is constant} = (r' = 0) \\ \Longleftarrow & \left|\left(\sum z, x \cdot (-1)^{\mathbf{x}\cdot\mathbf{z}+fx}/2^n \times |\mathbf{z}\rangle\right)\ r'\right|^2 \\ = & \ \mathbf{measure}\ \left(\sum x \cdot (-1)^{fx}/\sqrt{2^n} \times \left(\sum z \cdot (-1)^{\mathbf{x}\cdot\mathbf{z}}/\sqrt{2^n} \times |\mathbf{z}\rangle\right)\right)\ r \\ = & \ \mathbf{measure}\ (H^{\otimes n}(U_f(H^{\otimes n}|0\rangle^{\otimes n})))\ r \\ = & \ \psi := |0\rangle^{\otimes n};\ \psi := H^{\otimes n}\psi;\ \psi := U_f\psi;\ \psi := H^{\otimes n}\psi;\ \mathbf{measure}\ \psi\, r \end{aligned}$$

where H is the Hadamard transform defined by

$$H = \lambda\psi : 0, 1 \to \mathbb{C} \cdot i : 0, 1 \cdot (\psi 0 + (-1)^i \times \psi 1)/\sqrt{2}$$

and U_f is the generalized quantum oracle defined by

$$U_f = \lambda\psi : 0, 1 \to \mathbb{C} \cdot x : 0, 1 \cdot (-1)^{fx} \times \psi x$$

The complete solution is:

$$\psi := |0\rangle^{\otimes n};\ \psi := H^{\otimes n}\psi;\ \psi := U_f\psi;\ \psi := H^{\otimes n}\psi;\ \mathbf{measure}\ \psi\, r;\ b := (r' = 0)$$

Let us add to the specification a restriction on the number of calls to the oracle by introducing a time variable. Suppose the new specification is:

$$(f \textit{ is constant} \vee f \textit{ is balanced} \Longrightarrow\ b' = f \textit{ is constant}) \wedge (t' = t + 1)$$

where we charge 1 unit of time for each call to the oracle and all other operations are free. Clearly, the above quantum solution works.

Classically the specification is unimplementable. The strongest classically implementable specification is

$$(f \textit{ is constant} \vee f \textit{ is balanced} \Longrightarrow\ b' = f \textit{ is constant}) \wedge (t' \leq t + 2^{n-1} + 1)$$

3 Our contribution

Our approach to quantum programming amenable to formal analysis is very different from almost all of the existing proposals. Work of [7] is the only one which is similar to our work. The contribution of our work is twofold. Firstly, by building our theory on that in [3], we inherit the advantages it offers. The definitions of specification and program are simpler: a specification is a boolean (or probabilistic) expression and a program is a specification. The treatment of recursion is simple: there is no need for additional semantics of loops. The treatment of termination simply follows from the introduction of a time variable; if the final value of the time variable is ∞, then the program is a non-terminating one. Correctness and time and space complexity are proved in the same fashion; moreover, after proving them separately, we naturally obtain the conjunction. Secondly, the way probabilistic predicative programming is extended to quantum predicative programming is simple and intuitive. The use of Dirac-like notation makes it easy to write down specifications and develop algorithms. The treatment of computation with mixed states does not require any additional mechanisms. Quantum predicative programming fully preserves predicative programming's treatment of parallel programs and communication, which provides for a natural extension to reason about distributed quantum computation. Recent work defines the quantum system, introduces programming with the quantum system, and several well-known problems, their classical and quantum solutions, and their formal comparative time complexity analyses. Current work involves expressing quantum teleportation, dense coding, and various games involving entanglement, in a way that makes complexity analysis of these quantum algorithms simple and natural. We are interested in the possibilities of simple proofs and analysis of programs involving communication, both via quantum channels and exhibiting the LOCC (local operations, classical communication) paradigm. Future work involves formalising quantum cryptographic protocols, such as BB84 [8], in our framework and providing formal analysis of these protocols. This will naturally lead to formal analysis of distributed quantum algorithms.

References

1. Knill, E.: Conventions for quantum pseudocode. Technical Report LAUR-96-2724, Los Alamos National Laboratory (1996)
2. Hehner, E.: a Practical Theory of Programming. Second edn. Springer, New York (2004) Available free at `www.cs.utoronto.ca/~hehner/aPToP`.
3. Hehner, E.: Probabilistic predicative programming. In: Mathematics of Program Construction. (2004)
4. Ömer, B.: Quantum programming in QCL. Master's thesis, TU Vienna (2000)
5. Tafliovich, A., Hehner, E.: Quantum predicative programming. In: Mathematics of Program Construction. (2006)
6. Deutsch, D., Jozsa, R.: Rapid solution of problems by quantum computation. Proceedings of the Royal Society of London **439** (1992) 553–558
7. Sanders, J.W., Zuliani, P.: Quantum programming. In: Mathematics of Program Construction. (2000) 80–99
8. Bennet, C.H., Brassard, G.: Quantum cryptography: Public key distribution and coin tossing. In: IEEE Int. Conf. Computers, Systems and Signal Processing. (1984) 175–179