Towards Quantum Types

by

Éric Oliver Paquette
Université de Montréal

-Calgary-

CS-QIC 2005

Thesis: I believe that with a proper semantical (categorical) and topological / geometrical approach of quantum computation, one can characterize complexity classes in a brand new way and hopefully solve some open problems in complexity theory.

Towards quantum types

1. Classical and quantum types,

2. Classical and quantum information flow,

3. Quantum types as topological carriers.

1. Classical and quantum types

Type System

A *type theory* is a formal system such that each data considered within the limit of the theory is properly equipped with a type which characterizes the data.

During this talk, I will considered the set from which the data is taken to be the *type* while the elements of that set (type) will be called *terms*. To this, we add a countable number of variables $\{x_i^A\}$ for each type A.

For instance, just think of booleans, natural numbers, strings of symbols etc. These are all type with their own internal structure.

Here, I give a table with basic types (i.e. which are automatically contained in any type theory) together with their term forming operations:

1	N	PA	$A \times B$	Ω
$*$	0	$\{x \in A \mid \phi(x)\}$	$\langle a, b \rangle$	$\top$, $\bot$
	Sn			$a \wedge b$
				$a \vee b$
				$a \Rightarrow b$
				$\exists_x \phi(x)$
				$\forall_x \phi(x)$
				$a \in \alpha$

The question one may asks is why the types presented before are important (or fundamental)? Well,

- The singleton type is there for structural purpose, i.e. each term $a \in A$ can be seen as a unique function $a : 1 \to A$ which implies that type theory can be seen as a theory which is entirely functional.

- The natural number object (which we ask to be a model of Peano arithmetic) provides us the possibility of doing recursion.

- The product type gives us the possibility to consider functions with more than one input.

- The subobject classifier, in addition to provide to the type system a logic is also used to define subtyping. Indeed, suppose I want to define the subtype of *even numbers*; I can define a function $T : N \rightarrow \Omega$ which inputs a natural number n and tells me if this number is even or odd. The set of all even numbers is then a *subtype* of N as it is included in N. Note that this defines: $PN = \Omega^N$, where Ω^N is defined as the set of functions of type $N \rightarrow \Omega$. In general, for a type A, $PA = \Omega^A$.

Remark: We can describe a version of type theory with exponential types B^A as a type λ-calculus with equality and entailment relation.

Note that we can also add new types. For instance, consider that we add to our type theory the type $\mathcal{G}$ with the terms $a, b, c, \ldots$ (i.e. a formal alphabet), we can then express the notion of a graph within our theory where a vertex of a given graph is a term of $\mathcal{G}$ and an edge is a pair of type $\mathcal{G} \times \mathcal{G}$.

If one do quantum computing and wants to think at a higher level, the natural question to ask is whether or not it is possible to develop a *quantum* type theory.

The way the question is asked up there is quite important, the answer is:

$$|\mathrm{Answer}\rangle = \frac{1}{\sqrt{2}}(|\mathrm{Yes}\rangle + |\mathrm{No}\rangle)$$

The 'no' part is that one cannot expect a *pure* quantum type theory, that is, a theory that contains only quantum types.

Why? The reasons are well known:

- First, in a type theory, we need to have a cartesian product which means that product types comes equipped with canonical projection but, in the quantum world, we have entanglement thus, we cannot always project.

- Second, in a classical type theory, we can copy the information (with the diagonal map). As we cannot assume that quantum information is always perfectly known, we cannot make this assumption in a would-be quantum type theory (no cloning).

- Finally, we can not define a quantum analogue of the subobject classifier (Ω).

We then might conclude that classical and quantum information dont really like each other, but it does not matter as we cannot really think of a quantum computer without classical control.

The idea will therefore to augment our classical type with quantum types and this will define our quantum type theory. To stay as general as possible, I choose to work with the model of density matrices and superoperators.

1	N	PA	$A \times B$	Ω	n_s	$P \otimes Q$
$*$	0	$\{x \in A \mid \phi(x)\}$	$\langle a, b\rangle$	$\top$, $\bot$	ρ	$p \otimes q$
	Sn			$a \wedge b$		
				$a \vee b$		
				$a \Rightarrow b$		
				$\exists_x \phi(x)$		
				$\forall_x \phi(x)$		
				$a \in \alpha$		

Where $\rho \in n_s$ is a $n \times n$ density matrix. Thus, for all $n \neq 0$ there is a type n_s; hence n_s is *not* a sort of quantum natural number object.

Note that this structure is surprisingly minimal, it appears that we do not need the trace, the dagger and the dualisation operations. Actually, they are still present but imbedded in the superoperator formalism (i.e. function from a type to another) and therefore not needed at the level of types. Also, we drop the direct sum $\oplus$ present in the Hilbert space formalism as a term forming operation since it is trace increasing and thence, cannot be defined as a superoperator.

However, even if it is minimal, the setup is complete. It remains to define how the information flow behaves between the classical control and the quantum processor.

2. Information flow

From now on, we assume that classical data types are of finite cardinality, that is, there is only a finite number of terms for each type.

What is a complete computational process in the context of quantum computation with classical control?

Typically, the classical control initializes the quantum data. Then, we apply a circuit, we measure and finally, we send back the result to the classical control. In details,

1. The classical control (CC) initializes quantum data through an function $Q : CC \to n_c$. Thus, suppose we have a finite data type A whose only terms are a_1 and a_2 and that we send either term to the quantum processor, we get:

$$a_1 \mapsto \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \quad \text{or} \quad a_2 \mapsto \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$$

In addition, suppose that we have a product type $B = C \times D$ then for a B-term $b = \langle c, d \rangle$, we get $Q(b) = Q(\langle c, d \rangle) = Q(c) \otimes Q(d)$. Which tells us that the product structure in the classical type theory is transferred to a tensor product structure in the quantum world. This new type n_c is characterised by those $n \times n$ matrices who are 0 everywhere except at one position on the diagonal which coincide with the term initialised.

2. After the initialization, we apply a circuit, typically a unitary operation $U : n_c \to n_u$ or a superoperator $S : n_c \to n_s$ where n_u is the type of all pure state $n \times n$ density matrices and n_s is the type of all mixed state $n \times n$ density matrices.

3. Finally, we reduce the wave packet via a function $R_1 : n_s \to n_p$ (here, the domain of R_1 can also be n_u). The reduction of the wavepacket is given by a diagonalisation of the matrix e.g.:

$$\begin{pmatrix} a & b \\ \overline{b} & c \end{pmatrix} \xrightarrow{R_1} \begin{pmatrix} a & 0 \\ 0 & c \end{pmatrix}$$

One can then see that the type n_p is then the set of $n \times n$ density matrix with 0 everywhere except on the diagonal.

3. (contd.) After reduction, continuing with our example, we send back data a_1 to the classical classical control with probability a and a_2 with probability c. This part of the process is labeled by the operation R_2.

We summarize:

$$
\begin{tikzcd}
 & & n_s \arrow[dr, "R_1"] & & \\
CC \arrow[r, "Q"] & n_c \arrow[ur, "S"] \arrow[dr, "U"'] & & n_p \arrow[r, "R_2"] & CC \\
 & & n_u \arrow[ur, "R_1"'] & &
\end{tikzcd}
$$

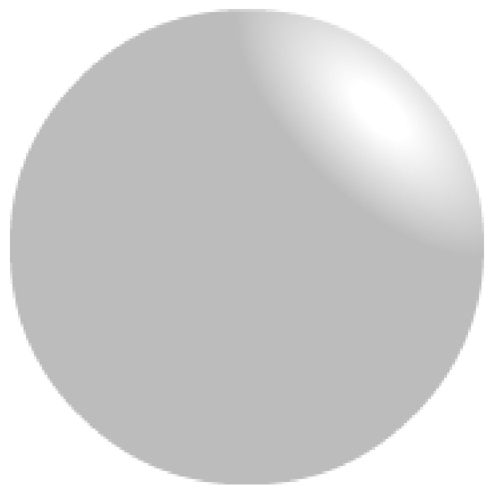
It is important to note here that the classical types are still present as an indexing system for the matrix entries! The first observation we could make from the view point of type theory is that quantum types are just a superposition of classical types but it goes further... We assert the following:

Conjecture: The Quantum part of our type theory is a linearisation and a quantisation of classical type theory in the sense that most (categorical) structures present in classical type theory are preserved up to a superposition provided that we satisfy the finiteness condition specified earlier (i.e. that we consider classical data type of finite cardinality).

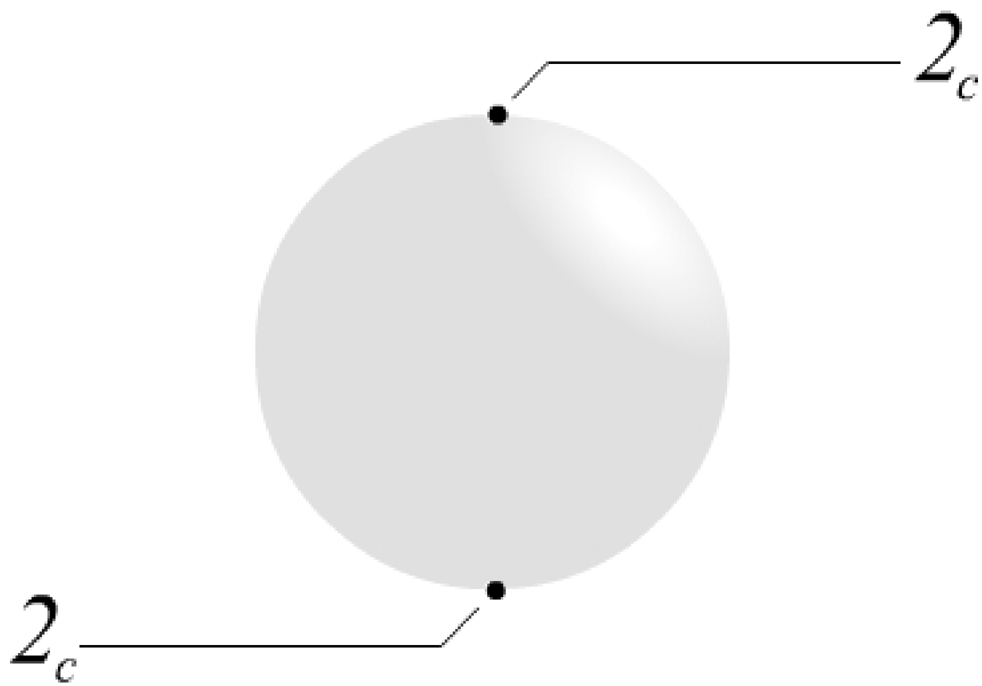
3. Quantum types as topological carriers

In this section, I will show that the quantum types introduced in the last section does not only carry typing information, but also some geometrical / topological information which are not, in general, present in a classical type theory.

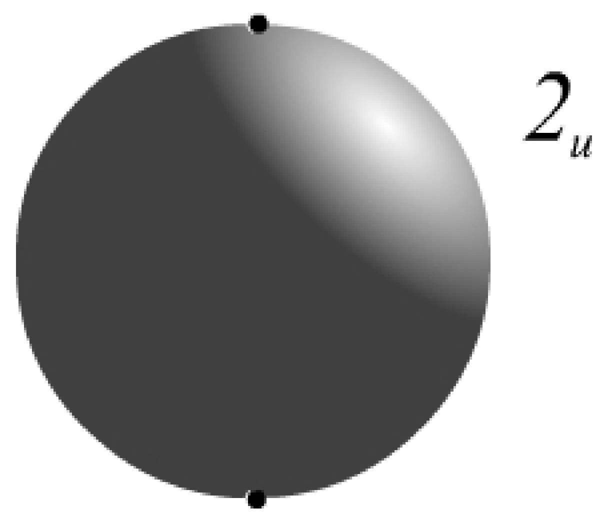
We previously introduced the types n_c, n_u, n_p and n_s. We now give a geometrical interpretation with a toy-model: the Bloch sphere, which corresponds to 2_s (i.e. the set of all 2×2 density matrices).



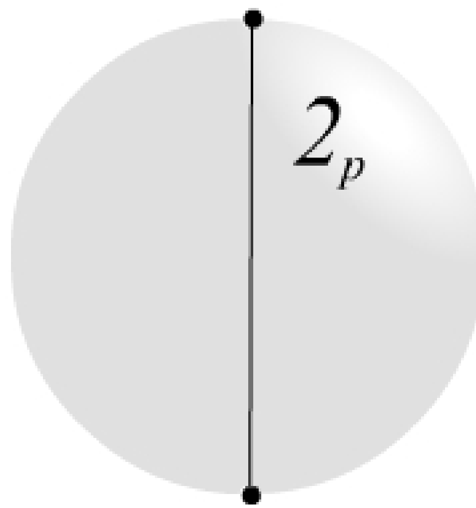
We previously introduced the types n_c, n_u, n_p and n_s. We now give a geometrical interpretation with a toy-model: the Bloch sphere, which corresponds to 2_s (i.e. the set of all 2×2 density matrices).



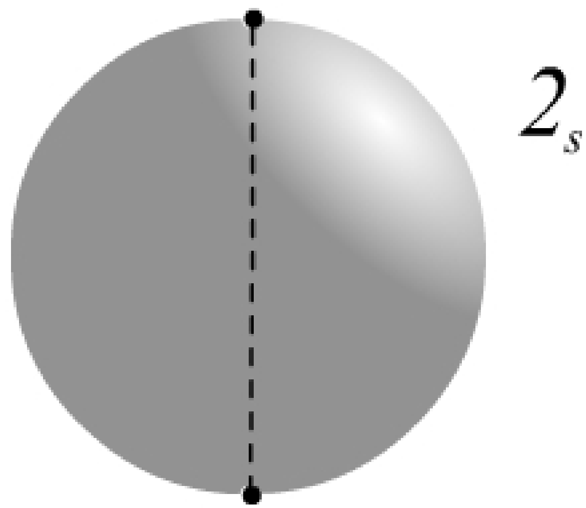
We previously introduced the types n_c, n_u, n_p and n_s. We now give a geometrical interpretation with a toy-model: the Bloch sphere, which corresponds to 2_s (i.e. the set of all 2×2 density matrices).



We previously introduced the types n_c, n_u, n_p and n_s. We now give a geometrical interpretation with a toy-model: the Bloch sphere, which corresponds to 2_s (i.e. the set of all 2×2 density matrices).



We previously introduced the types n_c, n_u, n_p and n_s. We now give a geometrical interpretation with a toy-model: the Bloch sphere, which corresponds to 2_s (i.e. the set of all 2×2 density matrices).



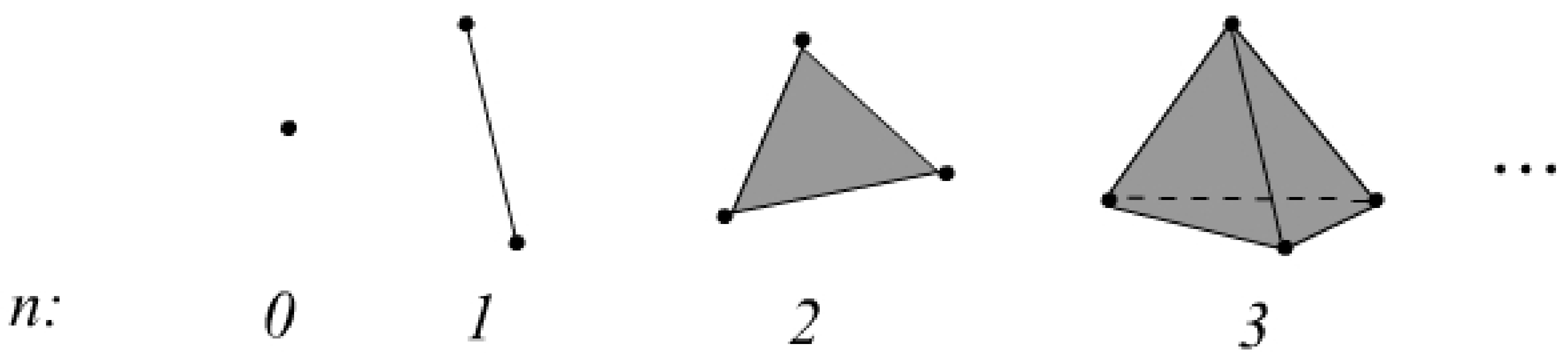
You probably noted that our 4 quantum types are not disjoint, for instance, a basis vector (in n_c) is a general density matrix (in n_s). Thus, we have for a fixed $n \neq 0$ the following subtyping relations:

$$\begin{tikzcd} n_c \arrow[r, hook] \arrow[d, hook] & n_p \arrow[d, hook] \\ n_u \arrow[r, hook] & n_s \end{tikzcd}$$

We now generalize the topological structures for each quantum types.

First, for the type n_c of basis vectors, there is not that much to say. These are *sets* of n elements without additional structure.

Second, for the type n_p of probabilistic data, we have *simplices*. By an n-simplex we mean the n-dimensional analogue of a triangle. Here are the few first simplices:



Third, for the type n_u of pure states, we have *manifolds* (smooth surfaces) which are always isomorphic to $\mathbb{CP}^n$, the n-dimensional complex projective plane i.e. the rays of a given Hilbert space.

Finally, for the type n_s of mixed states, we have a structure of *manifolds with boundaries*. Those manifolds with boundaries are isomorphic to $\mathbb{CP}^{n^2-1}/SU(n)$.

The subtyping relations presented as a square before remains valid for all $n \neq 0$. That is:

$$\begin{tikzcd} n_c \arrow[r, hook] \arrow[d, hook] & n_p \arrow[d, hook] \\ n_u \arrow[r, hook] & n_s \end{tikzcd}$$

This means that each n-set is included in an n-simplex which act as a 'skeleton' for n_s, a manifold with boundary. Also, the n-set is included in the manifold n_u which is the outer strata of n_s.

The key idea here is to note that superoperators are *not* just linear maps but also describes geometric / topological deformation on a given structure or a passage from one topological structure to another. How do we relate all this?

When I speak of passing from one structure to the other, I do not only speak of passing from structures of different types for a fixed n but also passing from one n to another m, we have:

$$0_\square \longrightarrow 1_\square \rightrightarrows 2_\square \; \substack{\longrightarrow \\ \longrightarrow \\ \longrightarrow} \; 3_\square \qquad \ldots$$

Where $\square$ is either c, u, p or s. Each of these arrow represent a way to apply the basis vectors of the type $n_\square$ to the basis vectors of $(n+1)_\square$ or, in other words, to inject the first structure in the second. The full structure of the topological transfer is there if one add the typing square presented previously. Here, $0_\square$ is the empty matrix.

Now, provided that we enrich our type system with an *effect algebra* (i.e. the interval $[0,1] \subset \mathbb{R}$) and that we reverse the arrows of the previous diagram, we obtain:

$$\begin{tikzcd} 0_\square & 1_\square \arrow[l] & 2_\square \arrow[l, shift left] \arrow[l, shift right] & 3_\square \arrow[l, shift left=2] \arrow[l] \arrow[l, shift right=2] & \ldots \end{tikzcd}$$

which is exactly the diagram of the projections. Again, one can paste to this diagram the typing square presented previously.

Remark: This type of diagram is another hint that there is a simplicial structure present in the quantum setup and this is usually very useful for combinatorial purposes.

Recap (types as topological carriers):

Type	Topological Structure	Type structure	Morphisms
n_s	Manifolds with boundaries	Density operators	Superoperators
n_u	Manifolds	Pure sates dens. ops	Unitary maps
n_p	Simplices	Diagonal density ops	Simplicial maps
n_c	Sets	$\mathbf{1}_i$	Classical gates

Where $\mathbf{1}_i$ stands for $n \times n$ density matrices with zeros everywhere except at the ii-th entry. Note that the morphisms considered above are only those who can be characterised as superoperators.

Open questions and future work

- To recast the superoperator formalism in a topolinear context, that is, to give a topological reinterpretation of the superoperators.

- Can we speak of *topological complexity*? By that I mean to recast (quantum) complexity theory in a topological / geometrical context.

- With this analysis can we improve our current notion of QPL?

- And many others much more categorical in essence...

Acknowledgement

Thanks to Hugue Blier and JC Boileau for inputs for preparing this talk. Also, I want to thank Gilles Brassard and Michel Boyer for advices and all members of the LITQ (Université de Montréal) for their ongoing patience with my divergent interests...