Quantum Probabilistic Dyadic Second-Order Logic*

A. Baltag, J. M. Bergfeld, K. Kishida, J. Sack, S. J. L. Smets, S. Zhong

Institute for Logic, Language and Computation, Universiteit van Amsterdam
Science Park 107, 1098XG Amsterdam, The Netherlands.

Abstract. We propose an expressive but decidable logic for reasoning about quantum systems. The logic is endowed with tensor operators to capture properties of composite systems, and with probabilistic predication formulas $P^{\geq r}(s)$, saying that a quantum system in state s will yield the answer 'yes' (i.e. it will collapse to a state satisfying property P) with a probability at least r whenever a binary measurement of property P is performed. Besides first-order quantifiers ranging over quantum states, we have two second-order quantifiers, one ranging over quantum-testable properties, the other over quantum "actions". We use this formalism to express the correctness of some quantum programs. We prove decidability, via translation into the first-order logic of real numbers.

1 Introduction

This paper introduces a powerful new logic for reasoning about quantum computation. Our *Quantum Probabilistic Dyadic Second-Order Logic* (**QPDSOL**) is *expressive enough* to capture superpositions, entanglements, measurements, quantum-logical gates and probabilistic features; it can express the correctness of a wide range of complex quantum protocols and algorithms; but at the same time it is logically tractable, in the sense of being *decidable*.

It is well-known that "classical" First-Order Logic is undecidable, and moreover that "classical" Second-Order Logic, as well as its monadic and dyadic fragments[1] are not even axiomatizable. By moving to the quantum world, it is natural to *extend* the range of first-order quantifiers to *quantum "states"* (i.e. superpositions of classical states), while at the same time it is natural to *restrict* the range of monadic second-order quantifiers to *quantum-testable properties* (closed linear subspaces of the state space), and to similarly restrict the range of dyadic second-order quantifiers to *quantum "actions"* (linear maps between

* The research of J. Bergfeld, K. Kishida and J. Sack has been funded by VIDI grant 639.072.904 of the NWO. The research of S. Smets is funded by the VIDI grant 639.072.904 of the NWO and by the FP7/2007-2013/ERC Grant agreement no. 283963. The research of S. Zhong has been funded by China Scholarship Council.

[1] *Monadic* Second-Order Logic is the fragment allowing quantification only over *unary* predicates, while the Dyadic fragment allows quantification only over *unary and binary* predicates.

state spaces). Indeed, it is widely accepted in the literature on Quantum Logic and on Foundations of Quantum Mechanics that quantum-testable properties are *the only* experimentally meaningful properties of a quantum system: any other (non-testable, non-linear) properties have no physical/experimental meaning in a quantum setting. Similarly, it is widely accepted in quantum computation that all meaningful quantum programs are obtainable by composing quantum gates (unitary maps) and quantum tests (measurements), and thus are quantum "actions" in the above sense.[2] So restricting the interpretations of the unary and binary predicates as above is a natural thing to do in a quantum setting: it only restricts the second-order quantifiers to properties/actions that are *physically meaningful*. The resulting logic *is indeed the natural "quantum analogue"* of classical (dyadic) second-order logic!

Surprisingly, this quantum analogue turns out to be much more tractable than its classical counterpart: the above well-justified and natural restrictions of range are enough to restore full decidability, even after the addition of "exotic" features such as probabilistic predication and tensors!

In a sense, this is not as surprising as it may first appear. Our semantics for second-order logic is "non-standard": not all sets of states (whose existence is guaranteed by the standard axioms of Set Theory) are accepted as "predicates". The second-order quantifiers are thus restricted to a limited range of predicates. Such non-standard variations of second-order logic have been studied before. Henkin's weak semantics for second-order logic [11] involves a restriction on the range of the second-order quantifiers (to some model-dependent class of admissible predicates), that restores the axiomatizability of the logic. Some variants of monadic second-order logic (for very restricted models) are even decidable [14].

But these classical results are conceptually very different from ours: none of these weaker logics can be considered to be a genuine and natural variant of second-order logic. In particular, Henkin's semantics (restricting second-order quantifiers to some arbitrary collections of subsets of the state space) is not a independently-justifiable restriction. It does not even provide a unique, canonical way to restrict the quantifiers (but a model-dependent one). In contrast, our restriction of quantifiers to quantum-testable properties (and quantum-performable operations) is natural, canonical (providing a unique collection for each dimension) and amply justified on independent grounds by a whole body of literature in Quantum Logic, Foundations of Quantum Mechanics and Quantum Computation.

[2] The converse is not obvious, and may even fail in practice. But from a theoretical perspective, one can argue that the converse is true in a sense: for any quantum action (linear map) f between systems $\mathcal{H}$ and $\mathcal{H}'$ there exists an entangled state s_f in $\mathcal{H} \otimes \mathcal{H}'$ with the property that, if a local measurement performed on the $\mathcal{H}$-subsystem of (a system in state) s_f yields state x, then after that a local measurement on the $\mathcal{H}'$-subsystem will yield the result $f(x)$. In this way, any such action f can be physically computed, in principle: first, prepare a large number of entangled states s_f; then perform local measurements on the $\mathcal{H}$-subsystem until one of them yields the desired input value x; and then perform a measurement on the $\mathcal{H}'$-subsystem, yielding the output-value $f(x)$.

Indeed, seen from the perspective of the quantum world, our "non-standard" semantics *looks like the "true" semantics* of second-order logic: it only eliminates the predicates that are "physically meaningless". Moreover, while in a sense being a restriction of the classical (standard) semantics, in a different sense this can be thought of as *an extension of the classical semantics*! Indeed, one can argue that, if we restrict ourselves to *classical states* (i.e. n-long tuples of bits $|0\rangle$ or $|1\rangle$, for any dimension n) then *all the standard predicates of such classical states are realized as quantum-testable predicates* (and hence fall within the range of our second-order quantifiers): for *every* set $A \subseteq \{|0\rangle, |1\rangle\}^n$, there exists a unique quantum-testable predicate (linear subspace[3]) $P_A \subseteq \mathcal{H}_2^{\otimes n}$ such that a classical n-state $s \in \{|0\rangle, |1\rangle\}^n$ satisfies P_A iff it belongs to the set A. So, insofar as *classical* states are concerned, our range restriction for second-order quantifiers *is not a restriction at all*: their range really includes (quantum counterparts of) *every set* of classical states. It is only when we look at non-classical (superposed) states that we see that the quantifier range is restricted (though in a natural way).

In conclusion, regardless of whether one considers it as a natural restriction of the classical semantics for (predicates of) quantum states, or as a huge extension of the classical semantics for (predicates of) classical states, we can still safely claim that *our logic really is the correct quantum (and probabilistic) counterpart of the classical (dyadic) second-order logic.*

As a consequence, we think that our decidability result is a significant contribution to the logical understanding of quantum mechanics: it shows in essence that, whereas the natural formulation of (dyadic) second-order logic in the *classical* world is undecidable, *the natural formulation of (dyadic) second-order logic for the quantum world is decidable.*

The fundamental reason for this tractability is the one severe constraint put by quantum mechanics on the "meaningful" properties and actions: *linearity.*[4] Once again, this does not really restrict the predicates/actions as far as classical states are concerned (since any two classical states of the same space are orthogonal to each other, a classical state cannot be written as a linear combination of other classical states). But linearity *does* constrain the behavior of "meaningful" predicates/actions on *superposed* states. And, in the end, linearity allows the reduction of all the "meaningful" second-order objects (predicates/actions) to their underlying linear expressions: matrices of (complex) numbers.

So this natural (and physically-imposed) linearity constraint reduces thus our quantum version of second-order logic to the *first-order theory* of complex numbers. And now, a classical result comes to our help: while first-order logic is in general undecidable (and the first-order theories of many useful structures, such as the ring of natural numbers, are not even axiomatizable), *the first-order theory of complex numbers is decidable.* This was pointed out by Alfred Tarski [17] as a corollary to the analogue result for the field of real numbers (proved in the same paper by quantifier-elimination).

[3] In fact, this is the linear subspace P_A generated by A.

[4] For unary predicates: having a linear subspace (not an arbitrary subset) as their extension; for actions: being induced by a linear map.

Our decidability proof makes essential use of Tarski's decidability result, as well as of the finite dimensionality; it translates effectively the probabilistic dyadic second-order logic of finite-dimensional quantum systems into the decidable first-order theory of reals. This proof method is inspired by the one given by Dunn et al. in [10], where the traditional (propositional) quantum logic of any finite-dimensional Hilbert space was proved to be decidable. However, the result in [10] required that we first fix a particular Hilbert space (model of a quantum system) of a finite dimension, so as to translate the logic of the space into the finitary language of reals, thus limiting the scope of application by fixing a finite dimension (and hence the number of *quantum bits* or *qubits*) throughout the discourse. In contrast, our logic overcomes this limitation by using types and tensors in the language, thus accommodating *an unbounded number of qubits*, while preserving the logical tractability.

Our results in this paper can be seen as part of a wider on-going effort towards bridging the gap between traditional quantum logic and the theory of quantum computation. On the one hand, traditional quantum logic (as originated in [7]) has focused on axiomatics and logical properties of the lattice of closed linear subspaces of an *infinite-dimensional* Hilbert space, with the goal being "to discover the logical structure one may hope to find in physical theories which, like QM, do not conform to classical logic" [7]. Quantum computation, on the other hand, concerns encoding and describing computations on the basis of quantum systems, and involves quantum ingredients such as superposition and entanglement, in order to perform certain tasks much faster than classical computers. The underlying theoretical framework for quantum computation is given by *finite-dimensional* Hilbert spaces. Among the few treatments of such finite-dimensional quantum logics and their decidability are the work of [8,10].

Another contrast between quantum logic and quantum computation lies in the treatment of "quantum entanglement". In traditional quantum logic, entanglement has been viewed as a problem-child, posing difficulties to the lattice-theoretic setting [2,15] (though naturally treatable in a category-theoretical setting [1,16]). In quantum computing, however, entanglement is viewed as a *computational resource*, that allows us to go beyond the world of classical computing. Among the papers that address this part of the gap between quantum logic and quantum computation are [3,8], and [9, Chapter 17]. Our work strengthens the connection further. The logic we propose in the following sections—dyadic second-order quantum logic—is fit to deal with multi-partite systems that exhibit quantum entanglement. Equipped with an explicitly typed language, with types for states, predicates, and actions, with tensor operators connecting them, as well as with probabilistic predication, our logic allows us to capture all the essential computational properties of composite quantum systems, and in particular it can encode the correctness of a wide range of quantum algorithms.

The design of dyadic second-order quantum logic in this paper builds further on the earlier work of Baltag and Smets on propositional dynamic quantum logics [5,6]. It is well known that standard Propositional Dynamic Logic (PDL), as well as its fragment called the Hoare Logic, plays an important role in classical

computing and programming. In particular, PDL and Hoare Logic are among the main logical formalisms used for classical program verification. The quantum version of PDL extends the area of applicability to the verification of quantum programs and quantum protocols. In [6], a quantum dynamic logic was designed that was expressive enough to prove the correctness of basic non-probabilistic quantum protocols such as teleportation and quantum secret sharing. The work of [4] used the tools of [10] to prove the decidability of such a propositional dynamic quantum logical system. While these results are important, note that the logic in [4] was unable yet to capture the correctness of any probabilistic quantum protocols. In this paper, we overcome this limitation and equip our logic with a *probabilistic predication operator*, indicating that a state of a quantum system will collapse to a state having property P with probability at least r whenever a measurement of property P is performed. This operator allows us to express the correctness of those quantum algorithms (such as quantum search) that make essential use of quantum probabilities.

A remark is in order regarding the fact that each given program in our syntax, and so each given sentence, uses only a given number of qubits (and thus it refers to a Hilbert space with a given finite number of dimensions). We would like to stress that our result is much more significant than, say, the decidability of checking the correctness of a classical circuit of a given size applied to a problem of given input size. This is because *we do not fix the size of the input, but only the dimension*. This point is important, since for a given fixed dimension (greater than 1) there are *infinitely* (in fact *uncountably*) many non-equivalent quantum states of that dimension (while typically there are only finitely many inputs of a given size). Hence, the algorithm for deciding satisfiability (on states of a space of given dimension) *cannot* simply proceed by exhaustive search over a finite domain (as in the case of models of bounded size). The correctness statements presented in this paper really capture the correctness of a program for uncountably many non-equivalent inputs![5]

2 Preliminaries

According to quantum theory (see e.g. [12]), any quantum system can be described by a Hilbert space $\mathcal{H}$ of appropriate dimension. Similar to the tradition of [13], we identify *(pure) states* of the system with the "rays" in $\mathcal{H}$ (i.e. the one-dimensional linear subspaces of $\mathcal{H}$) and the "impossible state" (zero-dimensional subspace, which we include as it allows us to discuss only total functions without loss of generality). Given a vector $|\psi\rangle \in \mathcal{H}$, we will write $\widehat{|\psi\rangle}$ for the state generated by $|\psi\rangle$. Given a state space $\mathcal{H}$ of some quantum system, we write $\Sigma_{\mathcal{H}}$

[5] Moreover, these correctness statements, even when translated back into the arithmetic of real numbers, do *not* boil down to simple equations involving addition and multiplication of *specific* real numbers and/or matrices. Instead, they reduce to complex first-order statements in the theory of real numbers, that involve in an essential way quantification over uncountably many objects. It just happens that (due to Tarski's theorem) this theory is still decidable!

for the set of all states, i.e. the set of all one-dimensional linear subspaces of $\mathcal{H}$ and $\widehat{\mathbf{0}_{\mathcal{H}}}$ (where $\mathbf{0}_{\mathcal{H}}$ is the zero vector).

Any change of the state of a quantum system can be described by a linear map on $\mathcal{H}$. There are two important kinds of linear maps: unitary operators and projectors. A *unitary operator* U is such that both $U^\dagger U$ and $UU^\dagger$ are the identity operator, where $(\cdot)^\dagger$ is the adjoint operation on linear maps. In quantum computation, unitary operators are the counterpart of logical gates in classical computation. An operator A is a *projector*, if it is bounded, idempotent, i.e. $AA = A$, and self-adjoint, i.e $A^\dagger = A$. Projectors are essential in describing quantum measurements, which are the only way we extract information from a quantum system. In this paper, our level of abstraction allows us to consider not only linear maps on a Hilbert space but also those between different Hilbert spaces. Every linear map $A : \mathcal{H} \to \mathcal{H}'$ from a quantum system $\mathcal{H}$ to a possibly different system $\mathcal{H}'$ naturally induces a unique function (also denoted by A) from the set of states $\Sigma_{\mathcal{H}}$ to the set of set of states $\Sigma_{\mathcal{H}'}$, given by $A(\widehat{|\psi\rangle}) := \widehat{A(|\psi\rangle)}$ for every $|\psi\rangle \in \mathcal{H}$. An *action* is any such function $A : \mathcal{H} \to \mathcal{H}'$ induced on state spaces by some linear map $A : \Sigma_{\mathcal{H}} \to \Sigma_{\mathcal{H}'}$. We can also define composition, tensor product and adjoint of actions in a natural way via composition, tensor product and adjoint of linear maps which induce the actions[6]. We will use the same symbols for operations on actions as those for linear maps.

In this paper, a *property* of a quantum system with state space $\mathcal{H}$ is just a subset of $\Sigma_{\mathcal{H}}$. However, according to quantum theory, not any subset of $\Sigma_{\mathcal{H}}$ represents a property of the system that can be tested. A property is *testable* iff it corresponds to a closed linear subspace W of $\mathcal{H}$ in such a way that the states in the property are exactly those generated by vectors in W. Since this correspondence is one-to-one and natural, we will always use the same symbol to denote a testable property and its corresponding closed linear subspace. Moreover, according to linear algebra, closed linear subspaces lie in one-to-one correspondence with projectors in the following sense:

1. For every projector A on $\mathcal{H}$, $\mathrm{ran}(A)$ (the range of A) is a closed linear subspace of $\mathcal{H}$, and for every vector $|\psi\rangle \in \mathcal{H}$, $|\psi\rangle \in \mathrm{ran}(A)$ iff $A(|\psi\rangle) = |\psi\rangle$.
2. For every closed linear subspace W of $\mathcal{H}$, there is a *unique* projector on $\mathcal{H}$, called *the projector onto* W and denoted by $?^{\mathcal{H}}(W)$, such that for every vector $|\psi\rangle \in \mathcal{H}$, $|\psi\rangle \in W$ iff $?^{\mathcal{H}}(W)(|\psi\rangle) = |\psi\rangle$.

The state space of a qubit, the unit of quantum information, is of dimension 2. Given a fixed orthonormal basis $\{|0\rangle, |1\rangle\}$ of the state space of a qubit, the two states generated by $|0\rangle$ and $|1\rangle$, respectively, correspond to the values 0 and 1 of a classical bit. Given several qubits indexed by elements in a finite set I, they form a compound quantum system, and the state space for I is the tensor product $\bigotimes_{i\in I} \mathcal{H}_i$ of the state space $\mathcal{H}_i$ for each qubit $i \in I$. A standard way of obtaining an orthonormal basis of this state space is to take tensor products of vectors in the fixed orthonormal bases of each $\mathcal{H}_i$. It is easy to see that there

[6] Note that different linear maps could induce the same action, but the operations on actions are still well-defined according to linear algebra.

are $2^{|I|}$ vectors in this basis, and we will index them by elements in ${}^{I}\mathbf{2}$, the set of all functions from I to $\mathbf{2} = \{0, 1\}$, in such a way that $|f\rangle = \otimes_{i \in I} |f(i)\rangle_i$, for each $f \in {}^{I}\mathbf{2}$. We call a state of a compound system *classical* if it is generated by a vector in this basis. Moreover, we write $|0\rangle_I$ for $\otimes_{i \in I} |0\rangle_i$.

It is well known that an n-dimensional Hilbert space is isomorphic to $\mathbb{C}^n$. In this case, every linear subspace is closed and every operator is bounded. Moreover, every state can be represented by n complex numbers if we pick a vector in the state as its representative. Every property, identified with its corresponding projector, can be represented by an $n \times n$ matrix of complex numbers. Every linear map from an n-dimensional Hilbert space to an m-dimensional one can be represented by an $m \times n$ matrix of complex numbers.

In this paper, for generality, we assume that we are supplied with countably infinitely many qubits indexed by elements in ω, the set of all natural numbers, which we take to be non-negative integers. We further assume that an orthonormal basis has been fixed for each qubit, and we obtain an orthonormal basis for compound systems consisting of a finite number of qubits by applying the tensor product in the way just described. Finally, we use $\mathcal{P}_{\text{fin}}(\omega)$ to denote the set of all finite, *non-empty* subsets of ω. For each $\tau \in \mathcal{P}_{\text{fin}}(\omega)$, by *$\tau$-system* we mean the quantum system consisting of qubits indexed by elements of τ. Whenever $\mathcal{H}_\tau$, the state space of the τ-system, appears as a superscript or subscript in a symbol, we simply write τ; for example, we write simply Σ_τ for $\Sigma_{\mathcal{H}_\tau}$.

Moreover, for each $\tau, \rho \in \mathcal{P}_{\text{fin}}(\omega)$ s.t. $\tau \subseteq \rho$, we know from linear algebra that $\mathcal{H}_\tau$ can be canonically embedded into $\mathcal{H}_\rho$, by "padding" all the vectors with $|0\rangle$'s for all the extra dimensions. Hence in this paper we write $\Theta_{\tau \to \rho} : \mathcal{H}_\tau \to \mathcal{H}_\rho$ for this canonical embedding

$$\Theta_{\tau \to \rho} = \textstyle\sum_{f \in {}^{\tau}\mathbf{2}} (|f\rangle \otimes |0\rangle_{\rho \setminus \tau}) \langle f|.$$

We also write $\Theta_{\rho \to \tau} : \mathcal{H}_\rho \to \mathcal{H}_\tau$ for the canonical projection that reverses the above embedding:

$$\Theta_{\rho \to \tau} = \textstyle\sum_{f \in {}^{\tau}\mathbf{2}} |f\rangle \, (\langle f| \otimes \langle 0|_{\rho \setminus \tau}).$$

Using the canonical embeddings and projections, one can *generalize projectors to arbitrary dimensions*: For every space $\mathcal{H}_\tau$ and every closed linear subspace W_ρ of some other space $\mathcal{H}_\rho$, we can define *the generalized projector of $\mathcal{H}_\tau$ onto W_ρ*, denoted by $?^\tau(W_\rho)$, by putting:

$$?^\tau(W_\rho) = \Theta_{\rho \cup \tau \to \rho} \circ \left(?^\rho(W_\rho) \otimes |0\rangle_{\tau \setminus \rho} \langle 0|_{\tau \setminus \rho} \right) \circ \Theta_{\tau \to \rho \cup \tau}$$

This is a linear map that takes a vector in $\mathcal{H}_\tau$ and "projects" it onto W_ρ. Physically, this action corresponds to *a successful measurement of a ρ-property performed on a τ-system.*

We introduce some notation. Given a binary relation R and a set $A \subseteq \text{dom}(R) = \{x \mid \exists y. \, (x, y) \in R\}$, let $R[A] \stackrel{\text{def}}{=} \{b \mid \exists a \in A. \, (a, b) \in R\}$ be the direct image of A under R. Given a set $B \subseteq \text{ran}(R) = \{y \mid \exists x. \, (x, y) \in R\}$, we let $[R]B \stackrel{\text{def}}{=} \{a \mid \forall b. \, (a, b) \in R \Rightarrow b \in B\}$ be the so-called weakest precondition of

B under R. Note that when R is a function instead of a relation in general, $[R]B$ is sometimes called the inverse image of B under R. In general, given two sets A and B, we write ${}^{A}B$ for the set of functions from A to B. Given a positive number N, let $\mathbf{N} = \{0, 1, \ldots, N-1\}$. Given a linear map T, let $\mathbf{T}$ be its matrix representation under the fixed bases.

3 Quantum Probabilistic Dyadic Second-Order Logic

Syntax of QPDSOL Our language consists of terms (for quantum states), predicates symbols (for quantum testable properties), and function symbols (for actions). The language is *typed*: each of these symbols comes with a type, which is an element of $\mathcal{P}_{\text{fin}}(\omega)$, indicating the underlying set of qubits involved in that state, property or action. E.g. terms of type τ refer to the possible (pure) states of the τ-system; predicate symbols of type τ are unary predicates referring to *quantum-testable properties* of the τ-system; function symbols of type $\tau \to \rho$ are dyadic predicates (restricted to functions) referring to *actions*. As the types range over all of $\mathcal{P}_{\text{fin}}(\omega)$, the entire domain of discourse involves infinitely many qubits; but each formula involves only finitely many types, each involving only finitely many qubits, so that a formula can only talk about finitely many qubits.

For each pair of elements $\tau, \rho \in \mathcal{P}_{\text{fin}}(\omega)$, we include in the language a countable set of *state variables* x_τ of type τ, a countable set of *state constants* c_τ of type τ, a countable set of *predicate variables* p_τ of type τ, a countable set of *predicate constants* T_τ of type τ, a countable set of *action variables* $a_{\tau\to\rho}$ of type $\tau \to \rho$, and a countable set of *action constants* $C_{\tau\to\rho}$ of type $\tau \to \rho$. It is assumed that these sets are pairwise disjoint, and that each of them is indexed by elements in ω without repetition.

Definition 3.1. *For any $\tau, \rho \in \mathcal{P}_{fin}(\omega)$, we define by (triple) mutual recursion the following sets of syntactic expressions: the set $Term_\tau$ of terms of type τ*

$$t_\tau ::= \; x_\tau \mid c_\tau \mid t_{\tau_1} \otimes t_{\tau_2} \mid \alpha_{\rho\to\tau}(t_\rho)$$

(where $\tau_1, \tau_2 \in \mathcal{P}_{fin}(\omega)$ are such that $\tau_1 \cup \tau_2 = \tau$, $\tau_1 \cap \tau_2 = \emptyset$), the set $\mathcal{P}_\tau$ of (unary) predicate symbols of type τ

$$P_\tau ::= p_\tau \mid T_\tau \mid t_\tau \mid {\sim}P_\tau \mid P_\tau \cap P_\tau \mid P_{\tau_1} \otimes P_{\tau_2} \mid \alpha_{\rho\to\tau}[P_\rho] \mid [\alpha_{\tau\to\rho}]P_\rho$$

(where $\tau_1, \tau_2 \in \mathcal{P}_{fin}(\omega)$ are such that $\tau_1 \cup \tau_2 = \tau$, $\tau_1 \cap \tau_2 = \emptyset$), and the set $\mathcal{A}_{\tau\to\rho}$ of function symbols of type $\tau \to \rho$

$$\alpha_{\tau\to\rho} ::= a_{\tau\to\rho} \mid C_{\tau\to\rho} \mid ?^\tau P_\rho \mid \alpha^\dagger_{\rho\to\tau} \mid \alpha_{\tau\to\mu}; \alpha_{\mu\to\rho} \mid \alpha_{\tau_1\to\rho_1} \otimes \alpha_{\tau_2\to\rho_2}$$

(where $\mu, \tau_1, \rho_1, \tau_2, \rho_2 \in \mathcal{P}_{fin}(\omega)$ are such that $\tau_1 \cup \tau_2 = \tau$, $\rho_1 \cup \rho_2 = \rho$ and $\tau_1 \cap \tau_2 = \rho_1 \cap \rho_2 = \emptyset$).

We write *Term* for the set $\bigcup_{\tau\in\mathcal{P}_{\text{fin}}(\omega)} Term_\tau$ of all terms, $\mathcal{P}$ for the set $\bigcup_{\tau\in\mathcal{P}_{\text{fin}}(\omega)} \mathcal{P}_\tau$ of all predicate symbols, and $\mathcal{A}$ for the set $\bigcup_{\tau,\rho\in\mathcal{P}_{\text{fin}}(\omega)} \mathcal{A}_{\tau\to\rho}$ of all function symbols. When $\tau = \rho$, we simply write $P_\rho?$ for the function symbol $?^\tau P_\rho$.

Definition 3.2. *We now define by induction the set $\mathcal{L}$ of formulas of our logic:*

$$\varphi ::= P_\tau^{\geq r}(t_\tau) \mid \neg\varphi \mid \varphi \wedge \varphi \mid \forall x_\tau \varphi \mid \forall p_\tau \varphi \mid \forall a_{\rho\to\tau}\varphi$$

where $\tau \in \mathcal{P}_{fin}(\omega)$, $t_\tau \in Term_\tau$, $P_\tau \in \mathcal{P}_\tau$ and $r \in [0,1]$ is a definable real number (described below before Definition 3.3).

The intended meaning of our basic formula $P_\tau^{\geq r}(t_\tau)$ is that a *quantum system in state t_τ will yield the answer 'yes'* (i.e. it will collapse to a state satisfying property P_τ) *with a probability at least r whenever a binary measurement of property P_τ is performed*. The rest of our logical formulas are built from such basic formulas using standard Boolean connectives, as well as three types of quantifiers: first-order quantifiers $\forall x_\tau$ ranging over quantum states, second-order quantifiers $\forall p_\tau$ over quantum (testable) predicates, and second-order quantifiers $\forall a_{\tau\to\rho}$ ranging over quantum actions.

The notions of free variables, bound variables, etc. are defined in the standard way. As usual, a formula $\varphi \in \mathcal{L}$ is called *closed* if it has no free (state, predicate or action) variables. A *pure* formula is a closed formula containing no (state, predicate or action) constants.

Semantics of QPDSOL Following standard practice, we introduce the notion of *frame* (also known as *structure* in the semantics of first-order logic), by which we mean a structure that fixes the (state, predicate and action) constants. Then, given a frame, we define a *model* on it (also known as an *interpretation* in the semantics of first-order logic), which can determine the denotation of each remaining term, predicate symbol and function symbol. Finally, we define the *satisfaction relation*.

Recall that we say that a real number r is *definable* if there is a formula $\phi(x)$ in the first-order language of $(\mathbb{R}, +, \cdot, 0, 1)$ such that $(\mathbb{R}, +, \cdot, 0, 1) \models \phi[r] \wedge \forall x(\phi(x) \to x = r)$. We also say that a complex number z is *simple* if $z = a + bi$ for definable real numbers a and b. Extending the terminology, we say that a state of the τ-system, a testable property of the τ-system and an action from the τ-system to ρ-system are *definable* if they can be represented under the fixed basis respectively by a $2^{|\tau|}$-tuple (with the state identified with the representative of it), a $2^{|\tau|} \times 2^{|\tau|}$-matrix (with the closed linear subspace identified with the corresponding projector), and a $2^{|\rho|} \times 2^{|\tau|}$-matrix (with the action identified with a linear map that induces it) of simple complex numbers.

Definition 3.3. *An $\mathcal{H}$-*valuation *is a function V defined on a subset of $\mathcal{P} \cup \mathcal{A} \cup Term$ and satisfying the following conditions:*

- *$V(t_\tau) \in \Sigma_\tau$ if $t_\tau \in Term_\tau$;*
- *$V(P_\tau)$ is a testable property of τ-system, if $P_\tau \in \mathcal{P}_\tau$;*
- *$V(\alpha_{\tau\to\rho})$ is an action from Σ_τ to Σ_ρ if $\alpha_{\tau\to\rho} \in \mathcal{A}_{\tau\to\rho}$.*

Definition 3.4. *A* frame *$\mathfrak{F}$ is an $\mathcal{H}$-valuation whose domain is the set of all (state, predicate and action) constants and whose values are all definable.*

Actually, for the decidability result to hold, a frame must be a computable function in some sense. We neglect this technicality here.

Definition 3.5. *A* model $\mathfrak{M}$ *on a frame* $\mathfrak{F}$ *is an* $\mathcal{H}$*-valuation whose domain is* $\mathcal{P} \cup \mathcal{A} \cup Term$*, that extends* $\mathfrak{F}$ *and that satisfies the following, for any terms* $t_\tau, t_{\tau_1}, t_{\tau_2}$*, function symbols* $\alpha_{\tau\to\rho}, \beta_{\rho\to\mu}, \alpha_{\tau_1\to\rho_1}, \alpha_{\tau_2\to\rho_2}$*, and predicate symbols* $P_\tau, Q_\tau, P_\rho, P_{\tau_1}, Q_{\tau_2}$ *such that* $\tau_1 \cap \tau_2 = \emptyset$ *and* $\rho_1 \cap \rho_2 = \emptyset$*:*

$\mathfrak{M}(t_{\tau_1} \otimes t_{\tau_2})$	$= \mathfrak{M}(t_{\tau_1}) \otimes \mathfrak{M}(t_{\tau_2})$
$\mathfrak{M}(\alpha_{\tau\to\rho}(t_\tau))$	$= \mathfrak{M}(\alpha_{\tau\to\rho})(\mathfrak{M}(t_\tau))$
$\mathfrak{M}(\alpha_{\tau\to\rho}; \beta_{\rho\to\mu})$	$= \mathfrak{M}(\beta_{\rho\to\mu}) \circ \mathfrak{M}(\alpha_{\tau\to\rho})$
$\mathfrak{M}(\alpha^\dagger_{\tau\to\rho})$	$= (\mathfrak{M}(\alpha_{\tau\to\rho}))^\dagger$
$\mathfrak{M}(\alpha_{\tau_1\to\rho_1} \otimes \alpha_{\tau_2\to\rho_2})$	$= \mathfrak{M}(\alpha_{\tau_1\to\rho_1}) \otimes \mathfrak{M}(\alpha_{\tau_2\to\rho_2})$
$\mathfrak{M}(?^\tau P_\rho)$	$= ?^\tau(\mathfrak{M}(P_\rho))$
$\mathfrak{M}(\sim P_\tau)$	$= \sim\mathfrak{M}(P_\tau)$
$\mathfrak{M}(P_\tau \cap Q_\tau)$	$= \mathfrak{M}(P_\tau) \cap \mathfrak{M}(Q_\tau)$
$\mathfrak{M}(P_{\tau_1} \otimes Q_{\tau_2})$	$= \mathfrak{M}(P_{\tau_1}) \otimes \mathfrak{M}(Q_{\tau_2})$
$\mathfrak{M}(\alpha_{\tau\to\rho}[P_\tau])$	$= \mathfrak{M}(\alpha_{\tau\to\rho})[\mathfrak{M}(P_\tau)]$
$\mathfrak{M}([\alpha_{\tau\to\rho}]P_\rho)$	$= [\mathfrak{M}(\alpha_{\tau\to\rho})]\mathfrak{M}(P_\rho)$

To interpret quantifiers, for each (state, predicate, or action) variable v we introduce an equivalence relation $\sim_v$ among models on the same frame such that $\mathfrak{M} \sim_v \mathfrak{M}'$ iff $\mathfrak{M}(v') = \mathfrak{M}'(v')$ for all variables v' except possibly v.

Definition 3.6. *The satisfaction relation between a model* $\mathfrak{M}$ *and a formula is defined recursively, where* v *is any (state, predicate, or action) variable,*

$$\begin{array}{lll}
\mathfrak{M} \models P^{\geq r}_\tau(t_\tau) & \iff & |\langle\psi|?^\tau(\mathfrak{M}(P_\tau))|\psi\rangle|^2 \geq r\||\psi\rangle\|^2\|?^\tau(\mathfrak{M}(P_\tau))|\psi\rangle\|^2, \\
 & & \text{for any vector } |\psi\rangle \in \mathfrak{M}(t_\tau) \\
\mathfrak{M} \models \neg\varphi & \iff & \mathfrak{M} \not\models \varphi, \\
\mathfrak{M} \models \varphi \wedge \psi & \iff & \mathfrak{M} \models \varphi \text{ and } \mathfrak{M} \models \psi, \\
\mathfrak{M} \models \forall v\varphi & \iff & \mathfrak{M}' \models \varphi, \text{ for all } \mathfrak{M}' \sim_v \mathfrak{M}.
\end{array}$$

Obviously, other Boolean connectives such as $\vee$, $\to$ and $\leftrightarrow$ can be defined in the usual manner. Existential quantifiers over states, predicates and actions can also be defined in the usual manner. Moreover, this logic is at least as expressive as the first-order language of the lattice $L(\mathbb{C}^{2^n})$, which is discussed in [10].

Now we introduce some useful abbreviations:

$$\begin{array}{ll}
P^{\leq r}_\tau(t_\tau) \stackrel{\text{def}}{=} (\sim P)^{\geq(1-r)}_\tau(t_\tau) & P^{=r}_\tau(t_\tau) \stackrel{\text{def}}{=} P^{\geq r}_\tau(t_\tau) \wedge P^{\leq r}_\tau(t_\tau) \\
P^{<r}_\tau(t_\tau) \stackrel{\text{def}}{=} \neg P^{\geq r}_\tau(t_\tau) & P^{>r}_\tau(t_\tau) \stackrel{\text{def}}{=} \neg P^{\leq r}_\tau(t_\tau) \\
s_\tau \perp t_\tau \stackrel{\text{def}}{=} s^{\leq 0}_\tau(t_\tau) &
\end{array}$$

$$s_\tau \doteq t_\tau \stackrel{\text{def}}{=} \left[s^{=1}_\tau(t_\tau) \wedge \neg(s_\tau \perp t_\tau)\right] \vee \left[(s_\tau \perp s_\tau) \wedge (t_\tau \perp t_\tau)\right]$$

Essentially, the meaning of $P^{\leq r}_\tau(t_\tau)$ (or respectively $P^{=r}_\tau(t_\tau)$, $P^{<r}_\tau(t_\tau)$, $P^{>r}_\tau(t_\tau)$) is that a quantum system in state t_τ will yield the answer 'yes' (i.e. it will collapse to a state satisfying property P_τ) with a probability $\leq r$ (or respectively $= r$, $< r$, $> r$) whenever a binary measurement of property P_τ is performed. Moreover, $\mathfrak{M} \models s_\tau \perp t_\tau$ iff s_τ and t_τ denote two orthogonal states. (Note that

the impossible state $\widehat{\mathbf{0}_\tau}$ is the only state that is orthogonal to itself.) Finally, we have $\mathfrak{M} \models s_\tau \doteq t_\tau$ iff s_τ and t_τ refer to *the same state*: the first disjunct ensures that s_τ and t_τ are equal but neither denotes $\widehat{\mathbf{0}_\tau}$ (note that $s_\tau^{=1}(t_\tau)$ and $s_\tau \perp t_\tau$ are together satisfiable where either s_τ or t_τ is interpreted by $\widehat{\mathbf{0}_\tau}$), while the second disjunct ensures that both s_τ and t_τ denote $\widehat{\mathbf{0}_\tau}$.

We now define the notion of validity.

Definition 3.7. *A formula φ of $\mathcal{L}$ is said to be valid in a frame $\mathfrak{F}$, written $\mathfrak{F} \models \varphi$, if $\mathfrak{M} \models \varphi$ for every model $\mathfrak{M}$ on $\mathfrak{F}$. A formula φ of $\mathcal{L}$ is said to be valid, written $\models \varphi$, if $\mathfrak{F} \models \varphi$ for every frame $\mathfrak{F}$.*

As in classical predicate logic, we have

Lemma 3.8. *For every* closed *formula φ in $\mathcal{L}$ and every frame $\mathfrak{F}$, $\mathfrak{F} \models \varphi$ iff there is a model $\mathfrak{M}$ on $\mathfrak{F}$ such that $\mathfrak{M} \models \varphi$. For every* pure *formula φ in $\mathcal{L}$, $\models \varphi$ iff there is a frame $\mathfrak{F}$ such that $\mathfrak{F} \models \varphi$.*

4 Examples

Here we show how our language can be used to express many properties of quantum algorithms. We start with introducing some notation that will be commonly used in the following examples.

First, for each qubit i, we introduce state constants 0_i and 1_i to denote the state generated by $|0\rangle_i$ and $|1\rangle_i$, respectively.

We furthermore have the following action constants for a single qubit i, and for some, we provide the matrix representation (in the fixed bases) of linear maps which are usually used to induce the actions interpreting these constants:

- I_i interpreted as the identity action,
- H_i the action induced by the Hadamard gate with matrix $\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
- X_i the action induced by the Pauli X gate $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
- Z_i the action induced by the Pauli Z gate $\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$

We furthermore have an action symbol $CNOT_{ij}$ $(i \neq j)$ for the *controlled-NOT* action with control qubit i and target qubit j usually induced by a linear map with the matrix

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

For any distinct i and j, we also define an abbreviation for an action that interchanges the states of qubits i and j:

$$FP_{ij} \stackrel{\text{def}}{=} CNOT_{ij}; CNOT_{ji}; CNOT_{ij}$$

We introduce an abbreviation $CS_\tau(t_\tau)$ for the formula saying that a state t_τ is a classical state:

$$CS_\tau(t_\tau) \stackrel{\text{def}}{=} \exists\{x_i \,|\, i \in \tau\}\Big(t_\tau \doteq \otimes_{i\in\tau} x_i \wedge \bigwedge_{i\in\tau}(x_i \doteq 0_i \vee x_i \doteq 1_i)\Big),$$

where $\exists\{x_i \,|\, i \in \tau\}$ means a sequence of existential quantifiers on state variables of type $i \in \tau$. Similarly, we introduce an abbreviation $Unit(\alpha_{\tau\to\tau})$ for the formula saying that the variable $\alpha_{\tau\to\tau}$ denotes (an action induced by) a unitary operator on a τ-system:

$$Unit(\alpha_{\tau\to\tau}) \stackrel{\text{def}}{=} \forall x_\tau(\alpha_{\tau\to\tau}; \alpha^\dagger_{\tau\to\tau}(x_\tau) \doteq x_\tau).$$

Next, we write $H^{\otimes\tau}$ for $\otimes_{i\in\tau} H_i$ and $I^{\otimes\tau}$ for $\otimes_{i\in\tau} I_i$. Finally, we recursively introduce an abbreviation $\alpha^n_{\tau\to\tau}$ for the action obtained by iterating the action $\alpha_{\tau\to\tau}$ for n times:

$$\begin{aligned} \alpha^0_{\tau\to\tau} &= I^{\otimes\tau} \text{ (the identity map on } \tau\text{-system)} \\ \alpha^{n+1}_{\tau\to\tau} &= \alpha^n_{\tau\to\tau}; \alpha_{\tau\to\tau} \text{ (for } n \geq 1) \end{aligned}$$

4.1 Quantum Teleportation

In quantum teleportation, Alice and Bob, who are separated by a long distance, share a pair of qubits in Bell state $\frac{1}{\sqrt{2}}(|0\rangle_2 |0\rangle_3 + |1\rangle_2 |1\rangle_3)$ (qubit 2 being with Alice, and 3 being with Bob). Alice would like to let Bob have a qubit whose state is the same as the state q of her qubit 1 (which we represent as a state variable of type $\{1\}$). She first interacts the qubit with her end of the Bell state. Define

$$PRE(q) \stackrel{\text{def}}{=} (CNOT_{12}; (H_1 \otimes I_2)) \otimes I_3\Big(q \otimes \big(CNOT_{23}; (H_2 \otimes I_3)(0_2 \otimes 0_3)\big)\Big).$$

She then measures her qubits 1 and 2, and depending on the result sends Bob instructions as to any further operation that must be performed on his qubit 3.

The *standard frame for Teleportation* is the frame that interprets as intended all the constants occurring in the Teleportation protocol: the constants 0_i and 1_i for each $i \in \{1, 2, 3\}$ as well as I_2, I_3, H_1, H_2 $CNOT_{12}$, $CNOT_{23}$ and FP_{13}.

The correctness of the Teleportation protocol is equivalent to the validity in its standard frame of the formula

$$\begin{aligned} &\forall q\big[(q \otimes 0_2 \otimes 0_3) \doteq (0_1? \otimes 0_2? \otimes I_3); (FP_{13} \otimes I_2)(PRE(q)) \\ &\quad \wedge (q \otimes 1_2 \otimes 0_3) \doteq (0_1? \otimes 1_2? \otimes I_3); (I_1 \otimes I_2 \otimes X_3); (FP_{13} \otimes I_2)(PRE(q)) \\ &\quad \wedge (q \otimes 0_2 \otimes 1_3) \doteq (1_1? \otimes 0_2? \otimes I_3); (I_1 \otimes I_2 \otimes Z_3); (FP_{13} \otimes I_2)(PRE(q)) \\ &\quad \wedge (q \otimes 1_2 \otimes 1_3) \doteq (1_1? \otimes 1_2? \otimes I_3); (I_1 \otimes I_2 \otimes (X_3; Z_3)); (FP_{13} \otimes I_2)(PRE(q))\big] \end{aligned}$$

4.2 Quantum Search Algorithm

In the search problem, we are given a unitary operator O, which is usually called an *oracle*, acting on $N+1$ qubits (we assume them to be indexed by elements in $\mathbf{N}+\mathbf{1}$), such that there is a classical state $|f_0\rangle$ with the property that, for each classical state $|f\rangle$ and $b \in \mathbf{2}$,

$$O(|f\rangle \otimes |b\rangle_N) = \begin{cases} |f\rangle \otimes |1-b\rangle_N, & \text{if } f = f_0, \\ |f\rangle \otimes |b\rangle_N, & \text{if } f \in {}^{\mathbf{N}}\mathbf{2} \setminus \{f_0\} \end{cases} \tag{1}$$

The aim of the algorithm is to find out the classical state $|f_0\rangle$.

To formalize the correctness of this algorithm, we use an action variable O of type $\mathbf{N}+\mathbf{1} \to \mathbf{N}+\mathbf{1}$ to denote the oracle. Moreover, we assume that we have an action constant $PS_{\mathbf{N}}$ of type $\mathbf{N} \to \mathbf{N}$ for the action induced by the conditional phase shift gate on the first N qubits, whose matrix under the fixed basis is the following:

$$\begin{bmatrix} \mathbf{Z} & \mathbf{O}_{2\times(N-2)} \\ \mathbf{O}_{(N-2)\times 2} & -\mathbf{I}_{(N-2)\times(N-2)} \end{bmatrix}$$

Here $\mathbf{O}_{2\times(N-2)}$ is the 2 by $N-2$ matrix of only 0 entries, and similarly for $\mathbf{O}_{(N-2)\times 2}$, and $\mathbf{I}_{(N-2)\times(N-2)}$ is the $N-2$ by $N-2$ identity matrix.

As before, the *standard frame* for the $(N+1)$-qubit quantum search algorithm is the one that interprets as intended all the above constants, as well as all the constants 0_i and 1_i. For convenience, we make the following abbreviation

$$\begin{aligned} &\mathit{Oracle}_{\mathbf{N}+\mathbf{1}}(O) \stackrel{\text{def}}{=} \mathit{Unit}(O) \wedge \exists x_{\mathbf{N}} \Big[\mathit{CS}_{\mathbf{N}}(x_{\mathbf{N}}) \wedge \forall y_{\mathbf{N}} \Big(\mathit{CS}_{\mathbf{N}}(y_{\mathbf{N}}) \\ &\to (x_{\mathbf{N}} \doteq y_{\mathbf{N}} \to O(y_{\mathbf{N}} \otimes 0_{N+1}) \doteq y_{\mathbf{N}} \otimes 1_{N+1} \wedge O(y_{\mathbf{N}} \otimes 1_{N+1}) \doteq y_{\mathbf{N}} \otimes 0_{N+1}) \\ &\wedge (x_{\mathbf{N}} \perp y_{\mathbf{N}} \to O(y_{\mathbf{N}} \otimes 0_{N+1}) \doteq y_{\mathbf{N}} \otimes 0_{N+1} \wedge O(y_{\mathbf{N}} \otimes 1_{N+1}) \doteq y_{\mathbf{N}} \otimes 1_{N+1}) \Big) \Big] \end{aligned}$$

for the formula saying that O is an action induced by an oracle acting on the $(\mathbf{N}+\mathbf{1})$-system satisfying Eq.(1).

The correctness of $(N+1)$-qubit Quantum Search Algorithm (with $N > 2$) is equivalent to the validity in its standard frame of the following formula:

$$\begin{aligned} &\forall O \forall x_{\mathbf{N}} \Big\{ \mathit{Oracle}_{\mathbf{N}+\mathbf{1}}(O) \wedge \mathit{CS}_{\mathbf{N}}(x_{\mathbf{N}}) \\ &\wedge O(x_{\mathbf{N}} \otimes 0_N) \doteq x_{\mathbf{N}} \otimes 1_N \wedge O(x_{\mathbf{N}} \otimes 1_N) \doteq x_{\mathbf{N}} \otimes 0_N \to (x_{\mathbf{N}} \otimes H_N(1_N))^{>0.5} \\ &\Big(H^{\otimes(\mathbf{N}+\mathbf{1})}; (O; ((H^{\otimes \mathbf{N}}; PS_{\mathbf{N}}; H^{\otimes \mathbf{N}}) \otimes I_N))^K (0_{\mathbf{N}} \otimes 1_N) \Big) \Big\}, \end{aligned}$$

where K is the largest natural number less than $\frac{\pi}{4}\sqrt{2^N}$.

4.3 Deutsch-Josza Algorithm

In the Deutsch-Josza problem, we are given a unitary operator O (usually called an oracle) acting on $N+1$ qubits (we assume them to be indexed by elements in $\mathbf{N}+\mathbf{1}$), which is known to satisfy one of the following properties:

(i) The oracle is *constant* (having the same value for all inputs): there is $i \in \{0,1\}$ s.t. $O(|f\rangle \otimes |b\rangle_N) = |f\rangle \otimes |b \oplus i\rangle_N$ for all $b \in \mathbf{2}$ and classical state $|f\rangle$, with $f \in {}^{\mathbf{N}}\mathbf{2}$;
(ii) The oracle is *balanced* (equal to 1 for exactly half of all the possible inputs, and 0 for the other half): there is $X \subseteq {}^{\mathbf{N}}\mathbf{2}$ s.t. $|X| = 2^{N-1}$ and $O(|f\rangle \otimes |b\rangle_N)$ is $|f\rangle \otimes |1-b\rangle_N$ if $f \in X$, and is $|f\rangle \otimes |b\rangle_N$, otherwise, for all $b \in \mathbf{2}$.

The goal of the algorithm is to determine which of the two properties holds for O.

To formalize the correctness of this algorithm, we use an action variable O of type $\mathbf{N}+\mathbf{1} \to \mathbf{N}+\mathbf{1}$ to denote the oracle. For convenience, we introduce some abbreviations: first, let us denote by $ConOra(O)$ the formula

$$\begin{array}{l} Unit(O) \wedge \Big[\\ \forall x_{\mathbf{N}} \Big(CS_{\mathbf{N}}(x_{\mathbf{N}}) \to O(x_{\mathbf{N}} \otimes 0_{N+1}) \doteq x_{\mathbf{N}} \otimes 0_{N+1} \wedge O(x_{\mathbf{N}} \otimes 1_{N+1}) \doteq x_{\mathbf{N}} \otimes 1_{N+1} \Big) \\ \vee \forall x_{\mathbf{N}} \Big(CS_{\mathbf{N}}(x_{\mathbf{N}}) \to O(x_{\mathbf{N}} \otimes 0_{N+1}) \doteq x_{\mathbf{N}} \otimes 1_{N+1} \wedge O(x_{\mathbf{N}} \otimes 1_{N+1}) \doteq x_{\mathbf{N}} \otimes 0_{N+1} \Big) \Big] \end{array}$$

saying that O is an action induced by a constant oracle; second, we denote by $BalOra(O)$ the formula (where $k = 2^{N-1}$)

$$\begin{array}{l} Unit(O) \wedge \exists x^1_{\mathbf{N}} ... \exists x^k_{\mathbf{N}} \Big[\Big(\bigwedge_{i=1}^{k} CS_{\mathbf{N}}(x^i_{\mathbf{N}}) \Big) \wedge \Big(\bigwedge_{1 \le i < j \le k} x^i_{\mathbf{N}} \perp x^j_{\mathbf{N}} \Big) \wedge \forall y_{\mathbf{N}} \Big(CS_{\mathbf{N}}(y_{\mathbf{N}}) \to \\ (\bigvee_{i=1}^{k} y_{\mathbf{N}} \doteq x^i_{\mathbf{N}} \to O(y_{\mathbf{N}} \otimes 0_{N+1}) \doteq y_{\mathbf{N}} \otimes 1_{N+1} \wedge O(y_{\mathbf{N}} \otimes 1_{N+1}) \doteq y_{\mathbf{N}} \otimes 0_{N+1}) \\ \wedge (\bigwedge_{i=1}^{k} y_{\mathbf{N}} \perp x^i_{\mathbf{N}} \to O(y_{\mathbf{N}} \otimes 0_{N+1}) \doteq y_{\mathbf{N}} \otimes 0_{N+1} \wedge O(y_{\mathbf{N}} \otimes 1_{N+1}) \doteq y_{\mathbf{N}} \otimes 1_{N+1})) \Big) \Big] \end{array}$$

saying that O is an action induced by a balanced oracle.

Finally, the *correctness of the $(N+1)$-qubit Deutsch-Jozsa algorithm* (for any natural number N) is equivalent to the assertion that the following formula is valid in its standard frame:

$$\begin{array}{l} \forall O \Big\{ ConOra(O) \vee BalOra(O) \to \\ \quad \Big[\Big(ConOra(O) \leftrightarrow H^{\otimes(\mathbf{N}+\mathbf{1})}; O; H^{\otimes(\mathbf{N}+\mathbf{1})}(0_{\mathbf{N}} \otimes 1_N) \doteq 0_{\mathbf{N}} \otimes 1_N \Big) \\ \quad \wedge \Big(BalOra(O) \leftrightarrow H^{\otimes(\mathbf{N}+\mathbf{1})}; O; H^{\otimes(\mathbf{N}+\mathbf{1})}(0_{\mathbf{N}} \otimes 1_N) \perp 0_{\mathbf{N}} \otimes 1_N \Big) \Big] \Big\} \end{array}$$

5 Decidability

The set of validities of **QPDSOL** on any *given frame* is decidable. Using the same proof strategy, the validity problem for *pure* formulas over (the class of) *all frames* is also decidable. In this section, we sketch the proofs of these results.

The basic technique for proving these decidability results is a generalization and extension of the method used in [10]: We express validity of formulas of $\mathcal{L}$ without free variables in a given frame $\mathfrak{F}$ via truth of first-order sentences of $(\mathbb{R}, +, \cdot, 0, 1)$; then the decidability of our logic follows from Tarski's theorem in [17] which states that the first-order theory of $(\mathbb{R}, +, \cdot, 0, 1)$ is decidable. This idea is unfolded into several technical steps.

In the first step, we need to deal with intersection of testable properties. For a function symbol of the form $(P_\tau \cap Q_\tau)?$, it is well known that calculating the matrix of the corresponding projector typically involves a process of taking limits and hence can not be expressed in the first-order theory of $(\mathbb{R}, +, \cdot, 0, 1)$. The key to solving this is the observation that complex predicate symbols, i.e. those built with $\cap$, $\otimes$, $\sim$ and other operations, can be recursively eliminated from our language with the help of quantifiers (over states). Let $\mathcal{L}^*$ be the result of this translation. Its formulas consist of those built as follows (where constraints on the types are those given in Definition 3.1 and 3.2, but with the additional requirement that for each singleton $\tau = \{i\}$, there exists a constant 0_τ that denotes $\widehat{|0\rangle}_i$, so as to facilitate the translation of generalized projectors):

$$\begin{aligned} t_\tau &::= x_\tau \mid c_\tau \mid x_{\tau_1} \otimes x_{\tau_2} \mid \alpha_{\rho\to\tau}(x_\rho) \\ P_\tau &::= p_\tau \mid T_\tau \\ \alpha_{\tau\to\rho} &::= a_{\tau\to\rho} \mid C_{\tau\to\rho} \mid a^\dagger_{\rho\to\tau} \mid a_{\tau_1\to\rho_1} \otimes a'_{\tau_2\to\rho_2} \mid P_\tau? \\ \varphi &::= x^{<r}_\tau(t_\tau) \mid x^{=r}_\tau(t_\tau) \mid \neg\varphi \mid \varphi \wedge \varphi \mid \forall x_\tau \varphi \mid \forall p_\tau \varphi \mid \forall a_{\rho\to\tau}\varphi \end{aligned}$$

With the possible exception of the constants 0_τ, we have that $\mathcal{L}^* \subseteq \mathcal{L}$, and the semantics of $\mathcal{L}^*$ is the same as for $\mathcal{L}$. One can define a function $\nabla : \mathcal{L} \to \mathcal{L}^*$ by recursion (and hence it is computable) s.t. $\mathfrak{M} \models \varphi \Leftrightarrow \mathfrak{M} \models \nabla(\varphi)$ for every model $\mathfrak{M}$. To illustrate why this is the case and how it helps to solve the problem, we exhibit one case in its definition:

$$\begin{aligned} \nabla(x^{=r}_\tau((P_\tau \cap Q_\tau)?(t_\tau))) = \exists y_\tau \exists z_\tau \big[\, & \nabla(t_\tau \doteq y_\tau \oplus z_\tau) \wedge y^{=0}_\tau(z_\tau) \wedge x^{=r}_\tau(y_\tau) \\ & \wedge \forall u_\tau \big(\nabla(P^{=1}_\tau(u_\tau)) \wedge \nabla(Q^{=1}_\tau(u_\tau)) \to z^{=0}_\tau(u_\tau)\big) \,\big] \end{aligned}$$

where x_τ is a state variable, t_τ is a term and $t_\tau \doteq y_\tau \oplus z_\tau$ is defined to be $\forall v_\tau(v^{=0}_\tau(y_\tau) \wedge v^{=0}_\tau(z_\tau) \to v^{=0}_\tau(t_\tau))$, which means that t_τ "lies on the plane generated by" y_τ and z_τ.

In the second step, we define for each frame $\mathfrak{F}$, a function $TR_\mathfrak{F} : \mathcal{L}^* \to \mathcal{L}_\mathbb{C}$, where $\mathcal{L}_\mathbb{C}$ is the first-order language of $(\mathbb{C}, +, \cdot, \bar{\cdot}, \prec, \mathbb{C})$, where $\bar{\cdot}$ is the conjugate operator, $\prec$ is a binary relation between complex numbers such that $a+bi \prec c+di$ iff $a < c$, and the last component $\mathbb{C}$ is the set of numbers named by a constant. Towards this aim, we first formalize in $\mathcal{L}_\mathbb{C}$ the matrix representation of the interpretation in $\mathfrak{F}$ of terms, predicate symbols and function symbols. This is possible because every term, predicate symbol and function symbol involves only finitely many qubits indicated by its type. In fact, one can define by recursion a computable function $\mathfrak{F}^\natural$ from the set of terms, predicate symbols and function symbols that can occur in formulas in $\mathcal{L}^*$ to the set of finite sets of terms in

$\mathcal{L}_{\mathbb{C}}$. For the base case, we define $\mathfrak{F}^\sharp(x_\tau)$, $\mathfrak{F}^\sharp(p_\tau)$ and $\mathfrak{F}^\sharp(a_{\tau\to\rho})$ to be the sets of variables indexed by ${}^\tau\mathbf{2}$, ${}^\tau\mathbf{2} \times^\tau \mathbf{2}$ and ${}^\rho\mathbf{2} \times^\tau \mathbf{2}$ in such a way that different state, predicate or action variables are mapped to disjoint sets of variables. Moreover, $\mathfrak{F}^\sharp(c_\tau)$, $\mathfrak{F}^\sharp(T_\tau)$ and $\mathfrak{F}^\sharp(C_{\tau\to\rho})$ are indexed in a similar way but they are sets of constants. Care must be taken to ensure that the constants are defined according to the interpretation in $\mathfrak{F}$. For complex symbols built with operations, we can mimic the manipulation of vectors and matrices. For example, assume that we have defined $\mathfrak{F}^\natural(x_\tau)$ to be the set of variables $\{\mathsf{x}[f] \mid f \in^\tau \mathbf{2}\}$ and $\mathfrak{F}^\natural(y_\rho)$ to be $\{\mathsf{y}[g] \mid g \in^\rho \mathbf{2}\}$ respectively, then we can mimic the Kronecker product of matrices and define $\mathfrak{F}^\natural(x_\tau \otimes y_\rho)$ to be the set of terms $\{\mathsf{x} \otimes \mathsf{y}[h] \mid h \in^{\tau\cup\rho} \mathbf{2}\}$ s.t. $\mathsf{x} \otimes \mathsf{y}[h] = \mathsf{x}[h\restriction\tau] \cdot_{\mathbb{C}} \mathsf{y}[h\restriction\rho]$, where $\cdot_{\mathbb{C}}$ is the symbol for multiplication in $\mathcal{L}_{\mathbb{C}}$. Using the function $\mathfrak{F}^\natural$, we proceed to define $TR_{\mathfrak{F}}$ in such a way that given a model $\mathfrak{M}$ on the frame $\mathfrak{F}$, $\mathfrak{M} \models \varphi$ iff $(\mathbb{C}, +, \cdot, \bar{}, \prec, \mathbb{C}) \models_{\mathfrak{M}} TR_{\mathfrak{F}}(\varphi)$, for every $\varphi \in \mathcal{L}^*$. Here the subscript in "$\models_{\mathfrak{M}}$" is an interpretation (added to the structure $(\mathbb{C}, +, \cdot, \bar{}, \prec, \mathbb{C})$) of the free variables in $TR_{\mathfrak{F}}(\varphi)$ according to the model $\mathfrak{M}$. In defining $TR_{\mathfrak{F}}$ as such, care is taken in order to verify that quantification over (finitely many) variables in $\mathfrak{F}^\sharp(x_\tau)$, $\mathfrak{F}^\sharp(p_\tau)$ or $\mathfrak{F}^\sharp(a_{\tau\to\rho})$ in the input formula really corresponds to quantification of x_τ, p_τ or $a_{\tau\to\rho}$ in the translated formula.

In the third step, we focus on the behaviour of $TR_{\mathfrak{F}}$ on the set of closed formulas. Since the definition of frames ensures that the matrix representation of the interpretation of constant symbols only has simple complex numbers as entries, the translation $TR_{\mathfrak{F}}(\varphi)$ of a closed formula φ of $\mathcal{L}^*$ in a given frame $\mathfrak{F}$ is actually a first-order sentence of $(\mathbb{C}, +, \cdot, \bar{}, \prec, \mathcal{S})$, where $\mathcal{S}$ is the set of simple complex numbers (see page 9). A consequence of this is that pure formulas of $\mathcal{L}$ are translated via $TR_{\mathfrak{F}}$ into first-order sentences of $(\mathbb{C}, +, \cdot, \bar{}, \prec)$, because there are no constants in a pure formula. Therefore, by Lemma 3.8 and the property of $TR_{\mathfrak{F}}$ by definition, we know that on a given frame $\mathfrak{F}$, $\mathfrak{F} \models \varphi$ iff $(\mathbb{C}, +, \cdot, \bar{}, \prec, \mathcal{S}) \models TR_{\mathfrak{F}} \circ \nabla(\varphi)$, for every closed formula $\varphi \in \mathcal{L}$.

The final step is to reduce the first-order theory of $(\mathbb{C}, +, \cdot, \bar{}, \prec, \mathcal{S})$ to the first-order theory of the reals. This is a simple translation, where each simple complex number is mapped to a pair of definable real numbers, and addition and multiplication are mapped according to complex arithmetic. Thus the decidability of our logic follows from these reductions and Tarksi's theorem. To summarize, we have the following decidability result.

Theorem 5.1. *The set $\{\varphi \in \mathcal{L} \mid \varphi$ is closed and $\mathfrak{F} \models \varphi\}$ is decidable, for any given frame $\mathfrak{F}$. Moreover, the set $\{\varphi \in \mathcal{L} \mid \varphi$ is pure and $\models \varphi\}$ is decidable.*

6 Conclusions

This paper extends decidability results from [10] and [4] to a language that is much more versatile in its ability to express quantum algorithms and their correctness. Our techniques can be applied to a wider range of quantum logics, giving a general recipe for showing decidability as long as definability of the sentences and operators can be done along the lines presented in this paper. In

addition we have described how to express the correctness of Quantum Teleportation, the Quantum Search algorithm and the Deutsch-Josza algorithm; however this is not an exhaustive list of algorithms whose correctness can be expressed in our language. The Fourier transform can easily be expressed in our language and this may lead to a wealth of further examples, notably those involving the hidden subgroup problem, such as order-finding and factoring; however we leave these for future work. Other future tasks involve finding a complete axiomatization and determining the complexity of the decision procedure.

References

1. S. Abramsky and B. Coecke. A categorical semantics of quantum protocols. In *Proceedings of the 19th IEEE conference on Logic in Computer Science (LiCS'04)*, pages 415–425. IEEE Press, 2004.
2. D. Aerts. Description of compound physical systems and logical interaction of physical systems. In E. Beltrametti and B. van Fraassen, editors, *Current Issues on Quantum Logic*, pages 381–405. Kluwer Academic, 1981.
3. A. Baltag, J. Bergfeld, K. Kishida, J. Sack, S. Smets, and S. Zhong. PLQP & company: Decidable logics for quantum algorithms. Submitted to the International Journal of Theoretical Physics, 2013.
4. A. Baltag, J. Bergfeld, K. Kishida, J. Sack, S. Smets, and S. Zhong. A Decidable Dynamic Logic for Quantum Reasoning. *EPTCS*, in print, 2012.
5. A. Baltag and S. Smets. Complete Axiomatizations for Quantum Actions. *International Journal of Theoretical Physics*, 44(12):2267–2282, 2005.
6. A. Baltag and S. Smets. LQP: The Dynamic Logic of Quantum Information. *Mathematical Structures in Computer Science*, 16(3):491–525, 2006.
7. G. Birkhoff and J. von Neumann. The Logic of Quantum Mechanics. *The Annals of Mathematics*, 37:823–843, 1936.
8. R. Chadha, P. Mateus, A. Sernadas, and C. Sernadas. Extending classical logic for reasoning about quantum systems. In K. Engesser, D. M. Gabbay, and D. Lehmann, editors, *Handbook of Quantum Logic and Quantum Structures: Quantum Logic*, pages 325–371. Elsevier, 2009.
9. M. L. Dalla Chiara, R. Giuntini, and R. Greechie. *Reasoning in quantum theory: sharp and unsharp quantum logics*, volume 22 of *Trends in logic*. Kluwer Acadamic Press, Dordrecht, 2004.
10. J. M. Dunn, T. J. Hagge, L. S. Moss, and Z. Wang. Quantum Logic as Motivated by Quantum Computing. *The Journal of Symbolic Logic*, 70(2):353–359, 2005.
11. L. Henkin. Completeness in the Theory of Types. *The Journal of Symbolic Logic*, 15:81–91, 1950.
12. M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2011.
13. C. Piron. *Foundations of Quantum Physics*. W.A. Benjamin Inc., 1976.
14. M. Rabin. Decidability of second order theories and automata on infinite trees. *Transactions of the American Mathematical Society*, pages 1–35, 1969.
15. C. Randall and D. Foulis. Tensor products of quantum logics do not exist. *Notices Amer. Math. Soc.*, 26(6), 1979.
16. P. Selinger. Towards a quantum programming language. *Mathematical Structures in Computer Science*, 14:527–586, 7 2004.
17. A. Tarski. *A Decision Method for Elementary Algebra and Geometry*. RAND Corporation, Santa Monica, California, 1948.