Quantum Term Rewrite Systems: Applications to Complexity Analysis

Kostia Chardonnet
Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France

Emmanuel Hainry
Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France

Romain Péchoux
Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France

Thomas Vinet
Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France

Abstract

Term Rewrite Systems (TRS) is a computational model offering a level of abstraction well-suited towards static analysis, e.g., termination or complexity analyses. In this paper, we introduce Quantum Term Rewrite Systems (QTRS), an extension of TRS to quantum computing, thus allowing to benefit from quantum advantage while being able to certify the complexity. We ensure that QTRS correspond to physically realizable processes and adapt techniques to obtain termination certificates or generic bounds on the reduction length. We delineate a class of terminating QTRS that can be compiled to uniform families of quantum circuits of size bounded by the reduction length. Conversely, this class is universal for quantum circuits. In particular, we show a characterization of the class of functions computable in quantum polynomial time, known as `FBQP`.

2012 ACM Subject Classification Theory of computation → Rewrite systems; Theory of computation → Quantum complexity theory

Keywords and phrases Term Rewrite Systems, Quantum Computing, Resource Analysis

1 Introduction

Motivations. The past few decades saw the emergence of quantum computation, a paradigm in which problems can be solved more efficiently compared to their classical counterparts [57]. This paradigm has been studied through multiple computational models, e.g., quantum circuits [64, 53], linear optics [43], and ZX-calculus [19]. Recently, a particular attention has been paid to the development of higher-level models, namely quantum programming languages. Quantum programs can be classified in two categories, depending on their control flow. A first approach, known as *classical control*, considered that only classical data, e.g., the outcome of a measurement, could influence the control flow [56, 33, 22, 29]. On the other hand, *quantum control* allows applying controlled operations depending on quantum data, yielding superpositions of programs. Quantum control gained interest as it provides a computational advantage over classical control [1, 59, 46], and has been approached through various languages [55, 27, 66, 9]. Thus, a natural issue is to develop and study *hybrid languages*, i.e., languages that feature both classical and quantum control flow/data [61, 65, 24, 18, 17].

The clear theoretical advantage of quantum computation is however mitigated by the hardness of hardware implementation. In particular, each type of hardware has its own set of constraints, and quantum gates may be more or less costly to implement depending on the hardware. Thus, providing a static analysis of the (low-level) resources used by a program (e.g., the size, depth, or gate count of its corresponding quantum circuit) is a relevant issue, and has been studied in [23, 6, 35, 20, 30]. These works fall within the field of *Implicit Computational Complexity* (ICC), by providing a language that characterizes exactly

a well-known complexity class. However, these languages are targeted towards a specific complexity class through typing or syntactic restrictions, and are thus not a good match for a generic complexity analysis.

With regard to classical programs, Term Rewriting Systems (TRS) [7] are a computational model that is sufficiently abstract and simple to allow for automatic and semi-automatic formal analyses of programs. In particular, numerous generic techniques exist for analyzing their termination (e.g., [2, 3]) as well as their runtime and complexity, (e.g., [13, 5]), making them well-suited for studying ICC-related properties. TRS have also proven to be a robust computational model that allows for simple and natural extensions to other paradigms, such as probabilistic rewriting [16, 3] or higher-order rewriting [50, 62].

The development of an extension to the quantum setting is still missing, which would allow to reuse existing techniques, to certify properties on the quantum resources.

Contributions. This paper fills the gap by introducing the notion of *quantum term rewrite systems* (QTRS). One advantage of QTRS, in contrast with aforementioned resource-aware languages, is that QTRS feature quantum control, and have no fixed initial set of quantum gates, making them prone to a general resource analysis. Furthermore, for an expressive fragment, QTRS correspond to quantum circuits, and their size can be related with the runtime-complexity of the QTRS. This paves the way for the reuse of any complexity technique on standard TRS. Our work contains the following contributions:

- We introduce QTRS as an extension of TRS, where the semantics achieves quantum parallelism [53], and the type system ensures the physicality of the terms. Standard properties are also proven, such as confluence (Lemma 10), subject reduction (Lemma 11), and a characterization of the normal forms (Lemma 14).
- We show that type inference is Π_2^0-hard, thus undecidable in the general case (Theorem 15). However, decidability can be recovered for an expressive subset of terms, on which type inference is decidable in polynomial time (Theorem 16).
- We exhibit a fragment of terminating QTRS, which is universal for quantum circuits (Theorem 21). In this fragment, QTRS can be compiled into uniform families of quantum circuits (Theorem 22). Furthermore, for QTRS with a specific recursive structure, the size of the circuit can be bounded by the runtime-complexity of the QTRS (Theorem 25). In particular, compilable QTRS terminating in polynomial time characterize exactly the class `FBQP` of functions computable in quantum polynomial time (Theorem 27).
- We study how complexity and termination properties can be inferred on QTRS by extending any ordering on standard TRS to the QTRS setting (Section 4.1). We show that existing techniques — e.g., polynomial interpretations [47] (Theorem 32), and dependency pairs [2] (Theorem 37) — can be adapted and reused on QTRS to guarantee termination and complexity properties of a QTRS.

Related work. Quantum complexity properties have already been studied for various languages with classical control: [23] characterizes (polynomial time) complexity classes on a variant of quantum lambda calculus; [6] adapts expectation transformers to a quantum imperative language to infer expected costs and values and has been implemented in [51]; [20] develops a dependent type system on a variant of the Quipper circuit description language [33].

With regard to quantum control, `FOQ` [35] is a hybrid imperative language characterizing functions terminating in quantum polynomial time, and has been refined in [30] to obtain a characterization of quantum polylogarithmic time. However, only qubit lists are featured,

while QTRS can express all standard inductive datatypes. Furthermore, FOQ does not feature general recursion, which is necessary to achieve a fully generic resource analysis.

TRS have been widely studied in order to guarantee ICC-related properties. Here we provide a non-exhaustive overview. One panel of studies is obtained through *interpretations*, yielding characterization of polynomial complexity [12]. *Quasi-interpretations* relax strict monotonicity conditions by incorporating path orderings [25], yielding polynomial time and space characterization [13]. Another set of techniques reside in orderings, which yield polynomial termination via light multiset path order [49] or polynomial path orders [5], while polynomial space can be obtained via Knuth-Bendix orders [14]. *Dependency pairs* [2], originally a tool for studying termination, have also been refined to analyze the runtime complexity of the TRS [39, 4, 54].

All the aforementioned techniques have also been extended to other paradigms. For example, termination has been studied for probabilistic TRS [15, 41], conditional TRS [48], and higher-order TRS [52]. Similarly, runtime complexity analysis has been performed on conditional [44] and higher-order TRS [45], and for TRS with a parallel reduction strategy [10]. In particular, characterizations of basic feasible functions [21] have been obtained for higher-order TRS [37, 8]. While these complexity results are often aimed at achieving upper bounds, some work has also been carried to obtain lower bounds [32] or to ensure constant runtime [31]. We believe that the very general concept of a quantum TRS presented in this paper can be adapted to these various paradigms.

2 Quantum Term Rewrite Systems

This section introduces *Superposed TRS* (STRS) as an extension of standard TRS [7] that features superpositions. The semantics of STRS is defined in term of a call-by-basis reduction strategy [26] mimicking quantum parallelism [53]. *Quantum TRS* are then introduced as a typed restriction ensuring physicality.

2.1 Syntax

Let $\mathcal{X}$ be a countable set of variables $\mathtt{x}, \mathtt{y}, \ldots$, and let $\mathcal{A}$, $\mathcal{C}$, $\mathcal{F}$ be three signatures containing *amplitude symbols* $\mathtt{a}$, *constructor symbols* $\mathtt{c}$, and *function symbols* $\mathtt{f}$, respectively. Recall that a *signature* is a set of symbols $\mathtt{b}$, together with a fixed *arity* $\mathtt{ar}(\mathtt{b}) \in \mathbb{N}$. The four sets are assumed to be disjoint.

First-order TRS are extended by allowing *term superposition*, that is, having a binary sum for superpositions ($+$) and multiplication by an amplitude ($\alpha\cdot$). This is formalized by the following grammar.

$$
\begin{array}{rrcl}
\text{(Natural numbers)} & \iota & ::= & \mathtt{x} \mid \mathtt{0} \mid \mathtt{S}(\iota) \\
\text{(Amplitudes)} & \alpha & ::= & \mathtt{a}(\iota, \ldots, \iota) \mid \alpha \cdot \alpha \mid \alpha + \alpha \\
\text{(Terms)} & \mathcal{T} \ni t & ::= & \mathtt{x} \mid \mathtt{c}(t, \ldots, t) \mid \mathtt{f}(t, \ldots, t) \mid \alpha \cdot t \mid t + t \\
\text{(Values)} & \mathcal{V} \ni v & ::= & \mathtt{c}(v, \ldots, v) \mid \underline{\alpha} \cdot v \mid v + v
\end{array}
$$

The distinction between constructor and function symbols is akin to *constructor TRS*, i.e., function symbols in $\mathcal{F}$ will be evaluated through rewrite rules, while constructor symbols in $\mathcal{C}$ are constants. In the following, we consider that $\mathcal{C}$ contains a unit constructor $()$; qubit constructor symbols $|0\rangle, |1\rangle$; constructors for natural numbers $\mathtt{0}$ and $\mathtt{S}$; a pair constructor $\mathtt{pr}$ of arity 2, written $(a, b) \triangleq \mathtt{pr}(a, b)$; as well as list constructors, $[\,]$ of arity 0, and $\mathtt{cons}$ of arity 2, written $h :: t \triangleq \mathtt{cons}(h, t)$.

Amplitude symbols of arity 0 correspond to fixed scalars. In what follows, $\mathtt{a}_\lambda$ will be the arity-0 amplitude symbol encoding the scalar $\lambda \in \mathbb{C}$. Amplitude symbols of greater arity will be used to express complex programs (see Example 3), where the amplitudes may vary.

Let $\mathcal{V}ar(r)$ be the set of variables occurring in any term or amplitude r. If $\mathcal{V}ar(r) = \emptyset$, r is said to be *ground*, and written $\underline{r}$. Let $\mathcal{T}_0$ denote the set of ground terms. A term is called *classical* if it has neither superposition, nor amplitude; denote $\mathcal{C}$ as the set of classical terms. Remark that a classical term can be viewed as a term of a standard TRS. The *size* of a term t, denoted $|t|$, is the number of occurrences of variables and symbols inside t.

A $\mathtt{f}$-*rewrite rule*, also called simply *rewrite rule*, is a pair $l \to r$, with $l, r \in \mathcal{T}$, such that $l = \mathtt{f}(p_1, \ldots, p_n)$, where p_i is a classical term with no function symbol, called *pattern*, and $\mathcal{V}ar(l) \supseteq \mathcal{V}ar(r)$. A set of rewrite rules is said to be *orthogonal*, if the rewrite rules are *left-linear*, i.e., each variable occurs at most once in the left-hand side of each rule, and *non-overlapping*, i.e., no pair of left-hand side of rules can overlap.

▶ **Definition 1.** *A* Superposed Term Rewrite System (STRS) $\mathfrak{R}$ *is a tuple* $\langle \mathcal{A}, \mathcal{C}, \mathcal{F}, \mathcal{X}, \mathcal{R} \rangle_{\mathtt{f}}$*, where* $\mathcal{R}$ *is a set of orthogonal rewrite rules* $l \to r$ *with* $l, r \in \mathcal{T}$*, and* $\mathtt{f} \in \mathcal{F}$ *is called the* main function symbol*. The set of rules of* $\mathfrak{R}$ *will be denoted by* $\mathcal{R}_{\mathfrak{R}} \triangleq \mathcal{R}$*.*

▶ **Example 2.** Quantum gates can be written as a STRS, by defining one rule for each action on a basis element. For example, define the STRS $\mathfrak{R}$ containing the rewrite rules $\mathcal{R}$ below. $\mathfrak{R}$ expresses the Clifford + T set of gates, which is known to be *universal* [53], i.e., any quantum gate on n qubits can be approximated by a sequence of gates from this set.

$$\mathcal{R} = \left\{ \begin{array}{ll} \mathtt{X}(|0\rangle) \to |1\rangle & \mathtt{X}(|1\rangle) \to |0\rangle \\ \mathtt{T}(|0\rangle) \to |0\rangle & \mathtt{T}(|1\rangle) \to \mathtt{a}_{e^{i\pi/4}} \cdot |1\rangle \\ \mathtt{H}(|0\rangle) \to \mathtt{a}_{\frac{1}{\sqrt{2}}} \cdot |0\rangle + \mathtt{a}_{\frac{1}{\sqrt{2}}} \cdot |1\rangle & \mathtt{H}(|1\rangle) \to \mathtt{a}_{\frac{1}{\sqrt{2}}} \cdot |0\rangle + \mathtt{a}_{-1} \cdot \mathtt{a}_{\frac{1}{\sqrt{2}}} \cdot |1\rangle \\ \mathtt{CNOT}(|0\rangle, \mathtt{q}) \to (|0\rangle, \mathtt{q}) & \mathtt{CNOT}(|1\rangle, \mathtt{q}) \to (|1\rangle, \mathtt{X}(\mathtt{q})) \end{array} \right\}$$

▶ **Example 3.** One can encode the Quantum Fourier Transform as a STRS with main function symbol $\mathtt{qft}$ and the following set of rules, where $\mathtt{a}(n)$ corresponds to the scalar $e^{2i\pi/2^n}$, and $(t_1, \ldots, t_n)$ corresponds to $n-1$ pair constructors. The second argument of the symbol $\mathtt{rec}$ allows to define the behavior of $\mathtt{rec}$ without an additional function symbol.

$$\left\{ \begin{array}{ll} \mathtt{inv}([\,], l) \to l & \mathtt{inv}(h :: t, l) \to \mathtt{inv}(t, h :: l) \\ \mathtt{phase}(|0\rangle, n) \to |0\rangle & \mathtt{phase}(|1\rangle, n) \to \mathtt{a}(n) \cdot |1\rangle \\ \mathtt{ctrl}((q, |0\rangle, t, l), n) \to (q, t, |0\rangle :: l) & \mathtt{ctrl}((q, |1\rangle, t, l), n) \to (\mathtt{phase}(q, n), t, |1\rangle :: l) \\ \mathtt{rot}((q, [\,], l), n) \to q :: \mathtt{inv}(l, [\,]) & \mathtt{rot}((q, h :: t, l), n) \to \mathtt{rot}(\mathtt{ctrl}((q, h, t, l), n), \mathtt{S}(n)) \\ \mathtt{rec}([\,], b) \to [\,] & \mathtt{rec}(h :: t, \mathtt{0}) \to \mathtt{rec}(\mathtt{rot}((\mathtt{Had}(h), t, [\,]), \mathtt{S}(\mathtt{S}(\mathtt{0}))), \mathtt{S}(\mathtt{0})) \\ \mathtt{rec}(h :: t, \mathtt{S}(b)) \to h :: \mathtt{rec}(t, b) & \mathtt{qft}(l) \to \mathtt{inv}(\mathtt{rec}(l, \mathtt{0}), [\,]) \end{array} \right\}$$

2.2 Semantics

Before introducing the semantics of a STRS, several steps must be performed. First, the Hilbert structure of quantum spaces requires us to consider terms modulo an *equivalence relation* on the vector space generated by term superpositions. to ensure reductions only consider meaningful terms. Third, a specific *evaluation strategy* will be fixed for classical terms.

Vector space structure. Defining a vector space structure requires checking properties on amplitudes, e.g., if an amplitude corresponds semantically to 0. To that end, each amplitude

$$\frac{}{t_1 + t_2 \equiv t_2 + t_1} \qquad \frac{}{t_1 + (t_2 + t_3) \equiv (t_1 + t_2) + t_3} \qquad \frac{[\![\alpha]\!] = 0}{s + \alpha \cdot t \equiv s} \qquad \frac{[\![\alpha]\!] = 1}{\alpha \cdot t \equiv t}$$

$$\frac{}{\alpha \cdot (\beta \cdot t) \equiv (\alpha \cdot \beta) \cdot t} \qquad \frac{}{\alpha \cdot (t_1 + t_2) \equiv \alpha \cdot t_1 + \alpha \cdot t_2} \qquad \frac{}{\alpha \cdot t + \beta \cdot t \equiv (\alpha + \beta) \cdot t}$$

$$\frac{}{\mathtt{b}(\dots, \alpha \cdot t, \dots) \equiv \alpha \cdot \mathtt{b}(\dots, t, \dots)} \qquad \frac{}{\mathtt{b}(\dots, t_1 + t_2, \dots) \equiv \mathtt{b}(\dots, t_1, \dots) + \mathtt{b}(\dots, t_2, \dots)}$$

$$\frac{s \equiv t}{C[s] \equiv C[t]}$$

Figure 1 Equivalence relation $\equiv$

symbol $\mathtt{a} \in \mathcal{A}$ comes with a total function $[\![\mathtt{a}]\!] : \mathbb{N}^{\mathtt{ar}(\mathtt{a})} \to \mathbb{C}$ called *interpretation*. The interpretation is then defined on ground amplitudes by canonical extension:

$$[\![\mathtt{0}]\!] \triangleq 0 \qquad [\![\mathtt{S}(\underline{\iota})]\!] \triangleq [\![\underline{\iota}]\!] + 1 \qquad [\![\mathtt{a}(\underline{\iota_1}, \dots, \underline{\iota_n})]\!] \triangleq [\![\mathtt{a}]\!]([\![\underline{\iota_1}]\!], \dots, [\![\underline{\iota_n}]\!])$$
$$[\![\underline{\alpha_1} + \underline{\alpha_2}]\!] \triangleq [\![\underline{\alpha_1}]\!] + [\![\underline{\alpha_2}]\!] \quad [\![\underline{\alpha_1} \cdot \underline{\alpha_2}]\!] \triangleq [\![\underline{\alpha_1}]\!] \times [\![\underline{\alpha_2}]\!]$$

Recall that a substitution is a map $\sigma : \mathcal{X} \to \mathcal{V}$. Let $r\sigma$ be the application of the substitution σ to the term or amplitude r. The interpretation can be extended to arbitrary amplitudes as follows: define $[\![\alpha]\!] \triangleq \lambda$ if for any substitution σ, if $\alpha\sigma$ is a ground amplitude, then $[\![\alpha\sigma]\!] = \lambda$. Two amplitudes α, β are *equivalent* if $[\![\alpha]\!] = [\![\beta]\!]$.

A (1-hole) *context* is a term C containing exactly one occurrence of the special symbol $\diamond$ and defined by the following grammar.

$$\text{(Contexts)} \qquad C ::= \diamond \mid \mathtt{c}(t_1, \dots, C, \dots, t_n) \mid \mathtt{f}(t_1, \dots, C, \dots, t_n) \mid \alpha \cdot C \mid C + t$$

Let $C[t]$ be the term obtained by substituting t to $\diamond$ in C. This notion can be generalized to *n-hole contexts* $C[\diamond_1, \dots, \diamond_n]$, which are terms containing one occurrence of each symbol $\diamond_1, \dots, \diamond_n$. A *classical context* is a context with no amplitude ($\alpha\cdot$), nor superposition ($+$).

Term equivalence $\equiv$ is defined as the smallest equivalence relation over $\mathcal{T}$ obtained from the rules of Figure 1. The two first lines of Figure 1 express the rules of a vector space structure and allow us to define unambiguously general summation as $\sum_{i=1}^{n} \alpha_i \cdot t_i \triangleq \alpha_1 \cdot t_1 + (\cdots + \alpha_n \cdot t_n)$. The remaining lines of Figure 1 highlight the linearity of quantum programs.

Canonical forms. Quantum parallelism will be defined using a restricted form of superposition to avoid non-physical reductions, e.g., reducing a term of amplitude 0. This is formalized below by the notion of *canonical form*.

▶ **Definition 4.** *A* canonical form *of a term t is any term $\sum_{i=1}^{n} \alpha_i \cdot t_i$ such that $t \equiv \sum_{i=1}^{n} \alpha_i \cdot t_i$, where t_i are pairwise distinct classical terms and $[\![\alpha_i]\!] \neq 0$. The set of canonical forms is denoted by* $\mathtt{CAN}$.

Evaluation strategy. Finally, the reduction of classical terms will be defined by fixing a given evaluation strategy. This is performed with the help of *evaluation contexts*.

$$\text{(Eval. contexts)} \quad E ::= \diamond \mid \mathtt{c}(v_1, \dots, v_{i-1}, E, t_{i+1}, \dots, t_n) \mid \mathtt{f}(v_1, \dots, v_{i-1}, E, t_{i+1}, \dots, t_n)$$

The semantics of a STRS $\mathfrak{R}$ is then defined inductively in Figure 2 as the relation $\to_{\mathfrak{R}}$ on ground terms. In Rule (Q), $\to_{\mathfrak{R}}$ relies on the intermediate relation $\to^{?}_{\mathfrak{R}}$, defined by

$$s \to^{?}_{\mathfrak{R}} t \iff (s \in \mathtt{NF}_{\mathfrak{R}} \wedge t = s) \vee s \to_{\mathfrak{R}} t$$

$$\frac{s \equiv s' \quad s' \to_{\mathfrak{R}} t' \quad t' \equiv t}{s \to_{\mathfrak{R}} t}\ (\mathrm{E}) \qquad \frac{\sum_{i=1}^{n} \alpha_i \cdot s_i \in \mathtt{CAN} \quad \forall i,\ s_i \to_{\mathfrak{R}}^{?} t_i \quad \exists i,\ s_i \to_{\mathfrak{R}} t_i}{\sum_{i=1}^{n} \alpha_i \cdot s_i \to_{\mathfrak{R}} \sum_{i=1}^{n} \alpha_i \cdot t_i}\ (\mathrm{Q})$$

$$\frac{l \to r \in \mathcal{R}_{\mathfrak{R}} \quad E[l\sigma] \in \mathcal{C}}{E[l\sigma] \to_{\mathfrak{R}} E[r\sigma]}\ (\mathrm{C})$$

Figure 2 Inference rules of the relation $\to_{\mathfrak{R}} \subseteq \mathcal{T}_0 \times \mathcal{T}_0$

where $\mathtt{NF}_{\mathfrak{R}} \triangleq \{s \in \mathcal{T}_0 \mid \nexists t,\ s \to_{\mathfrak{R}} t\}$ is the set of *normal forms*.

We will now provide some insights into the rules of Figure 2. Rule (E) allows one to perform reduction with respect to the equivalence relation on the vector space. Rule (Q) reduces superpositions only if expressed as canonical forms, ensuring that only meaningful reductions are fired. Furthermore, $\to_{\mathfrak{R}}^{?}$ implies that any term that is able to reduce will reduce. Therefore, $\to_{\mathfrak{R}}$ achieves quantum parallelism [53], similarly to [34]. An alternative reduction strategy not implementing parallelism would correspond to classical simulation, leading any program acting on n qubits to be evaluated in 2^n reduction steps. Finally, Rule (C) considers classical terms: variables are substituted by classical values, thus $\to_{\mathfrak{R}}$ follows a *call-by-basis* strategy [26]. The definition of evaluation contexts also imposes an evaluation of terms from left to right. While such choice is arbitrary, having a fixed strategy is required to obtain complexity results in Section 4.

Given a STRS $\mathfrak{R}$, define $\to_{\mathfrak{R}}^{*}$ as the reflexive and transitive closure of $\to_{\mathfrak{R}}$. We say that t *terminates in at most* k *steps*, and write $t \to_{\mathfrak{R}}^{\leq k} s$, if all chains of reduction starting from t reach a normal form in at most k steps. A term *terminates* if it terminates in at most k steps for some $k \in \mathbb{N}$. A STRS $\mathfrak{R}$ is said to be *terminating*, if any term $t \in \mathcal{T}_0$ terminates.

▶ **Example 5.** Define the unitary matrix Q_n, representing a single qubit gate, for any $n \in \mathbb{N}$, along the rewrite rules of its corresponding STRS:

$$Q_n \triangleq \frac{1}{\sqrt{2}} \cdot \begin{pmatrix} 1 & e^{i\pi/n} \\ e^{-i\pi/n} & -1 \end{pmatrix} \qquad \mathcal{R} \triangleq \left\{ \begin{matrix} \mathtt{f}(n, |0\rangle) \to \mathtt{sgn}^{+} \cdot |0\rangle + \mathtt{a}^{+}(n) \cdot |1\rangle \\ \mathtt{f}(n, |1\rangle) \to \mathtt{a}^{-}(n) \cdot |0\rangle + \mathtt{sgn}^{-} \cdot |1\rangle \end{matrix} \right\}$$

where $[\![\mathtt{sgn}^{\pm}]\!] \triangleq \pm\frac{1}{\sqrt{2}}$ and $[\![\mathtt{a}^{\pm}]\!](x) \triangleq \frac{1}{\sqrt{2}} e^{\pm i\pi/x}$. One can verify that applying twice $\mathtt{f}(n, \cdot)$ is equivalent to the identity, e.g., if applied to $|0\rangle$, for any ground natural number $\underline{\iota}$:

$$\begin{aligned} \mathtt{f}(\underline{\iota}, \mathtt{f}(\underline{\iota}, |0\rangle)) &\to_{\mathfrak{R}} \mathtt{f}(\underline{\iota}, \mathtt{sgn}^{+} \cdot |0\rangle + \mathtt{a}^{+}(\underline{\iota}) \cdot |1\rangle) \equiv \mathtt{sgn}^{+} \cdot \mathtt{f}(\underline{\iota}, |0\rangle) + \mathtt{a}^{+}(\underline{\iota}) \cdot \mathtt{f}(\underline{\iota}, |1\rangle) \\ &\to_{\mathfrak{R}} \left(\mathtt{sgn}^{+} \cdot \mathtt{sgn}^{+} + \mathtt{a}^{+}(\underline{\iota}) \cdot \mathtt{a}^{-}(\underline{\iota})\right) \cdot |0\rangle + \left(\mathtt{sgn}^{+} \cdot \mathtt{a}^{+}(\underline{\iota}) + \mathtt{a}^{+}(\underline{\iota}) \cdot \mathtt{sgn}^{-}\right) |1\rangle \\ &\equiv |0\rangle \end{aligned}$$

The first and second reductions respectively used the rules (C) and (Q) from Figure 2. The last equivalence can be obtained by seeing the reduced term as $\alpha \cdot |0\rangle + \beta \cdot |1\rangle$, and checking that $[\![\alpha]\!] = 1$ and $[\![\beta]\!] = 0$, e.g., see below for $[\![\alpha]\!] = 1$:

$$[\![\alpha]\!] = [\![\mathtt{sgn}^{+} \cdot \mathtt{sgn}^{+} + \mathtt{a}^{+}(\underline{\iota}) \cdot \mathtt{a}^{-}(\underline{\iota})]\!] \quad = [\![\mathtt{sgn}^{+}]\!] \times [\![\mathtt{sgn}^{+}]\!] + [\![\mathtt{a}^{+}]\!](\underline{\iota}) \times [\![\mathtt{a}^{-}]\!](\underline{\iota}) = 1$$

Therefore, using rule (E) from Figure 2, $\mathtt{f}(\underline{\iota}, \mathtt{f}(\underline{\iota}, |0\rangle))$ rewrites in two steps to $|0\rangle$.

2.3 Type System for Physicality

STRS extend standard TRS by allowing any general term superposition. However, quantum programs need to satisfy additional properties to be physically sound, e.g., linear use of data

$$\frac{\forall i,\ \Gamma;\varnothing \vdash_{\mathfrak{R}} \iota_i : \mathtt{nat} \qquad \mathtt{a} \in \mathcal{A} \qquad \mathtt{ar}(\mathtt{a}) = n}{\Gamma \Vdash_{\mathfrak{R}} \mathtt{a}(\iota_1, \dots, \iota_n)} \qquad \frac{\Gamma \Vdash_{\mathfrak{R}} \alpha_1 \qquad \Gamma \Vdash_{\mathfrak{R}} \alpha_2}{\Gamma \Vdash_{\mathfrak{R}} \alpha_1 + \alpha_2} \qquad \frac{\Gamma \Vdash_{\mathfrak{R}} \alpha_1 \qquad \Gamma \Vdash_{\mathfrak{R}} \alpha_2}{\Gamma \Vdash_{\mathfrak{R}} \alpha_1 \cdot \alpha_2}$$

$$\frac{x \in \mathcal{X}}{\Gamma, x : C; \varnothing \vdash_{\mathfrak{R}} x : C} \qquad \frac{x \in \mathcal{X}}{\Gamma; x : Q \vdash_{\mathfrak{R}} x : Q} \qquad \frac{\Gamma;\Delta \vdash_{\mathfrak{R}} s : T \qquad s \equiv t \qquad \mathtt{FV}(t) \subseteq \mathrm{Dom}(\Gamma \cup \Delta)}{\Gamma;\Delta \vdash_{\mathfrak{R}} t : T}$$

$$\frac{\mathrm{type}(\mathtt{b}) = T_1 \times \cdots \times T_n \to T \qquad \forall i,\ \Gamma;\Delta_i \vdash_{\mathfrak{R}} t_i : T_i \qquad \mathtt{b} \in \mathcal{C} \uplus \mathcal{F}}{\Gamma;\Delta_1, \dots, \Delta_n \vdash_{\mathfrak{R}} \mathtt{b}(t_1, \dots, t_n) : T}$$

$$\frac{\forall i,\ \Gamma;\Delta \vdash_{\mathfrak{R}} t_i : Q \qquad \forall i,\ \Gamma \Vdash_{\mathfrak{R}} \alpha_i \qquad \forall \sigma \in \mathrm{Sub}_0(\Gamma),\ \sum_{i=1}^n |[\![\alpha_i\sigma]\!]|^2 = 1 \qquad \forall i \neq j,\ t_i \perp_{\mathfrak{R}} t_j}{\Gamma;\Delta \vdash_{\mathfrak{R}} \sum_{i=1}^n \alpha_i \cdot t_i : Q}$$

Figure 3 Typing rules of terms and amplitudes

or norm preservation. This is tackled by introducing a type system, both on terms and amplitudes, allowing us to then define Quantum TRS (QTRS) as the STRS with a physical reality.

Towards that end, a unique *typed signature* $\mathrm{type}(\mathtt{b}) \triangleq T_1 \times \cdots \times T_{\mathtt{ar}(\mathtt{b})} \to T$ is assigned to each symbol $\mathtt{b} \in \mathcal{C} \uplus \mathcal{F}$, where $T, T_1, \dots, T_{\mathtt{ar}(\mathtt{b})}$ are taken from a countable set of basic types. The types $\mathbb{1}$ for unit, $\mathtt{Qbit}$ for qubits, or $\mathtt{nat}$ for natural numbers are examples of basic types. If $\mathtt{b}$ is of arity 0, its signature is written simply $\mathrm{type}(\mathtt{b}) = T$. The constructor symbols introduced in Section 2.1 can be equipped with the following signatures, for any basic type T, T_1, T_2:

$$\mathrm{type}(()) \triangleq \mathbb{1} \qquad \mathrm{type}(|0\rangle) = \mathrm{type}(|1\rangle) \triangleq \mathtt{Qbit}$$
$$\mathrm{type}(\mathtt{0}) \triangleq \mathtt{nat} \qquad \mathrm{type}(\mathtt{S}) \triangleq \mathtt{nat} \to \mathtt{nat}$$
$$\mathrm{type}([\,]_T) \triangleq \mathtt{List}(T) \qquad \mathrm{type}(\mathtt{cons}_{\mathtt{List}(T)}) \triangleq T \times \mathtt{List}(T) \to \mathtt{List}(T)$$
$$\mathrm{type}(\mathtt{pr}_{T_1,T_2}) \triangleq T_1 \times T_2 \to \times_{T_1,T_2}$$

Note that polymorphic types (e.g., lists or pairs) need to have multiple constructors defined (one for each possible input type). However, when clear from the context, the previous notations are used, e.g., $h :: t$ and (h, t). To ensure linearity of quantum data, the set of basic types is split disjointly between *quantum types* $Q \in \mathtt{Q}$ and *classical types* $C \notin \mathtt{Q}$. The set of quantum types $\mathtt{Q}$ is defined below:

$$\mathtt{Q} \triangleq \{\,\mathtt{Qbit}\,\} \cup \{\,T \mid \exists\, \mathtt{c} \in \mathcal{C}, \exists i \in \mathbb{N}, \mathrm{type}(\mathtt{c}) = T_1 \times \cdots \times T_n \to T \wedge T_i \in \mathtt{Q}\,\}$$

A quantum type is either the qubit type $\mathtt{Qbit}$, or any type having (at least) one constructor symbol with a quantum type in its signature.

Typing contexts Γ, Δ are sets of the shape $\{\, x_1 : T_1, \dots, x_n : T_n \,\}$, where x_i are pairwise distinct variables, and T_i are basic types. The *domain* of a context is defined as its set of variables, i.e., $\mathrm{Dom}(\Gamma) \triangleq \{x \mid x : T \in \Gamma\}$. Whenever we write Γ, Δ or $\Gamma;\Delta$, it is always assumed that Γ and Δ share no variables, i.e., $\mathrm{Dom}(\Gamma) \cap \mathrm{Dom}(\Delta) = \varnothing$. Given a typing context Θ, we say that a substitution σ is a Θ-*context substitution*, if for any $x : T \in \Theta$, $x\sigma$ is a well-typed value of type T. Let $\mathrm{Sub}_0(\Theta)$ be the set of Θ-context substitutions.

An *amplitude typing judgment* is written $\Gamma \Vdash_{\mathfrak{R}} \alpha$, indicating that α is a well-typed amplitude, with respect to $\mathfrak{R}$, under (non-linear) context Γ. A *term typing judgment* is

written $\Gamma;\Delta \vdash_{\mathfrak{R}} t : T$, indicating that t is well-typed with respect to $\mathfrak{R}$ and basic type T, under non-linear context Γ and linear context Δ; when $\Gamma = \Delta = \varnothing$, we may write $\vdash_{\mathfrak{R}} t : T$.

These judgments are derived inductively following the typing rules of Figure 3, for a given STRS $\mathfrak{R}$.

A term typing judgment aims to type normalized terms only. Towards that end, an *orthogonality predicate* on terms is introduced. It relies on the notion of *orthogonal Kronecker product*, which is a partial map on classical terms $\delta_\perp : \mathcal{C} \times \mathcal{C} \to \{0, 1\}$, where $\delta_\perp(t, t) \triangleq 1$, and $\delta_\perp(s, t) \triangleq 0$ if there exist a $n+1$-hole context C, $q \in \{0, 1\}$, terms $s_1, \ldots, s_n$ and $t_1, \ldots, t_n$ such that $s = C[|q\rangle, s_1, \ldots, s_n]$ and $t = C[|1-q\rangle, t_1, \ldots, t_n]$.

▶ **Definition 6** (Orthogonality). *Let $\mathfrak{R}$ be a STRS. Let $\Gamma;\Delta \vdash_{\mathfrak{R}} s : T$ and $\Gamma;\Delta \vdash_{\mathfrak{R}} t : T$ be two terms. We say that s and t are* orthogonal*, written $s \perp_{\mathfrak{R}} t$, if for any substitution $\sigma \in \mathrm{Sub}_0(\Gamma \cup \Delta)$, $s\sigma \to^*_{\mathfrak{R}} \sum_{i=1}^n \alpha_i \cdot v_i \in \mathtt{CAN}$, $t\sigma \to^*_{\mathfrak{R}} \sum_{j=1}^m \beta_j \cdot w_j \in \mathtt{CAN}$, and:*

$$\sum_{i=1}^{n}\sum_{j=1}^{m} [\![\alpha_i]\!][\![\beta_j]\!]^* \delta_\perp(v_i, w_j) = 0$$

where, given $\lambda \in \mathbb{C}$, λ^* is the complex conjugate of λ.

Note that the above definition requires $\delta_\perp$ to be defined for all (v_i, w_j). Typing a superposition requires the amplitudes to correspond semantically to a normalized complex vector, and that the normal forms of the terms are pairwise orthogonal.

Given a symbol $\mathtt{b} \in \mathcal{C} \biguplus \mathcal{F}$, define $\mathrm{In}(\mathtt{b})$ as the set of *well-typed inputs of* $\mathtt{b}$, i.e., $\overline{v} \triangleq (v_1, \ldots, v_n) \in \mathrm{In}(\mathtt{b})$ if $\mathrm{type}(\mathtt{b}) = T_1 \times \cdots \times T_n \to T$ and $\vdash_{\mathfrak{R}} v_i : T_i$, for all i.

Define $\overline{v} \perp_{\mathfrak{R}} \overline{w}$ if $v_i \perp_{\mathfrak{R}} w_i$ for some i. While typing imposes that terms are normalized, we still need to ensure that quantum programs preserve the norm, i.e., are *isometries*.

▶ **Definition 7.** *Given a STRS $\mathfrak{R}$ and a function symbol $\mathtt{f} \in \mathcal{F}$, we say that $\mathtt{f}$ is an* isometry*, if for all inputs $\overline{v}, \overline{w} \in \mathrm{In}(\mathtt{f})$, $\overline{v} \perp_{\mathfrak{R}} \overline{w}$ implies $\mathtt{f}(\overline{v}) \perp_{\mathfrak{R}} \mathtt{f}(\overline{w})$.*

Finally, quantum term rewrite systems are defined as STRS whose rewrite rules are well-typed, and where function symbols are isometries.

▶ **Definition 8.** *A STRS $\mathfrak{R}$ is called a* Quantum Term Rewrite System (QTRS) *if all function symbols are isometries and for any rule $l \to r \in \mathcal{R}_{\mathfrak{R}}$, typing contexts Γ, Δ, and type T:*

$$\Gamma;\Delta \vdash_{\mathfrak{R}} l : T \implies \Gamma;\Delta \vdash_{\mathfrak{R}} r : T$$

▶ **Example 9.** The STRS from Example 5 can be showed to be a QTRS. Indeed, each side of the rewrite rule can be typed identically. Furthermore, as both rewrite rules of $\mathtt{f}$ reduce to orthogonal terms, $\mathtt{f}$ can be shown to be an isometry. This is formalized in Appendix A.

3 Main Results

This section is devoted to showing that QTRS are well-behaved and enjoy standard properties (confluence, subject reduction, ...). It also discusses the hardness of type inference, which is undecidable in general, and shows that decidability can be recovered under some slight restrictions. Finally, we exhibit an expressive fragment of QTRS, which has the same computational power as quantum circuits: it is universal, and can be compiled to quantum circuits. Based on this fragment, we characterize the class of functions computable in quantum polynomial time, known as FBQP [11].

3.1 Standard Properties of QTRS

Because of orthogonality and the evaluation strategy, STRS are confluent up to equivalence.

▶ **Lemma 10** (Confluence). *Let $\mathfrak{R}$ be a STRS, and let t be a term. Suppose that there exist two terms t_1, t_2 such that $t \to^*_{\mathfrak{R}} t_1$ and $t \to^*_{\mathfrak{R}} t_2$. Then, there exist two terms t_3, t_4 such that $t_1 \to^*_{\mathfrak{R}} t_3$, $t_2 \to^*_{\mathfrak{R}} t_4$ and $t_3 \equiv t_4$.*

Typing is preserved through reduction for terminating or classical terms.

▶ **Lemma 11** (Subject reduction). *Let $\mathfrak{R}$ be a QTRS, and let $\vdash_{\mathfrak{R}} s : T$ be a well-typed term that is either terminating or classical. If $s \to_{\mathfrak{R}} t$, then $\vdash_{\mathfrak{R}} t : T$.*

▶ Remark 12. Termination is required to type superpositions obtained after reduction. For example, take the QTRS made of the three following rewrite rules:

$$\mathcal{R} = \{\Omega(\mathtt{0}) \to \Omega(\mathtt{S}(\mathtt{0})) \qquad \Omega(\mathtt{S}(n)) \to \Omega(n) \qquad \mathtt{f}(|i\rangle) \to |i\rangle\}$$

Take $t = (\mathtt{a}_{\frac{1}{\sqrt{2}}} \cdot |0\rangle + \mathtt{a}_{\frac{1}{\sqrt{2}}} \cdot \mathtt{f}(|1\rangle), \Omega(\mathtt{0}))$, with $[\![\mathtt{a}_{\frac{1}{\sqrt{2}}}]\!] = \frac{1}{\sqrt{2}}$. While t is well-typed, $t \to_{\mathfrak{R}} \mathtt{a}_{\frac{1}{\sqrt{2}}} \cdot (|0\rangle, \Omega(\mathtt{S}(\mathtt{0}))) + \mathtt{a}_{\frac{1}{\sqrt{2}}} \cdot (|1\rangle, \Omega(\mathtt{0}))$, which cannot be typed, as summands do not terminate.

A well-typed term always possesses a unique canonical form.

▶ **Lemma 13** (Canonical form). *Let $\mathfrak{R}$ be a QTRS, and let $\Gamma; \Delta \vdash_{\mathfrak{R}} t : T$ be a well-typed term. Then, t has a unique canonical form $\sum_{i=1}^{n} \alpha_i \cdot t_i$ up to reordering and amplitude equivalence, and $\Gamma; \Delta \vdash_{\mathfrak{R}} t_i : T$.*

As a consequence of Lemma 10 and 13, the orthogonality predicate (Definition 6) is sound.

A QTRS $\mathfrak{R}$ is said to be *total*, if for any function symbol $\mathtt{f}$ and any input $\overline{v} \in \mathrm{In}(\mathtt{f})$, there exist a substitution σ and a rule $l \to r \in \mathcal{R}_{\mathfrak{R}}$ such that $l\sigma = \mathtt{f}(\overline{v})$. For total QTRS, normal forms coincide with terms equivalent to values.

▶ **Lemma 14** (Normal forms). *Let $\mathfrak{R}$ be a total QTRS. Then, $\mathtt{NF}_{\mathfrak{R}}$ is exactly the set of well-typed ground terms equivalent to values.*

3.2 Type Inference

This section discusses the decidability of type inference. The typing rule for superpositions requires computing additions, multiplications, and nullity checks on complex numbers. Recall that algebraic numbers are complex numbers that are roots of a polynomial in $\mathbb{Q}[X]$, and write their sets as $\bar{\mathbb{C}}$. In the field of algebraic numbers, equality is decidable, and product and sum are computable [38]. In this section, we thus restrict ourselves to amplitudes symbols satisfying $[\![\mathtt{a}]\!] : \mathbb{N}^{\mathtt{ar(a)}} \to \bar{\mathbb{C}}$.

The typing rule for superpositions (Figure 3) also requires orthogonality checks $s \perp_{\mathfrak{R}} t$, which imply checking termination wrt a universal quantification over all possible substitutions (see Definition 6). Hence, typing is undecidable in the general case: we show that it is at least as hard as the Universal Halting Problem, i.e., Π^0_2-hard in the arithmetical hierarchy [28].

▶ **Theorem 15** (Undecidability of type inference). *Given a STRS $\mathfrak{R}$, deciding whether t can be typed is Π^0_2-hard, and belongs to Σ^0_3.*

Moreover, type inference without equivalence $\equiv$ is Π^0_2-complete. Decidability of type inference can be recovered by restricting the typing rules: a well-typed term t is said to be *easily typed*, if its typing derivation does not use any equivalence rule and, in any typing rule for superpositions, the orthogonality check $s \perp_{\mathfrak{R}} t$ is replaced by the check $\delta_{\perp}(s,t) = 0$. *Easy type inference* can be decided in polynomial time and implies that the term is well-typed.

▶ **Theorem 16** (Decidability of easy type inference). *Let $\mathfrak{R}$ be a terminating STRS, where $\mathcal{A}$ contains only symbols of arity 0. There is a polynomial P such that easy type inference of t is decidable in $P(|t|)$, and implies that t is well-typed.*

For example, all rewrite rules from Example 2 can be easily typed. However, Theorem 16 still requires termination, which is known to be undecidable. Nevertheless, Section 4 will discuss termination techniques that can be used to obtain a termination certificate, thus giving a semi-automatized procedure for type inference.

3.3 QTRS and Quantum Circuits

This section exhibits a relation between QTRS and quantum circuits. Towards that end, a fragment of QTRS is introduced (`QTRSCirc`, Definition 19). This fragment is universal for quantum circuits (Theorem 21), and can be compiled to families of quantum circuits (Theorem 22). For programs with a specific recursive structure, the size of the obtained circuit can be faithfully related with the runtime-complexity of the QTRS (Theorem 25), allowing to characterize precisely `FBQP` (Theorem 27), the natural complexity class for quantum polynomial time computable functions.

While function symbols of a QTRS preserve orthogonality, they do not always output a coherent number of qubits. As a counter-example, just consider the following QTRS $\{\mathtt{f}(|0\rangle) \to |0\rangle :: [\,], \mathtt{f}(|1\rangle) \to |1\rangle :: |1\rangle :: [\,]\}$. Furthermore, a naive padding with ancilla qubits could break the behaviour of a more general QTRS, e.g., if `f` is used as an input for another program. To tackle this problem, we introduce the notion of *structure*, which keeps only the classical shape of a value by replacing any occurrence of a qubit by the unit. Formally, the structure of a constructor is defined as $\mathtt{struct}(|0\rangle) = \mathtt{struct}(|1\rangle) \triangleq ()$ and $\mathtt{struct}(\mathtt{c}) \triangleq \mathtt{c}$ for $\mathtt{c} \in \mathcal{C} \setminus \{|0\rangle, |1\rangle\}$. The structure is then defined as a partial map on values by

$$\begin{aligned} \mathtt{struct}(\mathtt{c}(\overline{v})) &\triangleq \mathtt{struct}(c)(\mathtt{struct}(\overline{v})) \\ \mathtt{struct}(\alpha \cdot v) &\triangleq \mathtt{struct}(v) \\ \mathtt{struct}(v_1 + v_2) &\triangleq \mathtt{struct}(v_1) \qquad \text{if } \mathtt{struct}(v_1) = \mathtt{struct}(v_2) \end{aligned}$$

where $\mathtt{struct}(v_1, \ldots, v_n) \triangleq (\mathtt{struct}(v_1), \ldots, \mathtt{struct}(v_n))$. Two classical values have the same structure if they differ only on their qubit constructors. The above problem is solved by requiring that `f` reduces to values of identical structure, on inputs of identical structure.

▶ **Definition 17.** *Let $\mathfrak{R}$ be a terminating QTRS. We say that $\mathtt{f} \in \mathcal{F}$ is* structure preserving, *if for any $\overline{v}, \overline{w} \in \mathrm{In}(\mathtt{f})$ such that $\mathtt{struct}(\overline{v}) = \mathtt{struct}(\overline{w})$, $\mathtt{f}(\overline{v}) \to_{\mathfrak{R}}^* v', \mathtt{f}(\overline{w}) \to_{\mathfrak{R}}^* w'$, and $\mathtt{struct}(v') = \mathtt{struct}(w')$.*

Therefore, compilation of a function symbol `f` will differ depending on the considered structure of inputs. In particular, not all rewrite rules can be applied to a specific structure. Formally, a *`f`-structural set* S is a set of `f`-rewrite rules such that $\forall\, l \to r, l' \to r' \in S$, there exist two substitutions σ, σ' such that $l\sigma = \mathtt{f}(\overline{v})$, $l'\sigma' = \mathtt{f}(\overline{v'})$, and $\mathtt{struct}(\overline{v}) = \mathtt{struct}(\overline{v'})$.

Compiling a function symbol `f` can be done in two ways: either the considered rewrite rules are viewed as a single unitary gate, or each rewrite rule is compiled separately as a controlled statement. We say that `f` *encodes a unitary*, if any `f`-rewrite rule $l \to r \in \mathcal{R}$ satisfies $r \in \mathcal{V}$, allowing us to compile `f` as in the former case. To properly compile the latter case, it is required that all the considered rules treat the control qubits identically and leave them untouched. This property is ensured by the definition below.

▶ **Definition 18.** *Let $\mathfrak{R}$ be a QTRS, and let $\mathtt{f} \in \mathcal{F}$. We say that* $\mathtt{f}$ quantum controls, *if for any $\mathtt{f}$-structural set S, there exist a n-hole classical context C, and a $n+m$-hole classical context C' with no function symbol occurrence, such that for all $l_j \to r_j \in S$:*

$$l_j = C[|i_1\rangle, \ldots, |i_n\rangle] \quad \wedge \quad r_j = C'[|i_1\rangle, \ldots, |i_n\rangle, t_1^j, \ldots, t_m^j]$$

Finally, we restrict constructor symbols to the set $\mathcal{C}_Q \triangleq \{|0\rangle, |1\rangle, \mathtt{0}, \mathtt{S}, [\,], \mathtt{cons}, \mathtt{pr}\}$, so that values have a meaning circuit-wise. Altogether, define `QTRSCirc` as the intersection of all the above restrictions.

▶ **Definition 19.** *Define* `QTRSCirc` *as the set of QTRS $\mathfrak{R}$, where $\mathcal{C} \subseteq \mathcal{C}_Q$, and for all $\mathtt{f} \in \mathcal{F}$, $\mathtt{f}$ is structure preserving, and either $\mathtt{f}$ encodes a unitary, or $\mathtt{f}$ quantum controls.*

▶ **Example 20.** The QTRS from Example 2 belongs to `QTRSCirc`. $\mathtt{X}, \mathtt{T}, \mathtt{H}$ all output values, thus encode a unitary, while `CNOT` leaves the control qubit untouched, thus quantum controls. All symbols also are structure preserving, as they respectively output a qubit and a pair of qubits no matter their inputs. Similarly, each function symbol from Example 3 quantum controls and is structure preserving, as it outputs a qubit list of same size as its input, thus it belongs to `QTRSCirc`.

Let us now discuss the relation between `QTRSCirc` and quantum circuits. Recall that a family of quantum circuits $(C_n)_{n\in\mathbb{N}}$ is said to be *uniform*, if there exists a Turing machine which, on input n, outputs the circuit representation of C_n.

As term rewrite systems are known to be Turing complete [40], any uniform family of circuits can be expressed as a QTRS, with the notation $t_0 = \mathtt{0}$ and $t_{n+1} = \mathtt{S}(t_n)$ for all $n \in \mathbb{N}$, and by encoding any n-qubit state $|\phi\rangle = \sum_{x_1\ldots x_n \in \{0,1\}^n} \alpha_{x_1\ldots x_n} |x_1 \ldots x_n\rangle$ into the value $t_{|\phi\rangle}$ defined below:

$$t_{|\phi\rangle} \triangleq \sum_{x_1\ldots x_n \in \{0,1\}^n} \mathtt{a}_{x_1\ldots x_n} \cdot |x_1\rangle :: \cdots :: |x_n\rangle :: [\,], \qquad \text{with } [\![\mathtt{a}_{x_1\ldots x_n}]\!] \triangleq \alpha_{x_1\ldots x_n}.$$

▶ **Theorem 21** (Universality)**.** *Let $(C_n)_{n\in\mathbb{N}}$ be a uniform family of circuits. Then, there exists a QTRS $\mathfrak{R} \in$ `QTRSCirc` of main symbol $\mathtt{f}$ such that, for any $n \in \mathbb{N}$ and any n-qubit state $|\phi\rangle$, $\mathtt{f}(t_n, t_{|\phi\rangle}) \to_{\mathfrak{R}}^* t_{C_n|\phi\rangle}$.*

We now prove that any QTRS in `QTRSCirc` can be approximated by a uniform family of circuits. Towards that end, we encode naturally any value v with constructor symbols in $\mathcal{C}_Q$ to a quantum state $|v\rangle$, by discarding natural bumbers. This is extended to tuples $\overline{v} = (v_1, \ldots, v_n)$ by $|\overline{v}\rangle \triangleq |v_1\rangle \otimes \cdots \otimes |v_n\rangle$. Given any tuple $\overline{v}$ and any value v', a circuit C is said to *approximate* v' with probability $p \in [0,1]$ on input $\overline{v}$, if $C\,|\overline{v}\rangle$ evaluates to $|\phi\rangle$, and $|\langle v'|\phi\rangle|^2 \geq p$.

As discussed before, compilation of the QTRS will be done with respect to an input structure. Therefore, compilation yields a family of circuits indexed by the structure of the possible inputs of $\mathtt{f}$, i.e., by the set $\mathtt{S}(\mathtt{f}) \triangleq \{\, \mathtt{struct}(\overline{v}) \mid \overline{v} \in \mathrm{In}(\mathtt{f}) \,\}$. Note that the proof is constructive, thus actually generates the family of circuits $\mathtt{C}(\mathfrak{R})$.

▶ **Theorem 22** (Circuit compilation)**.** *Let $\mathfrak{R} \in$ `QTRSCirc` be a terminating QTRS of main symbol $\mathtt{f}$. Then, we can generate a family of circuits $\mathtt{C}(\mathfrak{R}) \triangleq (C_w)_{w\in\mathtt{S}(\mathtt{f})}$ s.t., for any input $\overline{v} \in \mathrm{In}(\mathtt{f})$, if $\mathtt{f}(\overline{v}) \to_{\mathfrak{R}}^* v'$ then $C_{\mathtt{struct}(\overline{v})}$ approximates v' with probability $\frac{2}{3}$ on input $\overline{v}$.*

We are now interested in obtaining bounds on the size of the generated circuit. Towards that end, an ordering $\succcurlyeq_{\mathfrak{R}}$ on function symbols is introduced: $\mathtt{f} \succcurlyeq_{\mathfrak{R}} \mathtt{g}$ holds if $\mathtt{f}$ calls $\mathtt{g}$ in its

reduction. Formally, $\succcurlyeq_{\mathfrak{R}}$ is defined as the smallest transitive relation satisfying $\mathtt{f} \succcurlyeq_{\mathfrak{R}} \mathtt{g}$ when there exists a context C such that $\mathtt{f}(\overline{s}) \to C[\mathtt{g}(\overline{t})] \in \mathcal{R}_{\mathfrak{R}}$. We also define its induced strict order and equivalence relation respectively by $\succ_{\mathfrak{R}}$ and $\approx_{\mathfrak{R}}$. The *rank* of a function symbol is defined inductively as follows, with the convention that $\max(\emptyset) = 0$:

$$\mathrm{rk}(\mathtt{f}) \triangleq \max_{\mathtt{f} \succ_{\mathfrak{R}} \mathtt{g}} (\mathrm{rk}(\mathtt{g}) + 1)$$

Function symbols are restricted by prohibiting mutal recursion and enforcing that recursive calls are always performed on the same input. This is defined formally below using the notation $\mathrm{Callee}(\mathtt{f}, S) \triangleq \{\mathtt{f}(\overline{s}) \mid l \to C[\mathtt{f}(\overline{s})] \in S\}$, for $\mathtt{f} \in \mathcal{F}$ and for a $\mathtt{f}$-structural set S.

▶ **Definition 23.** *A QTRS $\mathfrak{R}$ is said to be* simply recursive *if, for all function symbols $\mathtt{f}, \mathtt{g}$, $\mathtt{f} \approx_{\mathfrak{R}} \mathtt{g}$ implies $\mathtt{f} = \mathtt{g}$ and, for any $\mathtt{f}$-structural set S, $\#\mathrm{Callee}(\mathtt{f}, S) \leq 1$. Define $\mathtt{SRec}$ as the set of simply recursive QTRS.*

These conditions are akin to [36], allowing to merge recursive calls together and avoid an exponential blow-up in the size of the circuit; they are verified for the QTRS of Example 3. We define the following syntactic sugar for a tuple $\overline{v} = (v_1, \ldots, v_n)$: $|\overline{v}| \triangleq \sum_{i=1}^{n} |v_i|$.

▶ **Definition 24.** *Let $\tau : \mathbb{N} \to \mathbb{N}$ be a non-decreasing function. Define $\mathtt{Time}(\tau)$ as the set of QTRS of main symbol $\mathtt{f}$, where for any $\overline{v} \in \mathrm{In}(\mathtt{f})$, $\mathtt{f}(\overline{v})$ terminates in at most $\tau(|\overline{v}|)$ steps.*

On the fragment of QTRS defined below, the size of the circuits $\mathtt{C}(\mathfrak{R})$, generated by the proof of Theorem 22, can be upper-bounded and related with the runtime-complexity of the QTRS:

$$\mathtt{QTRS}(\tau) \triangleq \mathtt{QTRSCirc} \cap \mathtt{SRec} \cap \mathtt{Time}(\tau).$$

▶ **Theorem 25.** *Take $\mathfrak{R} \in \mathtt{QTRS}(\tau)$ of main symbol $\mathtt{f}$. Then, for any $\overline{v} \in \mathrm{In}(\mathtt{f})$, $|C_{\mathtt{struct}(\overline{v})}| = \mathcal{O}(\tau(|\overline{v}|)^{\mathrm{rk}(\mathtt{f})+1})$, where $C_{\mathtt{struct}(\overline{v})} \in \mathtt{C}(\mathfrak{R})$ is the compiled circuit of corresponding structure.*

We end this section by providing an implicit characterization of the complexity class $\mathtt{FBQP}$. Recall that a family of circuits $(C_n)_{n \in \mathbb{N}}$ is said to be *uniform polynomially-sized* if there exists $P \in \mathbb{N}[X]$ such that $|C_n| \leq P(n)$ and there is a polynomial-time Turing machine, which takes n as input and outputs a representation of C_n for any $n \in \mathbb{N}$.

▶ **Definition 26** ([11])**.** *A binary function $f : \{0,1\}^* \to \{0,1\}^*$ is said to be computed by a family of circuits $(C_n)_{n \in \mathbb{N}}$ if, for any $x \in \{0,1\}^*$, $C_{|x|}$ approximates $f(x)$ with probability $\frac{2}{3}$ on input x. $\mathtt{FBQP}$ is defined as the set of binary functions computed by a uniform polynomially-sized family of circuits.*

Define the fragment of QTRS terminating in polynomial time on lists of qubits.

$$\mathtt{QTRS}(Poly) \triangleq \cup_{P \in \mathbb{N}[X]} \{\langle \mathcal{A}, \mathcal{C}, \mathcal{F}, \mathcal{X}, \mathcal{R} \rangle_{\mathtt{f}} \in \mathtt{QTRS}(P) \mid \mathrm{type}(\mathtt{f}) = \mathtt{List}(\mathtt{Qbit}) \to \mathtt{List}(\mathtt{Qbit})\}$$

Using the characterization of $\mathtt{FBQP}$ from [63] and Theorem 25, $\mathtt{QTRS}(Poly)$ is shown to correspond exactly to $\mathtt{FBQP}$. Let $\{\!\{\mathtt{QTRS}(Poly)\}\!\}$ be the set of binary functions that can be computed by $\mathtt{C}(\mathfrak{R})$ for $\mathfrak{R} \in \mathtt{QTRS}(Poly)$.

▶ **Theorem 27** ($\mathtt{FBQP}$ characterization)**.** $\{\!\{\mathtt{QTRS}(Poly)\}\!\} = \mathtt{FBQP}$.

4 Complexity Analysis

The previous section emphasized that termination of a QTRS is a crucial hypothesis used for many results to hold, e.g., subject reduction (Lemma 11), type inference (Theorem 16), or circuit compilation (Theorem 22). In this section, we discuss how termination of a STRS can be proved, by adapting existing techniques from standard TRS. This is done by extending orders on classical terms to orders on any term with superposition (*worst path orderings*, Section 4.1). We show how these orders can be used to adapt well-known existing termination techniques — e.g., polynomial interpretations and dependency pairs — to the QTRS setting. Finally, we also show how we can obtain complexity results for polynomial time, yielding a decidable criterion to check whether a given QTRS computes a function in FBQP.

4.1 Worst Path Orderings

Recall first some definitions. A *quasi-order* $\succcurlyeq$ is a reflexive and transitive relation. A *strict order* $\succ$ is an irreflexive and transitive relation. For each quasi-order, we can associate the strict order $s \succ t \iff s \succcurlyeq t \wedge \neg(t \succcurlyeq s)$. If no infinite chain $t_1 \succ t_2 \dots$ can be built for a strict order, it is called *well-founded*. A STRS $\mathfrak{R}$ is said to be *compatible* with a strict order $\succ$, if for all rules $l \to r \in \mathcal{R}_{\mathfrak{R}}$, $l \succ r$. An order $\succ$ is said to be *monotonic*, if $s \succ t$ implies $\mathtt{b}(\dots, s, \dots) \succ \mathtt{b}(\dots, t, \dots)$ for all terms s, t and for all $\mathtt{b} \in \mathcal{C} \uplus \mathcal{F}$. An order $\succ$ is said to be *closed under substitutions*, if $s \succ t$ implies $s\sigma \succ t\sigma$, for all terms s, t and substitutions σ. A *rewrite order* is a monotonic order closed under substitutions.

On standard TRS, proofs of termination often include finding a well-founded rewrite order $\succ$, such that if s reduces to t, then $s \succ t$. However, existing orders are defined on classical terms. To extend these orders to any term of a STRS, remark the following lemma.

▶ **Lemma 28.** *Let $\mathfrak{R}$ be a STRS, and let $t = \sum_{i=1}^{n} \alpha_i \cdot t_i$ be a well-typed term. If t starts an infinite chain on the order induced by $\to_{\mathfrak{R}}$, then so does t_i for some $1 \leq i \leq n$.*

We also have the contrapositive result: if all t_i terminate, then so does t. Therefore, the idea is to allow the order to always take the worst path, i.e., the term of a superposition taking the most time to reduce; this can also be seen as taking the maximum over multiple paths. This is formalized below.

▶ **Definition 29.** *Let $\succcurlyeq$ be a quasi-order on classical terms. We define the* worst path extension *of $\succcurlyeq$ as the relation $\succcurlyeq_w$ defined inductively as follows:*

$$\frac{s, t \in \mathcal{C} \qquad s \succcurlyeq t}{s \succcurlyeq_w t} \qquad \frac{s \equiv \sum_{i=1}^{n} \alpha_i \cdot s_i \in \mathtt{CAN} \qquad \exists i,\ s_i \succcurlyeq_w t}{s \succcurlyeq_w t} \qquad \frac{t \equiv \sum_{j=1}^{m} \beta_j \cdot t_j \in \mathtt{CAN} \qquad \forall j,\ s \succcurlyeq_w t_j}{s \succcurlyeq_w t}$$

This relation enjoys the following properties. In particular, it is actually a (quasi-)order, hence the name worst path ordering.

▶ **Lemma 30.** *Given a quasi-order $\succcurlyeq_w$, the following properties are satisfied, for any $s \equiv \sum_{i=1}^{n} \alpha_i \cdot s_i \in \mathtt{CAN}$ and $t \equiv \sum_{j=1}^{m} \beta_j \cdot t_j \in \mathtt{CAN}$:*

- $s \succcurlyeq_w t \iff \forall j, \exists i, s_i \succcurlyeq t_j$;
- *If s, t are classical, $s \succcurlyeq_w t \iff s \succcurlyeq t$;*
- *$\succcurlyeq_w$ is a quasi-order.*
- *If $\succ$ is a rewrite order, then $\succ_w$ is a rewrite order too.*

Furthermore, it can be used to obtain termination of STRS compatible with the ordering.

▶ **Theorem 31.** *Let $\succcurlyeq$ be a quasi-order, where its induced strict order is a well-founded rewrite order. Then, any STRS compatible with the worst path extension $\succ_w$ terminates.*

Theorem 31 thus allows us to generate a termination technique, for each well-founded rewrite order $\succ$ on classical terms. Such orderings have been deeply-studied in the literature and we now show a few adaptations of some of these orderings to the quantum setting.

4.2 Polynomial Interpretations

Polynomial interpretations were introduced in [47] as a way to show termination of a TRS. The main idea is to associate each symbol with a monotonic function $(\!|-|\!)$, called *interpretation*, such that it decreases strictly over any rule $l \to r$, i.e., $(\!|l|\!) > (\!|r|\!)$. This is extended by defining interpretations over superpositions as the maximum of the interpretations.

Equip $\mathbb{N}$ with the natural strict ordering $\succ_{\mathbb{N}}$. Given two polynomials $P, Q \in \mathbb{N}[X_1, \ldots, X_n]$, define $P \succ_{\mathbb{N}} Q$ as follows:

$$P \succ_{\mathbb{N}} Q \iff \forall x_1, \ldots, x_n \in \mathbb{N}, P(x_1, \ldots, x_n) \succ_{\mathbb{N}} Q(x_1, \ldots, x_n)$$

An *assignment* $(\!|-|\!)$ is a function mapping each symbol in $\mathcal{C} \uplus \mathcal{F}$ to a multi-variate polynomial $(\!|\mathtt{b}|\!) \in \mathbb{N}[X_1, \ldots, X_{\mathtt{ar(b)}}]$. Define VAR as the countable set of indeterminates that can be used in polynomials, and fix a map $\rho : \mathcal{X} \to \mathrm{VAR}$ for the rest of the paper. Assignments are extended to any term of a STRS inductively as follows:

$$\begin{aligned} (\!|\mathtt{x}|\!) &\triangleq \rho(\mathtt{x}) \\ (\!|\mathtt{b}(s_1, \ldots, s_n)|\!) &\triangleq (\!|\mathtt{b}|\!)((\!|s_1|\!), \ldots, (\!|s_n|\!)) \\ (\!|t|\!) &\triangleq \max_{1 \le i \le n} (\!|t_i|\!), \qquad \text{provided } t \equiv \sum_{i=1}^{n} \alpha_i \cdot t_i \in \mathtt{CAN} \end{aligned}$$

where $\max_{1 \le i \le n} (\!|t_i|\!)$ is a polynomial P satisfying $P \succcurlyeq_{\mathbb{N}} (\!|t_i|\!)$. By unicity of the canonical form (Lemma 13), the assignment of a term is properly defined.

Given an assignment $(\!|-|\!)$, define the order $\succcurlyeq_{(\!|-|\!)}$ as $s \succcurlyeq_{(\!|-|\!)} t \iff (\!|s|\!) \succcurlyeq_{\mathbb{N}} (\!|t|\!)$. An assignment $(\!|-|\!)$ is *monotonic* if $\succ_{(\!|-|\!)}$ is monotonic. A STRS $\mathfrak{R}$ is *compatible* with an assignment $(\!|-|\!)$, if $\mathfrak{R}$ is compatible with its ordering $\succ_{(\!|-|\!)}$. As $\succcurlyeq_{(\!|-|\!)}$ can be showed to satisfy the properties of a worst path ordering, finding a compatible assignment provides a criterion for termination.

▶ **Theorem 32.** *Any STRS compatible with a monotonic assignment terminates.*

▶ **Example 33.** Define the STRS $\mathfrak{R}$ made with the following rules:

$$\mathcal{R} = \left\{ \begin{array}{r} \mathtt{X}(|i\rangle) \to |1-i\rangle \\ \mathtt{f}(\mathtt{x}, [\,]) \to \mathtt{x} :: [\,] \\ \mathtt{f}(|0\rangle, h :: t) \to \mathtt{sgn}^{+} \cdot |0\rangle :: \mathtt{f}(h, t) + \mathtt{sgn}^{+} \cdot |1\rangle : \mathtt{f}(\mathtt{X}(h), t) \\ \mathtt{f}(|1\rangle, h :: t) \to \mathtt{sgn}^{+} \cdot |0\rangle :: \mathtt{f}(h, t) + \mathtt{sgn}^{-} \cdot |1\rangle : \mathtt{f}(\mathtt{X}(h), t) \end{array} \right\}$$

Then, given the following monotonic assignment:

$$(\!||i\rangle|\!) = (\!|[\,]|\!) = 1 \qquad (\!|\mathtt{cons}|\!)(X, Y) = X + Y + 1 \qquad (\!|\mathtt{X}|\!) = X + 1 \qquad (\!|\mathtt{f}|\!)(X, Y) = X + 3Y$$

Checking that $\mathfrak{R}$ is compatible with this assignment is direct for the X-rewrite rules and the first rule of f; the check for the last two rules is done below:

$$(\!|f(|i\rangle, h :: t)|\!) = 4 + 3(\!|h|\!) + 3(\!|t|\!) \succ_{\mathbb{N}} 3 + (\!|h|\!) + 3(\!|t|\!) = (\!||1\rangle :: \mathtt{f}(\mathtt{X}(h), t)|\!) \succ_{\mathbb{N}} (\!||1\rangle :: \mathtt{f}(h, t)|\!)$$

Therefore, $\mathfrak{R}$ terminates.

4.3 Dependency Pairs

Dependency pairs [2] is another termination technique, which has the perk of dropping the strict monotonicity condition for rewrite rules. The idea is to define, for any rewrite rule $l \to r$ and any subterm s of r whose outermost symbol is a function symbol, a *dependency pair* $\langle l, s\rangle$. Proving termination is equivalent to finding a quasi-ordering $\succcurlyeq$ such that it decreases weakly on rewrite rules but strictly on dependency pairs, i.e., $l \succcurlyeq r$ and $l \succ s$.

This definition is adapted by requiring t to be a subterm of the canonical form of r, i.e., $r \equiv \sum_{i=1}^{n} \alpha_i \cdot r_i \in \mathtt{CAN}$ and $r_i = C[t]$ for some i, denoted $C[t] \lhd_{\mathtt{CAN}} r$ for conciseness. As for worst path orderings, this definition will be sound, but not complete.

For each function symbol $\mathtt{f} \in \mathcal{F}$, we associate a corresponding *tuple symbol* F, of same signature. For clarity, upper case symbols correspond to tuple symbols in the following.

▶ **Definition 34.** *Let $\mathfrak{R}$ be a STRS. If $\mathtt{f}(s_1, \dots, s_n) \to r \in \mathcal{R}_{\mathfrak{R}} \wedge C[\mathtt{g}(t_1, \dots, t_m)] \lhd_{\mathtt{CAN}} r$, for $\mathtt{g} \in \mathcal{F}$, then $\langle F(s_1, \dots, s_n), G(t_1, \dots, t_m)\rangle$ is called a dependency pair of $\mathfrak{R}$.*

Chains are then defined as successive dependency pairs.

▶ **Definition 35.** *Let $\mathfrak{R}$ be a STRS. A sequence of dependency pairs $\langle s_1, t_1\rangle, \langle s_2, t_2\rangle \dots$ is a* $\mathfrak{R}$-chain*, if there exists a substitution σ, such that $t_j\sigma \to^*_{\mathfrak{R}} r_j$ and $s_{j+1}\sigma \lhd_{\mathtt{CAN}} r_j$ for all j.*

The absence of infinite $\mathfrak{R}$-chain implies termination, as in [2].

▶ **Theorem 36.** *A STRS $\mathfrak{R}$ is terminating if no infinite $\mathfrak{R}$-chain exists.*

Equivalently, termination can be proven by finding orderings as defined below, which implies the absence of infinite $\mathfrak{R}$-chain.

▶ **Theorem 37.** *A STRS $\mathfrak{R}$ is terminating if there exists a well-founded weakly monotonic quasi-ordering $\succcurlyeq$ on classical terms, where $\succ$ and $\succcurlyeq$ are closed under substitution, such that:*

- *$l \succcurlyeq r_i$ for all rules $l \to \sum_{i=1}^{n} \alpha_i \cdot r_i \in \mathcal{R}_{\mathfrak{R}}$ and for all i, with $\sum_{i=1}^{n} \alpha_i \cdot r_i \in \mathtt{CAN}$;*
- *$s \succ t$ for all dependency pairs $\langle s, t\rangle$.*

4.4 Complexity Results

Sections 4.2 and 4.3 have provided different techniques, based on Lemma 28, for proving the termination of a STRS. We now are interested in obtaining complexity results on the runtime-complexity, which rely on the following lemma.

▶ **Lemma 38.** *Let $\mathfrak{R}$ be a STRS, and let $t = \sum_{i=1}^{n} \alpha_i \cdot t_i \in \mathcal{T}_0$. Assume each t_i terminates in at most k_i steps. Then t terminates in at most $\max_{1 \le i \le n} k_i$ steps.*

As a consequence of Lemma 38, the assignment of a term bounds its runtime complexity.

▶ **Lemma 39.** *Let $\mathfrak{R}$ be a QTRS compatible with an assignment $(\!|-|\!)$. Then, given any term t, it terminates in at most $(\!|t|\!)$ steps.*

Despite their name, polynomial interpretations are not necessarily bounded by polynomials. This can, however, be obtained by imposing a structure for constructor assignments.

▶ **Definition 40.** *Let $\mathfrak{R}$ be a QTRS, and let $(\!|-|\!)$ be an assignment. We say that it is an* additive assignment*, if for all $\mathtt{c} \in \mathcal{C}$ of arity $n > 0$, $(\!|\mathtt{c}|\!) = \sum_{i=1}^{n} X_i + \alpha_{\mathtt{c}}$, with $\alpha_{\mathtt{c}} \ge 1$.*

Such a restriction allows bounding any assignment by a polynomial, and thus to obtain polynomial time.

▶ **Theorem 41.** *Let $\mathfrak{R}$ be a QTRS compatible with an additive assignment and let $\mathtt{f} \in \mathcal{F}$. Then, there is a polynomial P such that for any $\overline{v} \in \mathrm{In}(\mathtt{f})$, $\mathtt{f}(\overline{v})$ terminates in time $P(|v|)$.*

As a consequence of Theorem 27, this yields a decidable way of checking whether a compilable QTRS computes a binary function in the quantum complexity class FBQP.

References

1 Mateus Araújo, Fabio Costa, and Časlav Brukner. Computational advantage from quantum-controlled ordering of gates. *Phys. Rev. Lett.*, 113:250402, December 2014. doi:10.1103/PhysRevLett.113.250402.

2 Thomas Arts and Jürgen Giesl. Termination of term rewriting using dependency pairs. *Theoretical Computer Science*, 236(1):133–178, 2000. doi:10.1016/S0304-3975(99)00207-8.

3 Martin Avanzini, Ugo Dal Lago, and Akihisa Yamada. On probabilistic term rewriting. *Science of Computer Programming*, 185:102338, January 2020. doi:10.1016/j.scico.2019.102338.

4 Martin Avanzini and Georg Moser. Dependency Pairs and Polynomial Path Orders. In *Rewriting Techniques and Applications*, pages 48–62, Berlin, Heidelberg, 2009. Springer. doi:10.1007/978-3-642-02348-4_4.

5 Martin Avanzini and Georg Moser. Polynomial Path Orders. *Logical Methods in Computer Science*, Volume 9, Issue 4, November 2013. doi:10.2168/LMCS-9(4:9)2013.

6 Martin Avanzini, Georg Moser, Romain Péchoux, Simon Perdrix, and Vladimir Zamdzhiev. Quantum expectation transformers for cost analysis. In *LICS '22: Symposium on Logic in Computer Science*, pages 10:1–10:13. ACM, 2022. doi:10.1145/3531130.3533332.

7 Franz Baader and Tobias Nipkow. *Term Rewriting and All That*. Cambridge University Press, March 1998. doi:10.1017/cbo9781139172752.

8 Patrick Baillot, Ugo Dal Lago, Cynthia Kop, and Deivid Vale. A characterization of basic feasible functionals through higher-order rewriting and tuple interpretations. *Logical Methods in Computer Science*, Volume 21, Issue 4, November 2025. doi:10.46298/lmcs-21(4:19)2025.

9 Kathleen Barsse, Romain Péchoux, and Simon Perdrix. Quantum Control and General Recursion Beyond the Unitary Case. In *41st Annual Symposium on Logic in Computer Science (LICS 2026)*, volume 380 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 14:1–14:26, Dagstuhl, Germany, July 2026. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.LICS.2026.14.

10 Thaïs Baudon, Carsten Fuhs, and Laure Gonnord. On complexity bounds and confluence of parallel term rewriting*. *Fundamenta Informaticae*, 192(2):121–166, 2024. doi:10.3233/fi-242191.

11 Ethan Bernstein and Umesh Vazirani. Quantum Complexity Theory. *SIAM Journal on Computing*, 26(5):1411–1473, October 1997. doi:10.1137/S0097539796300921.

12 Guillaume Bonfante, Adam Cichon, Jean-Yves Marion, and Hélène Touzet. Algorithms with polynomial interpretation termination proof. *J. Funct. Program.*, 11(1):33–53, January 2001. doi:10.1017/S0956796800003877.

13 Guillaume Bonfante, Jean-Yves Marion, and Jean-Yves Moyen. Quasi-interpretations a way to control resources. *Theoretical Computer Science*, 412(25):2776–2796, June 2011. doi:10.1016/j.tcs.2011.02.007.

14 Guillaume Bonfante and Georg Moser. *Characterising Space Complexity Classes via Knuth-Bendix Orders*, page 142–156. Springer Berlin Heidelberg, 2010. doi:10.1007/978-3-642-16242-8_11.

15 Olivier Bournez and Florent Garnier. *Proving Positive Almost-Sure Termination*, page 323–337. Springer Berlin Heidelberg, 2005. doi:10.1007/978-3-540-32033-3_24.

16 Olivier Bournez and Claude Kirchner. *Probabilistic Rewrite Strategies. Applications to ELAN*, page 252–266. Springer Berlin Heidelberg, 2002. URL: http://dx.doi.org/10.1007/3-540-45610-4_18, doi:10.1007/3-540-45610-4_18.

17 Kostia Chardonnet, Emmanuel Hainry, Romain Péchoux, and Thomas Vinet. Resource-aware quantum programming with general recursion and quantum control. In *11th International Conference on Formal Structures for Computation and Deduction (FSCD 2026)*, volume 378, pages 12:1–12:20, July 2026. doi:10.4230/LIPICS.FSCD.2026.12.

18 Christophe Chareton, Jad Issa, Mathieu Nguyen, Nicolas Blanco, and Sébastien Bardin. Hybrid path-sums for hybrid quantum programs. *Proceedings of the ACM on Programming Languages*, 10(PLDI):1687–1713, June 2026. doi:10.1145/3808314.

19 Bob Coecke and Ross Duncan. Interacting quantum observables: categorical algebra and diagrammatics. *New Journal of Physics*, 13(4):043016, April 2011. doi:10.1088/1367-2630/13/4/043016.

20 Andrea Colledan and Ugo Dal Lago. Flexible type-based resource estimation in quantum circuit description languages. *Proc. ACM Program. Lang.*, 9(POPL), January 2025. doi:10.1145/3704883.

21 Stephen A. Cook and Bruce .M. Kapron. Characterizations of the basic feasible functionals of finite type. In *30th Annual Symposium on Foundations of Computer Science*, page 154–159. IEEE, 1989. doi:10.1109/sfcs.1989.63471.

22 Andrew W. Cross, Lev S. Bishop, John A. Smolin, and Jay M. Gambetta. Open quantum assembly language, 2017. arXiv:1707.03429.

23 Ugo Dal Lago, Andrea Masini, and Margherita Zorzi. Quantum implicit computational complexity. *Theor. Comput. Sci.*, 411(2):377–409, 2010. doi:10.1016/J.TCS.2009.07.045.

24 Kinnari Dave, Louis Lemonnier, Romain Péchoux, and Vladimir Zamdzhiev. Combining quantum and classical control: syntax, semantics and adequacy. In *Foundations of Software Science and Computation Structures, FoSSaCS 2025*, pages 155–175, Berlin, Heidelberg, 2025. Springer-Verlag. doi:10.1007/978-3-031-90897-2_8.

25 Nachum Dershowitz. Orderings for term-rewriting systems. *Theoretical computer science*, 17(3):279–301, 1982. doi:10.1016/0304-3975(82)90026-3.

26 Alejandro Díaz-Caro, Mauricio Guillermo, Alexandre Miquel, and Benoît Valiron. Realizability in the unitary sphere. In *2019 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, page 1–13. IEEE, June 2019. doi:10.1109/lics.2019.8785834.

27 Alejandro Díaz-Caro and Octavio Malherbe. Quantum control in the unitary sphere: Lambda-s1 and its categorical model. *Log. Methods Comput. Sci.*, 18(3), 2022. doi:10.46298/LMCS-18(3:32)2022.

28 Jörg Endrullis, Herman Geuvers, and Hans Zantema. *Degrees of Undecidability in Term Rewriting*, pages 255–270. Springer Berlin Heidelberg, 2009. doi:10.1007/978-3-642-04027-6_20.

29 Yuan Feng and Mingsheng Ying. Quantum Hoare logic with classical variables. *ACM Transactions on Quantum Computing*, 2(4):1–43, 2021. doi:10.1145/3456877.

30 Florent Ferrari, Emmanuel Hainry, Romain Péchoux, and Mário Silva. Quantum Programming in Polylogarithmic Time. In *50th International Symposium on Mathematical Foundations of Computer Science (MFCS 2025)*, volume 345 of *LIPIcs*, pages 47:1–47:17, 2025. doi:10.4230/LIPIcs.MFCS.2025.47.

31 Florian Frohn and Jürgen Giesl. Constant runtime complexity of term rewriting is semi-decidable. *Information Processing Letters*, 139:18–23, November 2018. doi:10.1016/j.ipl.2018.06.012.

32 Florian Frohn, Jürgen Giesl, Jera Hensel, Cornelius Aschermann, and Thomas Ströder. Lower bounds for runtime complexity of term rewriting. *Journal of Automated Reasoning*, 59(1):121–163, December 2016. doi:10.1007/s10817-016-9397-x.

33 Alexander S. Green, Peter LeFanu Lumsdaine, Neil J. Ross, Peter Selinger, and Benoît Valiron. Quipper: a scalable quantum programming language. In *ACM SIGPLAN Conference on Programming Language Design and Implementation, PLDI '13*, pages 333–342. ACM, June 2013. doi:10.1145/2491956.2462177.

34 Stefano Guerrini, Simone Martini, and Andrea Masini. Quantum turing machines: Computations and measurements. *Applied Sciences*, 10(16):5551, August 2020. doi:10.3390/app10165551.

35 Emmanuel Hainry, Romain Péchoux, and Mário Silva. A Programming Language Characterizing Quantum Polynomial Time. In *Foundations of Software Science and Computation Structures*, pages 156–175, April 2023. doi:10.1007/978-3-031-30829-1_8.

36 Emmanuel Hainry, Romain Péchoux, and Mário Silva. Branch sequentialization in quantum polytime. In *Formal Structures for Computation and Deduction, FSCD 2025*, volume 337 of *LIPIcs*, pages 22:1–22:22, 2025. doi:10.4230/LIPICS.FSCD.2025.22.

37 Emmanuel Hainry and Romain Péchoux. Theory of higher order interpretations and application to basic feasible functions. *Logical Methods in Computer Science*, Volume 16, Issue 4, December 2020. doi:10.23638/lmcs-16(4:14)2020.

38 Vesa Halava, Tero Harju, Mika Hirvensalo, and Juhani Karhumäki. Skolem's problem - on the border between decidability and undecidability. Technical Report 683, Turku Center for Computer Science, 2005. URL: https://tucs.fi/publications/view/?pub_id=tHaHaHiKa05a.

39 Nao Hirokawa and Georg Moser. Automated Complexity Analysis Based on the Dependency Pair Method. In *Automated Reasoning*, pages 364–379, Berlin, Heidelberg, 2008. Springer. doi:10.1007/978-3-540-71070-7_32.

40 Gérard Huet and Dallas S. Lankford. On the uniform halting problem for term rewriting systems. Technical Report 283, IRIA, March 1978. URL: https://www.ens-lyon.fr/LIP/REWRITING/TERMINATION/Huet_Lankford.pdf.

41 Jan-Christoph Kassing and Jürgen Giesl. *Proving Almost-Sure Innermost Termination of Probabilistic Term Rewriting Using Dependency Pairs*, page 344–364. Springer Nature Switzerland, 2023. doi:10.1007/978-3-031-38499-8_20.

42 Alexei Y. Kitaev. Quantum computations: algorithms and error correction. *Russian Mathematical Surveys*, 52(6):1191–1249, December 1997. doi:10.1070/rm1997v052n06abeh002155.

43 Emanuel Knill, Raymond Laflamme, and Gerard J. Milburn. A scheme for efficient quantum computation with linear optics. *Nature*, 409(6816):46–52, January 2001. doi:10.1038/35051009.

44 Cynthia Kop, Aart Middeldorp, and Thomas Sternagel. Complexity of conditional term rewriting. *Logical Methods in Computer Science*, Volume 13, Issue 1, February 2017. doi:10.23638/lmcs-13(1:6)2017.

45 Cynthia Kop and Deivid Vale. Cost-size semantics for call-by-value higher-order rewriting. In *8th International Conference on Formal Structures for Computation and Deduction (FSCD 2023)*, volume 260, pages 15:1–15:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICS.FSCD.2023.15.

46 Hlér Kristjánsson, Tatsuki Odake, Satoshi Yoshida, Philip Taranto, Jessica Bavaresco, Marco Túlio Quintino, and Mio Murao. Exponential separation in quantum query complexity of the quantum switch with respect to simulations with standard quantum circuits, 2024. arXiv:2409.18420.

47 Dallas S. Lankford. On proving term rewriting systems are noetherien. Technical report, Department of Mathematics, Louisiana technical University, 1979. URL: https://www.ens-lyon.fr/LIP/REWRITING/TERMINATION/Lankford_Poly_Term.pdf.

48 Salvador Lucas and José Meseguer. Dependency pairs for proving termination properties of conditional term rewriting systems. *Journal of Logical and Algebraic Methods in Programming*, 86(1):236–268, January 2017. doi:10.1016/j.jlamp.2016.03.003.

49 Jean-Yves Marion. Analysing the implicit complexity of programs. *Information and Computation*, 183(1):2–18, May 2003. doi:10.1016/s0890-5401(03)00011-7.

50 Richard Mayr and Tobias Nipkow. Higher-order rewrite systems and their confluence. *Theoretical Computer Science*, 192(1):3–29, February 1998. doi:10.1016/S0304-3975(97)00143-6.

51 Georg Moser and Michael Schaper. Automated expected cost analysis for quantum programs. *CoRR*, 2026. doi:10.48550/arxiv.2604.03971.

52 Johannes Niederhauser and Aart Middeldorp. *The Computability Path Order for Beta-Eta-Normal Higher-Order Rewriting*, page 207–225. Springer Nature Switzerland, 2025. doi:10.1007/978-3-031-99984-0_12.

53 Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, June 2012. doi:10.1017/cbo9780511976667.

54 Lars Noschinski, Fabian Emmes, and Jürgen Giesl. Analyzing innermost runtime complexity of term rewriting by dependency pairs. *Journal of Automated Reasoning*, 51(1):27–56, March 2013. doi:10.1007/s10817-013-9277-6.

55 Amr Sabry, Benoît Valiron, and Juliana Kaizer Vizzotto. From symmetric pattern-matching to quantum control. In *Foundations of Software Science and Computation Structures*, pages 348–364, Cham, 2018. Springer International Publishing. doi:10.1007/978-3-319-89366-2_19.

56 Peter Selinger and Benoît Valiron. Semantic techniques in quantum computation. In *Quantum lambda calculus*, pages 135–172. Cambridge University Press, 2009. doi:10.1017/cbo9781139193313.005.

57 Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997. doi:10.1137/S0097539795293172.

58 Jakob Grue Simonsen. *The Π_2^0 -Completeness of Most of the Properties of Rewriting Systems You Care About (and Productivity)*, page 335–349. Springer Berlin Heidelberg, 2009. doi:10.1007/978-3-642-02348-4_24.

59 Márcio M. Taddei, Jaime Cariñe, Daniel Martínez, Tania García, Nayda Guerrero, Alastair A. Abbott, Mateus Araújo, Cyril Branciard, Esteban S. Gómez, Stephen P. Walborn, Leandro Aolita, and Gustavo Lima. Computational advantage from the quantum superposition of multiple temporal orders of photonic gates. *PRX Quantum*, 2:010320, February 2021. doi:10.1103/PRXQuantum.2.010320.

60 Terese. *Term rewriting systems*, volume 55 of *Cambridge tracts in theoretical computer science*. Cambridge University Press, 2003.

61 Finn Voichick, Liyi Li, Robert Rand, and Michael Hicks. Qunity: A unified language for quantum and classical computing. *Proc. ACM Program. Lang.*, 7(POPL), January 2023. doi:10.1145/3571225.

62 Toshiyuki Yamada. Confluence and Termination of Simply Typed Term Rewriting Systems. In *Rewriting Techniques and Applications*, pages 338–352, Berlin, Heidelberg, 2001. Springer. doi:10.1007/3-540-45127-7_25.

63 Tomoyuki Yamakami. A Schematic Definition of Quantum Polynomial Time Computability. *The Journal of Symbolic Logic*, 85(4):1546–1587, December 2020. doi:10.1017/jsl.2020.45.

64 Andrew Chi-Chih Yao. Quantum circuit complexity. In *Proceedings of 1993 IEEE 34th Annual Foundations of Computer Science*, pages 352–361, November 1993. doi:10.1109/SFCS.1993.366852.

65 Mingsheng Ying. *Foundations of quantum programming*. Elsevier, 2024. doi:10.1016/C2014-0-02660-3.

66 Charles Yuan, Agnes Villanyi, and Michael Carbin. Quantum control machine: The limits of control flow in quantum programming. *Proceedings of the ACM on Programming Languages*, 8(OOPSLA1):1–28, April 2024. doi:10.1145/3649811.

A Additional Material

▶ **Example 42.** Let us come back on the STRS defined in Example 5, and prove that it is a QTRS. The function symbol $\mathtt{f}$ comes with the signature $\mathrm{type}(\mathtt{f}) = \mathtt{nat} \to \mathtt{Qbit} \to \mathtt{Qbit}$. First, prove the typing condition; and suppose that $\Gamma; \Delta \vdash_{\mathfrak{R}} \mathtt{f}(n, |0\rangle) : T$. By Lemma 54, as the term terminates, its typing tree contains only classical rules. The only applyable rule is the symbol rule. Going upwards gives the following typing tree:

$$\dfrac{\dfrac{}{\Gamma, n : \mathtt{nat}; \varnothing \vdash_{\mathfrak{R}} n : \mathtt{nat}} \quad \dfrac{}{\Gamma, n : \mathtt{nat}; \varnothing \vdash_{\mathfrak{R}} |0\rangle : \mathtt{Qbit}}}{\Gamma, n : \mathtt{nat}; \varnothing \vdash_{\mathfrak{R}} \mathtt{f}(n, |0\rangle) : \mathtt{Qbit}}$$

Thus, imposing $T = \mathtt{Qbit}$ and $\Delta = \emptyset$. The right term can be typed identically as follows,

$$\dfrac{\dfrac{}{\Gamma, n : \mathtt{nat} \Vdash_{\mathfrak{R}} \mathtt{sgn}^+} \quad \dfrac{\dfrac{}{\Gamma, n : \mathtt{nat}; \varnothing \vdash_{\mathfrak{R}} n : \mathtt{nat}}}{\Gamma, n : \mathtt{nat} \Vdash_{\mathfrak{R}} \mathtt{a}^+(n)} \quad \dfrac{}{\Gamma, n : \mathtt{nat}; \varnothing \vdash_{\mathfrak{R}} |i\rangle : \mathtt{Qbit}} \quad |0\rangle \perp_{\mathfrak{R}} |1\rangle \; \forall \sigma \in \mathrm{Sub}_0(\Gamma, n : \mathtt{nat}), \|[\![\mathtt{sgn}^+]\!]\|^2 + \|[\![\mathtt{a}^+(\underline{\iota})]\!]\|^2 = 1}{\Gamma, n : \mathtt{nat}; \varnothing \vdash_{\mathfrak{R}} \mathtt{f}(n, |0\rangle) : \mathtt{Qbit}}$$

The amplitude condition can be verified straightforwardly, for any substitution of n by a ground natural number $\underline{\iota}$. This process can be done for the other rewrite rule, and validates the second condition of Definition 8. It now remains to show that $\mathtt{f}$ is an isometry. Consider two inputs $v, w \in \mathrm{In}(\mathtt{f})$. By typing, they are of the shape $v = (\alpha \cdot |0\rangle + \beta \cdot |1\rangle, \underline{\iota})$ and $w = (\gamma \cdot |0\rangle + \delta \cdot |1\rangle, \underline{\iota'})$. if $v \perp_{\mathfrak{R}} w$, this implies the following:

$$\begin{aligned} S = {} & [\![\alpha]\!][\![\gamma]\!]^* \delta_\perp((|0\rangle, \underline{\iota}), (|0\rangle, \underline{\iota'})) + [\![\alpha]\!][\![\delta]\!]^* \delta_\perp((|0\rangle, \underline{\iota}), (|1\rangle, \underline{\iota'})) \\ & + [\![\beta]\!][\![\gamma]\!]^* \delta_\perp((|1\rangle, \underline{\iota}), (|0\rangle, \underline{\iota'})) + [\![\beta]\!][\![\delta]\!]^* \delta_\perp((|1\rangle, \underline{\iota}), (|1\rangle, \underline{\iota'})) = 0 \end{aligned}$$

For $\delta_\perp$ to be defined, it imposes $\underline{\iota} = \underline{\iota'}$; and the equality implies $[\![\alpha]\!][\![\gamma]\!]^* + [\![\beta]\!][\![\delta]\!]^* = 0$. Now, it remains to prove that $\mathtt{f}(\alpha \cdot |0\rangle + \beta \cdot |1\rangle, \underline{\iota}) \perp_{\mathfrak{R}} \mathtt{f}(\alpha \cdot |0\rangle + \beta \cdot |1\rangle, \underline{\iota})$. Developping each term through $\equiv$, and then reducing them via (Q) yields the two reduced:

$$(\alpha \cdot \mathtt{sgn}^+ + \beta \cdot \mathtt{a}^-(\underline{\iota})) \cdot |0\rangle + (\alpha \cdot \mathtt{a}^+(\underline{\iota}) + \beta \cdot \mathtt{sgn}^-) \cdot |1\rangle$$

and

$$(\gamma \cdot \mathtt{sgn}^+ + \delta \cdot \mathtt{a}^-(\underline{\iota})) \cdot |0\rangle + (\gamma \cdot \mathtt{a}^+(\underline{\iota}) + \delta \cdot \mathtt{sgn}^-) \cdot |1\rangle$$

These terms are also expressed as canonical forms. Again, expressing the orthogonality yields the following, as $\delta_\perp(|i\rangle, |1 - i\rangle) = 0$:

$$S = [\![\alpha \cdot \mathtt{sgn}^+ + \beta \cdot \mathtt{a}^-(\underline{\iota})]\!][\![\gamma \cdot \mathtt{sgn}^+ + \delta \cdot \mathtt{a}^-(\underline{\iota})]\!]^* + [\![\alpha \cdot \mathtt{a}^+(\underline{\iota}) + \beta \cdot \mathtt{sgn}^-]\!][\![\gamma \cdot \mathtt{a}^+(\underline{\iota}) + \delta \cdot \mathtt{sgn}^-]\!]^*$$

By computing the amplitudes, we obtain $S = [\![\alpha]\!][\![\gamma]\!]^* + [\![\beta]\!][\![\delta]\!]^* = 0$. Therefore, $\mathtt{f}$ is an isometry, and $\mathfrak{R}$ is a QTRS.

B Proofs of Section 3

B.1 Proofs of Section 3.1

We first introduce an operator $\theta_s(t)$, called the *quantity*, which aims to measure the amplitude of a classical term s in a term t. This will be in particular useful to show Lemma 47. Suppose $\mathcal{A}$ contains two symbols $\mathtt{a}_0, \mathtt{a}_1$, where $[\![\mathtt{a}_0]\!] = 0$ and $[\![\mathtt{a}_1]\!] = 1$. Given a term t, denote $\mathrm{Sub}_0(\mathtt{FV}(t))$ as the set of substitutions such that $t\sigma$ is a ground term.

▶ **Definition 43.** *Let s be a classical term. We define the following map θ_s from terms to amplitudes as follows:*

$$\begin{aligned}
\theta_s(\mathtt{x}) &\triangleq \delta_{s,\mathtt{x}} \\
\theta_s(\mathtt{b}(t_1,\dots,t_n)) &\triangleq \begin{cases} \prod_{i=1}^n \theta_{s_i}(t_i) & \text{if } s = \mathtt{b}(s_1,\dots,t_n) \\ \mathtt{a}_0 & \text{else} \end{cases} \\
\theta_s(t_1+t_2) &\triangleq \theta_s(t_1)+\theta_s(t_2) \\
\theta_s(\alpha\cdot t) &\triangleq \alpha\cdot\theta_s(t)
\end{aligned}$$

where the notation $\delta_{s,t}$ outputs $\mathtt{a}_1$ if $s=t$, and $\mathtt{a}_0$ else.

▶ **Lemma 44.** *Let s be a classical term. Then $\theta_s(s)=\mathtt{a}_1$.*

Proof. Proven by induction on s, direct by definition of the quantity. ◀

▶ **Lemma 45.** *Let s be a classical term and t,t' be two terms. If $t\equiv t'$, then for any substitution $\sigma\in \mathrm{Sub}_0(\mathtt{FV}(t))\cap\mathrm{Sub}_0(\mathtt{FV}(t'))$, $[\![\theta_s(t)\sigma]\!]=[\![\theta_s(t')\sigma]\!]$.*

Proof. By induction on each rule defining $\equiv$ in Figure 1; the transitive and reflexive case can be derived from this. The first two lines of rules are verified directly as $\mathbb{C}$ is a vector space; while the linearity rules follow directly the case of summation in Definition 43. The last rule with the equivalence context can be proven by induction, on the construction of the context, the base case being verified above. ◀

▶ **Lemma 46.** *Let s,t be two classical terms. Then $\theta_s(t)=\delta_{s,t}$.*

Proof. Direct by induction on t; as it is classical, only the first two cases of Definition 43 are considered. ◀

As a consequence, if $t=\sum_{i=1}^n \alpha_i\cdot t_i$ with t_i being classical terms, $\theta_s(t)=\sum_{i=1}^n \alpha_i\cdot\delta_{s,t_i}$.

▶ **Lemma 47** (Unicity)**.** *If t_0 has a canonical form, then it is unique, up to reordering and amplitude equivalence.*

Proof. Suppose t_0 has two canonical forms $t=\sum_{i=1}^n \alpha_i\cdot t_i$ and $s=\sum_{j=1}^m \beta_j\cdot s_j$, thus $s\equiv t$. Take any substitution $\sigma\in\mathrm{Sub}_0(\mathtt{FV}(s))\cap\mathrm{Sub}_0(\mathtt{FV}(t))$, and any $1\le i_0\le n$. By definition of the quantity and by Lemma 46, $[\![\theta_{t_{i_0}}(t)\sigma]\!]=[\![\alpha_{i_0}\sigma]\!]$, which by definition of a canonical form, is different from 0. By Lemma 45, $[\![\theta_{t_{i_0}}(t)\sigma]\!]=[\![\theta_{t_{i_0}}(s)\sigma]\!]=\sum_{j=1}^m [\![\beta_j]\!]\cdot\delta_{t_{i_0},s_j}$. If $\delta_{t_{i_0},s_j}=\mathtt{a}_0$ for all j, then $[\![\theta_{t_{i_0}}(s)\sigma]\!]=0$, which will contradict Lemma 45; and if there are two j where it is $\mathtt{a}_1$, then we will have $s_j=t_{i_0}=s'_j$ which contradicts the fact that s is a canonical form. Therefore, there exists j_0 such that $t_{i_0}=s_{j_0}$, and α_{i_0} is equivalent to β_{j_0}. Thus, any t_i has a unique match in (s_j), therefore $n\le m$. The same process can be done the other way around, so $m\le n$, thus $m=n$. So each sum contains the same number of elements, such that there is a map from $1,\dots,n$ to $1,\dots,n$ that associates i with j such that $t_i=s_j$ and $\alpha_i=\beta_j$. Therefore, both sums are equal up to permutation of the t_i. ◀

▶ **Lemma 48.** *Let $\mathfrak{R}$ be a STRS, and t be a term. Then, either $t\equiv\mathtt{a}_0\cdot t'$, or $t\equiv\sum_{i=1}^n\alpha_i\cdot t_i\in\mathtt{CAN}$, and this canonical form is unique, up to sum reordering and amplitude equivalence.*

Proof. Unicity being direct from Lemma 47, let us prove either the existence of a canonical form or $t\equiv\mathtt{a}_0\cdot t'$, by induction on the syntax of t. Note that if $t\equiv\mathtt{a}_0\cdot t'$, then t has no canonical form.

- If $t = \mathtt{x}$, then t has $\mathtt{a}_1 \cdot \mathtt{x}$ as a canonical form, which is equivalent to t directly by $\equiv$.
- Suppose $t = \mathtt{b}(t_1, \dots, t_n)$. By induction, either $t_i \equiv \mathtt{a}_0 \cdot t_i$ is verified for some i, in which case $t \equiv \mathtt{a}_0 \cdot \mathtt{b}(t_1, \dots, t_n)$; else, all t_i possess a canonical form. Therefore, we have $t_i \equiv \sum_{j_i=1}^{m_i} \alpha_{j_i} \cdot t_{j_i}$. By developing t under $\equiv$, we have:

$$t \equiv \sum_{j_1=1}^{m_1} \cdots \sum_{j_n=1}^{m_n} \alpha_{j_1} \dots \alpha_{j_n} \cdot \mathtt{b}(t_{j_1}, \dots, t_{j_n})$$

  This can be seen as the following sum, where $\alpha_k = \alpha_{j_1} \dots \alpha_{j_n}$:

$$t \equiv \sum_{k=1}^{m_1 \times \cdots \times m_n} \alpha_k \cdot \mathtt{b}(t_1^k, \dots, t_n^k)$$

  In particular, $[\![\alpha_k]\!] \neq 0$ as $[\![\alpha_{j_i}]\!] \neq 0$; and as, for fixed i, all t_{j_i} are pairwise distinct, so are the $\mathtt{b}(t_1^k, \dots, t_n^k)$. Therefore, this is a canonical form.
- Suppose $t = \sum_{i=1}^{n} \alpha_i \cdot t_i$. By induction hypothesis, either $t_i \equiv \mathtt{a}_0 \cdot t_i$, in which case it can be removed through $\equiv$, or it possesses a canonical form. One can thus rewrite $t \equiv \sum_{j=1}^{m} \alpha'_j \cdot t'_j$, where t'_j are the t_i such that $t'_i \not\equiv \mathtt{a}_0 \cdot t''_i$. Let us denote p_k as the set of all classical terms present in the canonical form of at least one t'_j. Up to adding them with $\equiv$ ($t \equiv t + \mathtt{a}_0 \cdot p_k$), we can write each t'_j as $\sum_{k=1}^{l} \beta_{jk} \cdot p_k$, and thus write t as follows:

$$t \equiv \sum_{j=1}^{m} \alpha'_j \cdot (\sum_{k=1}^{l} \beta_{jk} \cdot p_k) \equiv \sum_{k=1}^{l} \gamma_k \cdot p_k$$

  with $\gamma_k = \sum_{j=1}^{m} \alpha'_j \beta_{jk}$. By definition, all p_k are classical and pairwise distinct. Now, either $[\![\gamma_k]\!] = 0$ for all k, and $t \equiv \mathtt{a}_0 \cdot t'$; or $[\![\gamma_k]\!] \neq 0$ holds for some k, and the obtained term is a canonical form, up to removing the other terms. ◀

▶ **Lemma 49** (Weakening). *Let $\mathfrak{R}$ be a QTRS, and let $\Gamma; \Delta \vdash_{\mathfrak{R}} t : T$ be a well-typed term. Then, for any $\Gamma' \supseteq \Gamma$ such that $\Gamma' \cap \Delta = \varnothing$, $\Gamma'; \Delta \vdash_{\mathfrak{R}} t : T$.*

Proof. Direct by induction on typing. ◀

▶ **Lemma 50.** *Let $\mathfrak{R}$ be a STRS, and let s, t be two terms. Suppose $s \equiv t$, $s \to_{\mathfrak{R}} s'$ and $t \to_{\mathfrak{R}} t'$. Then, $t \equiv t'$.*

Proof. This is proven by induction on the size of the derivation of both $\to_{\mathfrak{R}}$ relations.

If either s or t reduces by the rule (E), assuming s without loss of generality, then $s \equiv s_1$, $s_1 \to_{\mathfrak{R}} s'_1$ and $s'_1 \equiv s'$. Thus, $t \equiv s_1$, by induction hypothesis, $s'_1 \equiv t'$, and by transitivity of $\equiv$, $s' \equiv t'$. We now suppose that both s and t do not reduce via (E).

If s, t both reduce via the rule (C), then they are both classical. By quantity, $1 = [\![\theta_s(s)]\!] = [\![\theta_s(t)]\!] = [\![\delta_{s,t}]\!]$, thus $s = t$, with $s = E[l\sigma]$ and $t = E'[l'\sigma']$. As E reduces from left to right and $s = t$ then $E = E'$; and as $\mathcal{R}_{\mathfrak{R}}$ is orthogonal, there is a unique choice for the rewrite rule, thus $l = l'$, and $\sigma = \sigma'$. Therefore, by the rule (C), $s' = t'$.

Suppose s, t both reduce via (Q), thus $t = \sum_{i=1}^{n} \alpha_i \cdot t_i$, and $s = \sum_{j=1}^{m} \beta_j \cdot s_j$ (same for s', t' by adding $'$ on t_i and s_j). By Lemma 47, their canonical form are equal, up to reordering and amplitude equivalence; in particular, they have the same number of elements. Therefore, for any t_i, there exists s_j such that $t_i = s_j$; either they are normal forms, in which case $t'_i = s'_j$; or they both reduce, in which case by induction hypothesis $t'_i \equiv s'_j$. Furthermore, for these terms, α_i is equivalent to β_i; thus one can rewrite $t' \equiv t' + (\beta_j + \mathtt{a}_{min} \cdot \alpha_i) \cdot t_i$, with

$[\![\mathtt{a}_{min}]\!] = -1$, as $[\![\beta_j + \mathtt{a}_{min} \cdot \alpha_i]\!] = 0$; this thus transforms $\alpha_i \cdot t_i$ in $\beta_j \cdot t_i$. Therefore, starting from t', one can reorder all terms as in s', replace α_i by β_j, and replace t_i by s_j; all of this can be done via $\equiv$, and we reach s'. Therefore, $t' \equiv s'$.

Finally, suppose s reduces via (C) and t via (Q), thus $t = \sum_{i=1}^{n} \alpha_i \cdot t_i$. By unicity of the canonical form, as $\mathtt{a}_1 \cdot s$ is a canonical form, $t = \alpha \cdot t_1$, where $t' = s$ and $[\![\alpha]\!] = 1$. By definition, t_1 must reduce to some t_1'; by induction hypothesis, $s' \equiv t_1'$. As $[\![\alpha]\!] = 1$, $t' = \alpha \cdot t_1' \equiv t_1' = s'$, thus we conclude.

◀

▶ **Lemma 10** (Confluence). *Let $\mathfrak{R}$ be a STRS, and let t be a term. Suppose that there exist two terms t_1, t_2 such that $t \to_{\mathfrak{R}}^* t_1$ and $t \to_{\mathfrak{R}}^* t_2$. Then, there exist two terms t_3, t_4 such that $t_1 \to_{\mathfrak{R}}^* t_3$, $t_2 \to_{\mathfrak{R}}^* t_4$ and $t_3 \equiv t_4$.*

Proof. The fact that the canonical form is unique is direct by Lemma 47. Let us prove semi-confluence, which implies confluence. Let t be a term, suppose $t \to_{\mathfrak{R}} t_1$ and $t \to_{\mathfrak{R}}^* t_2$, and prove that there exist t_3, t_4 such that $t_1 \to_{\mathfrak{R}}^* t_3$, $t_2 \to_{\mathfrak{R}}^* t_4$, and $t_3 \equiv t_4$. This is done by induction on the number of rewrite steps of $t \to_{\mathfrak{R}}^* t_2$. If it is done in 0 steps, then take $t_4 = t_3 = t_1$. If it is done in one step, then by Lemma 50, $t_1 \equiv t_2$, thus take $t_3 = t_1$ and $t_4 = t_2$. If it is done in two steps or more, i.e., $t \to_{\mathfrak{R}} t' \to_{\mathfrak{R}} t'' \to_{\mathfrak{R}}^* t_2$, by Lemma 50, $t' \equiv t_1$; via (E) rule of Figure 2, $t_1 \to_{\mathfrak{R}} t''$. Therefore, take $t_3 = t_4 = t_2$. ◀

▶ **Lemma 51.** *Let $\mathfrak{R}$ be a QTRS, and let t be a term. Suppose $t \to_{\mathfrak{R}} t_1$. Then, $t \not\equiv \mathtt{a}_0 \cdot t'$.*

Proof. Direct by induction on $\to_{\mathfrak{R}}$; in the case where t is classical, conclude by Lemma 45 by checking $\theta_t(t)$ and $\theta_t(\mathtt{a}_0 \cdot t)$. ◀

▶ **Lemma 52.** *Let $\mathfrak{R}$ be a QTRS, and let s be a term. Suppose $s \to_{\mathfrak{R}} t$. Then $s \equiv \sum_{i=1}^{n} \alpha_i \cdot s_i \in \mathtt{CAN}$, $t \equiv \sum_{i=1}^{n} \alpha_i \cdot t_i$, and $s_i \to_{\mathfrak{R}}^? t_i$.*

Proof. First, Lemma 51 states that $s \not\equiv \mathtt{a}_0 \cdot s'$, and thus Lemma 48 implies that s has a canonical form $\sum_{i=1}^{n} \alpha_i \cdot s_i \in \mathtt{CAN}$. Now, by induction on the derivation of $s \to_{\mathfrak{R}} t$:

- If $s = C[\sigma l]$, as it is a classical term, $s \equiv \mathtt{a}_1 \cdot s \in \mathtt{CAN}$, and $t \equiv \mathtt{a}_1 \cdot t$, thus we conclude directly;
- If s is equivalent to some s', s' has the same canonical form as s, thus by induction hypothesis we can conclude.
- If s is directly a canonical form, we conclude. ◀

▶ **Lemma 10** (Confluence). *Let $\mathfrak{R}$ be a STRS, and let t be a term. Suppose that there exist two terms t_1, t_2 such that $t \to_{\mathfrak{R}}^* t_1$ and $t \to_{\mathfrak{R}}^* t_2$. Then, there exist two terms t_3, t_4 such that $t_1 \to_{\mathfrak{R}}^* t_3$, $t_2 \to_{\mathfrak{R}}^* t_4$ and $t_3 \equiv t_4$.*

To ease the notations, the rules of Figure 3 typing term equivalence, a symbol $\mathtt{b}$, and a superposition, are respectively denoted with labels (equiv), (symb), and (sup) in the following proofs.

A typing derivation is said to be *classical*, if it contains neither (sup) nor (equiv) typing rules.

▶ **Lemma 53.** *Let $\mathfrak{R}$ be a QTRS, and let $\Gamma; \Delta \vdash_{\mathfrak{R}} t : T$ be a well-typed classical term. Then, it can be typed by a classical typing derivation.*

Proof. First show that for any $\Gamma; \Delta \vdash_{\mathfrak{R}} t : T$ well-typed term, and any classical term s, if $[\![\theta_s(t)]\!] \neq 0$, then $\Gamma; \Delta \vdash_{\mathfrak{R}} s : T$ holds with a classical typing derivation. This is proven by induction on the typing of t.

- If t is typed as a variable, then it is classical, thus $[\![\theta_s(\mathtt{x})]\!] \neq 0$ implies that $s = \mathtt{x}$, thus we conclude.
- If $t = \mathtt{b}(t_1, \dots, t_n)$, then quantity implies that $s = \mathtt{b}(s_1, \dots, s_n)$ with $[\![\theta_{s_i}(t_i)]\!] \neq 0$. By induction hypothesis, we obtain typing of s_i with same type and context as t_i via a classical derivation, thus we conclude.
- If $t \equiv t'$, Lemma 45 implies that $[\![\theta_s(t')]\!] \neq 0$, thus we apply the induction hypothesis and conclude, as t' and t have the same type and context.
- Finally, suppose $t = \sum_{i=1}^{n} \alpha_i \cdot t_i$. By quantity, if $[\![\theta_s(t_i)]\!] = 0$ for all i, then $[\![\theta_s(t)]\!] \neq 0$ fails. Thus, apply the induction hypothesis on $\theta_s(t_i)$ where $[\![\theta_s(t_i)]\!] \neq 0$ holds, and conclude as t_i has same type and context as t.

We conclude by applying this result for $t = s$, as $\theta_s(s) = \mathtt{a}_1$ by Lemma 44. ◀

▶ **Lemma 54.** *Let $\mathfrak{R}$ be a QTRS, and let $\Gamma; \Delta \vdash_{\mathfrak{R}} t : T$ be a well-typed terminating term. Then, it can be typed as follows:*

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma; \Delta \vdash_{\mathfrak{R}} s_0 : T}}{\Gamma; \Delta \vdash_{\mathfrak{R}} s_1 : T}\ (sup)}{\vdots}}{\dfrac{}{\Gamma; \Delta \vdash_{\mathfrak{R}} s_n : T}\ (sup)} \qquad s_n \equiv t}{\Gamma; \Delta \vdash_{\mathfrak{R}} t : T}\ (equiv)$$

where t types s_0 via a classical typing derivation, and then is followed by $n \geq 0$ (sup) typing rules, and one (equiv) typing rule.

Proof. The derivation is built by modifying the original typing derivation of t, with the following remarks. Note that below, when we say that a typing rule (a) commutes with (b), we only mean that a derivation with root (b) and direct premise (a) can be rewritten as a derivation with a root (a) taking as direct premise (b).

- First, an (equiv) typing rule commutes with any other rule, thanks to the last rule of Figure 1; and by transitivity of $\equiv$, two (equiv) rules can be merged together. Any typing derivation for the following case can be considered with only one (equiv) rule at the end.
- We also need to prove that (sup) commutes with (symb) for terminating terms, up to adding an (equiv) rule. Assuming that (sup) happens in the first slot without loss of generality, if $\mathtt{b}(\sum_{i=1}^{n} \alpha_i \cdot t_i, \dots, s_m)$ is typable through (symb) and (sup), we want to prove that $\sum_{i=1}^{n} \alpha_i \cdot \mathtt{b}(t_i, \dots, s_m)$ is typable through (sup) and (symb), thus the initial term is typable through (equiv), (sup) and (symb). The contexts and types will be correct for the typing to happen, and the phases are normalized, but the difficult part is to show that if $t_i \perp_{\mathfrak{R}} t_j$, then $\mathtt{b}(t_i, \dots, s_m) \perp_{\mathfrak{R}} \mathtt{b}(t_j, \dots, s_m)$. However, if $\mathtt{b}$ is a constructor, then the terms will reduce to $\mathtt{b}(v_i, \dots, w_m)$, where $v_i \perp_{\mathfrak{R}} v_j$ as $t_i \perp_{\mathfrak{R}} t_j$, thus orthogonality is recovered; and if $\mathtt{b}$ is a function symbol, it must be an isometry by definition of a QTRS, and thus orthogonality is also preserved (up to reducing the t_i to their normal forms v_i, which can be done as they are orthogonal, thus terminate). Furthermore, considered terms always terminate. Indeed, either it is part of a construct from the original terminating term t; else, it must have been introduced through $\equiv$, but thus introduced a summation, which, if typed, implies that it terminates.

Using these remarks, one is able to commute the rules accordingly to obtain the expected structure. ◀

▶ **Lemma 55.** *Let $\mathfrak{R}$ be a QTRS, and let $\vdash_{\mathfrak{R}} s : T$ be a well-typed classical term. Suppose $s = t\sigma$, for t a left-linear classical term. Then, there exist Γ, Δ such that $\Gamma; \Delta \vdash_{\mathfrak{R}} t : T$, and $\sigma \in \mathrm{Sub}_0(\Gamma \cup \Delta)$.*

Proof. Recall that σ maps variables to classical values. By induction on the syntax of t, either $t = \mathtt{x}$, in which case $\Gamma; \Delta \vdash_{\mathfrak{R}} \mathtt{x} : T$ is valid through one of the two variable typing rules, taking $\mathrm{Dom}(\Gamma) \cup \mathrm{Dom}(\Delta) = \{\mathtt{x}\}$; and σ maps $\mathtt{x}$ to s, which is a classical and closed value of good type. Else, $t = \mathtt{b}(t_1, \dots, t_n)$, which implies that $s = \mathtt{b}(s_1, \dots, s_n)$ by action of a substitution, with $s_i = t_i\sigma$. As s is classical, Lemma 53 states that it has a classical typing derivation, thus we have $\mathrm{type}(\mathtt{b}) = T_1 \times \cdots \times T_n \to T$, and $\Gamma; \Delta_i \vdash_{\mathfrak{R}} t_i : T_i$. By induction on each t_i, $\Gamma; \Delta_i \vdash_{\mathfrak{R}} s_i : T_i$, and $\sigma \in \mathrm{Sub}_0(\Gamma \cup \Delta_i)$. Typing with (symb) yields the expected typing result; and as t is left-linear, variables in the t_i are disjoint, and thus we can conciliate all possible cases to obtain $\sigma \in \mathrm{Sub}_0(\Gamma \cup \Delta)$, and conclude. ◀

▶ **Lemma 56.** *Let $\mathfrak{R}$ be a QTRS, let $\Gamma; \Delta \vdash_{\mathfrak{R}} t : T$ be a well-typed term, and let $\Gamma \Vdash_{\mathfrak{R}} \alpha$ be a well-typed amplitude. Then, for any substitution $\sigma \in \mathrm{Sub}_0(\Gamma \cup \Delta)$, $\vdash_{\mathfrak{R}} t\sigma : T$ and $\Vdash_{\mathfrak{R}} \alpha\sigma$.*

Proof. By induction on typing of both amplitudes and terms.

- If $t = \mathtt{x}$, then $\mathtt{x} : T \in \Gamma \cup \Delta$, thus $\vdash_{\mathfrak{R}} s\sigma : T$ is direct by definition of a context substitution.
- Suppose $t \equiv s$, with $\Gamma; \Delta \vdash_{\mathfrak{R}} s : T$. By induction hypothesis, $\vdash_{\mathfrak{R}} s\sigma : T$, and it is easy to verify that $s\sigma \equiv t\sigma$, as σ maps to classical values, thus the structure stays identical. Therefore, $\vdash_{\mathfrak{R}} t\sigma : T$.
- If $t = \mathtt{b}(t_1, \dots, t_n)$, typing implies that $\Delta = \Delta_1, \dots, \Delta_n$, with $\Gamma; \Delta_i \vdash_{\mathfrak{R}} t_i : T_i$. Now, σ is, a fortiori, a $\Gamma \cup \Delta_i$-context substitution. Therefore, by induction hypothesis, $\vdash_{\mathfrak{R}} t_i\sigma : T_i$, thus $\vdash_{\mathfrak{R}} \mathtt{b}(t_1\sigma, \dots, t_n\sigma) : T$, and conclude as $\mathtt{b}(t_1\sigma, \dots, t_n\sigma) = t\sigma$.
- Suppose $t = \sum_{i=1}^{n} \alpha_i \cdot t_i$. By typing, $\Gamma; \Delta \vdash_{\mathfrak{R}} t_i : T$, thus $\vdash_{\mathfrak{R}} t_i\sigma : T$ by induction hypothesis. By typing, $\Gamma \Vdash_{\mathfrak{R}} \alpha_i$, thus $\Vdash_{\mathfrak{R}} \alpha_i\sigma$ by induction hypothesis; By typing, $\sum_{i=1}^{n} |[\![\alpha_i\delta]\!]|^2 = 1$ is true for any substitution $\delta \in \mathrm{Sub}_0(\Gamma)$, thus as σ is, a fortiori, a Γ-context substitution, it is true for σ, and now $\alpha_i\sigma$ are closed, thus no need to verify it for another substitution. Finally, $t_i \perp t_j$ implies that for any substitution $\delta \in \mathrm{Sub}_0(\Gamma \cup \Delta)$, t_i, t_j reduce to some value which verifies some properties. Again, this is true in particular for σ, and now $t_i\sigma$ is closed, thus orthogonality is verified between $t_i\sigma$ and $t_j\sigma$. Therefore, one is able to type $\vdash_{\mathfrak{R}} \sum_{i=1}^{n} (\alpha_i\sigma) \cdot t_i\sigma : T$, which, by definition of σ, is equal to $t\sigma$.
- If $\alpha = \mathtt{a}(\alpha_1^c, \dots, \alpha_n^c)$, this is direct by induction as for $\mathtt{b}(t_1, \dots, t_n)$.
- The case for $\alpha_1 + \alpha_2$ and $\alpha_1 \cdot \alpha_2$ are also direct by induction and by definition of a substitution. ◀

▶ **Lemma 11** (Subject reduction). *Let $\mathfrak{R}$ be a QTRS, and let $\vdash_{\mathfrak{R}} s : T$ be a well-typed term that is either terminating or classical. If $s \to_{\mathfrak{R}} t$, then $\vdash_{\mathfrak{R}} t : T$.*

Proof. By induction on $\to_{\mathfrak{R}}$.

- Suppose $s = C[l\sigma]$ and $t = C[r\sigma]$. As s is classical, Lemma 53 states that it can be typed by a classical typing derivation, which thus will coincide with each symbol of C. Therefore, $\vdash_{\mathfrak{R}} l\sigma : T'$; as QTRS are left-linear, so is l. By Lemma 55, there exist Γ, Δ such that $\Gamma; \Delta \vdash_{\mathfrak{R}} l : T'$, and $\sigma \in \mathrm{Sub}_0(\Gamma \cup \Delta)$. As $\mathfrak{R}$ is a QTRS, $\Gamma; \Delta \vdash_{\mathfrak{R}} r : T'$. By Lemma 56, $\vdash_{\mathfrak{R}} r\sigma : T'$; and one can type t by the same derivation as s, by replacing $l\sigma$ by $r\sigma$, as it has the same context and type, and it is a classical derivation, thus no need to derive orthogonality.
- Suppose that $s \equiv s'$ and $t \equiv t'$ with $s' \to_{\mathfrak{R}} t'$. As $\equiv$ preserves typing, $\Gamma; \Delta \vdash_{\mathfrak{R}} s' : T$; by induction hypothesis, $\Gamma; \Delta \vdash_{\mathfrak{R}} t' : T$; and by equivalence again, $\Gamma; \Delta \vdash_{\mathfrak{R}} t : T$.

- Finally, suppose that $s = \sum_{i=1}^{n} \alpha_i \cdot s_i \in \mathtt{CAN}$ and $t = \sum_{i=1}^{n} \alpha_i \cdot t_i$, with $s_i \to_{\mathfrak{R}}^{?} t_i$ for all i. As stated in Lemma 54, consider a typing derivation where all equivalence relations are at the end, thus $s \equiv s'$, where s' is well-typed, with no equivalence relation inside its derivation. If s' is classical and s is a canonical form, then $s = 1 \cdot s'$ by quantity; by induction hypothesis, this implies $\Gamma; \Delta \vdash_{\mathfrak{R}} t' : T$, and thus $\Gamma; \Delta \vdash_{\mathfrak{R}} t : T$ by $\equiv$, with $t = 1 \cdot t'$. Now, suppose s' is not classical, and thus s' terminates. Lemma 54 implies the following, with $s' \equiv s''$:

$$s'' = \sum_{i_1=1}^{m_1} \alpha_{i_1} \cdot (\cdots \sum_{i_n=1}^{m_n} \beta_{i_1 \dots i_n} \cdot p_{i_1 \dots i_n}),$$

  where the p_{ij} are classical terms. Developing all the sums yields a canonical form, up to removing potential terms with a phase equivalent to 0; and this is exactly s, by unicity of the canonical form (Lemma 47). By each (sup) rule, we also have $\Gamma; \Delta \vdash_{\mathfrak{R}} p_{i_1 \dots i_n} : T$, and such term is equal to some s_i. As $s_i \to_{\mathfrak{R}}^{?} t_i$, either $s_i = t_i$, and typing is direct, or $s_i \to_{\mathfrak{R}} t_i$, in which case we can apply the induction hypothesis. One can thus replace each $p_{i_1 \dots i_n}$ with its (potential) reduced $q_{i_1 \dots i_n}$; and as they are typed, and are reduction of orthogonal terms, thus still orthogonal, we are able to type back each (sup) rule, thus to type t', and conclude. ◀

▶ **Lemma 57.** *Let $t \equiv \sum_{i=1}^{n} \alpha_i \cdot t_i$. Suppose that $t \to_{\mathfrak{R}}^{*} v$. Then, there exist s_i such that $t_i \to_{\mathfrak{R}}^{*} s_i$ and $v \equiv \sum_{i=1}^{n} \alpha_i \cdot s_i$.*

Proof. By induction on the length k of the chain $t \to_{\mathfrak{R}}^{*} v$. Note that $\sum_{i=1}^{n} \alpha_i \cdot s_i$ is not a canonical form, as s_i may contain superpositions. The case $k = 0$ being direct, suppose $t \to_{\mathfrak{R}} t' \to_{\mathfrak{R}}^{k} v$. As t reduces, Lemma 51 implies that $t \not\equiv \mathtt{a}_0 \cdot t'$, thus t has a canonical form, which can be built as in the proof of Lemma 48, i.e., discard any t_i where $[\![\alpha_i]\!] = 0$ or $t_i \equiv \mathtt{a}_0 \cdot t_i'$, and for the remaining terms, write them as canonical forms up to complementation $t_i \equiv \sum_{j=1}^{m} \beta_{ij} \cdot p_j$. Therefore, $t \equiv \sum_{j=1}^{m} (\sum_{i=1}^{n} \alpha_i \beta_{ij}) \cdot p_j$. By Lemma 52, reduction of t implies that $t' \equiv \sum_{j=1}^{m} (\sum_{i=1}^{n} \alpha_i \beta_{ij}) \cdot q_j$, with $p_j \to_{\mathfrak{R}}^{?} q_j$, again up to discarding possibly some q_j where $[\![\sum_{i=1}^{n} \alpha_i \beta_{ij}]\!] = 0$. For each non-discarded t_i, remark that $t_i' = \sum_{j=1}^{m} \beta_{ij} \cdot q_j$ satisfies either $t_i' = t_i$ if all $p_j \in \mathtt{NF}_{\mathfrak{R}}$, or $t_i \to_{\mathfrak{R}} t_i'$. In any case, $t_i \to_{\mathfrak{R}}^{*} t_i'$ is satisfied, and up to readding the terms discarded at the beginning through $s \equiv t + 0 \cdot s'$, we obtain $t' \equiv \sum_{i=1}^{n} \alpha_i \cdot t_i'$, We can then conclude by applying the induction hypothesis on t', and thus $t_i \to_{\mathfrak{R}}^{*} t_i' \to_{\mathfrak{R}}^{*} s_i$, therefore $t_i \to_{\mathfrak{R}}^{*} s_i$. ◀

▶ **Lemma 13** (Canonical form). *Let $\mathfrak{R}$ be a QTRS, and let $\Gamma; \Delta \vdash_{\mathfrak{R}} t : T$ be a well-typed term. Then, t has a unique canonical form $\sum_{i=1}^{n} \alpha_i \cdot t_i$ up to reordering and amplitude equivalence, and $\Gamma; \Delta \vdash_{\mathfrak{R}} t_i : T$.*

Proof. We prove that $t \not\equiv \mathtt{a}_0 \cdot t'$ by induction on the typing rules of t. The fact that it implies that t has a unique canonical form is a consequence of Lemma 48. Typing of each t_i is obtained directly by induction on the typing of t, looking at the way the canonical form is built in Lemma 48.

- If $t = \mathtt{x}$, then we conclude directly as $\mathtt{a}_1 \cdot \mathtt{x}$ is a canonical form.
- It $t \equiv s$, then $s \not\equiv \mathtt{a}_0 \cdot s'$ by induction hypothesis, and thus $t \not\equiv \mathtt{a}_0 \cdot t'$ by quantity.
- Suppose $t = \mathtt{b}(t_1, \dots, t_n)$. Looking at the proof of Lemma 48, $t \equiv \mathtt{a}_0 \cdot t'$ would imply that $t_i \equiv \mathtt{a}_0 \cdot t_i'$ for some i, which is impossible by induction hypothesis.
- Finally, suppose $t = \sum_{i=1}^{n} \alpha_i \cdot t_i$. By typing $t_i \not\equiv \mathtt{a}_0 \cdot t_i'$, thus we can take their canonical form, and up to adding $\mathtt{a}_0 \cdot s$, we can write them as sharing the same classical terms: $t_i \equiv \sum_{j=1}^{m} \beta_{ij} \cdot s_j$, where s_j are classical terms and pairwise distinct, and thus $t \equiv$

$\sum_{j=1}^{m} \gamma_j \cdot s_j$, where $\gamma_j = \sum_{i=1}^{n} \alpha_i \beta_{ij}$. If $t \equiv 0 \cdot t$, this means that $\gamma_j = 0$ for all j. As $t_i \perp t_j$, given any substitution $\sigma \in \mathrm{Sub}_0(\Gamma \cup \Delta)$, $t_i\sigma \to_{\mathfrak{R}}^* v_i$. By Lemma 57, there exist q_j such that $v_i \equiv \sum_{j=1}^{m} \beta_{ij} \cdot q_j$. Therefore, $v \triangleq \sum_{i=1}^{n} \alpha_i \cdot v_i \equiv \sum_{j=1}^{m} \gamma_j \cdot q_j \equiv 0 \cdot v$. However, expressing each v_i in its canonical form, again by possibly complementing them, yields $v_i \equiv \sum_{k=1}^{l} \delta_{ik} \cdot r_k$, and thus $v \equiv \mathtt{a}_0 \cdot v'$ implies that for all k, $\sum_{i=1}^{n} \alpha_i \delta_{ik} = 0$. Again, as $t_i \perp t_j$, $v_i \perp v_j$, and thus $\sum_{k=1}^{l} \delta_{ik}\delta_{jk} = 0$. Furthermore, by Definition 6, v_i has a canonical form, thus $v_i \not\equiv \mathtt{a}_0 \cdot v_i'$. Therefore, $(\gamma_{ik})_k$ are non-zeros orthogonal vectors. One can thus invert the matrix $G = (\gamma_{ik})_{ik}$, and the equality $\sum_{i=1}^{n} \alpha_i \delta_{ik} = 0$ would imply that all $\alpha_i = 0$, which is not true as it is of norm 1. Therefore, $t \not\equiv \mathtt{a}_0 \cdot t'$. ◀

▶ **Lemma 14** (Normal forms). *Let $\mathfrak{R}$ be a total QTRS. Then, $\mathtt{NF}_{\mathfrak{R}}$ is exactly the set of well-typed ground terms equivalent to values.*

Proof. Prove the set equality by double inclusion, each time proving the contraposed result. Let us first show that any term that reduces is not equivalent to a value. This is done by induction on the reduction $\to_{\mathfrak{R}}$. If $t = C[l\sigma]$, then t is classical. In particular, if t is equivalent to a value v, by quantity, $1 = [\![\theta_t(t)]\!] = [\![\theta_t(v)]\!] = 0$, as they are ground terms, thus $t \not\equiv v$. If $t \equiv t_1$, with t_1 reducing, by induction hypothesis, t_1 is not equivalent to a value, thus so is t, else t_1 would be. Finally, if $t = \sum_{i=1}^{n} \alpha_i \cdot t_i \in \mathtt{CAN}$, reduction imposes that some t_i is not equivalent to a value. If t is equivalent to a value, by quantity, $0 \neq [\![\alpha_i]\!] = [\![\theta_{t_i}(t)]\!] = [\![\theta_{t_i}(v)]\!] = 0$, which concludes.

Now, prove that any term not equivalent to a value does reduce. To do so, we first prove that any well-typed classical term t that is not a value reduces through (C). By induction on the typing of $t = \mathtt{b}(t_1, \dots, t_n)$. Either t_i is not a value, and thus $t_i = C[l\sigma]$, and it reduces to $C[r\sigma]$ for $l \to r \in \mathcal{R}_{\mathfrak{R}}$. Taking $C' = \mathtt{b}(t_1, \dots, C, \dots, t_n)$ shows reduction of t. Else, if all t_i are values, then $\mathtt{b}$ must be a function symbol, else t is a value. As $\mathfrak{R}$ is total, there exist $l \to r \in \mathcal{R}_{\mathfrak{R}}$ and σ such that $\mathtt{f}(t_1, \dots, t_n) = l\sigma$, thus t reduces with $C = \diamond$. Now consider any general term t. As t is well-typed, then it possesses a canonical form $\sum_{i=1}^{n} \alpha_i \cdot t_i \in \mathtt{CAN}$ by Lemma 13. Now, as t is not equivalent to a value, at least one t_i must be distinct from a value; as it is classical, the above result tells us that t_i reduces. Therefore, $\sum_{i=1}^{n} \alpha_i \cdot t_i$ will reduce through (Q), as one element of the sum reduces; we conclude by reducing t through (E). ◀

B.2 Proofs of Section 3.2

▶ **Lemma 58.** *Given a term t, there is a procedure, which may not terminate, to generate its typing derivation without any equivalence rule.*

Proof. By induction on the syntax of t. If $t = \mathtt{x}$, return a typing derivation $*; * \vdash_{\mathfrak{R}} t : *$, indicating that t can be typed with any context or type. If t is the term we want to type, then it can be typed, by choosing any type. Suppose $t = \mathtt{b}(t_1, \dots, t_n)$. By induction hypothesis, either some t_i cannot be typed, in which case so does t. Now, suppose each t_i can be typed. By hypothesis, t_i must be of type T_i, with $\mathrm{type}(\mathtt{b}) = T_1 \times \cdots \times T_n \to T$, else it cannot be typed. In the case where $t_i = \mathtt{x}$, choose the according type, where its contexts containing a single variable, $\mathtt{x}$. Therefore, $\Gamma_i; \Delta_i \vdash_{\mathfrak{R}} t_i : T_i$. One need to check that for any variable appearing in one of the contexts, either it appears in a single linear-context, or in multiple non-linear contexts, with the same type; if not, t cannot be typed. This procedure is decidable, as the contexts are finite. Finally, denote $\Gamma = \cup_i \Gamma_i$; by weakening, $\Gamma; \Delta_i \vdash_{\mathfrak{R}} t_i : T_i$ holds, and thus, $\Gamma; \cup_i \Delta_i \vdash_{\mathfrak{R}} t : T$ holds.

Finally, consider $t = \sum_{i=1}^{n} \alpha_i \cdot t_i$. Again, if one t_i cannot be typed, then so does t. Else, consider each typing derivation $\Gamma_i; \Delta_i \vdash_{\mathfrak{R}} t_i : T_i$. If all T_i are not identical or do not yield

a quantum type, typing cannot be inferred. In the case where $t_i = \mathtt{x}$, either $n = 1$ and then type it as a qubit; if $n > 1$, t_i cannot be all variables, else orthogonality will not hold, thus take the type from the other terms. Now, one need to check that the contexts can be reconciled as previously. Furthermore, we need to check that the phases and orthogonality conditions are verified between terms, which may be or not decidable. ◀

▶ **Theorem 15** (Undecidability of type inference). *Given a STRS $\mathfrak{R}$, deciding whether t can be typed is Π_2^0-hard, and belongs to Σ_3^0.*

Proof. Define the free typing inference as the problem of whether a term t can be typed with a typing derivation without any equivalence rule. First, prove that free typing inference is Π_2^0-complete. [58] showed that deciding whether a constructor TRS is strongly normalizing, i.e., terminates on all inputs, is Π_2^0-complete. Suppose we have a STRS with one function symbol $\mathtt{f}$, and one wants to type $t = \mathtt{a}_{\frac{1}{\sqrt{2}}} \cdot (\mathtt{f}(x), |0\rangle) + \mathtt{a}_{\frac{1}{\sqrt{2}}} \cdot (\mathtt{f}(x), |1\rangle)$. If t is typed, it must be typed via the superposition rule, thus $(\mathtt{f}(\mathtt{x}), |0\rangle) \perp_{\mathfrak{R}} (\mathtt{f}(\mathtt{x}), |1\rangle)$, which imposes that $\mathtt{f}$ must terminate over any possible input, i.e., $\mathfrak{R}$ is strongly normalizing. Therefore, if one can decide $\perp_{\mathfrak{R}}$, thus typing, it can decide strongly normalization, thus free typing inference is Π_2^0-hard. To prove completeness, we must show that it belongs to Π_2^0. Taking the procedure from Lemma 58, the only undecidable properties to check are verifying the conditions in the superposition rule. Deciding orthogonality between two terms s, t is Π_2^0, as it can be seen as follows:

$$\forall \sigma, \exists k \in \mathbb{N}, s\sigma \to_{\mathfrak{R}}^* \sum_{i=1}^{n} \alpha_i \cdot v_i \in \mathtt{CAN} \wedge t\sigma \to_{\mathfrak{R}}^* \sum_{j=1}^{m} \beta_j \cdot w_j \in \mathtt{CAN} \wedge \sum_{i=1}^{n} \sum_{j=1}^{m} [\![\alpha_i]\!][\![\beta_j]\!]^* \delta_\perp(v_i, w_j) = 0$$

The condition on the two summations is decidable, as we have restricted ourselves to terms in $\bar{\mathbb{C}}$, where summation, product and zero equality are decidable. Furthermore, $\delta_\perp(-,-)$ is also decidable, as it can be checked inductively on the syntax of the term; therefore, it is Π_2^0. The test for amplitudes condition is, by definition, and as we consider only algebraic numbers, Π_1^0. The overall check is thus Π_2^0. As these undecidable checks are done a finite amount of times, as t has a finite syntax, the overall free type inference is Π_2^0.

Lemma 54 showed that any typing derivation can be written with an unique equivalence relation at the end, and with an above typing derivation which, in particular, has no equivalence relation. Therefore, type inference of t implies that there exists s such that $t \equiv s$ and s is typable with no equivalence rule, thus it is Σ_3^0 by definition of the arithmetical hierarchy. ◀

▶ **Theorem 16** (Decidability of easy type inference). *Let $\mathfrak{R}$ be a terminating STRS, where $\mathcal{A}$ contains only symbols of arity 0. There is a polynomial P such that easy type inference of t is decidable in $P(|t|)$, and implies that t is well-typed.*

Proof. Looking back at the procedure from Lemma 58, one is able to ensure the following points:

- When one types t, the contexts we choose contain exactly the variables in t. Therefore, the size of the contexts are bounded by the size of t, and any check on whether contexts can be conciliated is done in $P(|t|)$.
- The check on amplitude symbols is done in polynomial time, as amplitudes are ground, thus there is a unique substitution (the empty substitution) for which the check is done. On algebraic numbers, this is computable, in a constant time; therefore, as the number of amplitudes to sum is bounded by the size of the term, it can be checked linearly.

- Computing the orthogonal Kronecker product on classical terms is done linearly in the syntax of the term.

Therefore, at each step of the induction, one does a finite number of recursive calls, and does a finite number of checks, each in polynomial time (denote the total complexity Q) Therefore, there exist a polynomial P (depending on Q) such that type inference is done in $P(|t|)$.

To show that t is well-typed, we need to ensure that $\delta_\perp(s,t) = 0$ implies $s \perp_{\mathfrak{R}} t$. Suppose $s = C[|0\rangle, s_1, \dots, s_n]$ and $t = C[|1\rangle, t_1, \dots, t_n]$ without loss of generality, and take any substitution σ. As $\mathfrak{R}$ is terminating, we obtain $s\sigma \to^*_{\mathfrak{R}} C[|0\rangle, v_1, \dots, v_n]$ and $t\sigma \to^*_{\mathfrak{R}} C[|1\rangle, w_1, \dots, w_n]$; By linearity, develop each v_i, and obtain the related canonical form, denoted v, w. Note that as C is classical, the elements of their canonical forms are of the shape $v_i = C[|0\rangle, v_1^i, \dots, v_n^i]$ (similar for w). One thus needs to compute their inner product. However, for any v_i, w_j, they are still of the shape $C[|0\rangle, \dots]$ and $C[|1\rangle, \dots]$, thus $\delta_\perp(v_i, w_j) = 0$, yielding directly 0. Therefore, s, t are orthogonal. ◀

B.3 Proofs of Section 3.3

▶ **Theorem 21** (Universality). *Let $(C_n)_{n\in\mathbb{N}}$ be a uniform family of circuits. Then, there exists a QTRS $\mathfrak{R} \in \mathtt{QTRSCirc}$ of main symbol $\mathtt{f}$ such that, for any $n \in \mathbb{N}$ and any n-qubit state $|\phi\rangle$, $\mathtt{f}(t_n, t_{|\phi\rangle}) \to^*_{\mathfrak{R}} t_{C_n|\phi\rangle}$.*

Proof. By [60, Section 5.3.1], there exists a constructor TRS $\mathfrak{R}_1$ of main symbol $\mathtt{g}$ such that for any $n \in \mathbb{N}$, $\mathtt{g}(\mathtt{S}_n)$ outputs the representation of C_n, as $\to_{\mathfrak{R}}$ behaves standardly for classical terms. As it is made only of classical rewrite rules, it can be seen and typed as a QTRS. Furthermore, as it contains finitely many constant symbols, they can be represented using a combination of lists and natural numbers, thus in $\mathcal{C}_Q$. As it treats only classical data, two distinct rules do not share the same input structure, and thus function symbols are structure preserving. By the same argument, all symbols quantum controls, taking 0 control qubits. Let $\mathcal{U}$ be the universal set of gates used in the representation. Any circuit representation can be seen as a chain of wire swaps, applying a gate $U \in \mathcal{U}$, and another set of wire swaps. Any n-qubit unitary can be encoded as a QTRS with 2^n rules with no variables, and it will satisfy Definition 8 as it is an unitary; we can encode a swap between the first two elements of a list easily; any general swap can then be built from this. All these function encode an unitary by definition, and are structure preserving. This allows us to build a QTRS $\mathfrak{R}_2$ with main symbol $\mathtt{h}$, where $\mathtt{h}(C, t_{|\phi\rangle}) \to^*_{\mathfrak{R}} t_{C_n|\phi\rangle}$ for C a circuit representation of C_n. As $\mathtt{h}$ can be built with a unique rule $\mathtt{h}(x) \to r$, with r a composition of multiple gates, it satisfies the properties of $\mathtt{QTRSCirc}$. We then conclude by taking the QTRS $\mathfrak{R}$, where all the symbols and variables are joined, up to renaming, i.e. $\mathcal{A} = \mathcal{A} \uplus \mathcal{A}$, similarly for the other sets; and with one added function symbol $\mathtt{f}$, with one rewrite rule $\mathtt{f}(\mathtt{S}_n, x) \to \mathtt{h}(\mathtt{g}(\mathtt{S}_n), x)$, which is the main symbol of $\mathfrak{R}$. ◀

▶ **Theorem 25.** *Take $\mathfrak{R} \in \mathtt{QTRS}(\tau)$ of main symbol $\mathtt{f}$. Then, for any $\overline{v} \in \mathrm{In}(\mathtt{f})$, $|C_{\mathtt{struct}(\overline{v})}| = \mathcal{O}(\tau(|\overline{v}|)^{\mathrm{rk}(\mathtt{f})+1})$, where $C_{\mathtt{struct}(\overline{v})} \in \mathtt{C}(\mathfrak{R})$ is the compiled circuit of corresponding structure.*

Proof. Let $w \in \mathtt{S}(\mathtt{f})$; denote $N = |\mathcal{R}_{\mathfrak{R}}| + 1$ and $k = |w|$. We first produce inductively a circuit C_w^t by induction on the syntax of t, and $C_w^{\mathtt{b}}$ for $\mathtt{b} \in \mathcal{C} \uplus \mathcal{F}$; where the obtained circuit, for terms and function symbols, is of size bounded by $(|\mathcal{R}_{\mathfrak{R}}|T(|w|))^{\mathrm{rk}(\mathtt{g})} = K$, for $C_w^{\mathtt{g}}$ and C_w^t, where $\mathtt{g}$ is the function symbol in t of maximum rank; the size of the $C_w^{\mathtt{c}}$ will be constant. From that, we approximate each gate, using the chosen universal set of gates, up to a precision $1 - \epsilon$, such that the overall circuit has a precision $\frac{2}{3}$. This approximation, done on each gate, can be done using Solovay-Kitaev theorem [42], which is known to

approximate the original gate to a precision $1-\epsilon$ in time $\mathcal{O}(\log^c(1/\epsilon))$, for a constant c. Choosing $\epsilon = \frac{2}{3K}$ satisfies the overall precision [53], and thus, our approximated circuit is of size $\mathcal{O}(K\log^c(\frac{3K}{2})) = \mathcal{O}(T(k)^{\mathrm{rk}(\mathtt{f})+1})$.

We may label the wires when they correspond to variables, to be able to merge circuits easily. Note that, at any step of the induction, the structure of any variable is known, as we know the starting structure, and $\mathtt{FV}(l) \supseteq \mathtt{FV}(r)$ for any rewrite rule. At each step, we may also do simplification steps, such as removing a gate controlled on a wire that is $|0\rangle$. This removes any ill-path that would not terminate, or not in the correct number of steps.

Start the induction with the symbol $\mathtt{f}$. As each step of induction corresponds to one reduction, and the QTRS terminates (this is a consequence of the structure preserving property), the induction will terminate.

Suppose the induction step considers a classical term t. If $t = \mathtt{x}$, it yields the circuit with one wire, labelled with $\mathtt{x}$. If $t = \mathtt{b}(t_1, \ldots, t_n)$, then inductively compute the circuit for each t_i, stack them vertically (in parallel), and merge their outputs with the inputs of the circuit generated by $\mathtt{b}$. As the structure of each t_i is known, the structure of the inputs of $\mathtt{b}$ is also known, thus the number of input wires will be coherent with the total number of wires of the t_i, and thus wires can be merged. The number of rewrite steps satisfies $T(k) = \sum_{i=1}^n T_i + T'$, where T_i is the number of rewrite steps of t_i, and T' of $\mathtt{b}$. By induction, it thus satisfies $|C^t_w| = \sum_{i=1}^n |C^{t_i}_w| + |C^{\mathtt{b}}_w| \leq \sum_{i=1}^n (T_i N)^{\mathrm{rk}(\mathtt{g})} + (T_i N)^{\mathrm{rk}(\mathtt{g})} \leq ((N)\sum_{i=1}^n T_i + T')^{\mathrm{rk}(\mathtt{g})} = (T(|k|)N)^{\mathrm{rk}(\mathtt{g})}$, where $\mathtt{g}$ is the function symbol with the maximum rank between $\mathtt{b}$, and the function symbol of maximum rank in t.

Let us now construct the circuits for symbols $\mathtt{b} \in \mathcal{C} \biguplus \mathcal{F}$. Suppose that $\mathtt{b}$ is a constructor, and enumerate all possible cases: $\mathtt{0}, \mathtt{S}, [\,]$ yield no circuit nor wire; $|i\rangle$ initializes an ancillary qubit; for (h, t) and $h :: t$, concatenate vertically both the wires of h and of t; this is just wire concatenation, so no gate and thus of size 0.

Finally, consider the case where $\mathtt{b}$ is a function symbol $\mathtt{g}$. If $\mathtt{g}$-rewrite rules map to values, it can be implemented directly by an isometry, thus an unitary gate; remark that as we know the classical structure, any variable in an amplitude is fully known, and thus the amplitude can be interpreted as a scalar. Else, suppose that $\mathtt{g}$ quantum controls. Consider the $\mathtt{g}$-structural set of rewrite rules with the structure corresponding to the one fixed, and denote it $\mathcal{R}_{\mathtt{g}} = \{l_i \to r_i \in \mathcal{R}\}$. As it quantum controls, all of these rewrite rules only differ on their qubit constructors. Therefore, there exist contexts $C, C' \in \mathcal{C}$ such that $l_i = C[|j_{i1}, \ldots, |j_{im}\rangle\rangle]$ and $r_i = C'[|j_{i1}\rangle, \ldots, |j_{im}\rangle, t_{i1}, \ldots, t_{il}]$. One can thus view each r_i as $r_i = \mathtt{h}(|j_{i1}\rangle, \ldots, |j_{im}\rangle, t_{i1}, \ldots, t_{il})$, with $\mathtt{h}$ being a function symbol with one rule $\mathtt{h}(x_1, \ldots, x_m, q_1, \ldots, q_l) \to C'[x_1, \ldots, x_m, q_1, \ldots, q_l]$, which corresponds to the computation. Therefore, for each r_i, compile each t_{ik} to a circuit inductively, controlled by the $|j_{i1}\rangle, \ldots, |j_{im}\rangle$, except for any possible sub symbol $\mathtt{g}$; as there is no symbol equivalent to $\mathtt{g}$ other than itself, t_{ik}, apart from $\mathtt{g}$, will be of rank $\mathrm{rk}(\mathtt{g}) - 1$; furthermore, t_{ik} terminates in T_i steps, and $\sum_{k=1}^l T_{ik} \leq T(k)$. As this process is done for each i, thus at most $N-1$ times, this yields a total circuit of size $(N-1)(\sum_{k=1}^l (T_{ik}N)^{\mathrm{rk}(\mathtt{g})-1}) \leq (N-1)(T(k)N)^{\mathrm{rk}(\mathtt{g})-1}$. Then, following the merging process from [36, Theorem 9], as all calls to $\mathtt{g}$ are done on an identical input, one is able to rewrite all calls to $\mathtt{g}$ as a single call; then, one compiles $\mathtt{g}$ inductively. This induction step can be done at most $T(k)$ times, as we have done one rewrite step; therefore, the size of the circuit is bounded by $\frac{(T(k)N)^{\mathrm{rk}(\mathtt{g})}(N-1)}{N}$, which can be bounded by $(T(k)N)^{\mathrm{rk}(\mathtt{g})} - 1$, as $\mathrm{rk}(\mathtt{g}) \geq 1$. After all the controlled statements of each r_i, we compile the circuit corresponding to $\mathtt{h}$, corresponding to one gate, thus yielding the expected bound. Note that as $\mathtt{g}$ preserves the structure, each t_{ik} yields the same number of wires. Furthermore, as $\mathtt{g}$ quantum controls, each left-handside contains the same variables, thus

each r_i has the same variables. To connect each circuit of r_i, one just has to connect the control wires together, the variable wires together, and finally to possibly merge the added ancillary qubit wires, up to adding NOT gates to flip them to $|0\rangle$ or $|1\rangle$. Finally, with $\mathtt{h}$, merge the control wires together, and merge each wire from t_{ik} with the wire labelled by q_k. ◀

▶ **Theorem 22** (Circuit compilation). *Let $\mathfrak{R} \in \mathtt{QTRSCirc}$ be a terminating QTRS of main symbol $\mathtt{f}$. Then, we can generate a family of circuits $\mathtt{C}(\mathfrak{R}) \triangleq (C_w)_{w \in \mathtt{S(f)}}$ s.t., for any input $\overline{v} \in \mathrm{In}(\mathtt{f})$, if $\mathtt{f}(\overline{v}) \to^*_{\mathfrak{R}} v'$ then $C_{\mathtt{struct}(\overline{v})}$ approximates v' with probability $\frac{2}{3}$ on input $\overline{v}$.*

Proof. The same compilation process can be done as in the proof of Theorem 25; as $\mathfrak{R}$ terminates, it does in $\mathtt{Time}(\tau)$ for some τ; and $\mathtt{SRec}$ only plays a role in the size of the circuit. ◀

▶ **Theorem 27** ($\mathtt{FBQP}$ characterization). $\{\!\{\mathtt{QTRS}(Poly)\}\!\} = \mathtt{FBQP}$.

Proof. Let us first show soundness, i.e., let $\mathfrak{R} \in \mathtt{QTRS}(Poly)$, and show that any binary function computed by $\mathtt{C}(\mathfrak{R})$ belongs to $\mathtt{FBQP}$, i.e., $\mathtt{C}(\mathfrak{R})$ is an uniform polynomially-sized family of circuits. As $\mathfrak{R} \in \mathtt{QTRS}(Poly)$, there is a polynomial Q such that $\mathfrak{R}$ terminates in time Q. Therefore, by Theorem 25, the circuit is of size $\mathcal{O}(P(Q(n)))$, thus of polynomial size in n. Note that we have indexed the circuits by n rather than w, but $|w| = n$. The fact that this family is uniform is obtained by building the circuit as done in the proof of Theorem 25: one can build a Turing machine, taking as input the size n (thus the shape), and suppose that it contains initially the encoding of $\mathfrak{R}$, thus without the input, onto a tape of the QTM. As the circuit is build inductively on the syntax, and it differs only by the input structure, this creation is uniform, and is done in polynomial time, as the generated circuit is of polynomial size.

We now prove completeness, i.e., for any function f in $\mathtt{FBQP}$, there exist $\mathfrak{R} \in \mathtt{QTRS}(Poly)$ such that $\mathtt{C}(\mathfrak{R})$ computes f. We prove this result by using Yamakami's algebra [63]. This paper defines a function algebra, and then proves that it completely characterizes $\mathtt{FBQP}$. Formally, it defines the function class $\square_1^{\mathrm{QP}}$, which is the smallest class of functions including the gates below, for $\theta \in [0, 2\pi) \cap \tilde{\mathbb{C}}$ where $\tilde{\mathbb{C}}$ denotes the complex numbers whose real and imaginary parts can both be approximated by a polynomial-time Turing machine,

$$
\begin{aligned}
I(|\phi\rangle) &\triangleq |\phi\rangle \\
PHASE_\theta(|\phi\rangle) &\triangleq |0\rangle\langle 0|\phi\rangle + e^{i\theta}|1\rangle\langle 1|\phi\rangle \\
ROT_\theta(|\phi\rangle) &\triangleq \cos\theta\,|\phi\rangle + \sin\theta(|1\rangle\langle 0|\phi\rangle - |0\rangle\langle 1|\phi\rangle) \\
NOT(|\phi\rangle) &\triangleq |1\rangle\langle 0|\phi\rangle + |0\rangle\langle 1|\phi\rangle \\
SWAP(|\phi\rangle) &\triangleq \begin{cases} |\phi\rangle & \text{if } l(|\phi\rangle) \leq 1 \\ \sum_{a,b \in \{0,1\}} |ab\rangle\langle ba|\phi\rangle & \text{otherwise} \end{cases}
\end{aligned}
$$

and is closed under the following schemes:

$$
\begin{aligned}
Compo[g,h](|\phi\rangle) &\triangleq g \circ h(|\phi\rangle) \\
Branch[g,h] &\triangleq \begin{cases} |\phi\rangle & \text{if } l(|\phi\rangle) \leq 1 \\ |0\rangle \otimes g(\langle 0|\phi\rangle) + |1\rangle \otimes h(\langle 1|\phi\rangle) & \text{otherwise} \end{cases} \\
kQRec_t[g,h,p|\mathcal{F}_k](|\phi\rangle) &\triangleq \begin{cases} g(|\phi\rangle) & \text{if } l(|\phi\rangle) \leq t \\ h(\sum_{w \in \{0,1\}^k} |w\rangle \otimes f_w(\langle w|p(|\phi\rangle)\rangle)) & \text{otherwise} \end{cases}
\end{aligned}
$$

Here, $|\phi\rangle$ is a quantum state, i.e., belongs to the Hilbert space $\mathbb{C}^{2^n}$ for a given n written in Dirac notation. Its size $l(|\phi\rangle)$ is n. Given $w \in \{0,1\}^n$ with $n \leq l(|\phi\rangle)$, we may write $|\phi\rangle = \sum_i \alpha_i |w_i z_i\rangle$, where $w_i \in \{0,1\}^n$ and $z_i \in \{0,1\}^{l(|\phi\rangle)-n}$; we then abuse the notation by writing $\langle w|\phi\rangle = \sum_i \alpha_i \langle w|w_i\rangle |z_i\rangle$.

This class is proven to be FBQP complete, thus any function of FBQP can be written as a function $\square_1^{\mathrm{QP}}$. We therefore associate any function $\square_1^{\mathrm{QP}}$ with a given QTRS, $\overline{\square_1^{\mathrm{QP}}}$, and prove inductively that they belong in $\mathtt{QTRS}(Poly)$. By abuse of notation, we only write the corresponding set of rules of the QTRS below; these QTRS can be merged altogether, up to renaming. We also denote $\overline{f}_m$ as the main function symbol of the QTRS $\overline{f}$.

One can remark that for any function $f \in \square_1^{\mathrm{QP}}$, and any input $w \in \{0,1\}^*$, $f|w\rangle = v$, and the main symbol of $\overline{f}$, $\overline{f}_m$, satisfies $\mathtt{f}([\![w]\!]) \to_{\mathfrak{R}}^* v$, thus they compute the same function (here $[\![w]\!]$ is the notation from Theorem 21). As $\overline{f}$ is approximated by the family of circuits $\mathtt{C}(\overline{f})$ by Theorem 25, then f is also approximated by this family, with precision $\frac{2}{3}$, thus giving us the wanted result. The following phases are used: $[\![\mathtt{a}_\theta]\!] = e^{i\theta}$, $[\![\mathtt{a}^c_\theta]\!] = \cos\theta$, $[\![\mathtt{a}^s_\theta]\!] = \sin\theta$, $[\![\mathtt{sgn}]\!] = -1$.

$$
\begin{aligned}
\overline{I} &\triangleq \{\mathtt{g}(x) \to x\} \\
\overline{Ph_\theta} &\triangleq \{\mathtt{g}([\,]) \to [\,], \mathtt{g}(|0\rangle :: t) \to |0\rangle :: t, \mathtt{g}(|1\rangle :: t) \to \mathtt{a}_\theta \cdot |1\rangle :: t\} \\
\overline{Rot_\theta} &\triangleq \left\{ \begin{array}{c} \mathtt{g}([\,]) \to [\,] \\ \mathtt{g}(|0\rangle :: t) \to \mathtt{a}^c_\theta \cdot |0\rangle :: t + \mathtt{a}^s_\theta \cdot |1\rangle :: t \\ \mathtt{g}(|1\rangle :: t) \to \mathtt{a}^s_\theta \cdot \mathtt{sgn} \cdot |0\rangle :: t + \mathtt{a}^c_\theta \cdot |1\rangle :: t \end{array} \right\} \\
\overline{Not} &\triangleq \{\mathtt{g}([\,]) \to [\,], \mathtt{g}(|0\rangle :: t) \to |1\rangle :: t, \mathtt{g}(|1\rangle :: t) \to |0\rangle :: t\} \\
\overline{SWAP} &\triangleq \{\mathtt{g}([\,]) \to [\,], \mathtt{g}(h :: [\,]) \to h :: [\,], \mathtt{g}(h :: h' :: t) \to h' :: h :: t\} \\
\overline{COMP[f,g]} &\triangleq \{\mathtt{g}(x) \to \overline{f}_m(\overline{g}_m(x))\} \\
\overline{Branch[f,g]} &\triangleq \left\{ \begin{array}{c} \mathtt{g}([\,]) \to [\,] \\ \mathtt{g}(h :: [\,]) \to h :: [\,] \\ \mathtt{g}(|0\rangle :: h :: t) \to |0\rangle :: \overline{f}_m(h :: t) \\ \mathtt{g}(|1\rangle :: h :: t) \to |1\rangle :: \overline{g}_m(h :: t) \end{array} \right\}
\end{aligned}
$$

All terms belong in QTRSCirc by construction, and as functions have no recursive construct, they terminate in a fixed time, independently of the input, and thus they belong in $\mathtt{QTRS}(Poly)$. The definition of this function in [63] comes with $f_w \in \mathcal{F}_k$, where $\mathcal{F}_k$ is a set of functions $\{f_s\}_{s \in \{0,1\}^k}$, where each f_s is either I or $kQRec_t[g,h,p|\mathcal{F}_k]$ itself. Therefore, interpreting f_s as either $\overline{I}$ or rec, one is able to build the corresponding QTRS of main symbol rec.

$$
\overline{kQRec_t[g,h,p|\mathcal{F}_k]} \triangleq \left\{ \begin{array}{c} \mathtt{f}([\,]) \to [\,] \\ \vdots \\ \mathtt{f}(h_1 :: \ldots h_k :: [\,]) \to h_1 :: \cdots :: h_k :: [\,] \\ \mathtt{f}(s_1 :: \ldots s_k :: l) \to s_1 :: \cdots :: s_k :: \overline{f_{s_1 \ldots s_k m}}(l) \\ \mathtt{rec}([\,]) \to \overline{g}_m([\,]) \\ \vdots \\ \mathtt{rec}(h_1 :: \ldots h_t :: [\,]) \to \overline{g}_m(h_1 :: \cdots :: h_t :: [\,]) \\ \mathtt{rec}(h_1 :: \ldots h_t :: l) \to \overline{h}_m(\mathtt{f}(\overline{p}_m(h_1 :: \cdots :: h_t :: l))) \end{array} \right\}
$$

Note that h_i are variables, while s_i are qubit constructors, thus $\mathtt{f}$ has $2^k + 2$ rules. By definition, this QTRS belongs to $\mathtt{QTRSCirc}$, as the only structural set of rewrite rules with more than one rule is for the 2^k rules of $\mathtt{f}$ with different $s_1, \dots, s_k$ which satisfy the definition; all other satisfy the definition of quantum controls with $C' = \diamond$, As recursive calls are done by $\mathtt{f}_q$ on a similar input, then it also belongs to $\mathtt{SRec}$. Furthermore, terminating in polynomial time is guaranteed, as:

- By induction, each g, h, p computes in polynomial time;
- Either f_s is the identity, thus stops here, or is a recursive call on an input of size that decreases by k;
- Therefore, the compute time is roughly, for an input of size n, $\frac{n}{k}\max(P_g(n), P_h(p), P_p(n))$, where P_g, P_h, P_p is the compute time of respectively g, h, p, which is polynomial, by induction hypothesis. The obtain time is thus polynomial in n. ◀

C Proofs of Section 4

▶ **Lemma 28.** *Let $\mathfrak{R}$ be a STRS, and let $t = \sum_{i=1}^{n} \alpha_i \cdot t_i$ be a well-typed term. If t starts an infinite chain on the order induced by $\to_{\mathfrak{R}}$, then so does t_i for some $1 \leq i \leq n$.*

Proof. Suppose $t \to_{\mathfrak{R}} t_1 \to_{\mathfrak{R}} \dots$. By Lemma 52, it implies that $t \equiv \sum_{j=1}^{m} \beta_j \cdot s_j \in \mathtt{CAN}$, $t_1 \equiv \sum_{j=1}^{m} \beta_j \cdot s'_j$, and $s_j \to^{?}_{\mathfrak{R}} s'_j$. Now, if $t = \sum_{i=1}^{n} \alpha_i \cdot t_i$, one can rewrite t as its canonical form, i.e., remove amplitudes equivalent to 0, view $t_i \equiv \sum_{j=1}^{m} \gamma_{ij} \cdot s_j$ and $\beta_j = \sum_{i=1}^{n} \alpha_i \cdot \gamma_{ij}$. One can rewrite t_1 as $t_1 \equiv \sum_{i=1}^{n} \alpha_i \cdot (\sum_{j=1}^{m} \gamma_{ij} \cdot s'_j)$; denoting $t'_i = \sum_{j=1}^{m} \gamma_{ij} \cdot s'_j$, one thus has $t_i \to^{?}_{\mathfrak{R}} t'_i$ (as we have removed t_i where $t_i \equiv \mathtt{a}_0 \cdot t'_i$, thus they have a canonical form and may reduce). Therefore, for each reduction of the infinite chain, each t_i reduces via $\to^{?}_{\mathfrak{R}}$. We conclude as the reduction rule imposes that one s_j actually reduces via $\to_{\mathfrak{R}}$ and not $\to^{?}_{\mathfrak{R}}$, thus at least t_i truly reduces at each step; as there is a finite number of t_i, one must reduce infinitely. ◀

▶ **Lemma 30.** *Given a quasi-order $\succcurlyeq_w$, the following properties are satisfied, for any $s \equiv \sum_{i=1}^{n} \alpha_i \cdot s_i \in \mathtt{CAN}$ and $t \equiv \sum_{j=1}^{m} \beta_j \cdot t_j \in \mathtt{CAN}$:*

- *$s \succcurlyeq_w t \iff \forall j, \exists i, s_i \succcurlyeq t_j$;*
- *If s, t are classical, $s \succcurlyeq_w t \iff s \succcurlyeq t$;*
- *$\succcurlyeq_w$ is a quasi-order.*
- *If $\succ$ is a rewrite order, then $\succ_w$ is a rewrite order too.*

Proof. For the characterization of $\succcurlyeq_w$, the if condition can be shown by deriving $\succcurlyeq_w$ using all three rules from Definition 29 in the reverse order from the presentation; for the only if condition, it can be proven by induction hypothesis on the derivation of $\succcurlyeq_w$, by noting that any premise of any rule contains at least one classical term, which simplifies the characterization. All the other points can be seen as direct consequences of this characterization; for the last point, one can remark that $s\sigma \equiv \sum_{i=1}^{n} \alpha_i \cdot s\sigma_i \in \mathtt{CAN}$ as s_i are classical terms and σ maps to classical terms, and then conclude. ◀

▶ **Theorem 31.** *Let $\succcurlyeq$ be a quasi-order, where its induced strict order is a well-founded rewrite order. Then, any STRS compatible with the worst path extension $\succ_w$ terminates.*

Proof. Suppose there exists an infinite chain starting from t_1. By Lemma 28, writing its canonical form $\sum_{i=1}^{n} \alpha_i \cdot s_i \in \mathtt{CAN}$, there exists one s_i that starts an infinite chain, thus $s \succcurlyeq_w s_i$. As it is classical, then $s_i = C[\sigma l]$, and let us denote $t_2 = C[\sigma r]$. As $\mathfrak{R}$ is compatible with $\succ_w$, $l \succ_w r$; and as $\succ$ is a rewrite order, $\succ_w$ is a rewrite order from Lemma 30, thus $s_i \succ_w t_2$. We therefore have an infinite chain $t_1 \succcurlyeq_w s_i \succ_w t_2 \dots$. By transitivity, this yields

a chain of classical terms $p_1 \succ_w p_2 \dots$, which, by Lemma 30, implies that we have an infinite chain in $\succ$, which contradicts well-foundedness. ◀

▶ **Lemma 59.** *Let $(\!|-|\!)$ be a monotonic assignment. Let $\succ$ be the restriction of $\succcurlyeq_{(\!|-|\!)}$ to classical terms. Then, $\succcurlyeq_w \subset \succcurlyeq_{(\!|-|\!)}$, and $\succ$ is a well-founded rewrite order.*

Proof. Given $s \equiv \sum_{i=1}^{n} \alpha_i \cdot s_i \in \mathtt{CAN}$ and $t \equiv \sum_{j=1}^{m} \beta_j \cdot t_j \in \mathtt{CAN}$:

$$\begin{aligned}\forall j, \exists i, s_i \succcurlyeq_{(\!|-|\!)} t_i &\implies \forall j, \exists i, (\!|s_i|\!) \succcurlyeq_{\mathbb{N}} (\!|t_j|\!) \\ &\implies \max_{1\le i\le n} (\!|s_i|\!) \succcurlyeq_{\mathbb{N}} \max_{1\le j\le m} (\!|t_j|\!) \implies (\!|s|\!) \succcurlyeq_{\mathbb{N}} (\!|t|\!) \implies s \succcurlyeq_{(\!|-|\!)} t\end{aligned}$$

As this is a characterization of a worst path ordering by Lemma 30, $\succcurlyeq_w$ is thus encoded inside $\succcurlyeq_{(\!|-|\!)}$. For any context C and any term s, t satisfying $s \succ_{(\!|-|\!)} t$:

$$(\!|C[s]|\!) = (\!|C|\!)[(\!|s|\!)] \succ_{\mathbb{N}} (\!|C|\!)[(\!|t|\!)] = (\!|C[t]|\!) \implies C[s] \succ_{(\!|-|\!)} C[t]$$

where the first and last equality come from the definition of the assignment, and for the inequality, either C is empty in which case it is trivial, either it is made of multiple symbols, and we can use monotonicity of $(\!|-|\!)$. Given s, t two terms and a substitution σ, $s \succ_{(\!|-|\!)} t$ implies that $(\!|s|\!) \succ_{\mathbb{N}} (\!|t|\!)$, thus for any evaluation of the polynomials; and $(\!|s\sigma|\!)$ can be seen as $(\!|s|\!)$ where variables are evaluated correspondingly to σ. Finally, any infinite chain $t_1 \succ_w t_2 \dots$ yields an infinite chain in $\succ_{\mathbb{N}}$ (as all variables of t_i are included in those of t_1, thus we can choose a unique set of substitutions for all rules), which is impossible. ◀

▶ **Theorem 32.** *Any STRS compatible with a monotonic assignment terminates.*

Proof. Direct by Lemma 59 and Theorem 31. While $\succcurlyeq_{(\!|-|\!)}$ is not directly a worse path extension, it satisfies the induction hypothesis, thus the proof can be adapted to this specific case. ◀

▶ **Theorem 36.** *A STRS $\mathfrak{R}$ is terminating if no infinite $\mathfrak{R}$-chain exists.*

Proof. The proof is similar to the one from [2]. Let t be a chain starting an infinite reduction. By Lemma 28, there exists $p \lhd_{\mathtt{CAN}} t$ which also starts an infinite reduction. By a minimality argument, p possesses a subterm $f_1(\overrightarrow{u_1})$, starting an infinite reduction, and where the sequence $\overrightarrow{u_1}$ terminates to values $\overrightarrow{z_1}$. Again by Lemma 28, there exists $\overrightarrow{v_1} \lhd_{\mathtt{CAN}} \overrightarrow{z_1}$, such that $f_1(\overrightarrow{v_1})$ starts an infinite reduction (as $f_1(\overrightarrow{z_1})$ can be developed by linearity). Now, we have $f_1(\overrightarrow{w_1}) \to r_1 \in \mathcal{R}$, and a substitution σ_1, such that $f_1(\overrightarrow{w_1})\sigma_1 = f_1(\overrightarrow{v_1})$, thus $f_1(\overrightarrow{v_1}) \to_{\mathfrak{R}} r_1\sigma_1$, and $r_1\sigma_1$ starts an infinite reduction, and again, there is some classical term $y_1 \lhd_{\mathtt{CAN}} r_1$ that starts an infinite reduction; as σ_1 maps to classical values, we also have $y_1\sigma_1 \lhd_{\mathtt{CAN}} r_1\sigma_1$. As σ_1 maps variables to values, and $y_1\sigma_1$ starts an infinite reduction, there is a subterm, thus $C[f_2(\overrightarrow{u_2})] = y_1$ for some context C, such that $f_2(\overrightarrow{u_2})\sigma_1$ starts an infinite reduction, and $\overrightarrow{u_2}\sigma_1$ terminate. Therefore, let $\langle F_1(\overrightarrow{w_1}), F_2(\overrightarrow{u_2})\rangle$ be a valid dependency pair. We can repeat this process, as $f_2(\overrightarrow{u_2})$ starts an infinite reduction chain, thus obtaining an infinite sequence

$$\langle F_1(\overrightarrow{w_1}), F_2(\overrightarrow{u_2})\rangle, \langle F_2(\overrightarrow{w_2}), F_3(\overrightarrow{u_3})\rangle, \dots$$

Finally, we need to prove that this is an $\mathfrak{R}$-chain. By construction, $\overrightarrow{u_j}\sigma_{j-1} \to^*_{\mathfrak{R}} \overrightarrow{z_j}$, with $\overrightarrow{w_j}\sigma_j \lhd_{\mathtt{CAN}} \overrightarrow{z_j}$. One can assume, without loss of generality, that each rule is used with distinct variables, therefore we can take $\sigma = \sigma_1 \circ \sigma_2 \circ \dots$, i.e., the disjoint union of all substitutions σ_i. Therefore, $F_j(\overrightarrow{u_j})\sigma \to_{\mathfrak{R}} r$ with $F_j(\overrightarrow{w_j})\sigma \lhd_{\mathtt{CAN}} r$, thus this is an $\mathfrak{R}$-chain. ◀

▶ **Theorem 37.** *A STRS $\mathfrak{R}$ is terminating if there exists a well-founded weakly monotonic quasi-ordering $\succcurlyeq$ on classical terms, where $\succ$ and $\succcurlyeq$ are closed under substitution, such that:*

- *$l \succcurlyeq r_i$ for all rules $l \to \sum_{i=1}^{n} \alpha_i \cdot r_i \in \mathcal{R}_{\mathfrak{R}}$ and for all i, with $\sum_{i=1}^{n} \alpha_i \cdot r_i \in \mathtt{CAN}$;*
- *$s \succ t$ for all dependency pairs $\langle s, t\rangle$.*

Proof. Suppose there is an infinite $\mathfrak{R}$-chain $\langle s_1, t_1\rangle, \langle s_2, t_2\rangle, \dots$. Therefore, we have a substitution such that $t_j\sigma \to_{\mathfrak{R}}^{*} r_j$ and $s_{j+1}\sigma \lhd_{\mathtt{CAN}} r_j$ for all j. We will construct an infinite chain, using the worst path ordering generated from $\succcurlyeq$. In particular, as $l \succcurlyeq r_i$ for $l \to \sum_{i=1}^{n} \alpha_i \cdot r_i \in \mathtt{CAN} \in \mathcal{R}$, we have $l \succcurlyeq_w r$ for all rules. As $\succcurlyeq$ is weakly monotonic and closed under substitutions, so is $\succcurlyeq_w$ by Lemma 30, and $t \to_{\mathfrak{R}}^{*} s$ implies $t \succcurlyeq_w s$. Therefore, $t_j\sigma \succcurlyeq_w r_j\sigma \succcurlyeq_w s_{j+1}\sigma$, with the last inequality coming from the definition of a worst path ordering. Therefore, we can build the following chain:

$$s_1\sigma \succ_w t_1\sigma \succcurlyeq_w s_2\sigma \succ_w t_2\sigma \dots$$

thus, a chain $s_1\sigma \succ_w s_2\sigma \dots$ between classical terms, thus an infinite chain in $\succ$, which contradicts well-foundedness. We then conclude by Theorem 36. ◀

▶ **Lemma 38.** *Let $\mathfrak{R}$ be a STRS, and let $t = \sum_{i=1}^{n} \alpha_i \cdot t_i \in \mathcal{T}_0$. Assume each t_i terminates in at most k_i steps. Then t terminates in at most $\max_{1\le i\le n} k_i$ steps.*

Proof. This is proven by induction on the maximum number of reduction steps of the t_i, denoted k. If $k = 0$, they are all values, and we conclude. Suppose $k > 0$, thus one t_i is not a value. For each t_i such that $t_i \equiv \beta \cdot s$ with $[\![\beta]\!] = 0$, or where $[\![\alpha_i]\!] = 0$, we can remove it via $\equiv$, thus $t \equiv \sum_{k=1}^{l} \alpha_k \cdot t_k$. By writing each canonical form, up to completion (meaning we may add $\mathtt{a}_0 \cdot t'$ to the canonical forms so that they are defined on the same set of terms s_j), as $t_k \equiv \sum_{j=1}^{m} \beta_{k,j} \cdot s_j$, $s_j \to_{\mathfrak{R}} s'_j$ or $s_j = s'_j$ is a value, and $\sum_{k=1}^{l} \alpha_k \cdot \beta_{k,j} = \gamma_j$:

$$t \equiv \sum_{j=1}^{m} \gamma_j \cdot s_j \equiv \sum_{j,\gamma_j\neq 0} \gamma_j \cdot s_j \to_{\mathfrak{R}} \sum_{j,\gamma_j\neq 0} \gamma_j \cdot s'_j \equiv \sum_{j=1}^{m} \gamma_j \cdot s'_j \equiv \sum_{k=1}^{n} \alpha_k \cdot t'_k,$$

as the s_j are pairwise distinct, this is a canonical form. If the obtained form is a value, then we can stop here, and the result will be correct; even if t_i is not a value, the non-value parts will still be canceled via $\equiv$ at the end, as the reduction is deterministic. Else, by $\equiv$, and we can replace $\mathtt{a}_0 \cdot s_j$ by $\mathtt{a}_0 \cdot s'_j$. Furthermore, we can add back the removed terms from the beginning, adding back t'_i instead of t_i, to get that $\sum_{k=1}^{l} \alpha_k \cdot t'_k \equiv \sum_{i=1}^{n} \alpha_i \cdot t'_i$. Thus, by $\equiv$, t rewrites in one step to $\sum_{i=1}^{n} \alpha_i \cdot t'_i$; and all t'_i terminate in one less step, thus we conclude by induction hypothesis. ◀

▶ **Lemma 39.** *Let $\mathfrak{R}$ be a QTRS compatible with an assignment $(\!|-|\!)$. Then, given any term t, it terminates in at most $(\!|t|\!)$ steps.*

Proof. Let us prove by induction on $(\!|t|\!)$, which is a natural number, that there is no t such that t terminates in at least $(\!|t|\!) + 1$ steps; this implies the wanted result. Suppose $(\!|t|\!) = 0$, thus t reduces at least once. If t is classical, then $t \to_{\mathfrak{R}} s$, and by definition of the reduction rule (C), $(\!|t|\!) > (\!|s|\!)$; as $(\!|s|\!) < 0$ and belongs to $\mathbb{N}$, this is not possible. If $t \equiv \sum_{i=1}^{n} \alpha_i \cdot t_i \in \mathtt{CAN}$, at least one t_i must reduce, else Lemma 38 would imply that t terminates in 0 steps; thus t_i reduces, and $(\!|t_i|\!) \le (\!|t|\!)$, thus we can conclude as above. Now, suppose that the result is proven for any t' where $(\!|t'|\!) \le k$, and take t where $(\!|t|\!) = k+1$. If t is classical, then, as before, there exists s such that $t \to_{\mathfrak{R}} s$, thus $(\!|s|\!) < (\!|t|\!)$, and s terminates in at least $(\!|t|\!) \ge (\!|s|\!) + 1$

steps, which by induction hypothesis cannot happen. If t is non-classical, proceed as previous, by picking one t_i from the canonical form that reduces in at least $(\!|t|\!) + 1 \geq (\!|t_i|\!) + 1$ steps, which must exist by Lemma 38. ◀

▶ **Lemma 60.** *Let $\mathfrak{R}$ be a QTRS, and let $(\!|-|\!)$ be an assignment. Then, there exists $k \in \mathbb{N}$, such that for any classical value v, $(\!|v|\!) < k \times |v|$.*

Proof. By induction on the syntax of v; take $k = 1 + \max_{\mathtt{c} \in \mathcal{C}} \alpha_{\mathtt{c}}$. ◀

▶ **Theorem 41.** *Let $\mathfrak{R}$ be a QTRS compatible with an additive assignment and let $\mathtt{f} \in \mathcal{F}$. Then, there is a polynomial P such that for any $\overline{v} \in \mathrm{In}(\mathtt{f})$, $\mathtt{f}(\overline{v})$ terminates in time $P(|v|)$.*

Proof. By Lemma 39, $f(v_1, \ldots, v_n)$ terminates in at most $(\!|f(v_1, \ldots, v_n)|\!)$ steps. It satisfies:

$$(\!|f(v_1, \ldots, v_n)|\!) = (\!|\mathtt{f}|\!)((\!|v_1|\!), \ldots, (\!|v_n|\!)) < (\!|\mathtt{f}|\!)(k|v_1|, \ldots, k|v_n|) = P(|v_1|, \ldots, |v_n|),$$

where the second inequality comes from Lemma 60 and monotonicity of $(\!|-|\!)$, and the last inequality as $(\!|\mathtt{f}|\!)$ is a polynomial Q, and write P as the polynomial which replaces each indeterminate X by kX, which is also a polynomial. ◀