An imperative programming language characterizing FBQP

Emmanuel Hainry, Romain Péchoux, Mário Silva
Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France

Introduction

Quantum programming languages are a useful tool, not only for writing complex algorithms, but also for abstracting and reasoning about their properties and to learn more about what can be done efficiently by quantum computers [1]. However, merely being able to describe a quantum program does not inform us on its complexity, meaning that it may not ultimately be efficiently run on a quantum computer. Consequently, there is a need for static analysis tools and techniques for reasoning about and certifying the complexity of these quantum algorithms.

We introduce an imperative programming language called QPT (Quantum Poly-Time) with recursion rules that captures the bounded error class FBQP (Functions-Bounded-error Quantum Polytime), the class of functions computable in polynomial time by a quantum Turing machine with at most 1/3 probability of error, commonly accepted as the class of feasible problems for quantum computers. Our result takes advantage of a function algebra proposed by Yamakami [2] characterizing FBQP.

Rule for termination (R1)

Let \( \text{proc}_i \sim \text{proc}_j \) mean that \( \text{proc}_i \) and \( \text{proc}_j \) are mutually recursive procedures. The following condition ensures that a program with procedure declarations \( D \) will always terminate:

\[
\forall \text{Proc} \ proc(p) \in D,
\forall \text{call} \ proc(\sigma) \in S_n,
\text{proc}_i \sim \text{proc}_j \Rightarrow \sigma \text{ is a proper subset of } p,
\]

i.e., any call to a mutually recursive function must strictly decrease the amount of qubits available.

Rule for poly-time (R2)

We prevent an exponential number of procedure calls by limiting the number of recursive calls in each branch of a Case. Let \( \text{RCall}(\cdot) \) represent the maximum number of recursive calls for any branch, then

\[
\forall \text{Proc} \ proc(p) \in D, \text{RCall}(\text{proc}_i) \leq 1,
\]

i.e., multiple recursive calls can be done only on different branches of the Case structure.

QPT \( \sim \) FBQP

Soundness: For any program \( P \) in QPT following rules R1 and R2, there exists a poly-sized uniform family of circuits \( (C_n)_{n \in \mathbb{N}} \) for each input size \( n \) that simulates \( P \).

Completeness: For any function \( f \) in FBQP with size-bounding polynomial \( p \), and any constant \( \varepsilon \in [0, 1/2] \), there exists a program \( P \) in QPT following rules R1 and R2 such that, running \( P \) on polynomially extended input state \( \rho_n \), for \( x \in \{0, 1\}^n \), a measurement of the first \( |f(x)| \) of the output qubits will result in \( f(x) \) with probability at least \( 1 - \varepsilon \).

Building poly-sized circuits

Dealing with procedures that include recursive branching, a straightforward approach to building the circuit will readily require an exponential number of gates, e.g. in the following procedure:

\[
\text{Proc } f(p) \{ \\
\quad \text{if } |p| > 1 : \\
\quad \quad \text{Case } C(|p|) = 0 \text{ then } \\
\quad \quad \quad \text{call } f(\text{remove}(p, 1)) \\
\quad \quad \text{else Case } C(|p|) = 1 \text{ then } \\
\quad \quad \quad \text{call } f(\text{remove}(p, \{1, 2\})) \\
\quad \quad \text{else if } p[1] = q \text{ then } \\
\quad \quad \quad \text{call } f(q) \\
\}
\]

As a proof of Soundness, we provide an algorithm to build all programs following rule R2 with a polynomially large set of gates and wires, such as in the example of Figure 1 for the above function applied on an input of size \( n = 5 \).

Example: QFT

The quantum Fourier transform (QFT) is in FBQP, appearing as a subroutine of Shor’s algorithm. It contains two recursive patterns, following rules R1 and R2, highlighted in the circuit of Figure 2.

\[
\text{Proc } \text{QFT}(p) \{ \\
\quad p[0] = H; \\
\quad \text{call } \text{Chain}(2, p); \\
\quad \text{call } \text{QFT}(\text{remove}(p, 1)); \\
\}
\]

The circuit can be implemented with size \( O(n^2) \) gates, for an input with \( n \) qubits.

References
