It's Not Magic: I Can Prove It

Anya Tafliovich*
University of Toronto
10 King's College Road
Toronto, Canada
anya@cs.toronto.edu
http://www.cs.toronto.edu/~anya

ABSTRACT

Our work presents a new approach to developing, analyzing, and proving correctness of programs intended for execution on a quantum computer. We provide tools to write quantum as well as classical specifications, develop quantum and classical solutions for them, and analyses various properties of quantum specifications and quantum programs, such as implementability, time and space complexity, and probabilistic error analysis uniformly, all in the same framework.

The work also develops a formal framework for specifying, implementing, and analyzing quantum pseudo-telepathy: an intriguing phenomenon which manifests itself when quantum information theory is applied to communication complexity.

Categories and Subject Descriptors

F.3.1 [**Theory of Computation**]: Logics and Meanings of Programs; D.3.1 [**Software**]: Formal Definitions and Theory; F.1.2 [**Theory of Computation**]: Models of Computation—*Quantum Computing*

General Terms

quantum computing, quantum algorithms, quantum nonlocality, formal methods of software design, formal verification, quantum predicative programming

1. INTRODUCTION

Modern physics is dominated by concepts of quantum mechanics. Today, over seventy years after its recognition by the scientific community, quantum mechanics provides the most accurate known description of nature's behaviour. Surprisingly, the idea of using the quantum mechanical nature of the world to perform computational tasks is very new, less than thirty years old. 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, adiabatic computation, measurement-based quantum computation, theoretical quantum optics, and the very new quantum game theory. Experimental quantum information and communication has also been a fruitful field. Experimental quantum optics, ion traps, solid state implementations and nuclear magnetic resonance all add to the experimental successes of quantum computation.

The subject of this work is quantum programming — developing programs intended for execution on a quantum computer. We assume a model of a quantum computer proposed by Knill [21]: 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.

We look at programming in the context of formal methods of program development, or programming methodology. This is the field of computer science concerned with applications of mathematics and logic to software engineering tasks. In particular, the formal methods provide tools to formally express software specifications, prove correctness of implementations, and reason about various properties of specifications (e.g. implementability) and implementations (e.g. time and space complexity). Today formal methods are successfully employed in all stages of software development, such as requirements elicitation and analysis, software design, and software implementation.

In this work the theory of quantum programming is based on probabilistic predicative programming, a recent generalization of the well-established predicative programming [17, 18], 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 pro-

*Research in part supported by the Natural Sciences and Engineering Research Council of Canada

grams that implement these specifications, and reason about their comparative time and space complexity all in the same framework.

The work also develops a formal framework for specifying, implementing, and analyzing quantum pseudo-telepathy: an intriguing phenomenon which manifests itself when quantum information theory is applied to communication complexity. To demonstrate this phenomenon researchers in the field of quantum communication complexity devised a number of quantum non-locality games. The setting of these games is as follows: the players are separated so that no communication between them is possible and are given a certain computational task. When the players have access to a quantum resource called entanglement, they can accomplish the task: something that is impossible in a classical setting. To an observer who is unfamiliar with the laws of quantum mechanics it seems that the players employ some sort of telepathy; that is, they somehow exchange information without sharing a communication channel.

Quantum pseudo-telepathy, and quantum non-locality in general, are perhaps the most non-classical and the least understood aspects of quantum information processing. Every effort is made to gain information about the power of these phenomena. Quantum non-locality games in particular have been extensively used to prove separations between quantum and classical communication complexity. The need for a good framework for formal analysis of quantum non-locality is evident.

1.1 Related work

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.

Existing formal approaches to quantum programming include the language qGCL [24, 32, 33], process algebraic approaches developed in [4, 22, 19], tools developed in the field of category theory by [1, 2, 3, 10, 25], functional languages of [6, 7, 5, 29, 30], as well as work of [14, 15], [11], and [16]. A detailed discussion of the work related to quantum predicative programming is presented in [27]. Some researchers address the subject of formalizing quantum non-locality more directly than others (e.g. [32]). To the best of our knowledge, formal approaches to reasoning about quantum pseudo-telepathy games have not been considered.

1.2 Our contribution

Our approach to quantum programming amenable to formal analysis is very different from almost all of those mentioned above. Work of [24, 32, 33] is the only one which is similar to our work. The contribution of our research is threefold. Firstly, by building our theory on that in [18], 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 quantum communication protocols, such as BB84 ([8]), distributed quantum algorithms, such as distributed Shor's algorithm ([31]), as well as their time, space, and entanglement complexity. Finally, we apply our theory to specifying, implementing, and analyzing quantum pseudo-telepathy games.

2. IT'S NOT MAGIC: I CAN PROVE IT

The programming theory of our choice is quantum predicative programming. For a course in predicative programming the reader is referred to [17]. An introduction to probabilistic predicative programming can be found in [18]. Quantum predicative programming is developed in [27, 26]. Formal reasoning about non-locality as well as analysis of quantum pseudo-telepathy games is presented in [28].

In this section we apply our theory to formally prove correctness of quantum algorithms and reason about their complexity. We also demonstrate formal reasoning about quantum non-locality by implementing two quantum non-locality games.

2.1 Deutsch algorithm

Deutsch's algorithm [12] is one of the most famous quantum algorithms. The task is: given an oracle function $f : 0, 1 \to 0, 1$, compute $f0 \oplus f1$. The trick is: we are only allowed to call f once. Magic? Not at all: a quantum phenomenon. In our theory, we can prove:

$$\begin{aligned} & x' = f0 \oplus f1 \wedge t' = t + 1 \\ == \; & \psi := |0\rangle;\ \psi := H\psi;\ t := t + 1;\ \psi := U_f\psi;\ \psi := H\psi; \\ & \textbf{measure}\ \psi\ x \end{aligned}$$

The first line is the specification. It means "the computation should assign to x the value $f0 \oplus f1$ and it should take one time step to complete", where we charge 1 unit of time for each call to the oracle and all other operations are free. The second line is the quantum program. We initialize the quantum system to a zero state, apply appropriate unitary transformations, and measure the system in the appropriate basis. The fact that we can prove the equality of the two expressions means that the quantum program is correct and that it satisfies the restriction on the number of calls to the function f.

2.2 Deutsch-Jozsa algorithm

Deutsch-Jozsa's problem ([13]), an extension of Deutsch's Problem, is an example of the broad class of quantum algorithms that are based on the quantum Fourier transform ([20]). The task is: given a function $f : 0, ..2^n \to 0, 1$, such

that f is either constant or balanced, determine which case it is. As before, we are only allowed to query the oracle once. Formally, the 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. This looks like an impossible task. And it is impossible – in a classical setting. Classically the specification is unimplementable. The strongest classically implementable specification is

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

That is, we need at least $2^{n-1} + 1$ calls to f to determine whether f is constant or balanced.

The quantum solution is a direct generalization of Deutsch's algorithm. We can prove that the following implements the specification:

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

2.3 Pseudo-telepathy games

We formalize pseudo-telepathy games with n players as follows. For each player i, $0 \le i < n$, we have a domain D_i from which the inputs to player i are provided and a range R_i of player's possible output results. In addition we may have a promise P: a condition on the inputs to the players. If no promise is given, we set P to 1. The winning condition W can involve inputs as well as outputs for each player. The strategy S is a program, i.e. an implemented specification. The strategy S is winning if $P \wedge S \Rightarrow W$.

2.3.1 Deutsch-Jozsa game

The Deutsch-Jozsa pseudo telepathy game [9] is based on the Deutsch-Jozsa algorithm [13]. The setting of the game is as follows. Alice and Bob are separated several light years apart and are each presented with a 2^k-bit string. They are promised that either the strings are identical or they differ by exactly half of the bits. To win the game the players must each output a k-bit string, and these strings should be identical if and only if their input strings were identical.

We formalize the game as follows. We partition the space into the world of Alice (variables subscripted A) and the world of Bob (variables subscripted B). Then $D_A = D_B = \{0,1\}^{2^k}$ are the domain of inputs to Alice and Bob, $R_A = R_B = \{0,1\}^k$ are the range of outputs of Alice and Bob, $P = P_0 \vee P_1$, where P_0 states that the inputs are identical, $P_0 == \sum i : 0,..2^k \cdot ((x_A)_i = (x_B)_i) == 2^k$, and P_1 states that the inputs differ by half of the bits, $P_1 == \sum i : 0,..2^k \cdot ((x_A)_i = (x_B)_i) == 2^{k-1}$, is the promise on the inputs, and $W == (x_A = x_B) = (y'_A = y'_B)$ is the winning condition.

We demonstrate the quantum solution by implementing a specification S, so that $P \wedge S \Rightarrow W$:

$$S == \psi := \sum z : 0,..2^k \cdot |zz\rangle/\sqrt{2^k};\ (S_A\ ||_\psi\ S_B), \text{ where}$$

$$S_i == \psi_i := U_i^{\otimes k}\psi_i;\ \psi_i := H^{\otimes k}\psi_i; \textbf{measure}\ \psi_i\ y_i,$$

for unitary $U_i|z\rangle = (-1)^{(x_i)_z}|z\rangle$, where $i : A, B$. That is, Alice and Bob share an entangled state ψ and execute their programs *in parallel*.

To prove the solution correct we show:

$$S ==$$
$$\left|\sum u, v, z \cdot (-1)^{(x_A)_z + (x_B)_z + u\cdot z + v\cdot z}/\sqrt{2^k}^3 \times |uv\rangle\ (y_A y_B)'\right|^2$$

To demonstrate that S is winning, namely that $P \wedge S \Rightarrow W$, it is sufficient to show

$$P_0 \wedge S \Rightarrow (y'_A = y'_B) \qquad \text{and} \qquad P_1 \wedge S \Rightarrow (y'_A \neq y'_B)$$

2.3.2 Mermin's game

In a Mermin's game [23] there are three players. Each player i receives a bit x_i as input and outputs a bit y_i. The promise is that the sum of the inputs is even. The players win the game if the parity of the sum of the outputs is equal to the parity of half the sum of the inputs.

We formalize the game as follows: $D_i = R_i = \{0,1\}$, for $i : 0,1,2$. The promise is $P == (x_0 + x_1 + x_2) \mod 2 = 0$. The winning condition is $W == (y'_0 + y'_1 + y'_2) = (x_0 + x_1 + x_2)/2 \mod 2$.

We implement the following quantum strategy. The players share an entangled state $\psi = |000\rangle/\sqrt{1} + |111\rangle/\sqrt{2}$. After receiving the input, each player applies the operation U defined by $U|0\rangle = |0\rangle$ and $U|1\rangle = \sqrt{-1} \times |1\rangle$ to her qubit if the input is 1. The player then applies a Hadamard transform. The qubit is measured in the computational basis and the result of the measurement is the output.

The program is:

$$S == \psi := |000\rangle/\sqrt{2} + |111\rangle/\sqrt{2};\ S_0\ ||_\psi\ S_1\ ||_\psi\ S_2$$
$$S_i == \textbf{if}\ x_i = 1\ \textbf{then}\ \psi_i := U\psi_i\ \textbf{else}\ ok;\ \psi_i := H\psi_i;$$
$$\textbf{measure}\ \psi_i\ y_i$$

where $i : 0,1,2$. To prove that the solution is correct and that the strategy S is winning, we demonstrate:

$$S == \textbf{measure}\ \underset{y_0 y_1 y_2}{H^{\otimes 3}}(|000\rangle + (\sqrt{-1})^{x_0+x_1+x_2} \times |111\rangle)/\sqrt{2}$$
$$P \wedge S == y'_0 + y'_1 + y'_2 = (x_0 + x_1 + x_2)/2 \mod 2$$

3. CONCLUSION AND FUTURE WORK

We have presented a new approach to developing, analyzing, and proving correctness of quantum programs. Since we adopt Hehner's theory as the basis for our work, we inherit its advantageous features, such as simplicity, generality, and elegance. Our work extends probabilistic predicative programming in the same fashion that quantum computation extends probabilistic computation. We have provided tools to write quantum as well as classical specifications, develop quantum and classical solutions for them, and analyze various properties of quantum specifications and quantum programs, such as implementability, time and space complexity, and probabilistic error analysis uniformly, all in the

same framework. We have also presented a formal framework for specifying, implementing, and analyzing quantum pseudo-telepathy games.

Current research focuses on formal reasoning about complexity of distributed quantum algorithms (e.g. [31]). Research in the immediate future will focus on simple proofs and analysis of programs involving communication, both via quantum channels and exhibiting the LOCC (local operations, classical communication) paradigm. Future work involves formalizing quantum cryptographic protocols, such as BB84 [8], in our framework and providing formal analysis of these protocols.

4. REFERENCES

[1] S. Abramsky. High-level methods for quantum computation and information. In *Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science*, 2004.

[2] S. Abramsky and B. Coecke. A categorical semantics of quantum protocols. In *LICS 2004*, 2004.

[3] S. Abramsky and R. Duncan. A categorical quantum logic. In *QPL 2004*, pages 3–20, 2004.

[4] P. Adao and P. Mateus. A process algebra for reasoning about quantum security. In *QPL 2005*, 2005.

[5] T. Altenkirch and J. Grattage. A functional quantum programming language. In *Proceedings of the 20th Annual IEEE Symposium on Logic in Computer Science*, 2005.

[6] P. Arrighi and G. Dowek. Operational semantics for formal tensorial calculus. In *QPL 2004*, pages 21–38, 2004.

[7] P. Arrighi and G. Dowek. Linear-algebraic lambda-calculus. In *QPL 2005*, 2005.

[8] C. H. Bennet and G. Brassard. Quantum cryptography: Public key distribution and coin tossing. In *IEEE Int. Conf. Computers, Systems and Signal Processing*, pages 175–179, 1984.

[9] G. Brassard, R. Cleve, and A. Tapp. Cost of exactly simulating quantum entanglement with classical communication. *Physical Review Letters*, 83(9):1874–1878, 1999.

[10] B. Coecke. The logic of entanglement. 2004. quant-ph/0402014.

[11] V. Danos, E. D'Hondt, E. Kashefi, and P. Panangaden. Distributed measurement-based quantum computation. In *QPL 2005*, 2005.

[12] D. Deutsch. Quantum theory, the Church-Turing principle and the universal quantum computer. In *Proceedings of the Royal Society of London*, pages 97–117, 1985.

[13] D. Deutsch and R. Jozsa. Rapid solution of problems by quantum computation. *Proceedings of the Royal Society of London*, 439:553–558, 1992.

[14] E. D'Hondt and P. Panangaden. Quantum weakest precondition. In *QPL 2004*, pages 75–90, 2004.

[15] E. D'Hondt and P. Panangaden. Reasoning about quantum knowledge. 2005. quant-ph/0507176.

[16] S. J. Gay and R. Nagarajan. Communicating quantum processes. In *Proceedings of the 32nd ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages*, 2005.

[17] E. Hehner. *a Practical Theory of Programming*. Springer, New York, first edition, 1993. Current edn. (2007) Available free at `www.cs.utoronto.ca/~hehner/aPToP`.

[18] E. Hehner. Probabilistic predicative programming. In *Mathematics of Program Construction*, 2004.

[19] P. Jorrand and M. Lalire. Toward a quantum process algebra. In *Proceedings of the 1st ACM Conference on Computing Frontiers*, 2004.

[20] R. Jozsa. Quantum algorithms and the Fourier transform. *Proceedings of the Royal Society of London*, pages 323–337, 1998.

[21] E. Knill. Conventions for quantum pseudocode. Technical Report LAUR-96-2724, Los Alamos National Laboratory, 1996.

[22] M. Lalire and P. Jorrand. A process algebraic approach to concurrent and distributed quantum computation: operational semantics. In *QPL 2004*, pages 109–126, 2004.

[23] N. Mermin. Quantum mysteries revisited. *Americal Journal of Physics*, 58(8):731–734, 1990.

[24] J. W. Sanders and P. Zuliani. Quantum programming. In *Mathematics of Program Construction*, pages 80–99, 2000.

[25] P. Selinger. Towards a quantum programming language. *Mathematical Structures in Computer Science*, 2004.

[26] A. Tafliovich. Quantum programming. Master's thesis, University of Toronto, 2004.

[27] A. Tafliovich and E. Hehner. Quantum predicative programming. In *Proceedings of the 8th International Conference on Mathematics of Program Construction*, pages 433 – 455, 2006.

[28] A. Tafliovich and E. Hehner. Programming telepathy: Implementing quantum non-locality games. In *Proceedings of the 10th Brazilian Symposium on Formal Methods*, Ouro Preto, Brazil, 2007. To appear.

[29] B. Valiron. Quantum typing. In *QPL 2004*, pages 163–178, 2004.

[30] A. van Tonder. A lambda calculus for quantum computation. *SIAM Journal on Computing*, 33(5):1109–1135, 2004.

[31] A. Yimsiriwattana and S. J. L. Jr. Distributed quantum computing: A distributed Shor algorithm. 2004. `quant-ph/0403146`.

[32] P. Zuliani. Non-deterministic quantum programming. In *QPL 2004*, pages 179–195, 2004.

[33] P. Zuliani. Quantum programming with mixed states. In *QPL 2005*, 2005.