Quantum Alternation: Prospects and Problems

Costin Bădescu
McGill University
Montréal, Canada
cbades@cs.mcgill.ca

Prakash Panangaden
McGill University
Montréal, Canada
prakash@cs.mcgill.ca

We propose a notion of quantum control in a quantum programming language which permits the superposition of finitely many quantum operations without performing a measurement. This notion takes the form of a conditional construct similar to the `if` statement in classical programming languages. We show that adding such a quantum `if` statement to the QPL programming language [11] simplifies the presentation of several quantum algorithms. This motivates the possibility of extending the denotational semantics of QPL to include this form of quantum alternation. We give a denotational semantics for this extension of QPL based on Kraus decompositions rather than on superoperators. Finally, we clarify the relation between quantum alternation and recursion, and discuss the possibility of lifting the semantics defined by Kraus operators to the superoperator semantics defined by Selinger [11].

1 Introduction

The field of quantum programming languages emerged in the early 2000s as a result of researchers' interest in understanding quantum algorithms structurally. This interest is backed by the belief that a structural study of quantum algorithms may have the same positive effect on our understanding of quantum computing as the introduction of structured programming had on classical computation. This endeavor has two clear objectives: understanding how fundamental quantum resources such as quantum parallelism and entanglement fit into the theory of computation, and exploiting these resources to aid in designing new quantum algorithms which can outperform the existing classical ones.

Conforming to this structural approach, the present work casts quantum parallelism as a resource which can be used to determine the control flow of a program. This flow is usually built up by composing three primitive operations: sequencing, branching, and recursion. Of these three, branching is the only operation which depends on data supplied to the program. In quantum computing, this data can be a qubit whose state is unknown. In this case, a measurement is normally used to extract a Boolean value from the qubit and the transition to the next state depends on the measurement outcome. This procedure is similar to sampling a Bernoulli random variable where the distribution is determined by the state of the qubit. Hence, the form of quantum control implemented by measurements is of a probabilistic nature. A natural question to ask is whether there is a sensible notion of branching in a quantum programming language which operates at the quantum level,

To appear in EPTCS.

that is, without interference from the environment. This speculative type of branching is henceforth referred to as *quantum alternation* or *quantum control*. Investigating the viability of this concept is the main theme of this paper.

The idea of quantum control is not new. Indeed, in a quantum Turing machine [4] – the first formalism of quantum computation – the flow of execution is described by a constant unitary operator. Thus, both data and control may be "quantum." Nevertheless, the passage from the quantum control mechanism present in a quantum Turing machine to a structural notion of quantum branching in a programming language is not clear. The first programming language designed to support quantum control was defined by Altenkirch and Grattage in [1]. The language, called QML, provides a `case` statement which allows superposing several quantum operations without performing a measurement. However, the `case` statement can only be used in certain situations specified by the introduction rules of the type system which use an "orthogonality" judgement. A more recent work on quantum alternation is [13] where the authors propose a language called QGCL (after Dijkstra's Guarded Command Language) to support the paradigm of "superposition of programs." QGCL bases the definition of quantum control on the analogy with quantum random walks and introduces an auxiliary system of "quantum coins" which is used to perform branching. A more detailed discussion of both of these works and their relation to the work presented in this paper is deferred to the section on related work. For the moment, we note that there are many similarities and a few differences between our work and the work reported in [13].

We proceed to outline the basic properties that quantum alternation should possess. The notation used in the sequel follows the usual mathematical framework for open quantum systems: states are represented by density operators on some Hilbert space, and quantum operations are given by superoperators, i.e. completely positive (CP) trace-nonincreasing maps. All Hilbert spaces are assumed to be finite-dimensional, unless otherwise stated. If $\mathcal{H}$ is a Hilbert space, we denote by $S(\mathcal{H})$ the set of states on $\mathcal{H}$. Thus, a superoperator is a linear map $T : S(\mathcal{H}) \to S(\mathcal{K})$. The dynamics defined by a superoperator $T : S(\mathcal{H}) \to S(\mathcal{H})$ is said to be *reversible* if T can be represented as a pure unitary operation, viz. $T(\rho) = U\rho U^\dagger$ for some unitary operator $U : \mathcal{H} \to \mathcal{H}$. **qbit** is defined to be the 2-dimensional Hilbert space $\mathbb{C}^2$ with the computational basis $|0\rangle$ and $|1\rangle$. A qubit is a term q of type **qbit**, denoted $q : \mathbf{qbit}$. We define the classical states $\Pi_0 = |0\rangle\langle 0|$ and $\Pi_1 = |1\rangle\langle 1|$ corresponding to the elements of the computational basis.

We posit the following typing judgement for quantum alternation. Given a qubit $q : \mathbf{qbit}$ and two superoperators $T_0, T_1 : S(\mathcal{H}) \to S(\mathcal{K})$, the alternation of T_0 and T_1 with respect to q should be a superoperator $\mathrm{Alt}_q(T_0, T_1) : S(\mathbf{qbit} \otimes \mathcal{H}) \to S(\mathbf{qbit} \otimes \mathcal{K})$. Thus,

I. Quantum alternation has the following typing judgement, where Π is a procedure context and Γ and Γ' are typing contexts:

$$\frac{\Pi \vdash \langle \Gamma \rangle\, P\, \langle \Gamma' \rangle \qquad \Pi \vdash \langle \Gamma \rangle\, Q\, \langle \Gamma' \rangle}{\Pi \vdash \langle q : \mathbf{qbit}, \Gamma \rangle\, \mathbf{if}\ q\ \mathbf{then}\ P\ \mathbf{else}\ Q\, \langle q : \mathbf{qbit}, \Gamma' \rangle}$$

Note that, according to the typing judgement, the branches P and Q cannot access the

qubit q. There are at least two reasons for this particular choice. Firstly, we will require that the alternation of P and Q with respect to q is a reversible operation if P and Q are reversible, which is not necessarily the case if P and Q are allowed access to q. Secondly, q is a resource used to superpose different statements and, as with any type of resource, it should be in some sense consumed. This situation is not unlike the case of measurement where the state of the qubit collapses to the classical state observed. The difference here is that quantum branching does not extract any classical information from q, so the qubit does not collapse to a classical state.

The second fundamental property required of quantum alternation is that it should use the information encoded in the classical states of q. That is, the alternation should depend on a specific choice of basis for q and each branch must correspond to a distinct basis vector. The state of q should affect the superposition of quantum operations:

II. If the qubit q is in a classical state Π_i with $i \in \{0,1\}$, then $\mathrm{Alt}_q(T_0,T_1) = I \otimes T_i$, i.e. the alternation reduces to a local operation T_i on $S(\mathcal{H})$.

The second condition formalizes the intuition of classical alternation in this context. Since $\mathrm{Alt}_q(T_0,T_1)$ is a linear map, it follows that if ρ is a state on $\mathbf{qbit} \otimes \mathcal{H}$ then

$$\mathrm{Alt}_q(T_0,T_1) :: \rho = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \mapsto \begin{bmatrix} T_0 A & * \\ * & T_1 D \end{bmatrix}.$$

The off-diagonal asterisks represent entries which are not yet determined by anything other than the blocks on the diagonal and the condition that the result must be a positive operator. If these entries are null, then $\mathrm{Alt}_q(T_0,T_1)$ can be implemented by a measurement followed by merging. Hence, it is necessary to impose additional constraints to obtain a notion of branching which may be called "quantum." The final condition we impose, concerning the reversibility of alternation, addresses this issue:

III. If T_0 and T_1 are reversible, then $\mathrm{Alt}_q(T_0,T_1)$ is reversible.

The dynamics of a closed quantum-mechanical system is reversible, so this requirement is natural, if not compulsory, for any definition of quantum alternation. The reversibility condition also ensures that the implementation of alternation cannot be based on measurement.

Following the conditions introduced above, we can suggest a definition of quantum alternation in a *closed* quantum system:

Let $\mathcal{H}$ be a Hilbert space and let $U_0, U_1 : \mathcal{H} \to \mathcal{H}$ be unitary operators. Given a qubit $q : \mathbf{qbit}$, define the alternation $\mathrm{Alt}_q(U_0,U_1)$ with respect to q by

$$\mathrm{Alt}_q(U_0,U_1) = \Pi_0 \otimes U_0 + \Pi_1 \otimes U_1. \tag{1}$$

This definition of Alt meets all three conditions and generalizes immediately to a definition of quantum alternation controlled by a system of multiple qubits. Let $\mathbf{qbit}^n$ be the $n$fold

tensor product of **qbit** with itself and set $\ell = 2^n - 1$. Let $\Pi_0, \ldots, \Pi_\ell$ be the classical states of $\mathbf{qbit}^n$. Given $\bar{q} : \mathbf{qbit}^n$, the alternation of unitary operators $U_0, \ldots, U_\ell : \mathcal{H} \to \mathcal{H}$ with respect to $\bar{q}$ is defined by

$$\mathrm{Alt}_{\bar{q}}(U_0, \ldots, U_\ell) = \sum_{k=0}^{\ell} \Pi_k \otimes U_k. \tag{2}$$

This form of alternation corresponds to a quantum `case` statement. As we will see, the Deutsch–Jozsa algorithm can be obtained from Deutsch's algorithm essentially by replacing an `if` statement with a `case` statement.

(2) is a special case of a *measuring operator* [7]. In the definition of a measuring operator, the classical states Π_k can be replaced by projections onto pairwise orthogonal subspaces. Thus, it is possible to consider a slightly more general notion of quantum alternation where the superposition is controlled by a set of pairwise orthogonal projections rather than by a system of qubits; this idea is also introduced in [13].

The problem of defining quantum alternation in QPL amounts to finding an appropriate extension of the definition given above to open quantum systems which is structural, compositional, and satisfies the three aforementioned criteria.

2 Examples

Prior to defining a semantics for quantum control in open quantum systems, we present a few examples of QPL programs which make use of quantum alternation in a closed system. Thus, all quantum operations considered in this section are pure operations associated with a specific *unitary* operator defined within the program.

We briefly review the fragment of QPL which will be used in this paper. The state of a QPL program is a density matrix and a statement is interpreted as a superoperator. The primitives we will use are as follows: **skip** is the identity superoperator; $\bar{q} \mathrel{*}= U$ applies the unitary transformation U to the tuple of qubits $\bar{q}$; **new qbit** q allocates a new qubit register named q initialized to $|0\rangle$; **measure** q **then** P **else** Q measures the qubit register q and evaluates P or Q accordingly; **discard** q represents the partial trace over the component of the state space represented by q.

We will make use of two additional constructs to illustrate quantum alternation: an **if** q **then** P **else** Q statement interpreted as the superoperator defined by (1), and a **case** $\bar{q}$ **of** $\Pi_k \to P_k$ statement interpreted as the superoperator defined by (2). Note that all branches of an alternation (e.g. P, Q, etc.) are assumed to be pure unitary operations.

The simplest example using quantum alternation is the construction of controlled unitary operators. If U is a unitary operator and $q_0, q_1 : \mathbf{qbit}$ are two qubits, then

$$\textbf{if } q_0 \textbf{ then skip else } q_1 \mathrel{*}= U$$

implements a controlled-U operation. Thus, if N is the NOT gate, two nested `if` statements

can be used to implement the Toffoli gate:

$$\textbf{if } q_0 \textbf{ then skip else if } q_1 \textbf{ then skip else } q_2 \mathrel{*}= N$$

Implementing a controlled gate using an `if` statement allows for a more succint presentation of quantum circuits in QPL. For instance, given qubits $q_1, \ldots, q_n : \textbf{qbit}$, the following program implements an efficient circuit for the quantum Fourier transform (cf. [9, p. 219]):

$$\begin{aligned}
&\textbf{for } i = 1 \textbf{ to } n \textbf{ do}\\
&\quad q_i \mathrel{*}= H\\
&\quad \textbf{for } k = 2 \textbf{ to } n - i + 1 \textbf{ do}\\
&\qquad \textbf{if } q_{k+i-1} \textbf{ then skip else } q_i \mathrel{*}= R_k
\end{aligned}$$

Here R_k is the phase shift gate defined by $R_k = \Pi_0 + e^{i\theta}\Pi_1$ with $\theta = 2\pi/2^k$.

A more important example, exhibiting the relation between quantum parallelism and quantum alternation, is an implementation of *Deutsch's algorithm* [4]. The problem is to determine whether a given Boolean function $f : \{0,1\} \to \{0,1\}$ is constant.

For each $x \in \{0,1\}$, let $U_x : \textbf{qbit} \to \textbf{qbit}$ be the permutation operator transposing $|0\rangle$ with $|f(x)\rangle$ and fixing the rest of the basis. Let $x \oplus y$ denote the *exclusive or* of bits x and y. Note that $0 \oplus x = x$ and $1 \oplus x = \neg x$ for all $x \in \{0,1\}$. Thus, $U_x|y\rangle = |y \oplus f(x)\rangle$ for $x, y \in \{0,1\}$. Given qubits $q_0, q_1 : \textbf{qbit}$, consider the statement:

$$\textbf{if } q_0 \textbf{ then } q_1 \mathrel{*}= U_0 \textbf{ else } q_1 \mathrel{*}= U_1$$

Using definition (1), this statement is interpreted as the pure operation defined by the unitary:

$$U_f :: |0\rangle \otimes \psi_0 + |1\rangle \otimes \psi_1 \;\mapsto\; |0\rangle \otimes U_0\psi_0 + |1\rangle \otimes U_1\psi_1.$$

A simple calculation shows that U_f can also be defined by the map $|x, y\rangle \mapsto |x, y \oplus f(x)\rangle$. Therefore, Deutsch's algorithm can be implemented as follows.

$$\begin{aligned}
&\textbf{new qbit } q_0, q_1\\
&q_0 \mathrel{*}= H\\
&q_1 \mathrel{*}= H \circ N\\
&\textbf{if } q_0 \textbf{ then } q_1 \mathrel{*}= U_0 \textbf{ else } q_1 \mathrel{*}= U_1\\
&q_0 \mathrel{*}= H
\end{aligned}$$

The algorithm above can be modified to take as input a general Boolean function $f : \{0,1\}^n \to \{0,1\}$. A map such as f is said to be *balanced* if $\mathbf{P}[f(x) = 1] = \frac{1}{2}$ for a uniformly random $x \in \{0,1\}^n$. The *Deutsch–Jozsa algorithm* [5], a generalization of Deutsch's algorithm, determines whether a given Boolean function $f : \{0,1\}^n \to \{0,1\}$ is constant or not contingent upon the assumption that f either constant or balanced. An implementation

of this algorithm is obtained essentially by replacing the `if` statement above with a `case` statement. Indeed, for each $x \in \{0,1\}^n$, let U_x be the permutation operator transposing $|0\rangle$ with $|f(x)\rangle$ and fixing the rest of the basis. Suppose $\bar{q}_0 : \mathbf{qbit}^n$ and $q_1 : \mathbf{qbit}$ are given. The statement

$$\mathbf{case}\ \bar{q}_0\ \mathbf{of}\ |x\rangle \to q_1 \mathrel{*}= U_x \tag{3}$$

implements the unitary $\tilde{U}_f :: |x, y\rangle \mapsto |x, y \oplus f(x)\rangle$ with $x \in \{0,1\}^n$. Hence, the Deutsch–Jozsa algorithm can be written as:

$$\begin{aligned}
&\mathbf{new}\ \mathbf{qbit}^n\ \bar{q}_0\\
&\mathbf{new}\ \mathbf{qbit}\ q_1\\
&\bar{q}_0 \mathrel{*}= H^{\otimes n}\\
&q_1 \mathrel{*}= H \circ N\\
&\mathbf{case}\ \bar{q}_0\ \mathbf{of}\ |x\rangle \to q_1 \mathrel{*}= U_x\\
&\bar{q}_0 \mathrel{*}= H^{\otimes n}
\end{aligned}$$

The map which assigns the unitary operator $\tilde{U}_f$ to a Boolean function f appears in a number of quantum algorithms. For instance, if $f(x_0) = 1$ for some $x_0 \in \{0,1\}^n$ and $f(x) = 0$ otherwise, then $\tilde{U}_f$ is the "black box oracle" O used to implement Grover's search algorithm (see e.g. [9, p. 254]). Similarly, $\tilde{U}_f$ is used in the period-finding algorithm if f is a periodic function.

The ability of quantum computation to superpose multiple evaluations of a function f in a single application of a unitary operator is often referred to as quantum parallelism. Considering the permutation matrix U_x as an evaluation of f at x, the definition of $\tilde{U}_f$ as the `case` statement in (3) shows that quantum alternation embodies a form of quantum parallelism. Furthermore, the fact that an application of $\tilde{U}_f$ is considered a $O(1)$ operation is reflected in the syntactic representation of alternation as a conditional construct.

Finally, an elementary but important observation is that the conditional statement

$$\mathbf{if}\ q_0\ \mathbf{then}\ \mathbf{skip}\ \mathbf{else}\ q_1 \mathrel{*}= e^{i\theta}$$

implements a controlled phase. Since **skip** and $q_1 \mathrel{*}= e^{i\theta}$ are physically indistinguishable as quantum operations, it follows that quantum alternation is not directly physically realizable. Rather, it represents a conceptual semantic construct in a quantum programming language. Furthermore, this example shows that there is no structural semantics for quantum alternation which is based on superoperators with extensional equality.

3 Semantics

In this section, we give a definition of quantum alternation for open quantum systems and present a formal semantics for QPL with quantum control. We only define alternation with

respect to a single qubit $q:\mathbf{qbit}$ and two branches. A formula for the general case can be easily obtained using the same techniques.

Let $\mathcal{H}$, $\mathcal{K}$, and $\mathcal{L}$ be Hilbert spaces. A finite set $\mathcal{S}$ of nonzero bounded operators from $\mathcal{H}$ to $\mathcal{K}$ defines a superoperator $T : S(\mathcal{H}) \to S(\mathcal{K})$ by

$$T(\rho) = \sum_{E \in \mathcal{S}} E\rho E^\dagger \qquad \text{if} \qquad \sum_{E \in \mathcal{S}} E^\dagger E \leqslant I. \tag{4}$$

We will refer to $\mathcal{S}$ as a *decomposition of* T or, when the superoperator is implicit, as a *Kraus decomposition*. A well-known theorem of Kraus [8] states that every superoperator has a decomposition, but this decomposition is never unique. Thus, two Kraus decompositions $\mathcal{S}$ and $\mathcal{T}$ are said to be *extensionally equal*, denoted $\mathcal{S} \simeq \mathcal{T}$, if the corresponding superoperators are equal. The empty set $\varnothing$ corresponds to the 0 superoperator.

If $\mathcal{S} \subseteq B(\mathcal{K},\mathcal{L})$ and $\mathcal{T} \subseteq B(\mathcal{H},\mathcal{K})$ are Kraus decompositions, their *composition* $\mathcal{S} \circ \mathcal{T}$ is defined to be the set obtained from the multiset $\{E \circ F \mid E \in \mathcal{S}, F \in \mathcal{T}\}$ by replacing ℓ occurences of a bounded operator K with $\sqrt{\ell}K$ and removing any occurrence of the zero operator. Each Hilbert space $\mathcal{H}$ with identity operator $I : \mathcal{H} \to \mathcal{H}$ determines a unique Kraus decomposition $\mathrm{id}_{\mathcal{H}} = \{I\}$ which acts as the identity for composition. Thus, we can define a category $\mathbf{C}$ with Hilbert spaces $\mathcal{H},\mathcal{K}$ as objects and Kraus decompositions $\mathcal{S} \subseteq B(\mathcal{H},\mathcal{K})$ as morphisms $\mathcal{S} : \mathcal{H} \to \mathcal{K}$. A statement in QPL will be interpreted as a morphism in $\mathbf{C}$.

We define the *quantum alternation* of two morphisms[1] $\mathcal{S},\mathcal{T} : \mathcal{H} \to \mathcal{K}$ to be the morphism $\mathcal{S} \bullet \mathcal{T} : \mathbf{qbit} \otimes \mathcal{H} \to \mathbf{qbit} \otimes \mathcal{K}$ defined by

$$\mathcal{S} \bullet \mathcal{T} = \left\{ \Pi_0 \otimes \frac{E}{\sqrt{|\mathcal{T}|}} + \Pi_1 \otimes \frac{F}{\sqrt{|\mathcal{S}|}} \mid E \in \mathcal{S}, F \in \mathcal{T} \right\}.$$

Here the projections Π_0 and Π_1 are determined by the qubit $q:\mathbf{qbit}$ which is used in the alternation. It is easy to see that $\mathcal{S} \bullet \mathcal{T}$ satisfies condition (4). Moreover, if $\mathcal{S} = \{U_0\}$ and $\mathcal{T} = \{U_1\}$ where U_0 and U_1 are unitary operators, then $\mathcal{S} \bullet \mathcal{T}$ defines the same superoperator as $\mathrm{Alt}_q(U_0,U_1)$. Indeed, the elements of $\mathcal{S} \bullet \mathcal{T}$ are of the form $\mathrm{Alt}_q(\hat{E},\hat{F})$ where

$$\hat{E} = \frac{E}{\sqrt{|\mathcal{T}|}}, \quad \hat{F} = \frac{F}{\sqrt{|\mathcal{S}|}}, \quad \text{for } E \in \mathcal{S} \text{ and } F \in \mathcal{T}.$$

Thus, $\mathcal{S} \bullet \mathcal{T}$ can be understood operationally as randomly replacing a state ρ with $K\rho K^\dagger / \mathrm{tr}(K\rho K^\dagger)$ with probability $\mathrm{tr}(K\rho K^\dagger)$ where K is the "pure" quantum alternation $\mathrm{Alt}_q(\hat{E},\hat{F})$.

We briefly recall the definition of the category $\mathbf{Q}$ associated to the superoperator semantics of QPL. A *signature* σ is defined to be a tuple of positive integers $\sigma = (n_1,\ldots,n_s)$. If σ and τ are signatures, then their concatenation $\sigma \oplus \tau$ and tensor product $\sigma \otimes \tau$ are also signatures. To each such σ, we associate a complex vector space

$$V_\sigma = M(\mathbb{C},n_1) \times \ldots \times M(\mathbb{C},n_s),$$

[1] This equation also appears in [13].

where $M(\mathbb{C},k)$ denotes the vector space of $k \times k$ complex matrices. Clearly, $M(\mathbb{C},k) = B(\mathbb{C}^k)$, so the elements of V_σ are tuples of bounded operators. We define the trace of an element in V_σ to be the sum of the traces of its components and say that an element of V_σ is positive if all of its components are positive operators. Thus, a density operator in V_σ is a positive element with trace at most 1. The semantics of QPL, as defined in [11], is given by the category **Q** whose objects are signatures σ, τ and whose morphisms are superoperators $T : V_\sigma \to V_\tau$.

A semantics for QPL with quantum control is obtained by replacing the morphisms of **Q** with Kraus decompositions. The resulting category is the category **C** defined above. We assign to each QPL primitive a Kraus decomposition and define the semantics of an arbitrary program by structural induction. Although the choice of Kraus decomposition for a primitive may be arbitrary, we will rely on the fact that the computational basis for **qbit** is the "preferred" basis and give Kraus decompositions which are particularly simple to express using $|0\rangle$ and $|1\rangle$. For instance, let $\mathrm{in}_0, \mathrm{in}_1 : \sigma \to \sigma \oplus \sigma$ be the injections $\mathrm{in}_0(\rho) = (\rho, 0)$ and $\mathrm{in}_1(\rho) = (0, \rho)$. We can then define the semantics as follows.

$$
\begin{array}{lll}
[\![P; Q]\!] & : \sigma \to \tau & = [\![Q]\!] \circ [\![P]\!] \\
[\![\mathbf{skip}]\!] & : \sigma \to \sigma & = \{\mathrm{id}\} \\
[\![\mathbf{new\ bit}\ b := 0]\!] & : \sigma \to \sigma \oplus \sigma & = \{\mathrm{in}_0\} \\
[\![\mathbf{new\ qbit}\ q := 0]\!] & : \sigma \to \mathbf{qbit} \otimes \sigma & = \{|0\rangle \otimes -\} \\
[\![\mathbf{discard}\ q]\!] & : \mathbf{qbit} \otimes \sigma \to \sigma & = \{\langle 0| \otimes \mathrm{id}, \langle 1| \otimes \mathrm{id}\} \\
[\![\mathbf{merge}]\!] & : \sigma \oplus \sigma \to \sigma & = \{\mathrm{in}_0^\dagger, \mathrm{in}_1^\dagger\} \\
[\![\mathbf{measure}\ q]\!] & : \sigma \to \sigma \oplus \sigma & = \{\mathrm{in}_0 \circ \Pi_0, \mathrm{in}_1 \circ \Pi_1\} \\
[\![q *= U]\!] & : \sigma \to \sigma & = \{U\} \\
[\![\mathbf{if}\ q\ \mathbf{then}\ P\ \mathbf{else}\ Q]\!] & : \mathbf{qbit} \otimes \sigma \to \mathbf{qbit} \otimes \tau & = [\![P]\!] \bullet [\![Q]\!]
\end{array}
$$

The semantics defined above cannot be lifted to a semantics of superoperators, because quantum alternation does not preserve extensional equality. Indeed, the Kraus decompositions $\{U_0\} \bullet \{V_0\}$ and $\{U_1\} \bullet \{V_1\}$ are extensionally equal if and only if there exists a phase θ such that $U_0 = e^{i\theta} U_1$ and $V_0 = e^{i\theta} V_1$, so $\{U_0\} \bullet \{V_0\} \simeq \{U_1\} \bullet \{V_1\}$ may not hold even if $\{U_0\} \simeq \{U_1\}$ and $\{V_0\} \simeq \{V_1\}$. The failure of quantum alternation to preserve extensional equality shows that there is no compositional superoperator semantics which satisfies the definition of alternation given in the introduction. However, as the examples above and previous work [1] [13] show, that particular definition of quantum alternation for closed quantum systems is the most intuitive and practical.

An important part of the superoperator semantics for QPL is the ability to define recursion. The category **Q** is CPO-enriched [11], a fact which together with the $\oplus$ operation makes **Q** a *traced monoidal category*. Since each Kraus decomposition determines a unique superoperator, we can define an order on the Hom-sets of **C** using the order on the Hom-sets

of **Q**, viz. $\mathcal{S} \sqsubseteq \mathcal{T}$ if the relation holds for the corresponding superoperators. We can then try to adapt the situation to quantum alternation. But we have the following proposition.

Proposition. *Quantum alternation is not monotone with respect to the $\sqsubseteq$ order.*

Proof. Let $\mathcal{H}$ be the Hilbert space associated to a signature σ. Let U and V be two unitary operators on $\mathcal{H}$ defining Kraus decompositions $\mathcal{S} = \{U\}$ and $\mathcal{T} = \{V\}$. Let ρ be a state on $\mathbf{qbit} \otimes \mathcal{H}$ defined by

$$\rho = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$$

where $B \neq 0$. Then $\mathcal{S} \sqsubseteq \mathcal{S}$ and $\varnothing \sqsubseteq \mathcal{T}$, but

$$(\mathcal{S} \bullet \mathcal{T} - \mathcal{S} \bullet \varnothing)(\rho) = \begin{bmatrix} 0 & UBV^\dagger \\ VCU^\dagger & VDV^\dagger \end{bmatrix}.$$

Recall that if a diagonal entry of a positive matrix is zero, then the corresponding row and column must be all zero. Since $UBV^\dagger \neq 0$, it follows that $(\mathcal{S} \bullet \mathcal{T} - \mathcal{S} \bullet \varnothing)(\rho)$ is not positive. Therefore, $\mathcal{S} \bullet \varnothing \not\sqsubseteq \mathcal{S} \bullet \mathcal{T}$, but $\mathcal{S} \sqsubseteq \mathcal{S}$ and $\varnothing \sqsubseteq \mathcal{T}$. ■

This counter-example shows that quantum alternation is not compatible with the semantics for recursion defined in [11]. Since a CP map T is a pure operation $\rho \mapsto E\rho E^\dagger$ if and only if all operations completely dominated by it are its nonnegative multiples [10], it appears that the reversibility condition (III) makes quantum alternation fundamentally incompatible with the standard order on CP maps.

Quantum operations admit several equivalent representations based on the structure theory of CP maps [10]. Each representation illustrates a different aspect of the quantum operation. The rest of this section defines quantum alternation in terms of Stinespring representations. This alternative perspective will clarify the relation between our definition of alternation and that of [1].

Let $T : S(\mathcal{H}) \to S(\mathcal{K})$ be a superoperator. By Stinespring's theorem, T can be written as $T(\rho) = V^\dagger(\rho \otimes I_{\mathcal{A}})V$, where $\mathcal{A}$ is a Hilbert space called the *ancilla* and $V : \mathcal{K} \to \mathcal{H} \otimes \mathcal{A}$ is a bounded operator. The ancilla models the environment of the operation T. The pair $(\mathcal{A}, V)$ is called a *Stinespring representation* of T. Stinespring's theorem can be interpreted as saying that any quantum operation T can be implemented as a pure operation on a larger Hilbert space. Given a Kraus decomposition $\mathcal{S}$ defining a superoperator $T : S(\mathcal{H}) \to S(\mathcal{K})$, a Stinespring representation of T can be obtained from $\mathcal{S}$ as follows. Let $\mathcal{A}$ be a Hilbert space with basis $\{|E\rangle\}_{E \in \mathcal{S}}$ and define $V : \mathcal{K} \to \mathcal{H} \otimes \mathcal{A}$ by

$$V\psi = \sum_{E \in \mathcal{S}} E^\dagger \psi \otimes |E\rangle.$$

Then $(\mathcal{A}, V)$ is a Stinespring representation of T. Conversely, a representation $(\mathcal{A}, V)$ of T with a fixed basis for $\mathcal{A}$ determines a Kraus decomposition of T.

If $\mathcal{S}$ and $\mathcal{T}$ are Kraus decompositions, then there is a natural Stinespring representation for the superoperator determined by $\mathcal{S} \bullet \mathcal{T}$, viz. the pair $(\mathcal{E}, W)$ defined by $\mathcal{E} = \mathcal{A}' \otimes \mathcal{A}$ and

$$W\psi = \sum_{E \in \mathcal{S}, F \in \mathcal{T}} \mathrm{Alt}_q(\hat{E}, \hat{F})^\dagger \psi \otimes |F\rangle \otimes |E\rangle,$$

where $\mathcal{A}$ and $\mathcal{A}'$ are the ancillas of the Stinespring representations determined by $\mathcal{S}$ and $\mathcal{T}$, respectively. Thus, the environment of the quantum alternation is the tensor product of the environments of the quantum operations involved.

4 Related Work

Altenkirch and Grattage [1] defined QML, a quantum programming language with quantum control based on a new type of judgement called "orthogonality." The denotational semantics for QML is based on expressing superoperators $T : S(\mathcal{A}) \to S(\mathcal{B})$ in the form $T(\rho) = \mathrm{Tr}_{\mathcal{G}} U(\rho \otimes |\xi\rangle\langle\xi|)U^\dagger$, where $\mathcal{H}$ and $\mathcal{G}$ are Hilbert spaces, $\xi \in \mathcal{H}$ is a fixed unit vector, and $U : \mathcal{A} \otimes \mathcal{H} \to \mathcal{B} \otimes \mathcal{G}$ is an isometry. Defining the bounded operator $V : \mathcal{B} \to \mathcal{A} \otimes \mathcal{G}$ by $V\psi = U(\psi \otimes \xi)$, we obtain an equivalent Stinespring representation $(\mathcal{G}, V)$ of T. In QML, a *strict* morphism corresponds to a superoperator with $\dim \mathcal{G} = 1$. Thus, strict morphisms correspond to singleton Kraus decompositions in our semantics, i.e. pure operations $\rho \mapsto E\rho E^\dagger$ with $E^\dagger E \leqslant I$. Only strict morphisms may be alternated in QML. The alternation is further restricted by the orthogonality judgement, which is implemented by an incomplete set of introduction rules.

The work of Mingsheng Ying et al. [13] is very recent and closely related to ours, though their attitude is quite different. They also note that the superoperator semantics is not compositional, but they are content with this. They do not define a Kraus semantics as we do. However, our construction is essentially embedded inside their definition of their superoperator semantics. Perhaps, the right way to look at it is that we have both defined a Kraus semantics but they have gone on to give a superoperator semantics as an abstract interpretation of the Kraus semantics. In such a case it often happens that the resulting semantics is not compositional. The fact that quantum alternation is not monotone using the Löwer order is not noted by them. Ying has a different approach to recursion based on second quantization [12] which seems to avoid the difficulties noted here but we do not understand it well enough to comment on it here. Certainly, combining recursion with quantum alternation will require some radically new idea.

5 Conclusion

Superficially this may strike the reader as a very negative, or perhaps schizophrenic, paper. Certainly, we feel that quantum alternation as often casually discussed, is quite problematic

and some fix based on type theory or syntactic control will not serve to make it meaningful. On the other hand we see this as the start of some new directions.

Quantum alternation is not really physically meaningful. Even if it is, it seems incompatible with recursion. Is there some crisp no-go theorem here? If so, what *is* meaningful? Ideally one should start from physical systems and develop a structural understanding from which linguistic entities should emerge. It seems to us that quantum alternation is a fantasy arising from programming language semantics rather than from physics. What we propose is that one should look closely at, say, quantum optics where devices like Mach-Zehnder interferometers [6] provide physical situations that are reasonably viewed as alternation. Note that in MZ interferometers the system being split is the system on which the two alternate operations are applied; there is not a distinct control qubit.

On a more mathematical note one can question the arbitrariness of the Kraus semantics; different Kraus semantics correspond to the same operator so doesn't that mean that the semantics is making unobservable distinctions? However, this is not the case. Different Kraus decompostions correspond to different choices of measurement that an experimenter may choose to make. In the standard paradigm, with classical control, the contexts provided by the language do not make these differences visible but in the enriched language they do.

One can still ask whether there is a canonical decomposition one can associate to a superoperator which can be used to define alternation. Indeed there is and it involves more sophisticated mathematics; we choose not to include it in this note. There is an operator-algebra analogue of the Radon-Nikodym theorem due to Belavkin [3] and, independently, Arverson [2]. Given two CP maps S and T with $S \sqsubseteq T$, it gives a representation of S in terms of a chosen minimal Stinespring representation of T and a positive operator $\mathsf{D}_T(S)$, the Radon-Nykodim derivative of S with respect to T. Now there is a map, the tracial map, which can be proven to dominate any CP map from $\mathcal{B}(\mathcal{H})$ to $\mathcal{B}(\mathcal{K})$. This gives a canonical decomposition of an arbitrary CP map; we have worked out a denotational semantics of the language with quantum alternation based on this approach. The trouble, and the reason we have not included it here, is that the physical significance of this semantics is unclear to us.

Acknowledgements

Panangaden would like to thank Mingsheng Ying for discussions allowing us to understand the relationship between our semantics for quantum alternation. He would also like to thank Vincent Danos who was present at the discussion and made several insightful remarks sprinkled with some interesting non sequiturs. We thank the referees for their comments. We have both been supported by NSERC. Bădescu has also been supported by a scholarship by FQRNT. Panangaden acknowledges the generous support of the Chinese Academy of Sciences, Institute of Mathematics, during his stay in Beijing.

References

[1] Thorsten Altenkirch & Jonathan Grattage (2005): *A functional quantum programming language.* In: *Proceedings of the 20th Annual IEEE Symposium on Logic in Computer Science, 2005.*, IEEE, pp. 249–258.

[2] W. Arveson (1969): *Subalgebras of c^*-algebras. Acta Math* 123, pp. 141–224.

[3] V. P. Belavkin & P. Staszewski (1986): *Radon-Nikodym theorem for completely positive maps. Reports on Mathematical Physics* 24(1), pp. 49–55.

[4] D. Deutsch (1985): *Quantum theory, the Church-Turing Principle and the universal quantum computer. Proc. Roy. Soc. Lond. A* 400, p. 97.

[5] D. Deutsch & R. Jozsa (1992): *Rapid solution of problems by quantum computation. Proc. Roy. Soc. Lond. A* 439, p. 553.

[6] J. C. Garrison & R. Y. Chiao (2008): *Quantum Optics.* Oxford University Press.

[7] A. Yu. Kitaev, A. H. Shen & M. N. Vyalyi. (2002): *Classical and quantum computation.* Graduate Studies in Mathematics, American Mathematical Society, Providence, RI.

[8] K. Kraus (1983): *States, Effects and Operations. Lecture Notes in Physics* 190, Springer-Verlag.

[9] M. Nielsen & I. Chuang (2000): *Quantum Computation and Quantum Information.* Cambridge University Press.

[10] Maxim Raginsky (2003): *Radon-Nikodym derivatives of quantum operations. Journal of Mathematical Physics* 44(11), pp. 5003–5020.

[11] Peter Selinger (2004): *Towards a Quantum Programming Language. Mathematical Structures in Computer Science* 14(4), pp. 527–586.

[12] Mingsheng Ying (2014): *Quantum Recursion and Second Quantisation.* Available on the arXiv 1405.4443.

[13] Mingsheng Ying, Nengkun Yu & Yuan Feng (2014): *Alternation on quantum programming: from superposition of data to superposition of programs.* Available in arXiv as 1402.5172.