Quantum Computation, Categorical Semantics and Linear Logic

André van Tonder

Department of Physics, Brown University Box 1843, Providence, RI 02906 andre@het.brown.edu

October 27, 2003

Abstract

We develop a type theory and provide a denotational semantics for a simple fragment of the quantum lambda calculus, a formal language for quantum computation based on linear logic. In our semantics, variables inhabit certain Hilbert bundles, and computations are interpreted as the appropriate inner product preserving maps between Hilbert bundles. These bundles and maps form a symmetric monoidal closed category, as expected for a calculus based on linear logic.

Keywords: Quantum Computing, Lambda Calculus, Linear Logic, Denotational Semantics

1 Introduction

The quantum lambda calculus developed by the author in [1] may be regarded both as a programming language and as a formal algebraic system for reasoning about quantum algorithms. It provides a model of quantum computation that combines the universality of the quantum Turing machine [2, 3] and the compositionality of the quantum circuit models [4]. The calculus turned out to be closely related to the linear lambda calculi used in the

study of linear logic [5, 6, 7, 8]. In [1], we set up a computational model, or operational semantics, and an equational proof system for this calculus, and argued that it was equivalent to the quantum Turing machine, and therefore universal for quantum computation.

In the present article, we report some progress in developing a type theory and a denotational semantics for a fragment of the quantum lambda calculus.

In our semantics, variables inhabit certain Hilbert bundles, and computations are interpreted as the appropriate inner product preserving maps between Hilbert bundles. These bundles and maps form a symmetric monoidal closed category, as expected for a calculus based on linear logic [9, 10].

For simplicity, in this paper we restrict our attention to a purely linear fragment of the full quantum calculus. This fragment is not universal for quantum computation, but is at least as expressive as the quantum circuit model. Future work should address the full calculus.

2 A quantum fragment

We consider only a simple linear, multiplicative fragment λ_q^{\otimes} of the quantum lambda calculus. First, we define a grammar for types in figure 1.

$$A, B ::=$$
 $types:$ Qbit $qubit$ $(A \multimap B)$ $function$ $(A \otimes B)$ $tensor\ product$

Figure 1: Types of the quantum calculus λ_q^{\otimes}

The grammar for raw terms, given in figure 2, is based on a syntax for linear logic developed by Wadler in [6]. We have added constants representing qubit states and unitary maps, and we have provided primitives for addition and scalar mutiplication of terms, to enable us to express quantum superpositions.

In the following, we will omit brackets wherever convenient, with the convention that products, sums and applications associate to the left. For example, $(f\ u\ v\ w) \equiv (((f\ u)\ v)\ w)$.

```
patterns:
p,q ::=
                         variable
         (p \otimes q)
                         product
t, u :=
                         terms:
         x \mid y \mid \cdots
                         variable
         (\lambda p : A. t)
                         function abstraction
         (t u)
                         function application
         (t \otimes u)
                         product
         0 \mid 1
                         aubit
                         constant representing U, for every U \in U(2^n), n = 1, 2, ...
         c_U
         (t+u)
                         superposition
         \alpha \cdot t
                         scalar product, for \alpha \in \mathbf{C}
```

Figure 2: Syntax of the quantum calculus λ_q^{\otimes}

We will implicitly identify terms that differ only in the renaming of bound variables (called α -equivalence). In addition, we identify terms according to the equivalence relation generated by the equations of figure 3.

Notice that modulo the equivalence relation of figure 3, the set of terms becomes a complex vector space. Indeed, the postulates of a vector space are satisfied by construction. In addition, \otimes is by construction a bilinear tensor product, while the application bracket is bilinear and lambda abstraction linear over the vector space operations.

In the following, the notation t will be abused to denote the equivalence class to which t belongs under α -renaming and the linear equivalences of figure 3. As a consequence, we can freely perform linear manipulations on terms, such as commuting sums, distributing tensor products or application brackets over sums, and so on.

Not all terms constructible from the above grammar will represent physically meaningful computations. In order to encode the latter, we shall restrict attention to so-called well-typed terms. Well-typed terms are intended to represent physically realizable operations, and the rules for constructing them, based on the syntax provided in [6], are given in figure 4. The Qbit-I rule is used to form normalized superpositions of sequences of qubits. The Cut

$$t_{1}+t_{2} \sim t_{2}+t_{1} \qquad (comm_{+})$$

$$(t_{1}+t_{2})+t_{3} \sim t_{1}+(t_{2}+t_{3}) \qquad (assoc_{+})$$

$$\alpha \cdot (t_{1}+t_{2}) \sim \alpha \cdot t_{1}+\alpha \cdot t_{2} \qquad (dist)$$

$$\alpha \cdot (\beta \cdot t) \sim \alpha \beta \cdot t \qquad (assoc)$$

$$t_{1}+0 \cdot t_{2} \sim t_{1} \qquad (zero)$$

$$1 \cdot t \sim t \qquad (one)$$

$$(\alpha \cdot t_{1}+\beta \cdot t_{2}) \otimes u \sim \alpha \cdot t_{1} \otimes u + \beta \cdot t_{2} \otimes u \qquad (lin_{\otimes}^{1})$$

$$u \otimes (\alpha \cdot t_{1}+\beta \cdot t_{2}) \sim \alpha \cdot u \otimes t_{1}+\beta \cdot u \otimes t_{2} \qquad (lin_{\otimes}^{2})$$

$$((\alpha \cdot t_{1}+\beta \cdot t_{2}) u) \sim \alpha \cdot (t_{1} u) + \beta \cdot (t_{2} u) \qquad (lin_{app}^{1})$$

$$(u (\alpha \cdot t_{1}+\beta \cdot t_{2})) \sim \alpha \cdot (u t_{1}) + \beta \cdot (u t_{2}) \qquad (lin_{app}^{2})$$

$$\lambda p. (\alpha \cdot t_{1}+\beta \cdot t_{2}) \sim \alpha \cdot (\lambda p. t_{1}) + \beta \cdot (\lambda p. t_{2}) \qquad (lin_{\lambda})$$

Figure 3: Linear equivalences on terms

rule then allows us to form superpositions of more complex terms containing interspersed qubits in a way that guarantees that all terms in a superposition at most differ in positions containing 0's and 1's. We will call such superpositions congruent [1]. These are the terms that can be obtained in computations where reductions proceed in parallel in all the branches of a quantum superposition, as required by quantum mechanics [1].

Notice that the set of well-typed terms is not any longer a vector space, since the typing rules of figure 4 do not allow unrestricted superpositions of well-typed terms. For example, the term

$$\frac{1}{\sqrt{2}}(\mathbf{0} + \mathbf{0} \otimes \mathbf{0})$$

is not derivable from the above rules and is therefore not well-typed. Neither is the non-normalized term $\mathbf{0} + \mathbf{1}$.

In the following, we will often suppress the explicit type annotations in our terms for brevity.

$$\frac{}{x:A \vdash x:A} \tag{Id}$$

$$\frac{\Gamma, \ p: A, \ q: B, \ \Delta \vdash t: C}{\Gamma, \ q: B, \ p: A, \ \Delta \vdash t: C}$$
 (Exch)

$$\frac{\Gamma \vdash \sum_{i} \alpha_{i} \cdot t_{1}^{i} \otimes \cdots \otimes t_{n}^{i} : A_{1} \otimes \cdots \otimes A_{n} \qquad x_{1} : A_{1}, \dots, x_{n} : A_{n}, \Delta \vdash u : B}{\Gamma, \Delta \vdash \sum_{i} \alpha_{i} \cdot u \left[t_{1}^{i} / x_{1}, \dots, t_{n}^{i} / x_{n}\right] : B} \qquad (Cut)$$

$$\frac{\Gamma, p : A \vdash t : B}{\Gamma \vdash (\lambda p : A . t) : A \multimap B} \tag{\multimap-}I$$

$$\frac{\Gamma \vdash t : A \qquad y : B, \ \Delta \vdash u : C}{\Gamma, \ f : A \multimap B, \ \Delta \vdash u[(f \ t)/y] : C} \tag{$-\circ$-E})$$

$$\frac{\Gamma \vdash t_1 : A \qquad \Delta \vdash t_2 : B}{\Gamma, \ \Delta \vdash t_1 \otimes t_2 : A \otimes B} \tag{\otimes-I$}$$

$$\frac{\Gamma, \ p: A, \ q: B \vdash t: C}{\Gamma, \ p \otimes q: A \otimes B \vdash t: C} \tag{\otimes-E})$$

$$\frac{\sum |\alpha_{b_1 \cdots b_n}|^2 = 1}{\vdash \alpha_{0 \cdots 0} \cdot \mathbf{0} \otimes \cdots \otimes \mathbf{0} + \cdots + \alpha_{1 \cdots 1} \cdot \mathbf{1} \otimes \cdots \otimes \mathbf{1} : \mathrm{Qbit}^n}$$
(Qbit-I)

$$\frac{U \in U(2^n)}{\vdash c_U : \mathrm{Qbit}^n \longrightarrow \mathrm{Qbit}^n}$$
 (c_U-I)

Figure 4: Well-typed terms of the the quantum calculus λ_q^{\otimes}

Next we postulate a set of equations that we intend to be true in our calculus. These are given in figure 5.

These rules are intended to represent an equality relation on well-typed terms. In other words, we impose the constraint that the total term on either side of each = sign be well-typed. Otherwise we would be able to prove certain well-typed terms equal to terms that are not well-typed. This

$$t = t$$
 (refl)

$$\frac{t_1 = t_2}{t_2 = t_1} \tag{sym}$$

$$\frac{t_1 = t_2 \quad t_2 = t_3}{t_1 = t_3} \tag{trans}$$

$$\frac{t_1 = t_2 \quad u_1 = u_2}{\alpha \cdot t_1 + \beta \cdot u_1 = \alpha \cdot t_2 + \beta \cdot u_2}$$
 (lin)

$$\frac{t_1 = t_2 \quad u_1 = u_2}{(t_1 \ u_1) = (t_2 \ u_2)} \tag{app}$$

$$(\lambda x_1 \otimes \cdots \otimes x_n \cdot u) \sum_i \alpha_i \cdot t_1 \otimes \cdots \otimes t_n = \sum_i \alpha_i \cdot u \left[t_1 / x_1, \dots, t_n / x_n \right] \quad (\beta)$$

$$\frac{x \text{ not free in } t}{\lambda x. (t \ x) = t} \tag{\eta}$$

$$\frac{U \in U(2^n) \quad b_1, \dots, b_n \in \{\mathbf{0}, \mathbf{1}\}}{c_U \ b_1 \otimes \dots \otimes b_n = \sum_{b'_1, \dots, b'_n \in \{0, 1\}} U_{b_1 \dots b_n}^{b'_1 \dots b'_n} \cdot b'_1 \otimes \dots \otimes b'_n}$$
(U)

Figure 5: Equational proof system for the quantum calculus λ_q^{\otimes}

constraint avoids equalities such as

$$\frac{1}{\sqrt{2}}((\lambda x : \text{Qbit. } x) \ \mathbf{0} + (\lambda x : \text{Qbit. } x) \ \mathbf{1}) = \frac{1}{\sqrt{2}}(\mathbf{0} + (\lambda x : \text{Qbit. } x) \ \mathbf{1}),$$

which is invalid because the left hand side is well-typed (as follows from using the Qbit-I and Cut rules) while the right hand side is not, because terms in the superposition are not congruent. On the other hand,

$$\frac{1}{\sqrt{2}}((\lambda x : \text{Qbit. } x) \ \mathbf{0} + (\lambda x : \text{Qbit. } x) \ \mathbf{1}) = \frac{1}{\sqrt{2}}(\mathbf{0} + \mathbf{1})$$

is a valid equality, since substitution has been performed in parallel in both branches of the superposition.

In order to be able to satisfy this constraint, note that the rule (lin) is formulated so as to allow parallel substitution in all branches of the super-

position. The result has to be a normalized sum of congruent terms. Also, note that rule (β) cannot be formulated recursively in terms of binary sums only. Indeed, as in the (Cut) rule of figure 4, substitution has to occur in all branches of a superposition in parallel.

Note that the rules (app) and (β) imply the following substitution rules under \otimes and lambda abstraction:

$$\frac{t_1 = t_2 \quad u_1 = u_2}{t_1 \otimes u_1 = t_2 \otimes u_2} \qquad \frac{t_1 = t_2}{\lambda p. \, t_1 = \lambda p. \, t_2}$$

Notice that the rule (β) implies, for example, the equation

$$(\lambda p \otimes q. t) u \otimes v = \lambda p. ((\lambda q. t) v) u$$

relating uncurried and curried versions of a function.

3 An example

Deutsch's algorithm [2, 11] can be very simply expressed as follows:

deutsch
$$\equiv$$

$$\lambda U_f : (\mathrm{Qbit} \otimes \mathrm{Qbit} \multimap \mathrm{Qbit} \otimes \mathrm{Qbit}).$$

$$\mathbf{let} \ x \otimes y : \mathrm{Qbit} \otimes \mathrm{Qbit} = U_f \ (H \ \mathbf{0}) \otimes (H \ \mathbf{1}) \ \mathbf{in}$$

$$(H \ x) \otimes y$$

where we have used the common longhand

(let
$$p: A = t$$
 in u) $\equiv (\lambda p: A. u) t$

and where $H: \text{Qbit} \longrightarrow \text{Qbit}$ is the Hadamard gate [11].

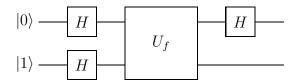


Figure 6: Deutsch's algorithm

Here the argument U_f is assumed to be a unitary map that takes $x \otimes y$ to $x \otimes (y + f(x) \mod 2)$, where f is some unknown function of one bit. For

example, if f is the identity function, then we should take U_f to be the *cnot* gate.

It is an easy exercise to check that **deutsch** is well-typed according to the rules of figure 4. The equational proof rules of figure 5 allow us to simplify the expression (**deutsch** *cnot*) just as we normally would in a pen and paper calculation, to obtain

deutsch
$$cnot = \mathbf{1} \otimes \frac{1}{\sqrt{2}} (\mathbf{0} + (-1) \cdot \mathbf{1}),$$

where the first qubit has value $1 = f(0) + f(1) \mod 2$, indicating that the function is balanced, as required.

4 Categorical semantics

We now address the main question of this paper, namely that of providing a denotational semantics for the fragment of the quantum lambda calculus considered here. An operational semantics for an untyped version of a full lambda calculus for universal quantum computation, including a description of how computations in the calculus can be physically realized on a quantum Turing machine, was provided in the earlier paper [1].

The fragment λ_q^{\otimes} considered here is not universal. For example, it cannot express recursion. However, it is already quite powerful. Indeed, any fixed-size acyclic quantum circuit can be straightforwardly encoded in λ_q^{\otimes} . In addition, the calculus can express higher-order functions, representing operations on quantum circuits, higher order operations on these operations, and so on.

The model for our interpretation will be a category. Types in the calculus will be interpreted as objects in the category, and judgments $\Gamma \vdash t : A$ will be interpreted as morphisms in the category [12, 13, 14, 15]. In the rest of this section we will give some motivation for the particular category that we will choose, which will then be constructed in more detail in the following sections.

To begin our search for the appropriate category, consider first the types Qbit^n , intended to represent qubit state vectors. To model these, it is clear that our category should have objects corresponding to 2^n -dimensional Hilbert spaces. As a first attempt, we may therefore try to model our calculus in the category of Hilbert spaces. States would correspond, up to a phase,

with unit vectors in a Hilbert space. The morphisms, corresponding to computations, would be unitary maps. Such a representation would indeed be familiar to any physicist.

However, this does not turn out to be the correct abstraction, since it does not correctly model function types such as Qbit \multimap Qbit and the higher-order operations between them. For example, it would be nice if we could model functions in Qbit \multimap Qbit as unit vectors in an appropriate Hilbert space. However, let us compare the operations that are available in Qbit with those that are available in Qbit \multimap Qbit. Up to normalization, Qbit is closed under addition. However, this is not true for elements of Qbit \multimap Qbit, if we interpret these as 2×2 unitary matrices. Indeed, a sum of two unitary matrices cannot in general be normalized to give another unitary matrix unless one is a multiple of the other. Instead of a Hilbert space, we therefore need a mathematical structure where the applicability of addition is restricted, in this case to collinear matrices.

In fact, the complex one-dimensional vector spaces consisting of scalar multiples of unitary matrices naturally fit together to give a vector bundle over base space U(2). We shall soon see that each fiber can be regarded as a one-dimensional Hilbert space. We are therefore led to interpret Qbit \multimap Qbit as a vector bundle whose fibers are Hilbert spaces, or a *Hilbert bundle*.

Note that a Hilbert space such as Qbitⁿ can be considered as a special case of a Hilbert bundle, where the base space is a single point.

For further evidence that the bundle abstraction is correct, consider the function

$$apply \equiv \lambda u \otimes x : (Qbit \multimap Qbit) \otimes Qbit. (u \ x)$$

with domain (Qbit \multimap Qbit) \otimes Qbit and range Qbit. As vector spaces, the dimension of the domain would be larger than that of the range, meaning that no inner-product-preserving map would exist to give a semantics to apply. However, interpreting Qbit \multimap Qbit as a Hilbert bundle with fiber dimension 1, with any reasonable definition of the tensor product the total fiber dimension of the domain would be 2, just like that of the range, and a fiberwise unitary map will exist that can be used to give a meaning to apply.

As another example, consider the function space $\mathrm{Qbit} \longrightarrow \mathrm{Qbit} \otimes \mathrm{Qbit}$, which contains, for example, the functions $x \mapsto x \otimes 0$ and $x \mapsto x \otimes 1$. These are both isometries (they conserve the Hilbert space inner product), but these maps can can be added and normalized to give another isometry. Once again restricting ourselves to subspaces of maps that can safely be added, we obtain

a Hilbert bundle whose fibers are two-dimensional Hilbert spaces.

5 Hilbert bundles

We have argued that just as qubits may be regarded as inhabiting Hilbert spaces, higher order types will inhabit appropriate Hilbert bundles. Operations on qubits, represented as unitary transformations between Hilbert spaces, will be generalized to a certain class of bundle morphisms. In this section we make these intuitions more precise.

Definition 5.1. A *Hilbert bundle* is a vector bundle in which the fibers are isomorphic to a fixed Hilbert space \mathcal{H} .

More precisely, a Hilbert bundle consists of a total space E, a base space B, a typical fiber \mathcal{H} , and a projection $\pi: E \to B$ such that for each point $b \in B$, there exists an open neighborhood U and a homeomorphism $h: \pi^{-1}(U) \to U \times \mathcal{H}$. Moreover, if h is another such homeomorphism whose domain intersects with that of h, then $h' \cdot h^{-1}$ is a fiberwise Hilbert space isomorphism.

Definition 5.2. The *rank* of a Hilbert bundle is the dimension of its fibers.

We will mainly deal with certain submanifolds of complex *Grassmann bundles*, which are built out of subspaces of a complex vector space as follows [16]:

Definition 5.3. Given a complex vector space V, we define the *Grassmann bundle* $\Gamma(k, V)$ to be the vector bundle with base the *Grassmann manifold*

$$B \equiv G(k, V) \equiv \{W \mid W \text{ a } k\text{-dimensional subspace of } V\},\$$

with total space

$$E \equiv \{(W, x) \mid x \in W\},\$$

and projection $\pi: E \to B: (W, x) = W$.

Notice that a Grassmann bundle comes with a canonical *Gauss map* [16] to the original vector space in which the fibers are embedded, given by

$$q: E \rightarrow V: (W, x) = x.$$

Abstracting this situation, we define:

Definition 5.4. An embedded Hilbert bundle with carrier space V, where V is a complex vector space, is a Hilbert bundle $\pi: E \to B$ together with a Gauss map $g: E \to V$ taking fibers to subspaces. Restricted to each fiber, we require that g be linear and one-to-one. We also require that if v and w are both in the intersection of the images of two fibers, the inner product of their preimages in the two fibers coincide.

Note that any Hilbert space \mathcal{H} may trivially be considered an embedded Hilbert bundle with a single fiber and carrier space \mathcal{H} .

Homomorphisms between Hilbert bundles are defined so as to preserve all the available structure:

Definition 5.5. Let $A \equiv \pi_1 : E_1 \to B_1$ and $B \equiv \pi_2 : E_2 \to B_2$ be Hilbert bundles. A function $f : E_1 \to E_2$ is a *Hilbert bundle homomorphism* if f takes fibers to fibers, is fiberwise linear, and preserves the Hilbert space inner product fiberwise.

If \mathcal{A} and \mathcal{B} are embedded Hilbert bundles with carrier spaces V_1 and V_2 and Gauss maps $g_1: E_1 \to V_1$ and $g_2: E_2 \to V_2$, then f is an embedded Hilbert bundle homomorphism if, in addition, there exists a unique linear function $\bar{f}: V_1 \to V_2$ such that $g_2 \cdot f = \bar{f} \cdot g_1$.

We are now ready to define $\mathcal{A} \longrightarrow \mathcal{B}$.

Definition 5.6. Let $\mathcal{A} \equiv \pi_1 : E_1 \to B_1$ and $\mathcal{B} \equiv \pi_2 : E_2 \to B_2$ be Hilbert bundles. Let

$$S \equiv \{c \cdot f \mid c \in \mathbf{C}, f : \mathcal{E}_1 \to \mathcal{E}_2 \text{ a Hilbert bundle homomorphism}\},$$

where scalar multiplication on bundle maps is defined via $(c \cdot f)(x) \equiv c f(x)$, and addition as $(f_1 + f_2)(x) \equiv f_1(x) + f_2(x)$ as long as $\pi_2(f_1(x)) = \pi_2(f_2(x))$ for all x. The set S is not in general closed under \cdot and +, but we may consider subsets W of S which are closed under these operations and therefore form vector spaces. We call such a vector space W maximal is it is not a proper subspace of another such W'. Under the condition that all maximal spaces $W \subseteq S$ have the same dimension, we may then define the Hilbert bundle $A \multimap B$ as follows:

Let the total space be

$$E \equiv \{(W, f) \mid W \subseteq S \text{ is maximal, } f \in W\},\$$

with base space projection

$$\pi: E \to B: (W, f) \to W,$$

and define a Hilbert space inner product on each fiber as that obtained from the norm

$$||cf|| \equiv |c|$$
,

for f a Hilbert bundle homomorphism, via the polarization formula

$$\langle x | y \rangle = \frac{1}{4} (\|x + y\|^2 + i \|x + iy\|^2 - \|x - y\|^2 - i \|x - iy\|^2).$$

If \mathcal{A} and \mathcal{B} are embedded Hilbert bundles with carrier spaces V_1 and V_2 and corresponding Gauss maps $g_1: E_1 \to V_1$ and $g_2: E_2 \to V_2$, we define the carrier space of $\mathcal{A} \multimap \mathcal{B}$ to be $V \equiv \operatorname{Lin}(V_1, V_2)$, and the Gauss map to be $g: E \to V: (W, f) \to \bar{f}$.

Note that, by construction, the bundle homomorphisms from \mathcal{A} to \mathcal{B} are exactly the unit vectors in the fibers of the Hilbert bundle $\mathcal{A} \multimap \mathcal{B}$. This generalizes the interpretation of quantum states as normalized vectors in a Hilbert space and is fundamental to our description of both primitive qubit types and function types in a unified framework.

Also note that homomorphisms f, which have to preserve the Hilbert space inner product on the fibers, only exist when the dimension of the fibers of \mathcal{A} is at most that of the fibers of \mathcal{B} . In other words, $\mathcal{A} \multimap \mathcal{B}$ is nonempty only when $\operatorname{rank}(\mathcal{A}) \leq \operatorname{rank}(\mathcal{B})$.

In fact, we have the following nice result relating the rank of $\mathcal{A} \multimap B$ to those of \mathcal{A} and \mathcal{B} .

Theorem 5.7. Let m be the rank of \mathcal{A} and n the rank of \mathcal{B} . Then $\mathcal{A} \multimap \mathcal{B}$ has rank at most $\lfloor n/m \rfloor$, the largest integer less than or equal to n/m.

Proof. As noted, $\mathcal{A} \multimap \mathcal{B}$ can be nonempty only if $n \geq m$. Assume this and let (W, f) be a normalized element of $\mathcal{A} \multimap \mathcal{B}$. With respect to local trivializations of \mathcal{A} and \mathcal{B} , we can represent f by a field of $n \times m$ matrices F(x) on each neighborhood of the base manifold of \mathcal{A} , satisfying $F(x)^{\dagger}F(x) = \mathbf{1}_{m \times m}$ independent of x. If n = km for some integer k, then by choosing a suitable orthonormal basis in each fiber, we can bring F(x) to block form

$$F(x) = \begin{pmatrix} U_1(x) \\ 0 \\ \vdots \end{pmatrix}$$
, with k blocks of $m \times m$ matrices. Let (W, g) , with

matrix representation G(x), be in the same fiber and orthonormal to (W, f) with respect to the inner product in $\mathcal{A} \multimap \mathcal{B}$. By definition of $\mathcal{A} \multimap \mathcal{B}$, the matrix representation of $\alpha \cdot f + \beta \cdot g$ has to satisfy

$$(\alpha F(x) + \beta G(x))^{\dagger} (\alpha F(x) + \beta G(x)) = \mathbf{1}_{m \times m}$$

for all $|\alpha|^2 + |\beta|^2 = 1$, independent of x. In other words, the columns of $\alpha F(x) + \beta G(x)$ must remain normalized and orthogonal as we vary α and β . It is not difficult to show that this means that G(x) can have no nonzero entries in the block occupied by $U_1(x)$ above. We can therefore further rotate

the basis in each fiber to make G(x) of the form $G(x) = \begin{pmatrix} 0 \\ U_2(x) \\ 0 \\ \vdots \end{pmatrix}$ while keeping F(x) invariant. Since the

keeping F(x) invariant. Since there are k vertical blocks, this can be repeated at most k times, so that the fiber to which (W, f) belongs can have dimension at most k = m/n.

In the case where m/n is not an integer, there is at most room for $\lfloor m/n \rfloor$ square vertical blocks, and the result follows.

All the Hilbert bundles that we will need will have ranks that are powers of two, so that $\lfloor n/m \rfloor = n/m$ when n > m, and $\lfloor n/m \rfloor = 0$ when n < m, in which case $\mathcal{A} \longrightarrow \mathcal{B}$ is the empty bundle.

Given two bundles, we can construct their *exterior tensor product* as follows:

Definition 5.8. Let $\mathcal{A} \equiv \pi_1 : E_1 \to B_1$ and $\mathcal{B} \equiv \pi_2 : E_2 \to B_2$ be Hilbert bundles. We define the *exterior tensor product bundle* $\mathcal{A} \boxtimes \mathcal{B}$ as the Hilbert bundle with total space

$$E \equiv \bigcup_{(x,y)\in B_1\times B_2} E_{(x,y)},$$

where the fiber $E_{(x,y)} \equiv E_x \otimes E_y$ is the Hilbert space tensor product of the fibers

$$E_x \equiv \pi_1^{-1}(x), \qquad E_y \equiv \pi_2^{-1}(y).$$

The base space is $B \equiv B_1 \times B_2$, with projection

$$\pi: E \to B: E_{(x,y)} \to (x,y).$$

If \mathcal{A} and \mathcal{B} are embedded Hilbert bundles with carrier spaces V_1 and V_2 and corresponding Gauss maps $g_1: E_1 \to V_1$ and $g_2: E_2 \to V_2$, the carrier space of $\mathcal{A} \boxtimes \mathcal{B}$ is the vector space tensor product $V \equiv V_1 \otimes V_2$, and the Gauss map is $g \equiv g_1 \otimes g_2$.

In other words, in forming the exterior tensor product, we are taking tensor products of every fiber of \mathcal{A} with every fiber of \mathcal{B} . Note that this construction is different from the usual tensor product of vector bundles over a fixed base manifold, since here we are also multiplying the base spaces, which can be distinct. When \mathcal{A} and \mathcal{B} are ordinary Hilbert spaces considered as Hilbert bundles, $\mathcal{A} \boxtimes \mathcal{B}$ reduces to the usual Hilbert space tensor product.

Theorem 5.9. Let m be the rank of A and n the rank of B. Then $A \boxtimes B$ has rank mn.

Definition 5.10. Let Unit denote the complex numbers \mathbf{C} considered as an embedded Hilbert bundle with carrier space \mathbf{C} . As its name indicates, *unit* acts as a unit for the exterior tensor product in the sense that $\mathcal{A} \boxtimes \mathsf{Unit} \simeq \mathsf{Unit} \boxtimes \mathcal{A} \simeq \mathcal{A}$.

We wrap up the section with a few constructions on bundle maps that will be very unseful in our semantics:

Definition 5.11. Let $f: \mathcal{A} \to \mathcal{B}$ and $g: \mathcal{C} \to \mathcal{D}$ be Hilbert bundle homomorphisms. Then we can define a Hilbert bundle homomorphism $f \boxtimes g: \mathcal{A} \boxtimes \mathcal{C} \to \mathcal{B} \boxtimes \mathcal{D}$ fiberwise by

$$(f \boxtimes g)|_{E_x \otimes E_y} = f|_{E_x} \otimes g|_{E_y}$$

where E_x and E_y denote the fibers of \mathcal{A} and \mathcal{C} above x and y respectively. \square

Definition 5.12. Given a Hilbert bundle homomorphism $f: \mathcal{C} \boxtimes \mathcal{A} \to \mathcal{B}$, let $x_i \otimes y_j$ be a fiberwise orthonormal basis of $C \boxtimes \mathcal{A}$ and write

$$f(\sum_{ij} \alpha_{ij} x_i \otimes y_j) = f(x_i \otimes \sum_{ij} \alpha_{ij} y_j) \equiv f_i(\sum_{ij} \alpha_{ij} y_j).$$

Given that the x_i are orthonormal, each f_i is a Hilbert bundle homomorphism $f_i: \mathcal{A} \to \mathcal{B}$. Let \tilde{f}_i be the corresponding element (W, f_i) of $\mathcal{A} \multimap \mathcal{B}$. Then the linear map $g: \mathcal{C} \to (\mathcal{A} \multimap \mathcal{B})$ defined by $g(x_i) = \tilde{f}_i$ is a Hilbert bundle homomorphism. We write

$$\operatorname{curry}(f) \equiv g.$$

The currying operation has an adjoint, defined as follows:

Definition 5.13. We define a Hilbert bundle homomorphism

$$\mathsf{apply} : (\mathcal{A} \multimap \mathcal{B}) \boxtimes \mathcal{A} \to \mathcal{B}$$

by

$$\operatorname{apply} \sum_{ij} \alpha_{ij} \left(W, f_i \right) \otimes x_j \equiv \sum_{ij} \alpha_{ij} \, f_i(x_j).$$

6 Bundle categories

Hilbert bundles form a category with particularly nice properties, making it very suitable for providing a semantics of our linear quantum lambda calculus [9, 10].

Theorem 6.1. The category consisting of (embedded) Hilbert bundles and (embedded) Hilbert bundle homomorphisms is a symmetric monoidal closed category.

Proof. The monoidal operation is provided by the exterior tensor product \boxtimes , which is symmetric and associative up to natural isomorphism, and has unit Unit. The existence of the connective \multimap and the functors curry and apply makes the category monoidal closed.

We now define the category that will form the arena for our semantics.

Definition 6.2. The category \mathbf{Q}^{\otimes} is a subcategory of the category of embedded Hilbert bundles. Like the latter, it has as objects embedded Hilbert bundles, and as arrows embedded Hilbert bundle homomorphisms. The objects of \mathbf{Q}^{\otimes} is the smallest set such that

- Unit $\in \mathbf{Q}^{\otimes}$.
- $Qbit \in Q^{\otimes}$, where Qbit is a fixed two-dimensional Hilbert space considered as an embedded Hilbert bundle with carrier space Qbit.
- $\mathcal{A} \boxtimes \mathcal{B} \in \mathbf{Q}^{\otimes}$ for every $\mathcal{A}, \mathcal{B} \in \mathbf{Q}^{\otimes}$.
- $\mathcal{A} \multimap \mathcal{B} \in \mathbf{Q}^{\otimes}$ for every $\mathcal{A}, \mathcal{B} \in \mathbf{Q}^{\otimes}$.

7 Semantics

We are now ready to define the semantics. The interpretation of types will be a function $[\cdot]$ from type denotations in our grammar to bundles in the category \mathbf{Q}^{\otimes} , defined inductively as follows:

Assumptions $\Gamma \equiv x_1 : A_1, \ldots, x_n : A_n$ will also be interpreted as bundles in the category as follows:

$$\llbracket \emptyset
rbracket = \mathsf{Unit}$$
 $\llbracket \Gamma
rbracket = \llbracket A_1
rbracket oxtimes \cdots oxtimes \llbracket A_n
rbracket$

The interpretation $[\Gamma \vdash t : A]$ of a judgment is a bundle morphism $[\Gamma] \xrightarrow{f} [A]$, inductively defined according to the rules in figure 7. These rules are in one-to-one correspondence to the rules for well-typed terms and follow the exposition of Wadler in [6]. To save notation, we write A for the bundle [A] denoted by A. We also use the notation \simeq to indicate natural isomorphisms which we do not explicitly name in the figure.

In the interpretation, the (Cut) rule corresponds to composition of morphisms in the category. The fact that the category is monoidal gives a meaning to products in the $(\otimes I)$ and $(\otimes E)$ rules. The symmetric monoidal structure of the category plays a role in various of the rules, providing the necessary natural isomorphisms needed in, for example, the interpretation of (Exch) and $(\multimap E)$. The fact that the category is monoidal closed provides the curry and apply operations needed for interpreting $(\multimap I)$ and $(\multimap E)$.

It is instructive to revisit at this point just how the necessity for a Hilbert bundle category arose. Note that in order to give a meaning to the rules $(Qbit^n-I)$ and (c_U-I) , we needed Hilbert spaces and objects containing unitary maps between any pair of Hilbert spaces, all of which should naturally fit into a symmetric monoidal category, as required by the structure of the calculus. These requirements are satisfied by the category of Hilbert bundles. Due to the linearity of our calculus, the Hilbert bundles that we obtain via our inductive construction are in fact embedded bundles with a linear carrier space.

Figure 7: Semantics of the quantum calculus λ_q^{\otimes}

It is straightforward to check that the semantics is sound with respect to the equational rules of figure 5.

8 Related work

We list some previous work on the semantics of quantum computation:

Abramsky and Coecke [17] described a realization of a categorical model of multiplicative linear logic via the quantum processes of entangling and deentangling by means of typed projectors. They discussed how these processes can be represented as terms of an affine lambda calculus.

Attempts to develop an interpretation of quantum computing using Chu space models of linear logic are described in [18] and [19].

The imperative language qGCL, developed by Sanders and Zuliani [20], is based on Dijkstra's guarded command language. It has a formal semantics and proof system.

An alternative approach to providing an operational semantics of quantum computation based on process algebras was developed in [21].

It would be interesting to relate our efforts to the work of Selinger [22], who constructed a semantics for quantum computation based on superoperators on density matrices, and of Girard [23], who developed a linear quantum logic based on similar technology. Closely related to Selinger's approach is the work done by Edalat [24] on the use of partial density operators to model quantum computations with a probability of nontermination. Coecke and Martin [25] introduced a domain structure for quantum information theory based on a partial order on density matrices, and in related work Kashefi [26] developed a denotational semantics for quantum computation based on a domain theory of completely positive maps over density matrices.

In contrast to the work based on density matrices, our calculus is based on the state vector formalism of quantum mechanics and does not express measurements as primitive operations within the calculus itself. However, we do not consider this a fundamental weakness, since it is known that measurements can always be deferred until the end of the computation without affecting universality [11]. Indeed, the untyped quantum lambda calculus of [1] is universal and equivalent to the quantum Turing machine (which does not have measurement primitives either). As such, our approach provides a somewhat more minimal model of quantum computation.

9 Conclusion

In this paper, we developed a type theory and provided a denotational semantics for a simplified version of the quantum lambda calculus first developed in [1]. In our interpretation, variables inhabit certain Hilbert bundles, and computations are interpreted as appropriate inner product preserving maps between Hilbert bundles.

There are various possible topics for future research: The interpretation considered here inductively assigns a morphism in the category \mathbf{Q}^{\otimes} to every judgment we can derive in our calculus. An interpretation for which the converse is true is called *full*. We conjecture that the Hilbert bundle interpretation is indeed full, although as of this writing we have not found a general proof of this conjecture.

As opposed to the full untyped quantum calculus developed in our previous work [1], the fragment considered here is not universal for quantum computation, although it is at least as expressive as the quantum circuit model. In order to achieve universality, we would need to model typed versions of the nonlinear variables appearing in the untyped calculus (also called exponentials), include additive connectives, and add some capability for recursion. This can most likely be done by relaxing the restriction that our Hilbert bundles be embedded in linear carrier spaces.

Linear logic, which explicitly models structural operations, is very suitable for complexity analysis, a topic of great interest in Quantum Computation. It is our hope that models of quantum computation based on linear logic may contribute to our understanding of complexity in Quantum Computation.

Acknowledgments

I would like to thank Prof. Antal Jevicki and the Brown University Physics department for their support.

References

[1] A. van Tonder, A lambda calculus for quantum computation, arXiv:quant-ph/0307150, (July 2003).

- [2] P. Benioff, The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines, J. Stat. Phys. 22 (5) (1980), 563-591.
- [3] D. Deutsch, Quantum theory, the Church-Turing principle and the universal quantum computer, Proceedings of the Royal Society of London A 400 (1985), 97-117.
- [4] D. Deutsch, Quantum Computational Networks, Proceedings of the Royal Society of London A 439 (1989), 553-558.
- [5] S. Abramsky, Computational interpretations of linear logic, Theoretical Computer Science 111 (1-2) (1993) 3-57.
- [6] P. Wadler, A syntax for linear logic, in Mathematical Foundations of Programming Semantics: 9th International Conference, New Orleans, LA, Proceedings 802 Springer-Verlag, New York (1993) 513-529.
- [7] J.-Y. Girard, Linear logic, Theoretical Computer Science 50 (1987) 1-102.
- [8] P. Wadler, A taste of linear logic, in Proceedings of the 18th International Symposium on Mathematical Foundations of Computer Science, Gdánsk, Springer-Verlag, New York (1993).
- [9] R.A.G. Seely, Linear logic, *-autonomous categories, and cofree coalgebras, in Categories in Computer Science and Logic, June 1989, AMS Contemporary Mathematics 92.
- [10] G.M. Bierman, What is a categorical model of intuitionistic linear logic?, in Proceedings of the conference on typed lambda calculus and applications, Editor M. Dezani, Springer-Verlag LNCS 902 (1995).
- [11] M.A. Nielsen and I.L. Chuang, Quantum Computation and Quantum Information, Cambridge University Press 2000.
- [12] J.C. Mitchell, Foundations of programming languages, MIT press 1996.
- [13] B.C. Pierce, Types and programming languages, MIT press 2002.
- [14] R.E. Davis, Truth, deduction and computation: logic and semantics for computer science, Computer Science Press, New York 1989.

- [15] C.A. Gunter, Semantics of programming languages: structures and techniques, MIT press 1992.
- [16] D. Husemoller, Fibre bundles, Graduate texts in mathematics, Springer-Verlag, New York (1966).
- [17] S. Abramsky and B. Coecke, Physical traces: Quantum vs. classical information processing, Electronic Notes in Theoretical Computer Science 69 (2003), arXiv:cs.CG/0207057 (2002).
- [18] V. Pratt, Linear logic for generalized quantum mechanics, in Proc. of Workshop on Physics and Computation (PhysComp'92), IEEE, Dallas (1992), 166-180.
- [19] M. Wehr, Quantum computing: A new paradigm and its type theory, Talk held at Quantum Computing Seminar, Lehrstuhl Prof. Beth, Universität Karlsruhe (1996).
- [20] J.W. Sanders and P. Zuliani, Quantum programming, in Mathematics of Program Construction, Springer LNCS, 1837:80-99, (2000).
- [21] P. Jorrand and M. Lalire, Towards a quantum process algebra, arXiv:quant-ph/0312067
- [22] P. Selinger, Towards a quantum programming language, to appear in Mathematical Structures in Computer Science (2003) 45 pages.
- [23] J.-Y. Girard, Between logic and quantic: A tract, (November 2003).
- [24] A. Edalat, An extension of Gleason's theorem for quantum computation, arXiv:quant-ph/0311070 (November 2003).
- [25] B. Coecke and K. Martin, A partial order on classical and quantum states, Programming research group research report RR0207 (2002).
- [26] E. Kashefi, Quantum domain theory definitions and applications, arXiv:quant-ph/0306077 (2003).