On the Hardness of Analyzing Quantum Programs Quantitatively

Martin Avanzini
Centre Inria d'Université Côte d'Azur, France
martin.avanzini@inria.fr

Georg Moser
Universität Innsbruck, Austria
georg.moser@uibk.ac.at

Romain Péchoux
CNRS-Inria-Université de Lorraine, LORIA, Nancy, France
romain.pechoux@loria.fr

Simon Perdrix
CNRS-Inria-Université de Lorraine, LORIA, Nancy, France
simon.perdrix@loria.fr

We study quantitative properties of quantum programs such as (positive) almost-sure termination, expected runtime or expected cost, that is, for example, the expected number of applications of a given quantum gate, etc. After studying the completeness of these problems in the arithmetical hierarchy over the Clifford+T fragment of quantum mechanics, we express these problems using a variation of a quantum pre-expectation transformer, a weakest pre-condition based technique that allows to symbolically compute these quantitative properties. Under a smooth restriction—a restriction to polynomials of bounded degree over a real closed field—we show that the quantitative problem, which consists in finding an upper-bound to the pre-expectation, can be decided in time double-exponential in the size of a program. Finally, we sketch how the latter can be transformed into an efficient synthesis method. This paper is a summary of a work which has been published at the 33rd European Symposium on Programming, ESOP 2024 and is available at https://arxiv.org/abs/2312.13657.

1 Introduction

Motivations. In this work, we study formal methods that can be used to obtain quantitative properties about the computations of a quantum program, such as the number of qubits used and the number of unitary operators used, thus enabling the corresponding compiled quantum circuit to be optimized (for example, by minimizing the use of gates that are hard to make fault-tolerant, or by reducing the number of qubits) or to avoid undesirable behavior such as non-termination. Another quantitative property of interest may also be the question whether or not a program *terminates almost-surely*, that is, whether its probability of non-termination is zero or not. Similarly, we could aim to capture the *expected values* of (classical) program variables upon program termination. The latter can also be employed to reason about the *expected runtime* or the *expected cost* of quantum programs, if we suitably instrument the code with counter variables.

To illustrate this, the program of Figure 1 implements a Repeat-Until-Success algorithm that can be used to simulate quantum unitary operators on input qubit $\mathtt{q}_1$ by using repeated measurements. The quantum step-circuit on the right part corresponds to one iteration of the loop. Variable `i` in the program just acts as a counter for `T`-gates. Hence an analysis on the expected value of variable `i` can be used to infer an upper-bound on the expected *T-count*, i.e., the expected number of times a `T`-gate is used by the program. As `T`-gates are known to be costly to implement fault-tolerantly [BK05, GKMR14], it illustrates that the study of quantitative properties is paramount.

In [AMP+22, LZBY22], new methodologies named *quantum expectation transformers* based on *predicate transformers* [Dij76, Koz85] and *expectation transformers* [MM05, GKM14] have been put

Submitted to:
QPL 2024

```
RUS ≜ i = 0;
      x = true;
      while x do {
        q2 = |0⟩ ;
        q2 *= H;
        q2 *= T;
        i = i + 1
        q2, q1 *= CNOT;
        q2 *= H;
        q2, q1 *= CNOT;
        q2 *= T;
        i = i + 1
        q2 *= H;
        x = meas q2
      }
```

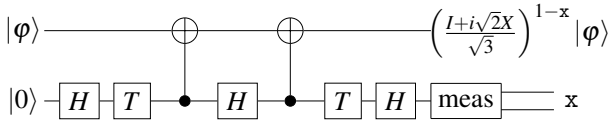

Figure 1: Repeat-until-success program RUS and step-circuit.

forward to naturally express and study the quantitative properties of quantum programs. However, no attempt was made to automate the corresponding techniques or delineate how complicated such an *automation* could be. Automation of these formal verification techniques in the context of quantum programs is a particularly difficult problem. Indeed, the consideration of Hilbert spaces as a mathematical framework for describing principles and laws of quantum mechanics makes it seemingly impossible to reason fully automatically about quantitative properties of quantum program: they involve computational objects of exponential dimensions (in the number of qubits) with scalars ranging over an uncountable domain (i.e., complex numbers $\mathbb{C}$). This problem is directly linked to the fact that the set $\mathbb{C}$ includes non-computable numbers [Wei12] and that testing the inequality $<$ or the equality $=$ of two real numbers is not decidable, even if one restricts their study to computable real numbers. Consequently, the particular nature of quantum programs and of their semantic domain, Hilbert spaces, makes it impossible to directly apply the results obtained in the classical and probabilistic setting [SS11, KK15].

Contributions. In this talk, we study the hardness of the quantitative properties of mixed classical-quantum programs and provide a first step towards their (full) automation using quantum expectation transformers. To this end, we restrict the considered quantum gates to the *Clifford+T fragment*, which is known to be the simplest approximately universal fragment of quantum mechanics [AG04]. Clifford+T makes it possible to only consider quantum states with algebraic amplitudes, thus restricting the study to a countable domain. It implies that our results can accommodate quantum gates employed in actual hardware, recently employed to claim *quantum advantage*, cf [AAe19]. Moreover, the obtained results are very general as it can be extended to any set of gates with algebraic coefficients.

As motivated, our first contribution is about the general hardness of deciding quantitative properties for mixed classical-quantum programs. For a given input state, we study properties such as:

- *(positive) almost-sure termination*, (P)AST for short, which consists in checking the a program terminates with probability 1 (and finite expected runtime);
- *testing problems*, $\text{TEST}_{\mathscr{R}}$, which consist in comparing a quantum expectation (for example, the mean value of a variable) with a given value (an algebraic and positive real number) with respect

	Standard		Universal	
	Problem	Class	Problem	Class
Testing	$\textsc{Test}_{>}$	Σ^0_1	$\textsc{UTest}_{>}$	Π^0_2
	$\textsc{Test}_{\geq}$	Π^0_2	$\textsc{UTest}_{\geq}$	Π^0_2
	$\textsc{Test}_{=}$	Π^0_2	$\textsc{UTest}_{=}$	Π^0_2
	$\textsc{Test}_{\leq}$	Π^0_1	$\textsc{UTest}_{\leq}$	Π^0_1
	$\textsc{Test}_{<}$	Σ^0_2	$\textsc{UTest}_{<}$	Π^0_3
Finiteness	$\textsc{Test}_{\neq\infty}$	Σ^0_2	$\textsc{UTest}_{\neq\infty}$	Π^0_3
Termination	AST	Π^0_2	UAST	Π^0_2
	PAST	Σ^0_2	UPAST	Π^0_3

Table 1: Completeness results for quantitative problems in the arithmetical hierarchy.

$$
\begin{aligned}
\mathtt{qet}[\mathtt{skip}]\{f\} &\triangleq f\\
\mathtt{qet}[\mathtt{x}=\mathtt{e}]\{f\} &\triangleq f[\mathtt{x}:=\mathtt{e}]\\
\mathtt{qet}[\mathtt{stm}_1\,;\,\mathtt{stm}_2]\{f\} &\triangleq \mathtt{qet}[\mathtt{stm}_1]\{\mathtt{qet}[\mathtt{stm}_2]\{f\}\}\\
\mathtt{qet}[\mathtt{if\ b\ then\ stm}_1\ \mathtt{else\ stm}_2]\{f\} &\triangleq \mathtt{qet}[\mathtt{stm}_1]\{f\} +_{[\![\mathtt{b}]\!]} \mathtt{qet}[\mathtt{stm}_2]\{f\}\\
\mathtt{qet}[\mathtt{while\ b\ do\ stm}]\{f\} &\triangleq \mathsf{lfp}\left(\lambda F.\mathtt{qet}[\mathtt{stm}]\{F\} +_{[\![\mathtt{b}]\!]} f\right)\\
\mathtt{qet}[\overline{\mathtt{q}} *= \mathtt{U}]\{f\} &\triangleq f[\Phi_{U_{\overline{\mathtt{q}}}}]\\
\mathtt{qet}[\mathtt{x}=\mathtt{meas\ q}_i]\{f\} &\triangleq f[\mathtt{x}:=0; m_{0,i}] +_{p_{0,i}} f[\mathtt{x}:=1; m_{1,i}].
\end{aligned}
$$

Figure 2: Quantum expectation transformer $\mathtt{qet}[\cdot]\{\cdot\}$

to the relation $\mathscr{R}$;

- the *finiteness problem*, $\textsc{Test}_{\neq\infty}$, which consists in checking that a quantum expectation is finite.

For each of those problems, we also study the related *universal problem*, which consists in checking the corresponding property for every input. We establish a precise mapping of the inherent complexity of each problem in the arithmetical hierarchy [Odi92] that is summarized in Table 1. E.g., AST is Π^0_2-complete while PAST is Σ^0_2-complete.

Our second contribution aims to overcome the aforementioned undecidability results. For that, we study approximations of *quantum expectation transformers* introduced in [AMP+22]. More precisely, we focus on *inferring* bounding functions (in general depending on the input) on the expected values of classical program variables upon termination.

The *quantum expectation transformer* consists in a program semantics mapping expectations to expectations in a continuation passing style

$$\mathtt{qet}[\cdot]\{\cdot\} : \mathtt{Stmt} \to (\mathtt{St} \to \mathbb{R}^{+\infty}) \to (\mathtt{St} \to \mathbb{R}^{+\infty}),$$

where $\mathtt{Stmt}$ is the set of program statements, $\mathtt{St}$ is the set of (classical and quantum) memory states, and

continuity	$\texttt{qet}[\texttt{stm}]\{\sup_i f_i\} = \sup_i \texttt{qet}[\texttt{stm}]\{f_i\}$
monotonicity	$f \leq g \Rightarrow \texttt{qet}[\texttt{stm}]\{f\} \leq \texttt{qet}[\texttt{stm}]\{g\}$
upper invariance	$(\neg \texttt{b} \Rightarrow f \leq g) \wedge (\texttt{b} \Rightarrow \texttt{qet}[\texttt{stm}]\{g\} \leq g) \Rightarrow \texttt{qet}[\texttt{while b do stm}]\{f\} \leq g$

Figure 3: Universal laws derivable for the quantum expectation transformer.

$\mathbb{R}^{+\infty}$ is the set of non-negative real numbers, including ∞. Quantum expectation transformers are defined inductively on statements in Figure 2.

Quantum expectation transformers provide a denotational semantics to quantum programs (as described in [AMP+22]) and, hence, are not computable in general. This drawback is a consequence of the least fixpoint computation for while loops in Figure 2 and can be avoided by considering upper-bound approximations as described by the *upper invariance rule* of Figure 3. Using this rule, the decision problem has thus been altered to an inference problem. Further, we restrict the set of potential bounding functions. As a suitable class of functions, we consider polynomials over the real-closed field of the algebraic numbers. The restriction to algebraic numbers guarantees that comparison operations between real numbers remain decidable. On the other hand, for any real closed field, quantifier elimination for formulas over polynomials is decidable, that is, there exists a double-exponential algorithm computing a quantifier-free formula equivalent to the original formula [HRS90]. This recasting of the problem and restriction of the solution space suffices to render the problem decidable. The inference algorithm established remains double-exponential, thus of similar complexity as the underlying quantifier elimination procedure.

Finally, our last contribution studies effective automation of the inference of upper bounds on the expected values of program variables. To improve upon the double-exponential complexity, we further restrict the class of polynomials considered, that is, to degree-2 polynomials and sketch how techniques from optimization theory can be employed. Several simple quantum algorithms such as program `RUS` can be analyzed using this approach. This further reduction in expressivity allows the encoding of the problem in SMT and thus paves the way towards (full) automation.

Acknowledgments.

This work is supported by the HORIZON 2020 project NEASQC and by the Inria associate team TC(Pro)^3. It is also supported by the Plan France 2030 through the PEPR integrated project EPiQ ANR-22-PETQ-0007 and the HQI initiative ANR-22-PNCQ-0002, the ANR PRC project PPS ANR-19-CE48-0014, as well as FWF Project AUTOSARD P 36623.

References

[AAe19] F. Arute, Kunal Arya & et al. (2019): *Quantum supremacy using a programmable superconducting processor*. *Nature* 574, p. 505–510, https://doi.org/10.1038/s41586-019-1666-5.

[AG04] Scott Aaronson & Daniel Gottesman (2004): *Improved simulation of stabilizer circuits*. *Physical Review A* 70(5), p. 052328.

[AMP+22] Martin Avanzini, Georg Moser, Romain Péchoux, Simon Perdrix & Vladimir Zamdzhiev (2022): *Quantum Expectation Transformers for Cost Analysis*. In: *LICS '22: 37th Annual ACM/IEEE*

Symposium on Logic in Computer Science, Haifa, Israel, August 2 - 5, 2022, pp. 10:1–10:13, https://doi.org/10.1145/3531130.3533332.

[BK05] Sergey Bravyi & Alexei Kitaev (2005): *Universal quantum computation with ideal Clifford gates and noisy ancillas*. *Physical Review A* 71(2), p. 022316, https://doi.org/10.1103/PhysRevA.71.022316.

[Dij76] Edsger W. Dijkstra (1976): *A discipline of programming*. Prentice-Hall Englewood Cliffs.

[GKM14] Friedrich Gretz, Joost-Pieter Katoen & Annabelle McIver (2014): *Operational versus weakest pre-expectation semantics for the probabilistic guarded command language*. *Performance Evaluation* 73, pp. 110–132, https://doi.org/10.1016/j.peva.2013.11.004.

[GKMR14] David Gosset, Vadym Kliuchnikov, Michele Mosca & Vincent Russo (2014): *An algorithm for the T-count*. *Quantum Information & Computation* 14(15-16), pp. 1261–1276, https://doi.org/10.26421/QIC14.15-16-1.

[HRS90] Joos Heintz, Marie-Françoise Roy & Pablo Solernó (1990): *Sur la complexité du principe de Tarski-Seidenberg*. *Bulletin de la Société mathématique de France* 118(1), pp. 101–126.

[KK15] B. L. Kaminski & J.-P. Katoen (2015): *On the Hardness of Almost-Sure Termination*. In: *MFCS 2015, Part I*, LNCS, Springer, pp. 307–318, https://doi.org/10.1007/978-3-662-48057-1_24.

[Koz85] Dexter Kozen (1985): *A probabilistic PDL*. *Journal of Computer and System Sciences* 30(2), pp. 162–178.

[LZBY22] Junyi Liu, Li Zhou, Gilles Barthe & Mingsheng Ying (2022): *Quantum Weakest Preconditions for Reasoning about Expected Runtimes of Quantum Programs*. arXiv:1911.12557.

[MM05] Annabelle McIver & Carroll Morgan (2005): *Abstraction, refinement and proof for probabilistic systems*. Springer Science & Business Media.

[Odi92] Piergiorgio Odifreddi (1992): *Classical recursion theory: The theory of functions and sets of natural numbers*. Elsevier.

[SS11] Andreas Schnabl & Jakob Grue Simonsen (2011): *The Exact Hardness of Deciding Derivational and Runtime Complexity*. In: *Proc. 20th CSL, LIPIcs* 12, pp. 481–495, https://doi.org/10.4230/LIPIcs.CSL.2011.481.

[Wei12] Klaus Weihrauch (2012): *Computable analysis: an introduction*. Springer Science & Business Media.