On the Hardness of Analyzing Quantum Programs Quantitatively

Martin Avanzini[1], Georg Moser[2], Romain Péchoux[3], and Simon Perdrix[3]

[1] Centre Inria d'Université Côte d'Azur, France
[2] Universität Innsbruck, Austria
[3] CNRS-Inria-Université de Lorraine, LORIA, Nancy, France
martin.avanzini@inria.fr, georg.moser@uibk.ac.at,
{romain.pechoux,simon.perdrix}@loria.fr

Abstract. In this paper, we study quantitative properties of quantum programs. Properties of interest include (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, thus providing, despite its great complexity, one of the first decidable results on the analysis and verification of quantum programs. Finally, we sketch how the latter can be transformed into an efficient synthesis method.

1 Introduction

Motivations. Quantum computation is a promising and emerging computational paradigm which can efficiently solve problems considered to be intractable on classical computers [41,20]. However, the unintuitive nature of quantum mechanics poses challenging questions for the design and analysis of corresponding quantum programming. Indeed, the quantum program dynamics are considerably more complicated compared to the behavior of classical or probabilistic programs. Therefore, formal reasoning requires the development of novel methods and tools, a development that has already started and recently gathered momentum in various areas, like *design automation* [43,22], *programming languages* [39,2,31,23,15], *verification* [36,11], etc.

Among these formal methods, those that allow us to obtain quantitative properties on quantum programs are particularly interesting. They can be used to obtain relevant information about the computations of a quantum program,

```
RUS ≜ i^N = 0;
      x^B = tt;
      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
      }
.
```

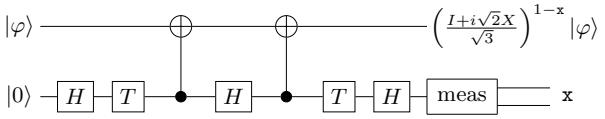

Fig. 1. Repeat-until-success program RUS and step-circuit.

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 in the fully compiled quantum circuit. Such an approach offers the advantage to allow the programmer to implement quantum programs using fewer `T`-gates, which are costly to implement fault-tolerantly [10,16], and it therefore provides a simple quantum program to illustrate that the study of quantitative properties is paramount.

In [6,30], new methodologies named *quantum expectation transformers* based on *predicate transformers* [13,28] and *expectation transformers* [32,17] have been put 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. Automa-

tion 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 [42] and that testing the inequality $\leq$ 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 [37,24].

Contributions. In this paper, 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 [1]. 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 [3]. 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; *testing problems*, $\text{TEST}_{\mathcal{R}}$, 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) wrt the relation $\mathcal{R}$; and the *finiteness problem*, $\text{TEST}_{\neq\infty}$, 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 (Theorem 1) of the inherent complexity of each problem in the arithmetical hierarchy [34] that is summarized in Table 1 (provided in Section 3). E.g., AST is Π_2^0-complete while PAST is Σ_2^0-complete.

Our second contribution aims to overcome the aforementioned undecidability results. For that, we study approximations. 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 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 num-

bers 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 [21]. This recasting of the problem and restriction of the solution space suffices to render the problem decidable. The inference algorithm established remains double-exponential (Theorem 4), thus of similar complexity as the underlying quantifier elimination procedure.

Finally, our last contribution (Section 5) 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 (Example 6). This further reduction in expressivity allows the encoding of the problem in SMT and thus paves the way towards (full) automation.

Related Work. Predicate transformers [13,28]—on which our work is based—were introduced as a method for reasoning about the semantics of imperative programs. They have been adapted to the probabilistic setting, leading to the notion of expectation transformer [32,17], which has been used to reason about expected values [26,8], runtimes [27,33], and costs [7,4,33], and to the quantum paradigm, leading to the notion of quantum pre-expectation transformer [35,30,6].

The problem of studying the difficulty of analyzing quantitative program properties has been deeply studied in the classical setting. To mention a few, [14] and [37] study termination properties and runtime/derivational properties of first-order programs, respectively. Further, in [24] completeness results for various quantitative properties of (pure) probabilistic programs have been established. The inference problem of expectation transformers, i.e., establishing an implementation that automates the search for pre-expectations, has been studied extensively. Examples of successful implementation are presented in [33,7,8]. Up to now, however, no practical, feasible studies have been carried out on quantum languages. Among the techniques using quantum expectation transformers, we believe [6] to be the most amenable to automation. Indeed, by lifting *upper invariants* of [27] to the quantum setting, it enables approximate reasoning and eliminate the need to reason about fixpoints or limits, stemming from the semantics of loops.

2 Quantum Programming Language

In this section, we introduce the syntax and operational semantics of the considered mixed-quantum imperative programming language.

Syntax. We make use of three basic datatypes $\mathcal{B}$, $\mathcal{N}$ and $\mathcal{Q}$ for Boolean, numbers (non-negative integers), and qubit data, respectively. Let $\mathcal{K}$ be an arbitrary

$$
\begin{array}{rll}
\mathcal{N}\texttt{Exp} \ni \texttt{n}, \texttt{n}_1, \texttt{n}_2 & ::= & \texttt{x}^{\mathcal{N}} \mid n \in \mathbb{N} \mid \texttt{n}_1 + \texttt{n}_2 \mid \texttt{n}_1 - \texttt{n}_2 \mid \texttt{n}_1 \times \texttt{n}_2 \\
\mathcal{B}\texttt{Exp} \ni \texttt{b}, \texttt{b}_1, \texttt{b}_2 & ::= & \texttt{x}^{\mathcal{B}} \mid \texttt{tt} \mid \texttt{ff} \mid \texttt{n}_1 = \texttt{n}_2 \mid \texttt{n}_1 < \texttt{n}_2 \mid \neg\texttt{b} \mid \texttt{b}_1 \wedge \texttt{b}_2 \mid \texttt{b}_1 \vee \texttt{b}_2 \\
\texttt{Exp} \ni \texttt{e}, \texttt{e}_1, \texttt{e}_2 & ::= & \texttt{n} \mid \texttt{b} \\
\texttt{Stmt} \ni \texttt{stm}, \texttt{stm}_1, \texttt{stm}_2 & ::= & \texttt{skip} \mid \texttt{x}^{\mathcal{K}} = \texttt{e}^{\mathcal{K}} \mid \texttt{stm}_1\texttt{;}\ \texttt{stm}_2 \mid \texttt{if}\ \texttt{b}^{\mathcal{B}}\ \texttt{then}\ \texttt{stm}_1\ \texttt{else}\ \texttt{stm}_2 \\
& & \mid\ \texttt{while}\ \texttt{b}^{\mathcal{B}}\ \texttt{do}\ \texttt{stm} \mid \overline{\texttt{q}}^{\mathcal{Q}} *= \texttt{U} \mid \texttt{x}^{\mathcal{B}} = \texttt{meas}\ \texttt{q}^{\mathcal{Q}}
\end{array}
$$

Fig. 2. Syntax of quantum programs.

classical type in $\{\mathcal{B}, \mathcal{N}\}$. Each program variable comes with a fixed datatype and can be optionally annotated by its type as a superscript. In what follows, we will use $\texttt{x}, \texttt{x}', \texttt{y}, \ldots$ to denote classical variables of type $\mathcal{K}$ and $\texttt{q}, \texttt{q}', \ldots$ to denote quantum variables of type $\mathcal{Q}$. A program, denoted $\texttt{P}$, is simply a statement; see Figure 2. Program statements are either classical assignments, conditionals, sequences, loops, *quantum assignments* $\overline{\texttt{q}}^{\mathcal{Q}} *= \texttt{U}$, or *measurements* $\texttt{x}^{\mathcal{B}} = \texttt{meas}\ \texttt{q}^{\mathcal{Q}}$. A quantum assignment consists in the application of a quantum unitary gate $\texttt{U}$ of arity $ar(\texttt{U})$ to a sequence of qubits $\overline{\texttt{q}} \triangleq \texttt{q}_1, \ldots, \texttt{q}_{ar(\texttt{U})}$. As we will see in the semantics section, a unitary matrix U will be associated with each quantum gate $\texttt{U}$. A measurement performs a single qubit measurement of $\texttt{q}$ in the computational basis: the outcome is a Boolean value and the quantum state evolves accordingly. For a given syntactic construct t, let $\mathcal{B}(t)$ (respectively $\mathcal{N}(t)$, $\mathcal{Q}(t)$) be the set of Boolean (respectively, number, qubit) variables in t.

Notice that the language encompasses qubit-initializing in the basis states. In particular, we will use $\texttt{q}^{\mathcal{Q}} = |0\rangle$ as syntactic sugar for $\texttt{x} = \texttt{meas}\ \texttt{q}\texttt{;}\ \texttt{if}\ \texttt{x}\ \texttt{then}\ \texttt{q} *= \texttt{X}\ \texttt{else}\ \texttt{skip}$, for $\texttt{X}$ being the Pauli X gate and for some fresh variable $\texttt{x}$ of type $\mathcal{B}$.

Example 1. Consider the program of Figure 3, adapted from [6], as a simple leading example. Let H be the unitary operator computing the Hadamard gate. This program simulates coin tossing by repeatedly measuring the qubit $\texttt{q}$, until the measurement outcome $\texttt{ff}$ occurs. The probability to terminate within n steps depends on the initial state $\rho = \left(\begin{smallmatrix} \alpha & \beta \\ \gamma & \delta \end{smallmatrix}\right)$ (a density matrix in $\mathbb{C}^{2\times 2}$, which implies $\alpha + \delta = 1$ and $\gamma = \bar{\beta}$) of the qubit $\texttt{q}$. Variable $\texttt{i}$ is increased by one at each iteration, and hence, when the program terminates, $\texttt{i}$ stores as final value the number of loop iterations performed. The overall probability of termination is 1. The mean value of variable $\texttt{i}$, that is, the expected number of loop iterations, depends on the program input, in particular on the initial quantum state. After termination, for an initial state $\rho = \left(\begin{smallmatrix} \alpha & \beta \\ \bar{\beta} & \delta \end{smallmatrix}\right)$, its expected value is given by

$$F(\rho) = p_0 \times 1 + \sum_{i=1}^{\infty} \frac{p_1}{2^i}(i+1) = p_0 + p_1 + 2p_1 = 1 + (\alpha - \beta - \bar{\beta} + \delta) = 2 - (\beta + \bar{\beta}),$$

where $p_0 = \frac{\alpha+\beta+\bar{\beta}+\delta}{2} = \frac{1+\beta+\bar{\beta}}{2}$ and $p_1 = 1 - p_0$ are the probabilities of measuring $|0\rangle$ and $|1\rangle$, respectively, on the first iteration of the loop. For instance, for a qubit

```
Cntoss  ≜  x^B = tt;
           i^N = 0;
           while x do {
             i = i + 1;
             q^Q *= H;        } ≜ stm
             x = meas q
           }
```

with $H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$

Fig. 3. Quantum Coin tossing

initialized in state $|\phi\rangle = \sqrt{1/3}\,|0\rangle + \sqrt{2/3}\,|1\rangle$, the corresponding density matrix is $\rho_{|\phi\rangle} = |\phi\rangle\langle\phi| = \begin{pmatrix} 1/3 & \sqrt{2}/3 \\ \sqrt{2}/3 & 2/3 \end{pmatrix}$ and hence the expected number of loop iterations is $F(\rho_{|\phi\rangle}) = 2 - 2\sqrt{2}/3$. It will be simply 2 in the case of an initialization in the computational basis $|\phi\rangle = |0\rangle$ or $|\phi\rangle = |1\rangle$.

Operational Semantics. Following [6], we model the dynamics of our language as a probabilistic abstract reduction system (see [9]), a transition system where reduction is defined as a relation over probability distributions.

Probabilistic abstract reduction systems. Given a subset $\mathbb{K}$ of $\mathbb{R}$, let $\mathbb{K}^+$ be the set of non-negative numbers in $\mathbb{K}$, i.e., $\mathbb{K}^+ \triangleq \mathbb{K} \cap \{x \mid x \geq 0\}$ and let $\mathbb{K}^\infty$ be defined by $\mathbb{K}^\infty \triangleq \mathbb{K} \cup \{\infty\}$.

A discrete (sub)*distribution* δ over a set A is a function $\delta : A \to [0,1]$ with countable support $\mathrm{supp}(\delta) \triangleq \{a \in A \mid \delta(a) \neq 0\}$ that maps an element a of A to a probability $\delta(a)$ such that $|\delta| \triangleq \sum_{a \in \mathrm{supp}(\delta)} \delta(a) = 1$ ($|\delta| \leq 1$). Any (sub)distribution δ can be written as $\{\delta(a) : a\}_{a \in \mathrm{supp}(\delta)}$. The set of subdistributions over A, denoted by $\mathcal{D}(A)$, is closed under denumerable convex combinations $\sum_i p_i \cdot \delta_i \triangleq \lambda a. \sum_i p_i \delta_i(a)$, with $p_i \in [0,1]$ and $\sum_i p_i \leq 1$. Slightly simplifying standard notation, given $f : A \to \mathbb{R}^{+\infty}$ and a subdistribution $\delta \in \mathcal{D}(A)$, we define $\mathbb{E}_\delta(f)$, *the expectation of* f *on* δ, by $\mathbb{E}_\delta(f) \triangleq \Sigma_{a \in supp(\delta)} \delta(a) f(a)$. Note that $\mathbb{E}_\delta(f) \in \mathbb{R}^{+\infty}$ is always defined, since the images of f are non-negative reals.

Bournez and Garnier [9] introduced the notion of *Probabilistic Abstract Reduction System* (PARS) as a means to study reduction systems that evolve probabilistically. A PARS $\to$ on A is a binary relation $\cdot \to \cdot \subseteq A \times \mathcal{D}(A)$. The intended meaning is that when $a \to \delta$, then a reduces to $b \in \mathrm{supp}(\delta)$ with probability $\delta(b)$. Here, we focus on *deterministic* PARSs, i.e., PARSs $\to$ with $a \to \delta_1$ and $a \to \delta_2$ implies $\delta_1 = \delta_2$. An object $a \in A$ is called *terminal* if there is no rule $a \to \delta$, which we write as $a \not\to$.

Every deterministic PARS $\to$ over A naturally lifts to a reduction relation $\twoheadrightarrow$ over distributions so that $\delta \twoheadrightarrow \varepsilon$, if the reduct distribution ε is obtained from δ by replacing reducts in $\mathrm{supp}(\delta)$ according to the PARS $\to$. In fact, we define this lifting in terms of a ternary relation $\cdot \dot{\twoheadrightarrow} \cdot \subseteq \mathcal{D}(A) \times \mathbb{R}^+ \times \mathcal{D}(A)$ on

distributions, where in a step $\delta \xrightarrow{c}\!\!\!\!\rightarrow \varepsilon$ the *weight* c signifies the probability that a reduction has occurred. This relation is defined wrt. the following three rules.

$$\frac{a \not\to}{\{1:a\} \xrightarrow{0}\!\!\!\!\rightarrow \{1:a\}} \qquad \frac{a \to \delta}{\{1:a\} \xrightarrow{1}\!\!\!\!\rightarrow \delta} \qquad \frac{\delta_i \xrightarrow{c_i}\!\!\!\!\rightarrow \epsilon_i \quad \sum_i p_i \leq 1}{\sum_i p_i \cdot \delta_i \xrightarrow{\sum_i p_i c_i}\!\!\!\!\rightarrow \sum_i p_i \cdot \epsilon_i}$$

We may sometimes use the n-fold ($n \geq 0$) composition of $\dot{\to}$, denoted $\dot{\to}^n$, given by $\delta \xrightarrow{c}\!\!\!\!\rightarrow^n \epsilon$ if $\delta \xrightarrow{c_1}\!\!\!\!\rightarrow \cdots \xrightarrow{c_n}\!\!\!\!\rightarrow \epsilon$ and the weights satisfy $c = \sum_{i=1}^{n} c_i$. Notice that since $\to$ is deterministic, so is $\xrightarrow{c}\!\!\!\!\rightarrow$, in the sense that $\delta \xrightarrow{c_1}\!\!\!\!\rightarrow \epsilon_1$ and $\delta \xrightarrow{c_2}\!\!\!\!\rightarrow \epsilon_2$ implies $c_1 = c_2$ and $\epsilon_1 = \epsilon_2$. Thus, in particular, for every $a \in A$ there is precisely one (infinite) reduction

$$\{1:a\} = \delta_0 \xrightarrow{c_0}\!\!\!\!\rightarrow \delta_1 \xrightarrow{c_1}\!\!\!\!\rightarrow \delta_2 \xrightarrow{c_2}\!\!\!\!\rightarrow \delta_3 \twoheadrightarrow \cdots .$$

For any $b \in A$, $\delta_i(b)$ gives the probability that a reduces to b in i steps. Note that when b is terminal, this probability only increases along reductions (i.e., $\delta_i(b) \leq \delta_{i+1}(b)$ for all i). This justifies that we define the *terminal distribution* of a as the distribution $\delta(b) \triangleq \lim_{i\to\infty} \delta_i(b)$. Note that $\delta(b)$ gives the probability that a reaches b in an arbitrary (but finite) number of steps. Since the weights c_i indicate the probability that a step has been performed from δ_i to δ_{i+1}, the infinite sum $\sum_{i=0}^{\infty} c_i \in \mathbb{R}^{+\infty}$ gives the expected number of reduction steps carried out, the *expected derivation length of* a [5].

For a PARS $\to$, we denote by $\mathsf{term}_{\to} : A \to \mathcal{D}(A)$ the function associating with each $a \in A$ its terminal distribution. The *expected derivation length function* $\mathsf{edl}_{\to} : A \to \mathbb{R}^{+\infty}$ associates each $a \in A$ to its expected derivation length. The PARS $\to$ is *almost surely terminating* [40] (*a.s. terminating* for short) if $a \in A$ reduces to a terminal object $b \not\to$ with probability 1, that is, if $|\mathsf{term}_{\to}(a)| = 1$ for every a. It is *positive almost surely terminating*, if the expected derivation length is always finite, that is, $\mathsf{edl}_{\to}(a) < \infty$ for all $a \in A$.

Apart from termination, we are interested also in questions related to functional correctness, such as (i) what is the probability that a reaches a terminal b, (ii) what is the probability that a reaches a terminal satisfying predicate P, and more generally, (iii) which value does a function $f : A \to \mathbb{R}^{+\infty}$ take, in expectation, when fully reducing an object a. In the literature [32], one tool to answer all of these are given by *weakest pre-expectation transformers*, the natural generalization of classical weakest pre-condition transformers to a quantitative, probabilistic setting. We suite this notion to PARSs.

Definition 1 (Weakest pre-expectation). *The* weakest pre-expectation *for a PARS* $\to$ *over* A *is given by the function*

$$\mathsf{wp}_{\to} : (A \to \mathbb{R}^{+\infty}) \to (A \to \mathbb{R}^{+\infty})$$
$$\mathsf{wp}_{\to} \triangleq \lambda f.\lambda a.\ \mathbb{E}_{\mathsf{term}_{\to}(a)}(f).$$

For $\mathbf{1}_b$ the indicator function evaluating to 1 on argument b and to 0 otherwise, and by seeing a predicate P as a $0,1$-valued function, $\mathsf{wp}_{\to}\ \mathbf{1}_b\ a$ answers

$$\frac{}{(\mathtt{skip}, s, \rho) \to_{\mathrm{Q}} \{1 : (\downarrow, s, \rho)\}}\ (\mathrm{Skip}) \qquad \frac{}{(\mathtt{x} = \mathtt{e}, s, \rho) \to_{\mathrm{Q}} \{1 : (\downarrow, s[\mathtt{x} := [\![\mathtt{e}]\!]^s], \rho)\}}\ (\mathrm{Exp})$$

$$\frac{}{(\overline{\mathtt{q}} *= \mathtt{U}, s, \rho) \to_{\mathrm{Q}} \{1 : (\downarrow, s, \Phi_{U_{\overline{\mathtt{q}}}}(\rho))\}}\ (\mathrm{Op})$$

$$\frac{}{(\mathtt{x} = \mathtt{meas}\ \mathtt{q}_i, s, \rho) \to_{\mathrm{Q}} \{tr(M_{k,i}\rho) : (\downarrow, s[\mathtt{x} := k], m_{k,i}(\rho))\}_{k \in \{0,1\}}}\ (\mathrm{Meas})$$

$$\frac{(\mathtt{stm}_1, s, \rho) \to_{\mathrm{Q}} \{p_i : (\mathtt{stm}_{\downarrow}^i, s^i, \rho^i)\}_{i \in I}}{(\mathtt{stm}_1\mathtt{;}\ \mathtt{stm}_2, s, \rho) \to_{\mathrm{Q}} \{p_i : (\mathtt{stm}_{\downarrow}^i\mathtt{;}\ \mathtt{stm}_2, s^i, \rho^i)\}_{i \in I}}\ (\mathrm{Seq})$$

$$\frac{[\![\mathtt{b}]\!]^s \in \{0,1\}}{(\mathtt{if\ b\ then\ stm}_1\ \mathtt{else\ stm}_0, s, \rho) \to_{\mathrm{Q}} \{1 : (\mathtt{stm}_{[\![\mathtt{b}]\!]^s}, s, \rho)\}}\ (\mathrm{Cond})$$

$$\frac{[\![\mathtt{b}]\!]^s = 0}{(\mathtt{while\ b\ do\ stm}, s, \rho) \to_{\mathrm{Q}} \{1 : (\downarrow, s, \rho)\}}\ (\mathrm{Wh}_0)$$

$$\frac{[\![\mathtt{b}]\!]^s = 1}{(\mathtt{while\ b\ do\ stm}, s, \rho) \to_{\mathrm{Q}} \{1 : (\mathtt{stm;\ while\ b\ do\ stm}, s, \rho)\}}\ (\mathrm{Wh}_1)$$

Fig. 4. Operational semantics in terms of PARS.

question (i), $\mathsf{wp}_{\to}\ P\ a$ answers (ii), and generally $\mathsf{wp}_{\to}\ f\ a$ answers question (iii). Note also that a PARS is a.s. terminating iff $\mathsf{wp}_{\to}\ (\lambda b.\, 1)\ a = 1$ for each $a \in A$. On the other hand, positive a.s. termination cannot be expressed through an application of $\mathsf{wp}_{\to}$.

Quantum programs as PARSs. We now endow quantum programs with an operational semantics defined in terms of a PARS. Given a totally ordered set of qubits $Q = \{\mathtt{q}_1, \ldots, \mathtt{q}_n\}$, let $\mathcal{H}_Q$ be the 2^n-dimensional Hilbert space defined by $\mathcal{H}_Q \triangleq \otimes_{i=1}^n \mathcal{H}_{\mathtt{q}_i}$, with $\mathcal{H}_{\mathtt{q}} = \mathbb{C}^2$ being the vector space of computational basis $\{|0\rangle, |1\rangle\}$ and $\otimes$ being the tensor product. With $\langle k|$ we denote the transpose conjugate of $|k\rangle$, for $k \in \{0,1\}$. Let $\mathcal{M}(\mathcal{H}_Q)$ be the set of complex square matrices acting on the Hilbert space $\mathcal{H}_Q$, i.e., $\mathcal{M}(\mathcal{H}_Q) = \mathbb{C}^{2^n \times 2^n}$. Given $M \in \mathcal{M}(\mathcal{H}_Q)$, $M^\dagger$ denotes the transpose conjugate of M, and I_{2^n} denotes the identity matrix over $\mathcal{M}(\mathcal{H}_Q)$. We will write I when the dimension is clear from the context.

Let $\mathfrak{D}(\mathcal{H}_Q) \subsetneq \mathcal{M}(\mathcal{H}_Q)$ be the set of all *density operators* (or quantum states), i.e., positive semi-definite matrices of trace equal to 1 on $\mathcal{H}_Q$. Density operators can be viewed as the mathematical representation of a (mixed) *quantum state*. A *unitary operator* U is a matrix in $\mathcal{M}(\mathcal{H}_Q)$ such that $UU^\dagger = U^\dagger U = I$. A *superoperator* $\Phi_U : \mathfrak{D}(\mathcal{H}_Q) \to \mathfrak{D}(\mathcal{H}_Q)$, an endomorphism over density operators, is attached to each unitary operator U and defined by $\Phi_U \triangleq \lambda\rho.U\rho U^\dagger$. By definition, Φ_U is a completely positive trace preserving linear map. Indeed, $tr(U\rho U^\dagger) = tr(\rho)$, by unitarity. Hence $U\rho U^\dagger$ is a density operator in $\mathfrak{D}(\mathcal{H}_Q)$ for each $\rho \in \mathfrak{D}(\mathcal{H}_Q)$.

Regarding measurements, for each i, $1 \leq i \leq card(Q)$, we define $M_{k,i} \in \mathcal{M}(\mathcal{H}_Q)$, with $k \in \{0,1\}$, by $M_{0,i} \triangleq I_{2^{i-1}} \otimes (|0\rangle\langle 0|) \otimes I_{2^{n-i}}$ and $M_{1,i} \triangleq I - M_{0,i}$. The measurement of the qubit q_i (in the computational basis) of a density matrix $\rho \in \mathfrak{D}(\mathcal{H}_Q)$, produces the classical outcome $k \in \{0,1\}$ with probability $tr(M_{k,i}\rho)$. The (normalized) quantum state, after the measurement, is defined by

$$m_{k,i}(\rho) \triangleq \begin{cases} \frac{M_{k,i}\rho M_{k,i}^\dagger}{tr(M_{k,i}\rho)}, & \text{if } tr(M_{k,i}\rho) \neq 0, \\ \frac{I}{2^n} & \text{otherwise.} \end{cases}$$

Note that for all $\rho \in \mathfrak{D}(\mathcal{H}_Q)$, $m_{k,i}(\rho) \in \mathfrak{D}(\mathcal{H}_Q)$, as it holds that $tr(m_{k,i}(\rho)) = 1$. Indeed, $tr(M_{k,i}\rho M_{k,i}^\dagger) = tr(M_{k,i}^2\rho) = tr(M_{k,i}\rho)$, as $M_{k,i}$ is a projection. Hence $m_{k,i}$ is a map in $\mathfrak{D}(\mathcal{H}_Q) \to \mathfrak{D}(\mathcal{H}_Q)$.

We set $[\![\mathcal{B}]\!] \triangleq \{0,1\}$ and $[\![\mathcal{N}]\!] \triangleq \mathbb{N}$. The classical state is modeled as a (well-typed) *store* s of domain $dom(s)$ mapping each variable $\mathtt{x}$ of type $\mathcal{K}$ to a value in $[\![\mathcal{K}]\!]$. With $\mathtt{Store}$, we denote the set of all such stores. Let $s[\mathtt{x}^{\mathcal{K}} := k]$ with $k \in [\![\mathcal{K}]\!]$ be the store obtained from s by updating the value assigned to $\mathtt{x}$ in the map s. Given a store s, let $[\![-]\!]^s : \mathcal{K}\mathtt{Exp} \to [\![\mathcal{K}]\!]$ be the map associating to each expression $\mathtt{e}$ of type $\mathcal{K}$ and such that $\mathcal{B}(\mathtt{e}) \cup \mathcal{N}(\mathtt{e}) \subseteq dom(s)$, a value in $[\![\mathcal{K}]\!]$, defined in the obvious way. For example $[\![\mathtt{x}]\!]^s \triangleq s(\mathtt{x})$, $[\![n]\!]^s \triangleq n$, $[\![\mathtt{tt}]\!]^s \triangleq 1$, $[\![\mathtt{n}_1 - \mathtt{n}_2]\!]^s \triangleq \max(0, [\![\mathtt{n}_1]\!]^s - [\![\mathtt{n}_2]\!]^s)$, etc.

Let $\downarrow$ be a special symbol for termination. A *configuration* μ, for (extended) statement $\mathtt{stm} \in \mathtt{Stmt} \cup \{\downarrow\}$, store $s \in \mathtt{Store}$, and a quantum state $\rho \in \mathcal{H}_Q$, has the form $(\mathtt{stm}, s, \rho)$. Let $\mathtt{Conf}$ be the set of configurations. A configuration $(\mathtt{stm}, s, \rho)$ is well-formed with respect to the sets of variables B, V, and Q if $\mathcal{B}(\mathtt{stm}) \subseteq B$, $\mathcal{N}(\mathtt{stm}) \subseteq V$, $\mathcal{Q}(\mathtt{stm}) \subseteq Q$, $dom(s) = B \cup V$, and $\rho \in \mathfrak{D}(\mathcal{H}_Q)$. Throughout the paper, we only consider configurations that are well-formed with respect to the sets of variables of the program under consideration.

The operational semantics is described in Figure 4 as a PARS $\to_{\mathrm{Q}}$ over objects in $\mathtt{Conf}$, where terminal objects are precisely the configurations of the shape $(\downarrow, s, \rho)$. The (classical or quantum) state of a configuration can only be updated by the three rules (Exp), (Op), and (Meas). Rule (Exp) updates the classical store wrt the value of the evaluated expression. Rule (Op) updates the quantum state to a new quantum state $\Phi_{U_{\overline{\mathtt{q}}}}(\rho) = U_{\overline{\mathtt{q}}}\rho U_{\overline{\mathtt{q}}}^\dagger$, where $U_{\overline{\mathtt{q}}}$ is the unitary operator in $\mathcal{M}(\mathcal{H}_Q)$ computed by extending the quantum gate $\mathtt{U}$ to the entire set of qubits Q. Rule (Meas) performs a measurement on qubit $\mathtt{q}_i$. This rule returns a distribution of configurations corresponding to the two possible outcomes, $k = 0$ and $k = 1$, with their respective probabilities $tr(M_{k,i}\rho)$ and, in each case, updates the classical store and the quantum state accordingly. In the particular case where $tr(M_{k_0,i}\rho) = 0$ for some $k_0 \in \{0,1\}$, $\{tr(M_{k,i}\rho) : (\downarrow, s[\mathtt{x} := k], m_{k,i}(\rho))\}_{k\in\{0,1\}} = \{1 : (\downarrow, s[\mathtt{x} := 1 - k_0], m_{1-k_0,i}(\rho))\}$. Rule (Seq) governs the execution of a sequence of statements $\mathtt{stm}_1\,\mathtt{;}\,\mathtt{stm}_2$, under the covenant that $\downarrow\ \mathtt{;}\ \mathtt{stm} \triangleq \mathtt{stm}$, for each statement $\mathtt{stm}$. The rule accounts for potential probabilistic behavior when $\mathtt{stm}_1$ performs a measurement and it is otherwise standard. All the other rules are standard.

In a configuration $\mu = (\mathtt{stm}, s, \rho)$, the pair $\sigma \triangleq (s, \rho)$ is called a state. Let $\mathtt{St}^{\mathtt{stm}}$ be the set of states $\sigma, \tau, \ldots$ that are well-formed wrt statement $\mathtt{stm}$. For simplicity, we will denote this set by $\mathtt{St}$ when $\mathtt{stm}$ is clear from the context. To ease the presentation, we sometimes write $(\mathtt{stm}, \sigma)$ for the configuration μ.

We will be interested in expectation-based reasoning on quantum programs. In what follows, we also call functions $f : \mathtt{Conf} \to \mathbb{R}^{+\infty}$ *expectations*, for brevity.

Definition 2. *For a statement* $\mathtt{stm}$ *and* $f : \mathtt{St} \to \mathbb{R}^{+\infty}$*, we overload the notions of expected derivation length and weakest pre-expectation by:*

$$\mathsf{edl}_{\mathtt{stm}} : \mathtt{St} \to \mathbb{R}^{+\infty} \qquad \mathsf{qwp}_{\mathtt{stm}} : (\mathtt{St} \to \mathbb{R}^{+\infty}) \to (\mathtt{St} \to \mathbb{R}^{+\infty})$$
$$\mathsf{edl}_{\mathtt{stm}} \triangleq \lambda\sigma.\mathsf{edl}_{\to_{\mathrm{Q}}}(\mathtt{stm}, \sigma) \qquad \mathsf{qwp}_{\mathtt{stm}} \triangleq \lambda f.\lambda\sigma.\mathsf{wp}_{\to_{\mathrm{Q}}}(f_{st})(\mathtt{stm}, \sigma),$$

where $f_{st}(\mathtt{stm}, \tau) = f(\tau)$.

Example 2. Consider the program $\mathtt{Cntoss}$ given Figure 3. In the setting of the program $\mathtt{Cntoss}$, $Q = \{\mathtt{q}\}$, $M_{0,1} = \left(\begin{smallmatrix} 1 & 0 \\ 0 & 0 \end{smallmatrix}\right)$ and $M_{1,1} = \left(\begin{smallmatrix} 0 & 0 \\ 0 & 1 \end{smallmatrix}\right)$. On an initial state $\sigma = (s, \rho)$, the reduction starts deterministically as in the classical setting, performing the initialization $\mathtt{x} = \mathtt{tt}$ and $\mathtt{i} = 0$. From there, evaluation reaches the loop $\mathtt{while\ x\ do\ stm}$. At each loop iteration, the loop counter $\mathtt{i}$ is incremented, and the Hadamard gate applied to the quantum variable $\mathtt{q}$. The loop guard is obtained through measuring $\mathtt{q}$.

To see how this is reflected in the semantics, let us first look at an iteration of the loop. If $\mathtt{x}$ was set to false, that is $\mathtt{x}$ holds the value 0, by rule (Wh_0) the loop terminates within one step:

$$\{1 : (\mathtt{while\ x\ do\ stm}, [\mathtt{x}{:=}0, \mathtt{i}{:=}i], \rho)\} \xrightarrow{1}\!\!\!\!\rightarrow_{\mathrm{Q}} \{1 : (\downarrow, [\mathtt{x}{:=}0, \mathtt{i}{:=}i], \rho)\}. \tag{0}$$

On the other hand, when $\mathtt{x}$ was previously set to true, the loop executes its body. Precisely, we have:

$$\begin{aligned}
&\{1 : (\mathtt{while\ x\ do\ stm}, [\mathtt{x}{:=}1, \mathtt{i}{:=}i], \rho)\} \\
&\quad \xrightarrow{1}\!\!\!\!\rightarrow_{\mathrm{Q}} \{1 : (\mathtt{i = i{+}1;\ q = H;\ x = meas\ q;\ while\ x\ do\ stm}, [\mathtt{x}{:=}1, \mathtt{i}{:=}i], \rho)\} && (1)\\
&\quad \xrightarrow{1}\!\!\!\!\rightarrow_{\mathrm{Q}} \{1 : (\mathtt{q = H;\ x = meas\ q;\ while\ x\ do\ stm}, [\mathtt{x}{:=}1, \mathtt{i}{:=}i+1], \rho)\} && (2)\\
&\quad \xrightarrow{1}\!\!\!\!\rightarrow_{\mathrm{Q}} \{1 : (\mathtt{x = meas\ q;\ while\ x\ do\ stm}, [\mathtt{x}{:=}k, \mathtt{i}{:=}i+1], \Phi_H(\rho))\} && (3)\\
&\quad \xrightarrow{1}\!\!\!\!\rightarrow_{\mathrm{Q}} \{p_k : (\mathtt{while\ x\ do\ stm}, [\mathtt{x}{:=}k, \mathtt{i}{:=}i+1], \rho_k))\}_{k\in\{0,1\}}, && (4)
\end{aligned}$$

where in the last step, the probability p_k equals $\mathrm{tr}(M_{k,1}\Phi_H(\rho))$, while the normalized quantum state ρ_k is given as $m_{k,1}(\Phi_H(\rho))$. The above reduction is obtained by applying the rules of Figure 4: rule (Wh_1) for reduction (1); rules (Exp) and (Seq) for reduction (2); rules (Op) and (Seq) for reduction (3); and finally rules (Meas) and (Seq) for reduction (4).

For an arbitrary initial quantum state $\rho = \left(\begin{smallmatrix} \alpha & \beta \\ \gamma & \delta \end{smallmatrix}\right) \in \mathfrak{D}(\mathcal{H}_Q)$ (where $\alpha, \beta, \gamma, \delta \in \mathbb{C}$ and $tr(\rho) = \alpha + \delta = 1$, $\gamma = \overline{\beta}$, etc.), it follows that

$$p_0 = tr(M_{0,1}H\rho H^\dagger) = tr(\left(\begin{smallmatrix} 1 & 0 \\ 0 & 0 \end{smallmatrix}\right)\frac{1}{2}\left(\begin{smallmatrix} \alpha+\beta+\gamma+\delta & \alpha-\beta+\gamma-\delta \\ \alpha+\beta-\gamma-\delta & \alpha-\beta-\gamma+\delta \end{smallmatrix}\right)) = \frac{1+\beta+\gamma}{2},$$

and that, $p_1 = 1 - p_0 = \frac{1-(\beta+\gamma)}{2}$. Using $\rho_k = \frac{M_{k,1} H \rho H^\dagger M_{k,1}^\dagger}{tr(M_{k,1} H \rho H^\dagger)} = \frac{(M_{k,1}H)\rho(M_{k,1}H)^\dagger}{p_k}$,

$$\rho_0 = \frac{\left(\begin{smallmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 0 & 0 \end{smallmatrix}\right)\left(\begin{smallmatrix} \alpha & \beta \\ \gamma & \delta \end{smallmatrix}\right)\left(\begin{smallmatrix} 1/\sqrt{2} & 0 \\ 1/\sqrt{2} & 0 \end{smallmatrix}\right)}{p_0} = \left(\begin{smallmatrix} 1 & 0 \\ 0 & 0 \end{smallmatrix}\right) \quad \rho_1 = \frac{\left(\begin{smallmatrix} 0 & 0 \\ 1/\sqrt{2} & 1/\sqrt{2} \end{smallmatrix}\right)\left(\begin{smallmatrix} \alpha & \beta \\ \gamma & \delta \end{smallmatrix}\right)\left(\begin{smallmatrix} 0 & 1/\sqrt{2} \\ 0 & 1/\sqrt{2} \end{smallmatrix}\right)}{p_1} = \left(\begin{smallmatrix} 0 & 0 \\ 0 & 1 \end{smallmatrix}\right).$$

Summarizing (1)–(4) we thus get:

$$\begin{aligned} \{1 : (\texttt{while x do stm}, [\texttt{x}{:=}1, \texttt{i}{:=}i], \left(\begin{smallmatrix} \alpha & \beta \\ \gamma & \delta \end{smallmatrix}\right))\} \\ \xrightarrow{4}\!\!\!\!\twoheadrightarrow^4_{\mathrm{Q}} \{p_0 &: (\texttt{while x do stm}, [\texttt{x}{:=}0, \texttt{i}{:=}i+1], \rho_0), \\ p_1 &: (\texttt{while x do stm}, [\texttt{x}{:=}1, \texttt{i}{:=}i+1], \rho_1)\}. \end{aligned}$$

Putting everything together, we have

$$\begin{aligned} (\texttt{Cntoss}, s, \left(\begin{smallmatrix} \alpha & \beta \\ \gamma & \delta \end{smallmatrix}\right)) \xrightarrow{2}\!\!\!\!\twoheadrightarrow^2_{\mathrm{Q}} & \{1 : (\texttt{while x do stm}, [\texttt{x}{:=}1, \texttt{i}{:=}0], \rho)\} \\ \xrightarrow{4}\!\!\!\!\twoheadrightarrow^4_{\mathrm{Q}} & \{p_0 : (\texttt{while x do stm}, [\texttt{x}{:=}0, \texttt{i}{:=}1], \rho_0), \\ & \quad p_1 : (\texttt{while x do stm}, [\texttt{x}{:=}1, \texttt{i}{:=}1], \rho_1)\} \\ \xrightarrow{p_0+4p_1}\!\!\!\!\twoheadrightarrow^4_{\mathrm{Q}} & \{p_0 : \underline{(\downarrow, [\texttt{x}{:=}0, \texttt{i}{:=}1], \rho_0)}, \\ & \quad \tfrac{p_1}{2} : (\texttt{while x do stm}, [\texttt{x}{:=}0, \texttt{i}{:=}2], \rho_0), \\ & \quad \tfrac{p_1}{2} : (\texttt{while x do stm}, [\texttt{x}{:=}1, \texttt{i}{:=}2], \rho_1)\} \\ \xrightarrow{\frac{p_1}{2}+4\frac{p_1}{2}}\!\!\!\!\twoheadrightarrow^4_{\mathrm{Q}} & \{p_0 : \underline{(\downarrow, [\texttt{x}{:=}0, \texttt{i}{:=}1], \rho_0)}, \\ & \quad \tfrac{p_1}{2} : \underline{(\downarrow, [\texttt{x}{:=}0, \texttt{i}{:=}2], \rho_0)}, \\ & \quad \tfrac{p_1}{4} : (\texttt{while x do stm}, [\texttt{x}{:=}0, \texttt{i}{:=}3], \rho_0), \\ & \quad \tfrac{p_1}{4} : (\texttt{while x do stm}, [\texttt{x}{:=}1, \texttt{i}{:=}3], \rho_1)\} \\ \xrightarrow{\frac{p_1}{4}+4\frac{p_1}{4}}\!\!\!\!\twoheadrightarrow^4_{\mathrm{Q}} & \quad \ldots \end{aligned}$$

where terminal configurations are underlined. This reduction converges to the terminal distribution

$$\mathsf{term}_{\texttt{Cntoss}}(s, \rho) = \{p_0 : (\downarrow, [\texttt{x}{:=}0, \texttt{i}{:=}1], \rho_0)\} + \{\tfrac{p_1}{2^i} : (\downarrow, [\texttt{x}{:=}0, \texttt{i}{:=}i+1], \rho_0)\}_{i \geq 1},$$

with an expected derivation length of

$$\mathsf{edl}_{\texttt{Cntoss}}(s, \left(\begin{smallmatrix} \alpha & \beta \\ \gamma & \delta \end{smallmatrix}\right)) = 2 + 4 + (p_0 + 4p_1) + \sum_{i=1}^{\infty} \frac{5p_1}{2^i} = 7 + 8p_1 = 11 - 4(\beta + \gamma).$$

For expectation $f(s, \rho) \triangleq s(\texttt{i})$, measuring the iteration counter $\texttt{i}$, we have

$$\mathsf{qwp}_{\texttt{Cntoss}}\ f\ (s, \left(\begin{smallmatrix} \alpha & \beta \\ \gamma & \delta \end{smallmatrix}\right)) = p_0 \times 1 + \sum_{i=1}^{\infty} \frac{p_1}{2^i}(i+1) = p_0 + p_1 + 2p_1 = 2 - (\beta + \gamma),$$

that is, the mean value held by $\texttt{i}$ holds after execution is $2 - (\beta + \gamma)$. The termination probability is

$$\mathsf{qwp}_{\texttt{Cntoss}}\ (\lambda\sigma.1)\ (s, \left(\begin{smallmatrix} \alpha & \beta \\ \gamma & \delta \end{smallmatrix}\right)) = p_0 \times 1 + \sum_{i=1}^{\infty} \frac{p_1}{2^i} \times 1 = p_0 + p_1 = 1,$$

i.e., the program is almost surely terminating.

3 Weakest Pre-expectations and Arithmetical Hierarchy

In this section, we study the hardness of some natural quantitative problems for weakest pre-expectations and expected derivation length.

Computability-Aimed Restrictions. This subsection is devoted to putting some restrictions on programs and on the considered notion of expectation to overcome the issues of computability, mentioned in the introduction.

Algebraic numbers. Towards this end, our solution is to target a subset of complex numbers, where simple operations like equality are decidable. We consider the set $\overline{\mathbb{Q}}$ of *algebraic numbers*, i.e., complex numbers in $\mathbb{C}$ that are roots of a non-zero polynomial in $\mathbb{Q}[X]$. Let $\mathbb{A} \triangleq \overline{\mathbb{Q}} \cap \mathbb{R}$ be the real closed field of real algebraic numbers in $\mathbb{R}$. The following inclusions trivially hold (i) $\mathbb{N} \subseteq \mathbb{Q} \subseteq \mathbb{A} \subseteq \mathbb{R} \subseteq \mathbb{C}$ and (ii) $\overline{\mathbb{Q}} \subseteq \mathbb{C}$. It was proved in [18, Proposition 2.2] that equality over $\overline{\mathbb{Q}}$ and inequality over $\mathbb{A}$ are decidable using Cohn's representation [12]. It is well-known that the product and sum over $\overline{\mathbb{Q}}$ are computable in polynomial time.

We now restrict the program semantics to matrices and density operators over algebraic numbers. Given a totally ordered set of qubits $Q = \{\mathtt{q}_1, \ldots, \mathtt{q}_n\}$, let $\tilde{\mathcal{H}}_Q$ be the Hausdorff pre-Hilbert space $\overline{\mathbb{Q}}^{2^n}$ (i.e., the completeness requirement on Hilbert spaces is withdrawn) of n qubits defined by $\tilde{\mathcal{H}}_Q \triangleq \otimes_{i=1}^{n} \tilde{\mathcal{H}}_{\mathtt{q}_i}$, with $\tilde{\mathcal{H}}_{\mathtt{q}} \triangleq \overline{\mathbb{Q}}^2$ being the vector space of computational basis $\{|0\rangle, |1\rangle\}$ over the field $\overline{\mathbb{Q}}$. Let $\mathcal{M}(\tilde{\mathcal{H}}_Q)$ and $\mathfrak{D}(\tilde{\mathcal{H}}_Q)$ be the set of matrices and density operators on $\tilde{\mathcal{H}}_Q$, respectively.

Clifford+T gates. For the program semantics to be defined on the space $\mathfrak{D}(\tilde{\mathcal{H}}_Q)$, the considered quantum gates are now restricted to gates whose corresponding unitary operators are in $\mathcal{M}(\tilde{\mathcal{H}}_Q)$, i.e., have a matrix representation over the algebraic numbers. To this end, we consider a restriction to the *Clifford+T gates*: `I`, `X`, `Y`, `Z`, `H`, `S`, `CNOT`, and `T`, whose unitary matrices are given below:

$$I \triangleq \left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right),\ X \triangleq \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right),\ Y \triangleq \left(\begin{smallmatrix} 0 & -\mathrm{i} \\ \mathrm{i} & 0 \end{smallmatrix}\right),\ Z \triangleq \left(\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}\right),\ H \triangleq \frac{1}{\sqrt{2}} \left(\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right),$$

$$S \triangleq \frac{1}{\sqrt{2}} \left(\begin{smallmatrix} 1 & 0 \\ 0 & \mathrm{i} \end{smallmatrix}\right),\ CNOT \triangleq \left(\begin{smallmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{smallmatrix}\right),\ T \triangleq \left(\begin{smallmatrix} 1 & 0 \\ 0 & e^{\mathrm{i}\frac{\pi}{4}} \end{smallmatrix}\right).$$

The Clifford+T fragment is the set of unitary transformations generated by sequential (matrix multiplication) and parallel (Kronecker product) compositions of the gates H, S, $CNOT$, and T. This constitutes a reasonable restriction for unitary operators as it is known to be the simplest approximately universal fragment of quantum mechanics [1].

A central observation is that the superoperator associated with a unitary operator of the Clifford+T fragment is an endomorphism over density operators in $\mathfrak{D}(\tilde{\mathcal{H}}_Q)$.

Lemma 1. *The Clifford+T fragment preserves $\mathfrak{D}(\tilde{\mathcal{H}}_Q)$, i.e., there exist Q and $\overline{\mathtt{q}} \in Q$ such that for each unitary operator U of the Clifford+T fragment $\Phi_{U_{\overline{\mathtt{q}}}} \in \mathfrak{D}(\tilde{\mathcal{H}}_Q) \to \mathfrak{D}(\tilde{\mathcal{H}}_Q)$.*

Notice that, while a restriction to Clifford+T is reasonable in terms of quantum mechanics and universality, our result can be extended by adding any quantum gate preserving the above lemma. For example, the phase shift gate, defined by $P_\varphi \triangleq \left(\begin{smallmatrix} 1 & 0 \\ 0 & e^{\mathrm{i}\varphi} \end{smallmatrix}\right)$, preserves $\mathfrak{D}(\tilde{\mathcal{H}}_Q)$ whenever $\varphi = r\pi$, for any $r \in \mathbb{Q}$.

Let $\mathtt{Stmt}_{\mathtt{CT}}$ be the set of statements restricted to quantum gates computing Clifford+T unitary operators (hence a subset of $\mathtt{Stmt}$), $\mathtt{St}_{\mathtt{CT}}$ be the set of states whose quantum state is in $\mathfrak{D}(\tilde{\mathcal{H}}_Q)$, and $\mathtt{Conf}_{\mathtt{CT}}$ be the set of well-formed configurations in $(\mathtt{Stmt}_{\mathtt{CT}} \cup \{\downarrow\}) \times \mathtt{St}_{\mathtt{CT}}$. Let $\mathtt{St}_{\mathtt{CT}}^{\mathtt{stm}}$ be the set of states in $\mathtt{St}_{\mathtt{CT}}$ that are well-formed wrt statement $\mathtt{stm}$. Once again, by abuse of notation, we will denote this set by $\mathtt{St}_{\mathtt{CT}}$ when $\mathtt{stm}$ is clear from the context.

A consequence of Lemma 1 is that $\mathtt{Conf}_{\mathtt{CT}}$ is closed under reduction, in the following sense. Let $\mathcal{D}^{\mathrm{fin}}_{\mathbb{A}^+}(A) \subseteq \mathcal{D}(A)$ be the set of *finitely supported* subdistributions δ with algebraic probabilities, i.e., $\delta(a) \in \mathbb{A}^+$ for all $a \in A$.

Lemma 2. *The set $\mathcal{D}^{fin}_{\mathbb{A}^+}(\mathtt{Conf}_{\mathtt{CT}})$ is stable under reduction, more precisely, if $\delta \in \mathcal{D}^{fin}_{\mathbb{A}^+}(\mathtt{Conf}_{\mathtt{CT}})$ and $\delta \xrightarrow{c}\!\!\!\!\rightarrow_{\mathtt{Q}} \varepsilon$, then $\varepsilon \in \mathcal{D}^{fin}_{\mathbb{A}^+}(\mathtt{Conf}_{\mathtt{CT}})$ and $c \in \mathbb{A}^+$.*

Computable expectations. We also restrict the expectation codomain to algebraic numbers. Hence the considered expectations will be functions in $\mathtt{St}_{\mathtt{CT}} \to \mathbb{A}^+$. On its own, this restriction is not sufficient for our concerns, as the set $\mathtt{St}_{\mathtt{CT}} \to \mathbb{A}^+$ is not countable. It implies that there exist expectations in $\mathtt{St}_{\mathtt{CT}} \to \mathbb{A}^+$ that are not computable functions. To resolve this issue, we restrict the space of expectations further to computable ones:

$$\mathtt{E}_{\mathtt{CT}} \triangleq \{f \mid f : \mathtt{St}_{\mathtt{CT}} \to \mathbb{A}^+,\ f \text{ computable}\}.$$

An immediate consequence of Lemma 2 is that $\mathsf{term}_{\mathtt{stm}}(\sigma) \in \mathcal{D}(\mathtt{Conf}_{\mathtt{CT}})$ for any $\mathtt{stm} \in \mathtt{Stmt}_{\mathtt{CT}}$ and $\sigma \in \mathtt{St}_{\mathtt{CT}}$. In consequence, $\mathsf{qwp}_{\mathtt{stm}}\ f\ \sigma$ is well-defined for all $f \in \mathtt{St}_{\mathtt{CT}}$. This justifies that in our treatment below, we restrict expectations to the class $\mathtt{E}_{\mathtt{CT}}$. However, keep in mind that despite Lemma 2, the subdistribution $\mathsf{term}_{\mathtt{stm}}(\sigma)$, obtained at the limit, does not fall within $\mathcal{D}^{\mathrm{fin}}_{\mathbb{A}^+}(A)$. It is neither finite nor are probabilities algebraic ($\mathbb{A}^+$ is not complete). In particular, in general $\mathsf{qwp}_{\mathtt{stm}}\ f\ \sigma$ is a real number, rather than an algebraic one.

Quantitative Problems. We now define formally the quantitative problems that we study.

Testing problems. Some natural quantitative problems related to weakest pre-expectations are to determine for a given program $\mathtt{stm}$, a given state σ, a given expectation f, and a given algebraic number a, whether the corresponding weakest pre-expectation $\mathsf{qwp}_{\mathtt{stm}}\ f\ \sigma$ is smaller than or equal to a. In this setting, it makes sense to consider any possible relation in the set $\{<, \leq, =, \geq, >\} \subseteq \mathcal{P}(\mathbb{A} \times \mathbb{A})$ as one could be interested in finding precise values, (strict) upper- or lower-bounds.

Definition 3. *The* testing problem *sets* $\textsc{Test}_{\mathcal{R}} \subseteq \mathtt{Conf}_{\mathtt{CT}} \times \mathtt{E}_{\mathtt{CT}} \times \mathbb{A}^+$*, for* $\mathcal{R} \in \{<, \leq, =, \geq, >\}$*, are defined by:*

$$(\mathtt{stm}, \sigma, f, a) \in \textsc{Test}_{\mathcal{R}} :\iff (\mathsf{qwp}_{\mathtt{stm}}\ f\ \sigma)\ \mathcal{R}\ a.$$

The consideration of both $\textsc{Test}_{\leq}$ and $\textsc{Test}_{>}$ may seem redundant, as $\textsc{Test}_{>}$ can be viewed as the complement of $\textsc{Test}_{\leq}$. However, it makes perfect sense to distinguish both properties, when considering the corresponding universal problems, as we do in a moment.

Finiteness problem. Another problem of interest consists in checking whether the weakest pre-expectations produces some finitary output.

Definition 4. *The* finiteness problem *set* $\textsc{Test}_{\neq\infty} \subseteq \mathtt{Conf}_{\mathtt{CT}} \times \mathtt{E}_{\mathtt{CT}}$ *is defined by:*

$$(\mathtt{stm}, \sigma, f) \in \textsc{Test}_{\neq\infty} :\iff \mathsf{qwp}_{\mathtt{stm}}\ f\ \sigma < \infty.$$

Termination problems. We also define two termination problems for almost sure termination and positive almost sure termination:

Definition 5. *The sets of (*positive*)* almost-sure terminating *configurations* $\textsc{Ast} \subseteq \mathtt{Conf}_{\mathtt{CT}}$ *(*$\textsc{Past} \subseteq \mathtt{Conf}_{\mathtt{CT}}$*) are defined by:*

$$(\mathtt{stm}, \sigma) \in \textsc{Ast} :\iff |\mathsf{term}_{\mathtt{stm}}(\sigma)| = 1$$
$$(\mathtt{stm}, \sigma) \in \textsc{Past} :\iff \mathsf{edl}_{\mathtt{stm}}(\sigma) < \infty.$$

It is well-known that $\textsc{Past} \subsetneq \textsc{Ast}$, cf. [9].

Universal problems. Another kind of natural problems arises if one tries to check some properties for each possible program input (i.e., for each state σ). We can thus define universal properties for each of the sets described previously.

Definition 6. *The sets of universal testing, finiteness and (positive) a.s. termination problems are defined by:*

$$(\mathtt{stm}, f, g) \in \textsc{UTest}_{\mathcal{R}} \subseteq \mathtt{Stmt}_{\mathtt{CT}} \times \mathtt{E}_{\mathtt{CT}}^2 \iff \forall \sigma \in \mathtt{St}_{\mathtt{CT}},\ (\mathtt{stm}, \sigma, f, g(\sigma)) \in \textsc{Test}_{\mathcal{R}}$$
$$(\mathtt{stm}, f) \in \textsc{UTest}_{\neq\infty} \subseteq \mathtt{Stmt}_{\mathtt{CT}} \times \mathtt{E}_{\mathtt{CT}} \iff \forall \sigma \in \mathtt{St}_{\mathtt{CT}},\ (\mathtt{stm}, \sigma, f) \in \textsc{Test}_{\neq\infty}$$
$$\mathtt{stm} \in \textsc{UAst} \subseteq \mathtt{Stmt}_{\mathtt{CT}} \iff \forall \sigma \in \mathtt{St}_{\mathtt{CT}},\ (\mathtt{stm}, \sigma) \in \textsc{Ast}$$
$$\mathtt{stm} \in \textsc{UPast} \subseteq \mathtt{Stmt}_{\mathtt{CT}} \iff \forall \sigma \in \mathtt{St}_{\mathtt{CT}},\ (\mathtt{stm}, \sigma) \in \textsc{Past}$$

Example 3. We have $\mathtt{Cntoss} \in \textsc{UAst}$ and $\mathtt{Cntoss} \in \textsc{UPast}$, for the program `Cntoss` of Figure 3. Indeed, it was shown in Example 2 that `Cntoss` terminates with probability 1 and a finite expected derivation length. This property holds for any input of the domain. In the same example, we have proven $(\mathtt{Cntoss}, f) \in \textsc{Test}_{\neq\infty}$ for $f(s, \rho) = s(\mathtt{i})$. Indeed, we have shown the stronger property $(\mathtt{Cntoss}, f, g) \in \textsc{Test}_{=}$, where $g(s, \left(\begin{smallmatrix} \alpha & \beta \\ \gamma & \delta \end{smallmatrix}\right)) = 2 - (\beta + \gamma)$.

	Standard		Universal	
	Problem	Class	Problem	Class
Testing	$\textsc{Test}_{>}$	Σ_1^0	$\textsc{UTest}_{>}$	Π_2^0 (‡)
	$\textsc{Test}_{\geq}$	Π_2^0 (‡)	$\textsc{UTest}_{\geq}$	Π_2^0 (‡)
	$\textsc{Test}_{=}$	Π_2^0	$\textsc{UTest}_{=}$	Π_2^0 (‡)
	$\textsc{Test}_{\leq}$	Π_1^0 (‡)	$\textsc{UTest}_{\leq}$	Π_1^0 (‡)
	$\textsc{Test}_{<}$	Σ_2^0	$\textsc{UTest}_{<}$	Π_3^0 (‡)
Finiteness	$\textsc{Test}_{\neq\infty}$	Σ_2^0	$\textsc{UTest}_{\neq\infty}$	Π_3^0 (‡)
Termination	AST	Π_2^0	UAST	Π_2^0
	PAST	Σ_2^0	UPAST	Π_3^0

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

Completeness Results in the Arithmetical Hierarchy. In what follows, we place the introduced quantitative problems within the *arithmetical hierarchy* [34]. The arithmetical hierarchy is a means to classify and relate *undecidable* problems wrt. to their inherent difficulty, measured in terms of the number of (unbounded) quantifier alternations needed to state the problem as a formula in first-order arithmetic, based on a decidable (recursive) predicate.

Reminder on the arithmetical hierarchy. Classes of the arithmetical hierarchy are defined inductively as follows:

$$\begin{aligned}
&\Pi_0^0 = \Sigma_0^0 \triangleq \textsc{rec}, \quad \textsc{rec} \text{ being the class of decidable problems (recursive sets)}\\
&\Pi_{n+1}^0 \triangleq \{\psi \mid \exists \phi \in \Sigma_n^0,\ \forall \overline{x}.(\psi(\overline{x}) \iff \forall \overline{y}.\phi(\overline{x},\overline{y}))\},\\
&\Sigma_{n+1}^0 \triangleq \{\psi \mid \exists \phi \in \Pi_n^0,\ \forall \overline{x}.(\psi(\overline{x}) \iff \exists \overline{y}.\phi(\overline{x},\overline{y}))\}.
\end{aligned}$$

For each n, Π_n^0 is the complement of Σ_n^0 (i.e., $\Pi_n^0 = \text{co-}\Sigma_n^0$, and vice versa) and it is a well-known result that Σ_1^0 and Π_1^0 correspond to the classes RE of recursively enumerable (i.e., semi-decidable) problems and co-RE of co-recursively enumerable (i.e., co-semi-decidable) problems, respectively. Given the sets $A \subseteq X$ and $B \subseteq Y$, we write $A \leq_{\mathsf{m}} B$ (A is many-one reducible to B) if there exists a computable function $f : X \to Y$ such that $\forall x \in X,\ x \in A \iff f(x) \in B$. Given a class C of the arithmetical hierarchy and a set A, A is C-*hard* if $\forall B \in \text{C},\ B \leq_{\mathsf{m}} A$. A set A is C-*complete* if $A \in \text{C}$ and A is C-hard. It is well-known that if a set A is C-complete then its complement, noted co-A, is (co-C)-complete.

Results. Table 1 associates the quantum decision problems to the corresponding classes in the arithmetical hierarchy for which we have proven them *complete*, that is, we have proven membership and hardness for the corresponding class. Some of the results may seem surprising. For instance, the testing problem $\textsc{Test}_{>}$, i.e., deciding $\mathsf{qwp}_{\mathtt{stm}}\ f\ \sigma > a$ within the Clifford+T fragment, turns out to be recursive enumerable. It is thus classified identical to the (classical) *halting*

problem $\mathcal{H}$.[§] Remarkable, through the restriction to the Clifford+T fragment, corresponding problems are ranked within the arithmetical hierarchy identical to their non-quantum counterparts (see [37,24]). This observation holds for all problems apart those marked with (‡) which, to the best of our knowledge, have not been studied in a classical/probabilistic setting. Π_2^0- and Π_3^0-completeness of the universal testing problems, given relations $>$ and $<$ respectively, has been conjectured by Kaminski in his PhD thesis [25] for probabilistic programs.

A crucial observation towards these results is that, restricting to the Clifford+T fragment, the weakest pre-expectation of a program $\mathtt{P}$ can be *approximated* through *computable* transformers $\mathsf{qwp}^{\leq n}_{\mathtt{stm}} : \mathtt{E}_{\mathtt{CT}} \to \mathtt{E}_{\mathtt{CT}}$ that limit execution of $\mathtt{stm}$ to at most $n \in \mathbb{N}$ reduction steps. That is,

$$\mathsf{qwp}^{\leq n}_{\mathtt{stm}}\ f\ \sigma \triangleq \mathbb{E}_{\mathsf{term}^{\leq n}_{\mathtt{stm}}(\sigma)}(f),$$

for $\mathsf{term}^{\leq n}_{\mathtt{stm}}(\sigma)$ the distribution of terminal configurations obtained within n reduction steps, when evaluating $(\mathtt{stm}, \sigma)$. With regards to the above mentioned $\textsc{Test}_{>} \in \Sigma_1^0$ for instance, observe that:

$$\begin{aligned}(\mathtt{stm}, f, \sigma, a) \in \textsc{Test}_{>} &\iff \mathsf{qwp}_{\mathtt{stm}}\ f\ \sigma > a \\ &\iff \lim_{i\to\infty} \mathsf{qwp}^{\leq n}_{\mathtt{stm}}\ f\ \sigma > a \\ &\iff \exists n \in \mathbb{N},\ \exists \delta \in \mathbb{A}^+ \setminus \{0\},\ \mathsf{qwp}^{\leq n}_{\mathtt{stm}}\ f\ \sigma \geq a + \delta.\end{aligned}$$

Crucially, the predicate $\mathsf{qwp}^{\leq n}_{\mathtt{stm}}\ f\ \sigma \geq a+\delta$ becomes computable. In essence, this is a consequence of Lemma 2: The n-th step normal form distribution $\mathsf{term}^{\leq n}_{\mathtt{stm}}(\sigma)$ is finite and computable, as f is computable so is thus $\mathsf{qwp}^{\leq n}_{\mathtt{stm}}\ f\ \sigma$. From here, the result follows now as equality on $\mathbb{A}$ is decidable. The proof of this, as well as all completeness proofs listed in Table 1 can be found in the Appendix. The following constitutes our first main result.

Theorem 1. *All completeness results in Table 1 hold.*

4 Quantum Expectation Transformers

In what follows, we are interested in deliniating subclasses of testing problems that lead to decidability. To this end, we now define a notion of *quantum expectation transformer* as a means to compute symbolically the weakest pre-expectation of a program. We first introduce some preliminary notations in order to lighten the presentation.

Notations. For any expression $\mathtt{e}$, $[\![\mathtt{e}]\!]$ is a shorthand notation for the function $\lambda(s,\rho).[\![\mathtt{e}]\!]^s \in \mathtt{St} \to \mathbb{R}^{+\infty}$. We will also use $f[\mathtt{x} := \mathtt{e}]$ for the expectation $\lambda(s,\rho).f(s[\mathtt{x} := [\![\mathtt{e}]\!]^s], \rho)$. Similarly, for a given map $\chi : \mathfrak{D}(\mathcal{H}_Q) \to \mathfrak{D}(\mathcal{H}_Q)$,

[§] In our context the halting set $\mathcal{H}$ can be defined as the class of classical programs and states $(\mathtt{P}, \sigma)$ for which $\mathtt{P}$ is halting on σ.

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

Fig. 5. Quantum expectation transformer $\mathtt{qet}[\,\cdot\,]\{\cdot\}$

$f[\chi] \triangleq \lambda(s,\rho).f(s,\chi(\rho))$. We will also sometimes group such state modifications, for instance, $f[\mathtt{x} := \mathtt{e};\, \chi]$ stands for $(f[\mathtt{x} := \mathtt{e}])[\chi]$ and $f[\mathtt{x} := \mathtt{e}, \mathtt{y} := \mathtt{e}']$ stands for $(f[\mathtt{x} := \mathtt{e}])[\mathtt{y} := \mathtt{e}']$.

For $p \in \mathtt{St} \to [0,1]$ and $f, g \in \mathtt{St} \to \mathbb{R}^{+\infty}$, $f +_p g$ denotes the function $\lambda\sigma.p(\sigma) \cdot f(\sigma) + (1 - p(\sigma)) \cdot g(\sigma) \in \mathtt{St} \to \mathbb{R}^{+\infty}$, similar we use $f \cdot g$ to denote $\lambda\sigma.f(\sigma) \cdot g(\sigma) \in \mathtt{St} \to \mathbb{R}^{+\infty}$. Thus, for instance, $f[\mathtt{x} := \mathtt{x} + 1] +_{[\![\mathtt{x}=1]\!]} f$ behaves like f, except that $\mathtt{x}$ is first incremented when applied to states with classical variable $\mathtt{x}$ equal to 1. In correspondence to the normalization of quantum state $m_{k,i}$, we define probabilities $p_{k,i} \triangleq \lambda\rho.tr(M_{k,i}\rho M_{k,i}^\dagger)$. We overload this function from $\mathfrak{D}(\mathcal{H}_Q)$ to $\mathtt{St}$ s.t. $p_{k,i}(s,\rho) = p_{k,i}(\rho)$. In this way, $f[\mathtt{x} := 0;\, m_{0,i}] +_{p_{0,i}} f[\mathtt{x} := 1;\, m_{1,i}]$ computes precisely the expected value of f on the distribution of states obtained by measuring the i-th qubit and assigning the outcome to classical variable $\mathtt{x}$.

Finally, we denote by $\leq$ also the pointwise extension of the order from $\mathbb{R}^{+\infty}$ to functions, that is, $f \leq g$ holds iff $\forall\sigma \in \mathtt{St},\ f(\sigma) \leq g(\sigma)$.

Definition 7 (Quantum expectation transformer). *The* quantum expectation transformer *consists in a program semantics mapping expectations to expectations in a continuation passing style*

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

and is defined inductively on statements in Figure 5.

This transformer corresponds to the notion of *expected value transformer* of [6] on the Kegelspitze $\mathtt{S} = (\mathbb{R}^{+\infty}, +_{\mathtt{f}})$, with $+_{\mathtt{f}}$ being the forgetful addition. In the case of loops, the least fixed point lfp is defined with respect to the pointwise ordering on the function space $\mathtt{St} \to \mathbb{R}^{+\infty}$. Equipped with this ordering, this space forms a ω-CPO. As the quantum transformer can be shown to be ω-continuous, the fixed-point is always defined, cf. [44].

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

Fig. 6. Universal laws derivable for the quantum expectation transformer.

Theorem 2 (Adequacy). *The following holds:*

$$\forall \mathtt{stm} \in \mathtt{Stmt},\ \forall f : \mathtt{St} \to \mathbb{R}^{+\infty},\ \mathtt{qwp}_{\mathtt{stm}}(f) = \mathtt{qet}[\mathtt{stm}]\{f\}\,.$$

Apart from continuity, the quantum expectation transformer satisfies several useful laws, see Figure 6. The (*monotonicity*) Law permits us to reason modulo upper-bounds: actual expectations can be always substituted by upper-bounds. It is in fact an immediate consequence from the (*continuity*) Law, which is defined for any ω-chain $(f_i)_i$. The (*upper invariance*) Law constitutes a generalization of the notion of invariant stemming from Hoare calculus. It is used to find closed-form upper-bounds g to expectations f of loops. The pre-conditions state that g should dominate f on states where the loop would immediately exist, and otherwise, should remain invariant under iteration. It is worth mentioning that this proof rule is not only sound, but also complete, in the sense that any upper-bound satisfies the two constraints. The following example illustrates the use of this rule on the running example.

Example 4. Following Example 2, we over-approximate $\mathtt{qet}[\mathtt{Cntoss}]\{f\}$, for $f(s,\rho) = s(\mathtt{i})$ the post-expectation measuring the classical variable $\mathtt{i}$.

To this end, observe that the function $g : \mathtt{St} \to \mathbb{R}^{+\infty}$ is an upper-invariant (Figure 6) to the while loop $\mathtt{while\ x\ do\ stm}$, given a post-expectation $f : \mathtt{St} \to \mathbb{R}^{+\infty}$. Recall that the loop body $\mathtt{stm}$ comprises $(\mathtt{i} = \mathtt{i}{+}1;\ \mathtt{q} \mathrel{*}= \mathtt{H};\ \mathtt{x} = \mathtt{meas\ q})$. To fulfill the conditions of the (*upper invariance*) Law the following inequalities have to be met:

$$[\![\neg \mathtt{x}]\!] \cdot f \leq g \qquad [\![\mathtt{x}]\!] \cdot \mathtt{qet}[\mathtt{i} = \mathtt{i}{+}1;\ \mathtt{q} \mathrel{*}= \mathtt{H};\ \mathtt{x} = \mathtt{meas\ q}]\{g\} \leq g. \tag{5}$$

By unfolding the definition, we see

$$\begin{aligned}
&\mathtt{qet}[\mathtt{i} = \mathtt{i}{+}1;\ \mathtt{q} \mathrel{*}= \mathtt{H};\ \mathtt{x} = \mathtt{meas\ q}]\{g\}\\
&\quad = \mathtt{qet}[\mathtt{i} = \mathtt{i}{+}1]\{\mathtt{qet}[\mathtt{q} \mathrel{*}= \mathtt{H}]\{\mathtt{qet}[\mathtt{x} = \mathtt{meas\ q}]\{g\}\}\}\\
&\quad = \mathtt{qet}[\mathtt{i} = \mathtt{i}{+}1]\big\{\mathtt{qet}[\mathtt{q} \mathrel{*}= \mathtt{H}]\big\{g[\mathtt{x}{:=}0;\ m_{0,1}] +_{p_{0,1}} g[\mathtt{x}{:=}1;\ m_{1,1}]\big\}\big\}\\
&\quad = \mathtt{qet}[\mathtt{i} = \mathtt{i}{+}1]\big\{g[\mathtt{x}{:=}0;\ m_{0,1};\ \Phi_H] +_{p_{0,1}\cdot\Phi_H} g[\mathtt{x}{:=}1;\ m_{1,1};\ \Phi_H]\big\}\\
&\quad = g[\mathtt{x}{:=}0;\ m_{0,1};\ \Phi_H;\ \mathtt{i}{:=}\mathtt{i}{+}1] +_{p_{0,1}\cdot\Phi_H} g[\mathtt{x}{:=}1;\ m_{1,1};\ \Phi_H;\ \mathtt{i}{:=}\mathtt{i}{+}1]\\
&\quad = \lambda(s,\rho).\ \sum_{k\in\{0,1\}} p_{k,1}(\Phi_H(\rho)) \cdot g(s[\mathtt{x} = k, \mathtt{i}{:=}\mathtt{i}{+}1], m_{k,1}(\Phi_H(\rho)))\,.
\end{aligned}$$

By using the identities computed already in Example 2, we thus obtain

$$\mathtt{qet}[\,\mathtt{stm}\,]\{g\}\,(s,\left(\begin{smallmatrix}\alpha & \beta\\ \gamma & \delta\end{smallmatrix}\right)) = \sum_{k\in\{0,1\}} p_k \cdot g(s[\mathtt{x}=k, \mathtt{i}{:=}\mathtt{i}{+}1], \rho_k), \tag{6}$$

where, as in Example 2, $p_0 = \frac{1+\beta+\gamma}{2}$, $p_1 = \frac{1-(\beta+\gamma)}{2}$, $\rho_0 = \left(\begin{smallmatrix}1 & 0\\ 0 & 0\end{smallmatrix}\right)$ and $\rho_1 = \left(\begin{smallmatrix}1 & 0\\ 0 & 0\end{smallmatrix}\right)$

We claim that $g(s, \left(\begin{smallmatrix}\alpha & \beta\\ \gamma & \delta\end{smallmatrix}\right)) \triangleq s(\mathtt{i}) + s(\mathtt{x}) \cdot (2-(\beta+\gamma))$ is an upper-bound to the pre-expectation of the while loop wrt. to the post expectation f. To this end, we check (5). The first inequality is trivially satisfied. Concerning the second, notice that by definition,

$$g(s[\mathtt{x}=0, \mathtt{i}{:=}\mathtt{i}{+}1], \left(\begin{smallmatrix}1 & 0\\ 0 & 0\end{smallmatrix}\right)) = s(\mathtt{i}){+}1 \quad \text{and} \quad g(s[\mathtt{x}=1, \mathtt{i}{:=}\mathtt{i}{+}1], \left(\begin{smallmatrix}0 & 0\\ 0 & 1\end{smallmatrix}\right)) = s(\mathtt{i}){+}3.$$

By (6) we have

$$\begin{aligned}\mathtt{qet}[\,\mathtt{stm}\,]\{g\}\,(s,\left(\begin{smallmatrix}\alpha & \beta\\ \gamma & \delta\end{smallmatrix}\right)) &= \frac{1+\beta+\gamma}{2}(s(\mathtt{i})+1) + \frac{1-(\beta+\gamma)}{2}(s(\mathtt{i})+3)\\ &= (s(\mathtt{i})+2) - (\beta+\gamma) = g(s,\left(\begin{smallmatrix}\alpha & \beta\\ \gamma & \delta\end{smallmatrix}\right)),\end{aligned}$$

from which now the second constraint follows by case analysis on the value of $\mathtt{x}$. Hence $\mathtt{qet}[\,\mathtt{while\ x\ do\ stm}\,]\{f\} \leq g$ and, by monotonicity (Figure 6),

$$\begin{aligned}\mathtt{qet}[\,\mathtt{Cntoss}\,]\{f\}\,(s,\left(\begin{smallmatrix}\alpha & \beta\\ \gamma & \delta\end{smallmatrix}\right)) &\leq \mathtt{qet}[\,\mathtt{x}=\mathtt{tt};\ \mathtt{i}=0\,]\{g\}\,(s,\left(\begin{smallmatrix}\alpha & \beta\\ \gamma & \delta\end{smallmatrix}\right))\\ &= g([\mathtt{x}{:=}1, \mathtt{i}{:=}0], \left(\begin{smallmatrix}\alpha & \beta\\ \gamma & \delta\end{smallmatrix}\right)) = 2-(\beta+\gamma).\end{aligned}$$

Note that, in this case, the computed bound is exact.

One question of interest is to find the $\mathtt{qet}[\,\cdot\,]\{\cdot\}$ of a given statement. We obtain the following completeness results as a corollary of Theorem 1 and Theorem 2 on the Clifford+T fragment.

Corollary 1. *The following completeness results hold:*

- $\{(\mathtt{stm}, f, g) \in \mathtt{Stmt}_{\mathtt{CT}} \times \mathtt{E}^2_{\mathtt{CT}} \mid \forall\sigma,\ \mathtt{qet}[\,\mathtt{stm}\,]\{f\}\,(\sigma) \ = g(\sigma)\}$ *is* Π^0_2*-complete.*
- $\{(\mathtt{stm}, f, g) \in \mathtt{Stmt}_{\mathtt{CT}} \times \mathtt{E}^2_{\mathtt{CT}} \mid \forall\sigma,\ \mathtt{qet}[\,\mathtt{stm}\,]\{f\}\,(\sigma) \ \leq g(\sigma)\}$ *is* Π^0_1*-complete.*

The same kind of result can be straightforwardly obtained for each of the quantitative problems defined in previous section. All the corresponding sets are undecidable: they are at best (co-)semi-decidable as illustrated by Figure 1. This motivates us for restricting the problem a bit further to find a class of functions for which the quantitative problems for $\mathtt{wp}_{\mathtt{stm}}\ f$ can be decided.

5 Decidability of qet Inference over a Real Closed Field

Corollary 1 illustrates that it is not sufficient to relax the problem of finding the quantum expectation transformer of a given statement to upper-bounds, in order

$$\mathtt{qinf}[\mathtt{skip}]\{F\} \triangleq F$$
$$\mathtt{qinf}[\mathtt{x} = \mathtt{e}]\{F\} \triangleq F[\mathtt{x} := \mathtt{e}]$$
$$\mathtt{qinf}[\mathtt{stm}_1\,;\,\mathtt{stm}_2]\{F\} \triangleq \mathtt{qinf}[\mathtt{stm}_1]\{\mathtt{qinf}[\mathtt{stm}_2]\{F\}\}$$
$$\mathtt{qinf}\begin{bmatrix}\mathtt{if}^{\ell}\ \mathtt{b}\\ \mathtt{then}\ \mathtt{stm}_1\\ \mathtt{else}\ \mathtt{stm}_2\end{bmatrix}\{F\} \triangleq X_\ell, \text{ with side-cond. } \begin{cases}\mathtt{b} \vdash \mathtt{qinf}[\mathtt{stm}_1]\{F\} \leq X_\ell\\ \neg\mathtt{b} \vdash \mathtt{qinf}[\mathtt{stm}_2]\{F\} \leq X_\ell\end{cases}$$
$$\mathtt{qinf}\big[\mathtt{while}^{\ell}\ \mathtt{b}\ \mathtt{do}\ \mathtt{stm}\big]\{F\} \triangleq X_\ell, \text{ with side-cond. } \begin{cases}\mathtt{b} \vdash \mathtt{qinf}[\mathtt{stm}]\{X_\ell\} \leq X_\ell\\ \neg\mathtt{b} \vdash F \leq X_\ell\end{cases}$$
$$\mathtt{qinf}[\overline{\mathtt{q}} *= \mathtt{U}]\{F\} \triangleq F[\phi_{U_{\overline{\mathtt{q}}}}]$$
$$\mathtt{qinf}\big[\mathtt{x} = \mathtt{meas}^{\ell}\ \mathtt{q}_i\big]\{F\} \triangleq X_\ell, \text{ with side-cond. } \begin{cases}p_{0,i} = 0 \vdash F[\mathtt{x} := 1;\, m_{1,i}] \leq X_\ell\\ p_{1,i} = 0 \vdash F[\mathtt{x} := 0;\, m_{0,i}] \leq X_\ell\\ p_{k,i} \neq 0 \vdash F[\mathtt{x} := 0;\, m_{0,i}]\\ \qquad +_{p_{0,i}} F[\mathtt{x} := 1;\, m_{1,i}] \leq X_\ell\end{cases}$$

Fig. 7. Term representations of $\mathtt{qinf}[\cdot]\{\cdot\}$ and their corresponding side-conditions.

to make it decidable. The undecidability of finding the quantum expectation transformer of a given program is due to two other issues: 1) *Issue 1:* The computation of a fixpoint for $\mathtt{qet}[\cdot]\{\cdot\}$ in the case of while loops, 2) *Issue 2:* The check for inequalities over functions in $\mathtt{E}_{\mathtt{CT}}$, whose first-order theory is not decidable. This section is devoted to overcoming these two issues, by finding an expressive fragment on which the inference of an upper-bound of the quantum expectation transformer becomes decidable.

Symbolic Inference. As a first step towards automated inference, we define a symbolic variant of the quantum expectation transformer in Figure 7. In the case of conditionals, loops, and measurements, we will use fresh variables for expectations; side conditions will guarantee that these variables indeed denote (upper-bounds to) the corresponding expectations. This means that the symbolic version yields correct results only when the expectations assigned to these variables satisfy all the side conditions. By solving the generated constraints, viz., by finding an interpretation of ascribed variables that satisfy the imposed side-conditions, we effectively arrive at an inference procedure overcoming Issue 1.

To formalize this approach, we associate a unique label ℓ with each loop, conditional, and measurement, occurring in the considered program. Notationally, we write $\mathtt{while}^{\ell}\ \mathtt{b}\ \mathtt{do}\ \mathtt{stm}$ / $\mathtt{if}^{\ell}\ \mathtt{b}\ \mathtt{then}\ \mathtt{stm}_1\ \mathtt{else}\ \mathtt{stm}_2$ / $\mathtt{meas}^{\ell}\ \mathtt{q}$. Such labels permit us to associate a unique expectation variable X_ℓ to each of these constructs. Given a set of such expectation variables EVar, the set of terms ETerm, upon which the symbolic quantum expectation transformer operates, is defined according to the following grammar:

$$\mathsf{ETerm}\ F, G ::= X \mid F[\mathtt{x} := \mathtt{e}] \mid F[\chi] \mid F +_p G,$$

where X stand for an arbitrary expectation variable in EVar. As stressed above, X will be used to denote certain expectations wrt. loops, conditionals, and measurements. We have already introduced the notations $F[\mathtt{x} := \mathtt{e}]$ and $F[\chi]$ to represent updates to the classical and quantum state, respectively. Here, χ will always denote a finite composition of superoperators ϕ_U and measurements $m_{k,i}$. By ensuring that normalization of quantum states $m_{k,i}(\rho)$ is never considered in the degenerate case of a zero-probability measurement $p_{k,i}(\rho)$, it will thereby always be possible to write χ as $\lambda\rho.\frac{M\rho M^\dagger}{tr(N\rho N^\dagger)}$, for some $M \in \mathcal{M}(\tilde{\mathcal{H}}_Q)$ in the Clifford+T fragment. Finally, following the same reasoning, in the barycentric sum $F +_p G$ the probability p is a function in the quantum state, and will always be of general form $\lambda\rho.\frac{tr(M\rho M^\dagger)}{tr(N\rho N^\dagger)}$, for some $M, N \in \mathcal{M}(\tilde{\mathcal{H}}_Q)$ in the Clifford+T fragment. Similar to before, we may group updates such as in $F[\mathtt{x} := \mathtt{e};\ \chi]$.

The symbolic variation of the expectation transformer can now be defined as

$$\mathtt{qinf}[\,\cdot\,]\{\cdot\} : \mathsf{Stmt} \to \mathsf{ETerm} \to \mathsf{ETerm},$$

generating also a set of side-conditions of the shape $\Gamma \vdash F \leq G$, with the intended meaning that G binds F on all input states that satisfy the predicate Γ. The full definition of qinfer is given in Figure 7. As already hinted, the side conditions ensure that introduced variables X_ℓ indeed yield an upper-bound on the corresponding expectation, in the case of conditionals by case-analysis, and in the case of loops via an application of the upper-invariant law from Figure 6. In the case of measurements, $m_{k,i}$ and $p_{k,i}$ are defined exactly as before. Here, we single out the two cases where the probability of a measurement, either $p_{0,i}(\rho) = tr(M_{0,i}\rho) = tr(M_{0,i}\rho M_{0,i}^\dagger)$ or $p_{1,i}(\rho) = 1 - p_{0,i}(\rho)$, is zero. This way, we avoid the case analysis underlying the definition of $m_{k,i}$ and may, wlog., assume that it is indeed of the form $\lambda\rho.\frac{M_{k,i}\rho M_{k,i}^\dagger}{tr(M_{k,i}\rho M_{k,i}^\dagger)}$, with non-zero trace $tr(M_{k,i}\rho M_{k,i}^\dagger)$.

Example 5. In correspondence to Example 4, let us consider the application of the inference procedure on the program `Cntoss`, wrt. to the post-expectation $f(s,\rho) = s(\mathtt{i})$. We label the loop and measurement with $\mathtt{m}$ and $\mathtt{w}$, respectively.

Let X denote the post-expectation f. Unfolding the definition, we see

$$\begin{aligned}\mathtt{qinf}[\mathtt{Cntoss}]\{X\} &= \mathtt{qinf}[\mathtt{x} = \mathtt{tt};\ \mathtt{i} = 0;\ \mathtt{while}^{\mathtt{w}}\ \mathtt{x}\ \mathtt{do}\ \mathtt{stm}]\{X\}\\ &= X_{\mathtt{w}}[\mathtt{x}{:=}1;\ \mathtt{i}{:=}0],\end{aligned}$$

generating the side-conditions $\mathtt{x} \vdash X_{\mathtt{m}}[\Phi_H; \mathtt{i}{:=}\mathtt{i}{+}1] \leq X_{\mathtt{w}}$ and $\neg\mathtt{x} \vdash X \leq X_{\mathtt{w}}$. The left-hand side of the first constraint is obtained from

$$\begin{aligned}\mathtt{qinf}[\mathtt{stm}]\{X_{\mathtt{w}}\} &= \mathtt{qinf}[\mathtt{i} = \mathtt{i}{+}1]\{\mathtt{qinf}[\mathtt{q} *{=} H]\{\mathtt{qinf}[\mathtt{meas}^{\mathtt{m}}\ \mathtt{q}]\{X_{\mathtt{w}}\}\}\}\\ &= X_{\mathtt{m}}[\Phi_H;\ \mathtt{i}{:=}\mathtt{i}{+}1].\end{aligned}$$

Note that this expansion generates further constraints, this time on $X_{\mathtt{m}}$ representing the measurement. Specifically, it yields the following constraints:

$$p_{1-k,1} = 0 \vdash X_{\mathtt{w}}[\mathtt{x}{:=}k; m_{k,1}] \leq X_{\mathtt{m}}, \qquad (\text{for } k \in \{0,1\}),$$
$$p_{0,1} \neq 0 \neq p_{1,1} \vdash X_{\mathtt{w}}[\mathtt{x}{:=}0; m_{0,1}] +_{p_{0,1}} X_{\mathtt{w}}[\mathtt{x}{:=}1; m_{1,1}] \leq X_{\mathtt{m}}.$$

Using the analysis from Example 4, we interpret X_{w} and X_{m} as:

$$\alpha(X_{\mathsf{w}}) \triangleq \lambda(s, \left(\begin{smallmatrix}\alpha & \beta\\ \gamma & \delta\end{smallmatrix}\right)).\ s(\mathtt{i}) + s(\mathtt{x})(2-(\beta+\gamma)),$$
$$\alpha(X_{\mathsf{m}}) \triangleq \lambda(s, \left(\begin{smallmatrix}\alpha & \beta\\ \gamma & \delta\end{smallmatrix}\right)).\ s(\mathtt{i}) + 2 - 2\alpha.$$

Furthermore, we interpret the input variable X as f, i.e., $\alpha(X) \triangleq \lambda(s,\rho).\ s(\mathtt{i})$. Notice how $\alpha(X_{\mathsf{w}})$ just corresponds to the upper-invariant g derived in Example 4. Using the assignment, it is now standard to check that it is a solution to the five constraints. For instance, considering states $\sigma = (\{\mathtt{i}{:=}n, \mathtt{x}{:=}x\}, \left(\begin{smallmatrix}\alpha & \beta\\ \gamma & \delta\end{smallmatrix}\right))$, the ultimate constraint amount to the implication

$$\alpha \neq 0 \neq \delta \Rightarrow n +_\alpha (n+2) \leq n + 2 - 2\alpha,$$

which trivially holds. Finally, recall $\mathtt{qinf}[\,\mathtt{Cntoss}\,]\{X\} = X_{\mathsf{w}}[\mathtt{x}{:=}1\mathtt{;}\ \mathtt{i}{:=}0]$. This term is interpreted as $\lambda(s, \left(\begin{smallmatrix}\alpha & \beta\\ \gamma & \delta\end{smallmatrix}\right)).\ 2-(\beta+\gamma)$, yielding the optimal bound computed in Example 4.

Example 6. Re-consider program `RUS` depicted in Figure 1. Here, we are interested in an upper-bound on the number of T-gates, counted by the program variable `i`. As before, we label the loop and measurement with m and w, respectively. Let

$$\mathtt{stm} = \overbrace{\mathtt{q}_2 = |0\rangle\ \mathtt{;}\ \ldots}^{\mathtt{stm}_0}\mathtt{;}\ \mathtt{x} = \mathtt{meas}^{\mathsf{m}}\ \mathtt{q}_2,$$

be the body of the while loop statement (see Figure 1). We proceed with the analysis backwards. By the rules of Figure 7 it holds that $\mathtt{qinf}[\,\mathtt{stm}_0\,]\{F\} = F[\Phi\mathtt{;}\ \mathtt{i}{:=}\mathtt{i}{+}2]$ for any F, where Φ gives the quantum state updates within $\mathtt{stm}_0$. Unfolding definitions, we have $\mathtt{qinf}[\,\mathtt{RUS}\,]\{X\} = X_{\mathsf{w}}[\mathtt{x}{:=}0\mathtt{;}\ \mathtt{i}{:=}1]$ with $\mathtt{x} \vdash X_{\mathsf{m}}[\Phi;\mathtt{i}{:=}\mathtt{i}{+}2] \leq X_{\mathsf{w}}$ and $\neg\mathtt{x} \vdash X \leq X_{\mathsf{w}}$, since, by the above observation,

$$\mathtt{qinf}[\,\mathtt{stm}\,]\{X_{\mathsf{w}}\} = \mathtt{qinf}[\,\mathtt{stm}_0\,]\{\mathtt{qinf}[\,\mathtt{x} = \mathtt{meas}^{\mathsf{m}}\ \mathtt{q}_2\,]\{X_{\mathsf{w}}\}\} = X_{\mathsf{m}}[\Phi\mathtt{;}\ \mathtt{i}{:=}\mathtt{i}{+}2],$$

subject to the following additional constraints stemming from measurements:

$$p_{1-k,2} = 0 \vdash X_{\mathsf{w}}[\mathtt{x}{:=}k; m_{k,2}] \leq X_{\mathsf{m}}, \qquad (\text{for } k \in \{0,1\}),$$
$$p_{0,2} \neq 0 \neq p_{1,2} \vdash X_{\mathsf{w}}[\mathtt{x}{:=}0; m_{0,2}] +_{p_{0,2}} X_{\mathsf{w}}[\mathtt{x}{:=}1; m_{1,2}] \leq X_{\mathsf{m}}.$$

Taking $\alpha(X) \triangleq \lambda(s,\rho).\ s(\mathtt{i})$ and solving the constraints yields a constant upper bound of $8/3$ on the expected number of T-gates used by the program. This is due to the fact that the probability of the internal measurement is always $\frac{3}{4}$. Note that this bound is tight.

The transformer qinfer can be linked to qet of course only when variables X_ℓ are interpreted in a way that the side conditions generated by infer are met. To spell this out formally, let $\alpha : \mathsf{EVar} \to \mathtt{E}_{\mathtt{CT}}$ be an *assignment* of expectations to variables in EVar, and let $[\![F]\!]^\alpha : \mathtt{E}_{\mathtt{CT}}$ denote the interpretation of $F \in \mathsf{ETerm}$

under α defined in the natural way, e.g., $[\![X_\ell]\!]^\alpha = \alpha(X_\ell)$, $[\![F[\chi]]\!]^\alpha = [\![F]\!]^\alpha[\chi]$, etc.

We say that a constraint $\Gamma \vdash F \leq G$ is *valid under* α if $[\![F]\!]^\alpha(\sigma) \leq [\![G]\!]^\alpha(\sigma)$ holds for all states $\sigma \in \mathtt{St}_{\mathtt{CT}}$ with $\Gamma(\sigma)$. An assignment α is a *solution* to a set of constraints $\mathcal{C}$, if it makes every constraint in $\mathcal{C}$ valid. Finally, we say α is a solution to $\mathtt{qinf}[\,\mathtt{stm}\,]\{f\}$ if it is a solution to the set of constraints generated by $\mathtt{qinf}[\,\mathtt{stm}\,]\{f\}$. We have the following correspondence:

Theorem 3. *For any $\alpha \in \mathsf{EVar} \to \mathtt{E}_{\mathtt{CT}}$, if α is solution to $\mathtt{qinf}[\,\mathtt{stm}\,]\{F\} = G$, then it holds that $\mathtt{qet}[\,\mathtt{stm}\,]\{[\![F]\!]^\alpha\} \leq [\![G]\!]^\alpha$.*

It is worth mentioning that the above procedure could have been defined without restriction to the full space $\mathtt{St} \to \mathbb{R}^{+\infty}$ of expectations. In this case, this symbolic approach is also complete, in the sense that if $\mathtt{qet}[\,\mathtt{stm}\,]\{f\} = g$ then $\mathtt{qinf}[\,\mathtt{stm}\,]\{X\} = G$ for some G such that the side-conditions have a solution α, with $\alpha(X) = f$ and $[\![G]\!]^\alpha = g$. As our main focus is on decidability, however, we have made the choice to restrict ourself to the Clifford+T setting.

Restriction to Polynomials over the Real Closed Field $\mathbb{A}$. We now turn our eyes towards constraint solving, addressing the remaining Issue 2 through restricting the domain of expectations to *polynomials over algebraic numbers*. To be more precise, we consider the following problem.

Definition 8. *Let $E \subseteq \mathtt{E}_{\mathtt{CT}}$ be a class of expectations. The* inference problem $\textsc{QInfer}(E) \subseteq \mathtt{Stmt}_{\mathtt{CT}} \times E \times (\mathsf{EVar} \to E)$ *is given by*

$$(\mathtt{stm}, f, \alpha) \in \textsc{QInfer}(E) \iff \alpha[X := f] \text{ is solution to } \mathtt{qinf}[\,\mathtt{stm}\,]\{X\}$$

In the above definition, $(\mathtt{stm}, f, \alpha) \in \textsc{QInfer}(E)$ is satisfied if the statement $\mathtt{stm}$ has solution $\alpha[X := f]$ wrt. the expectation f. Hence it can be seen as checking whether f is a post-expectation for $\mathtt{stm}$. In particular, any solution $\alpha[X := f]$ constitutes an upper bound on the weakest pre-expectation of f (see Theorem 3). We will now see that $\textsc{QInfer}(E)$ is decidable, for E the set of *(real algebraic) polynomial expectations* of (arbitrary but fixed) degree d. For states $\mathtt{St}_{\mathtt{CT}}$ over n classical variables $\mathtt{y}_1, \ldots, \mathtt{y}_n$ and m qubits, let $\mathbb{A}^d[\mathtt{St}_{\mathtt{CT}}]$ denotes the class of functions of *polynomial expectations* of the form

$$\lambda(\{\mathtt{y}_i := Y_i\}_{1 \leq i \leq n}, (A_{j,k} + \mathtt{i}B_{j,k})_{1 \leq j,k \leq 2^m}).\ P, \tag{7}$$

where variables Y_i refer to the classical, and variables $A_{j,k}$ and $B_{j,k}$ refer to the real part and imaginary part, respectively, of each algebraic coefficient in the quantum state. Further, $P \in \mathbb{A}[Y_1, \ldots, Y_n, A_{1,1}, \ldots, A_{2^m,2^m}, B_{1,1}, \ldots, B_{2^m,2^m}]$ is a multivariate polynomial with coefficients in $\mathbb{A}$. The index d refers to the (total) degree of the underlying polynomial P. For instance,

$$\lambda(\{\mathtt{x} := X\,;\ \mathtt{i} := I\}, \begin{pmatrix} A_{1,1}+\mathtt{i}B_{1,1} & A_{1,2}+\mathtt{i}B_{1,2} \\ A_{2,1}+\mathtt{i}B_{2,1} & A_{2,2}+\mathtt{i}B_{2,2} \end{pmatrix}).\ I + X(2 - (A_{1,2} + A_{2,1})) \in \mathbb{A}^2[\mathtt{St}_{\mathtt{CT}}]$$

One important remark here is that we allow for possibly negative polynomials whereas expectations only output positive real algebraic numbers. Consequently, some side conditions are put on the admissible coefficients $A_{j,k}$ and $B_{j,k}$ of the input density matrix to preserve this condition (the matrix is positive, has trace 1, is hermitian). For example, $\sum_{i=1}^{2^m} A_{i,i} = 1$, $\sum_{i=1}^{2^m} B_{i,i} = 0$ (trace is 1) and $\forall i,\ k,\ A_{i,k} = A_{k,i}$ and $B_{i,k} = -B_{k,i}$ (self-adjointness). One can easily check that the expectations defined in Example 5 are in $\mathbb{A}^d[\mathtt{St}_{\mathtt{CT}}]$, for $d \geq 1$.

The restriction to polynomials is made on purpose, as quantifier elimination is decidable in the theory of real closed fields, a well known result due to Tarski and Seidenberg. Recall that the theory of real closed fields is the first-order theory in which the primitive operations are multiplication, addition, the order relation $\leq$, and the constants 0 and 1. Consequently, the only numbers that can be defined are the real algebraic numbers. Specifically, we will make use of the following result, quantifying the complexity of the quantifier elimination decision procedure as a function exponential in number of variables, and double-exponential in the number of quantifier alternations.

Proposition 1 ([21, Theorem 6]). *Let* $\mathbf{A}$ *be an integral ring over a real closed field* $\mathbf{R}$*. Let* $\psi = Q_1 \boldsymbol{x}_1 . Q_2 \boldsymbol{x}_2 . \cdots Q_l \boldsymbol{x}_l .\ \phi$ *be a formula in prenex-normal form, where* $\forall k,\ Q_k \in \{\forall, \exists\}$*,* $Q_k \neq Q_{k+1}$*, and* ϕ *is a quantifier-free formula over* i *variables and* j *atomic propositions of the shape* $P \geq 0$*, each* P *being a polynomial of degree at most* d *with coefficients in* $\mathbf{A}$*. There exists an algorithm computing a quantifier-free formula equivalent to* ψ *in time* $O(|\psi|) \cdot (jd)^{i^{O(l)}}$*.*

As $\mathbb{A}$ constitutes both an integral ring and a real closed field, the above theorem is in particular applicable taking $\mathbf{A} = \mathbf{R} \triangleq \mathbb{A}$. In the particular case where ψ is a closed formula, the resulting quantifier-free formula is simply a Boolean combination of inequalities over constants from $\mathbb{A}$. Since we already observed that these can be decided in polynomial time, the above proposition thus implies that validity of ψ is decidable under the given time bound.

By restricting assignment α to polynomial expectations, it becomes decidable to check that α is a solution to a given constraint set C. Indeed, under such a polynomial assignment α, a constraint $\Gamma \vdash F \leq G$ becomes expressible as a formula in the theory of real closed field $\mathbb{A}$. By letting α range over polynomial expectations with undetermined coefficients, we can this way arrive at the main decidability result of this section.

Theorem 4. *For any degree* $d \in \mathbb{N}$*,* $d \geq 1$*, the problem* $\textsc{Qinfer}(\mathbb{A}^d[\mathtt{St}_{\mathtt{CT}}])$ *is decidable in time* $2^{2^{dO(n)}}$*, where* n *is the size of the considered program.*

Practical Algorithm. Theorem 4 established a computable algorithm on the inference of upper bounds on weakest pre-expectation on quantitative program properties of any given mixed classical-quantum program. Nevertheless, the complexity of this algorithm — double-exponential in the program size — is forbiddingly high. In order to turn this procedure into a practical algorithm, we have to tame this inherent complexity. For this, significant further restrictions on the

class of bounding functions are necessary. We propose to proceed as follows. (1) *Bounding functions:* in (7) we restricted the class of expectations to polynomials, which in turn yield a bound on the weakest pre-expectation. Based on an analysis of concrete examples considered in the literature (e.g., [30,6]), this can be tightened further to degree 2 polynomials. (2) *Approximate solutions*: Theorem 4 rests upon (the decidability) of quantifier elimination. Thus the constraints C induced through the symbolic inference of $\texttt{qinf}[\texttt{stm}]\{X\} = G$ ($G, X \in \mathsf{ETerm}$) are solved exactly. Over-approximation, however, suffices, if we are only interested in soundness of the inference mechanism.

The restriction of the class of bounding functions is in essence a question of applicability of the automation, taking into account particular use-cases. With respect to approximate solutions, we observe that the actual constraints C considered have at most one quantifier alternation and admit a quantifier prenex of the form $\exists^*\forall^*$, that is, a sequence of existential quantifier follows by a sequence of universal quantifiers. Roughly speaking the existential quantifiers refer to the inference of coefficients in the bounding polynomials, while the universal quantifiers refer to program variables. It is well-known that universal quantification in optimization problems can be turned into existential quantification, like Farka's lemma or generalizations thereof, cf. [38,19]. (E.g., [7,29] for instances of this approach for the inference of expected program costs.)

Summarizing, the inference mechanism detailed in Section 5 can be over-approximated to generate purely existential constraints. The latter can be effectively solved via SMT. We expect that (full) automation of the inference mechanism can captialize on these ideas. Working out the details and in particular implementation of an effective prototype is subject to future work.

6 Conclusion and Future Work

We have studied the complexity and inference of techniques for obtaining qualitative program properties. One particular property of interest would be the cost of quantum programs, that is average time, average number of gates, mean value of a variable, etc. We show that these problems were undecidable in general by placing them in the arithmetic hierarchy and saw that inference could become decidable on a restricted fragment: quantum gates in Clifford+T and a function space with a decidable theory (polynomials of bounded degree over a real closed field). Further, we sketch how the latter can be transformed into an efficient synthesis method.

Many open questions remain. The studied notion of expectation transformer describes *local* properties of the quantum state, while it would be interesting to extend this technique to the *global* state so as to study a mixed state in a quantum-only setting (without classical variables and stores). Another question of interest is to what extent a characterization of the quantum class ZBQP, the class of problems computed by quantum programs in polynomial expected runtime, could be obtained using this tool.

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

1. Aaronson, S., Gottesman, D.: Improved simulation of stabilizer circuits. Physical Review A **70**(5), 052328 (2004)
2. Altenkirch, T., Grattage, J.: A functional quantum programming language. In: 20th IEEE Symposium on Logic in Computer Science (LICS 2005), 26-29 June 2005, Chicago, IL, USA, Proceedings. pp. 249–258. IEEE Computer Society (2005). https://doi.org/10.1109/LICS.2005.1
3. Arute, F., Arya, K., et al.: Quantum supremacy using a programmable superconducting processor. Nature **574**, 505–510 (2019). https://doi.org/10.1038/s41586-019-1666-5
4. Avanzini, M., Barthe, G., Lago, U.D.: On continuation-passing transformations and expected cost analysis. Proc. of the ACM on Programming Languages **5**(ICFP), 1–30 (2021). https://doi.org/10.1145/3473592
5. Avanzini, M., Lago, U.D., Yamada, A.: On probabilistic term rewriting. Science of Computer Programming **185** (2020). https://doi.org/10.1016/j.scico.2019.102338
6. Avanzini, M., Moser, G., Péchoux, R., Perdrix, S., Zamdzhiev, V.: 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 (2022). https://doi.org/10.1145/3531130.3533332
7. Avanzini, M., Moser, G., Schaper, M.: A modular cost analysis for probabilistic programs. Proc. of the ACM on Programming Languages **4**(OOPSLA), 172:1–172:30 (2020). https://doi.org/10.1145/3428240
8. Avanzini, M., Moser, G., Schaper, M.: Automated expected value analysis of recursive programs. Proceedings of the ACM on Programming Languages **7**(PLDI), 1050–1072 (2023). https://doi.org/10.1145/3591263
9. Bournez, O., Garnier, F.: Proving Positive Almost-Sure Termination. In: Proc. of RTA 2005. LNCS, vol. 3467, pp. 323–337. Springer (2005). https://doi.org/10.1142/S0129054112400588
10. Bravyi, S., Kitaev, A.: Universal quantum computation with ideal clifford gates and noisy ancillas. Physical Review A **71**(2), 022316 (2005). https://doi.org/10.1103/PhysRevA.71.022316
11. Chen, Y.F., Chung, K.M., Lengál, O., Lin, J.A., Tsai, W.L., Yen, D.D.: An automata-based framework for verification and bug hunting in quantum circuits. Proceedings of the ACM on Programming Languages **7**(PLDI), 1218–1243 (2023). https://doi.org/10.1145/3591270
12. Cohn, P.M.: Further algebra and applications. Springer Science & Business Media (2002)
13. Dijkstra, E.W.: A discipline of programming. Prentice-Hall Englewood Cliffs (1976)
14. Endrullis, J., Geuvers, H., Zantema, H.: Degrees of undecidability in term rewriting. In: Grädel, E., Kahle, R. (eds.) Computer Science Logic, 23rd international Workshop, CSL 2009, 18th Annual Conference of the EACSL, Coimbra, Portugal,

September 7-11, 2009. Proceedings. Lecture Notes in Computer Science, vol. 5771, pp. 255–270. Springer (2009). https://doi.org/10.1007/978-3-642-04027-6_20
15. Fu, P., Kishida, K., Ross, N.J., Selinger, P.: Proto-quipper with dynamic lifting. POPL **7**, 309–334 (2023)
16. Gosset, D., Kliuchnikov, V., Mosca, M., Russo, V.: An algorithm for the t-count. Quantum Information & Computation **14**(15-16), 1261–1276 (2014). https://doi.org/10.26421/QIC14.15-16-1
17. Gretz, F., Katoen, J.P., McIver, A.: Operational versus weakest pre-expectation semantics for the probabilistic guarded command language. Performance Evaluation **73**, 110–132 (2014). https://doi.org/10.1016/j.peva.2013.11.004
18. Halava, V., Harju, T., Hirvensalo, M., Karhumäki, J.: Skolem's problem - on the border between decidability and undecidability. Tech. Rep. 683, Turku Center for Computer Science (2005)
19. Handelman, D.: Representing Polynomials by Positive Linear Functions on Compact Convex Polyhedra. PJM **132**(1), 35–62 (1988). https://doi.org/http://dx.doi.org/10.2140/pjm.1988.132.35
20. Harrow, A.W., Hassidim, A., Lloyd, S.: Quantum algorithm for linear systems of equations. Physical Review Letters **103**, 150502 (2009). https://doi.org/10.1103/PhysRevLett.103.150502
21. Heintz, J., Roy, M.F., Solernó, P.: Sur la complexité du principe de tarski-seidenberg. Bulletin de la Société mathématique de France **118**(1), 101–126 (1990)
22. Hillmich, S., Zulehner, A., Kueng, Richardand Markov, I.L., Wille, R.: Approximating decision diagrams for quantum circuit simulation. ACM Trans. Quantum Comput. **3**(4), 1–21 (2022)
23. Jia, X., Kornell, A., Lindenhovius, B., Mislove, M.W., Zamdzhiev, V.: Semantics for variational quantum programming. Proc. ACM Program. Lang. **6**(POPL), 1–31 (2022). https://doi.org/10.1145/3498687
24. Kaminski, B.L., Katoen, J.P.: On the hardness of almost-sure termination. In: MFCS 2015, Part I. pp. 307–318. LNCS, Springer (2015). https://doi.org/10.1007/978-3-662-48057-1_24
25. Kaminski, B.L.: Advanced weakest precondition calculi for probabilistic programs. Ph.D. thesis, RWTH Aachen University, Germany (2019), http://publications.rwth-aachen.de/record/755408
26. Kaminski, B.L., Katoen, J.P.: A weakest pre-expectation semantics for mixed-sign expectations. In: LICS 2017. pp. 1–12. IEEE (2017). https://doi.org/10.1109/LICS.2017.8005153
27. Kaminski, B.L., Katoen, J., Matheja, C., Olmedo, F.: Weakest precondition reasoning for expected run-times of probabilistic programs. In: ESOP 2016. LNCS, vol. 9632, pp. 364–389. Springer (2016). https://doi.org/10.1007/978-3-662-49498-1_15
28. Kozen, D.: A probabilistic PDL. Journal of Computer and System Sciences **30**(2), 162–178 (1985)
29. Leutgeb, L., Moser, G., Zuleger, F.: Automated Expected Amortised Cost Analysis of Probabilistic Data Structures. In: Proc. of 34th CAV. LNCS, vol. 13372, pp. 70–91 (2022). https://doi.org/10.1007/978-3-031-13188-2_4
30. Liu, J., Zhou, L., Barthe, G., Ying, M.: Quantum weakest preconditions for reasoning about expected runtimes of quantum programs (2022)
31. Malherbe, O., Díaz-Caro, A.: Quantum control in the unitary sphere: Lambda-s1 and its categorical model. Logical Methods in Computer Science **18**(3) (2022)
32. McIver, A., Morgan, C.: Abstraction, refinement and proof for probabilistic systems. Springer Science & Business Media (2005)

33. Ngo, V.C., Carbonneaux, Q., Hoffmann, J.: Bounded expectations: resource analysis for probabilistic programs. ACM SIGPLAN Notices **53**(4), 496–512 (2018). https://doi.org/10.1145/3296979.3192394
34. Odifreddi, P.: Classical recursion theory: The theory of functions and sets of natural numbers. Elsevier (1992)
35. Olmedo, F., Díaz-Caro, A.: Runtime analysis of quantum programs: A formal approach. In: PLanQC 2020 (2020)
36. Perdrix, S.: Quantum entanglement analysis based on abstract interpretation. In: SAS. pp. 270–282. LNCS (2008). https://doi.org/10.1007/978-3-540-69166-2_18
37. Schnabl, A., Simonsen, J.G.: The exact hardness of deciding derivational and runtime complexity. In: Proc. 20th CSL. LIPIcs, vol. 12, pp. 481–495 (2011). https://doi.org/10.4230/LIPIcs.CSL.2011.481
38. Schrijver, A.: Theory of linear and integer programming. Wiley (1999)
39. Selinger, P.: Towards a quantum programming language. Mathematical Structures in Computer Science **14**(4), 527–586 (2004). https://doi.org/10.1017/S0960129504004256
40. Sharir, M., Pnueli, A., Hart, S.: Verification of probabilistic programs. SIAM Journal on Computing **13**(2), 292–314 (1984)
41. Shor, P.W.: Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. SIAM Review **41**(2), 303–332 (1999). https://doi.org/10.1137/S0036144598347011
42. Weihrauch, K.: Computable analysis: an introduction. Springer Science & Business Media (2012)
43. Wille, R., Van Meter, R., Naveh, Y.: IBM's Qiskit Tool Chain: Working with and Developing for Real Quantum Computers. In: 2019 Design, Automation & Test in Europe Conference & Exhibition (DATE). pp. 1234–1240 (2019). https://doi.org/10.23919/DATE.2019.8715261
44. Winskel, G.: The formal semantics of programming languages - an introduction. Foundation of computing series, MIT Press (1993)