CPS and Simulation of Quantum Processes

Jerzy Karczmarczuk
University of Caen, France
karczma@info.unicaen.fr

1 Introduction

Independently of the hope to speed-up some hard algorithms, and of the possibility that quantum programmable devices will help to simulate other quantum systems, computer scientists continue their interest in the domain of *quantum computing* and its simulation because implementable constructions help to shorten the gap between our formal understanding of quantum entities, and the sense of their models. Besides, the theory of algorithms realizable on quantum devices is *per se* an interesting study topic.

We implemented in Haskell a formalism permitting to "put into the computer" quantum states and operators, using *functional objects* . Our approach relies on standard properties of quantum theory, we simply "jumped into" the category of Hilbert spaces in order to implement some standard quantum calculi. Within this framework, the chain of transformations of quantum states bears a striking resemblance to the continuation-passing protocol with linear continuations. While this is a preliminary work, and not everything is clear, this resemblance is not accidental, and may be fruitful. We will arrive at CPS in a natural way, using universal (thus: weak) properties of functions representing quantum entities. The main motivation to present this work is that we haven't seen this analogy in easily accessible literature. We don't plan to add anything new to the theory of programming with continuations, we just hope that this protocol is a promising approach in applications related to the modelling of quantum processes.

2 Quantization, the Functional Way

The configuration (state) of a classical system is a set α of parameters: the position $\vec{p}$ of a particle, the excitation level n of an atom, the values $\downarrow$ or $\uparrow$ of a binary variable q which represents some model of a bit, etc. In the quantum world the **state is a normalized vector** in a Hilbert space, whose basis is usually parameterized by some classical configurations, so that we can call the basis vectors the "classically measurable states", say, a "ket" $|\uparrow\rangle$ in the Dirac *bra-ket* notation.

A *qubit* or another system may find itself in a superposed state $|\psi\rangle = g|\downarrow\rangle + h|\uparrow\rangle$ or $g|0\rangle + h|1\rangle$ (with indices '0' and '1' used for obvious reasons). The scalar product squared: $|\langle 0|\psi\rangle|^2$ gives the probability that in this state the measurement yields '0'. The *bra* vector $\langle\psi|$ is the dual (co-vector) of $|\psi\rangle$. A quantum system evolves by acting upon them with linear unitary operators: $|\psi\rangle' = U|\psi\rangle$. The "evolution stops" when the observer projects the final state on some physically justified basis, and gets some answer. It may be an oracle-type answer (spin-up or down), or it may take the form of some probabilistic measure, after having repeated the experiment many times on identically prepared states. Quantum states of independent subsystems are tensor products of individual vectors: $|100\rangle \equiv |1\rangle \otimes |0\rangle \otimes |0\rangle$, and operators acting on

them are also tensor products. For interacting systems states are superpositions of such tensor states; their separation into components may not be easy.

Computer (classical) realizations of quantum entities have many variants. In view of the simplicity of the evolution process – just a linear algebra – the functional approaches seem particularly appropriate. The data types used to store quantum states may be algebraic. But we took another approach, whose advantage is that the vector structure of the state models arises *naturally*, that it is adaptable to any quantum systems, not only to qubits (or other discrete, finite sets of configurations), and moreover, it corresponds faithfully to models of quantum calculi used by physicists. States and operators are higher-order functions.

A functor which takes a configuration α to a Hilbert space vector is the Haskell functional we call **`ket`**. If a qubit is defined by the **`data QBit = B0 | B1`**, then **`ket B1`** is $|1\rangle$. The construction of **`ket`** is indirect, we shall construct first the co-vectors $\langle\alpha|$, which we call *axes*, equivalent (but not identical) to Dirac bras, which will come later, as linear functionals over kets. We use a variant of construction known sometimes as *Kolmogorov dilation theorem*. Suppose that the "classical sector" of the world is equipped by a particular measure, which discerns between identical and different configurations. We may define a kernel function called **`bracket`**

```
bracket a b = if a==b then 1 else 0
```

i.e., the Kronecker delta. (We won't need here more general definitions, but they ***are*** useful.) The values 0 and 1 should be treated here — as it is usual — to be complex numbers; we assume that they belong to the type **`Scalar`**. If we rename **`axis = bracket`**, the form **`(axis alpha)`** is a unary function, and as such it constitutes naturally an element of a vector space with operations **`(*>), (<+>)`**:

```
axis a <+> axis b = \c -> axis a c + axis b c
h *> axis a = \c -> h*axis a c
```

(We omit the definition of classes specifying the overloading.) In our framework **`axis alpha`** represents $\langle\alpha|$. In geometry typically co-vectors are functionals over vectors, but since the Hilbert space permits a natural definition of dual objects, we reverse the construction. Kets are functionals over axes:

```
ket alpha = \ax -> ax alpha
```

from which we see that kets: (**`ket alpha`**), being unary functions are natural vectors. We can superpose them. Also, that they ***are linear functions***:

```
ket a (ax1 <+> ax2) = (ax1 <+> ax2) a = ket a ax1 + ket a ax2
```

We notice that the definition of **`ket`** is equivalent to the standard Fischer-Plotkin lifting functional of elementary values into the CPS domain! We may say that a state is a "quantum value" awaiting its final continuation (the measurement), and that this continuation is the appropriate axis which yields the answer, the quantum probabilistic amplitude. The type of **`ket`** is **`ket :: a -> (a->Scalar) -> Scalar`**, where **`a`** is the type of "classical values", and the **`Scalar`** is the "answer" type. (In fact, it is useful to keep **`ket`** more polymorphic, needed for a more general lifting.)

2.1 Duality

Since $\langle\psi|\alpha\rangle = \langle\alpha|\psi\rangle^*$, we may immediately construct an axis dual to a ket **`psi`**:

```
dual psi = \a -> conjugate (psi (axis a))
```

It is also easy to *lift* kets to their dual space, functionals are bi-dual, thus isomorphic to axes, acting on kets. This space is considered as the domain of Dirac bras. A bra is defined through

```
coax psi = \phi -> phi(dual psi)
```

or, shorter, through a lifting from axes: **`br = ket ax`**. Of course, **`coax=ket . dual`**. Bras can be lowered to axes by

```
lower br = \a -> br (ket a)        --    ⟨br|α⟩
```

and the diagram at the right commutes. The proof that **`ket`** and **`lower`** are inverse is easy, but not trivial, it holds for elementary axes and kets, and is generalized thanks to the linearity. We have thus a correspondence/duality between a quantum state and its measurement context.

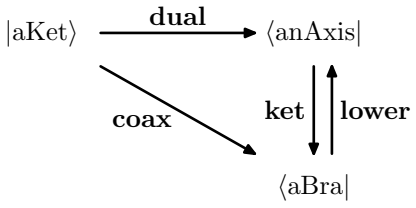

Figure 1: Duality conversions

One has to be cautious. In the language of categories and in algebra we have the terms "duality" or "adjoints" with related, (or subsumed) but not identical meaning, since we work within one, concrete category. We shall see other similar cases, e.g. the term "linearity", which means something different in geometry, and in the continuation theory and in logic.

3 Operators and Circuits

A quantum process in which the initial state $|\psi\rangle_0$ is sequentially acted upon by some operators $\hat{A}, \hat{B}, \hat{C}$, and results in a state $|\psi\rangle' = \hat{A}\hat{B}\hat{C}|\psi\rangle_0$, may be depicted as the following diagram: The time goes from

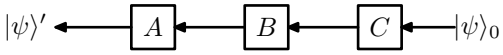

Figure 2: Chain of operators

right to the left. In practical cases, the states are compound, disconnected lines represent tensor products of states. We have also tensor product operators. But in the presence of interactions, the lines get connected. Fig. 3 represents two elementary quantum gates, the "controlled not" and the Toffoli gate, and also their combination: a 1-bit half-adder. More complicated circuits are needed in order to implement some non-

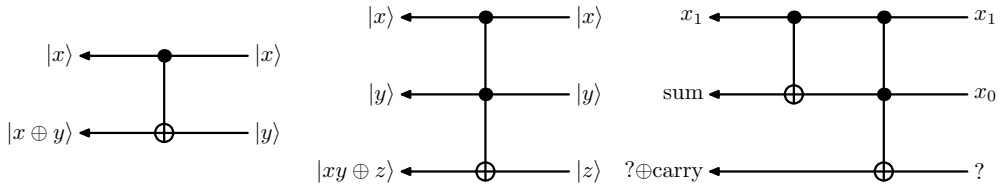

Figure 3: Controlled-not, Toffoli gate, and half-adder

trivial computations, it suffices to look the teleporting circuit on Fig. 4, where L, R, etc. boxes are one-qubit operators which rotate, or otherwise transform the state. E.g., $S = i|0\rangle\langle 0| - |1\rangle\langle 1|$, $L = R^+ =$

$(|1\rangle\langle 0| - |0\rangle\langle 1| + 1)/\sqrt{2}$, etc. We cannot discuss the details. One should be careful: the lines *do not transport classical data*. What is conceptually interesting, is the observation:

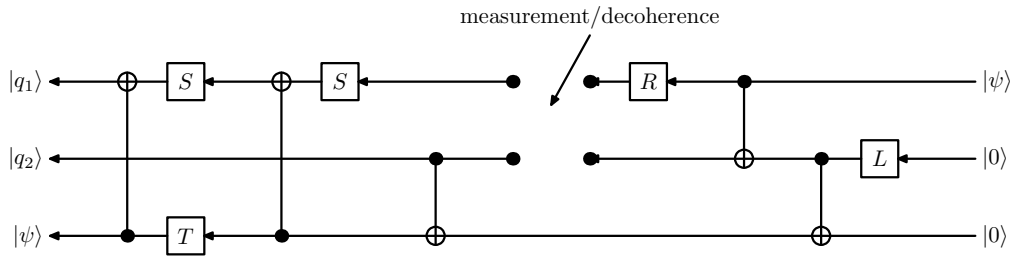

Figure 4: Brassard teleporting circuit

- The evolution is fully functional, moreover it is singly-threaded, 'linear' from the computational point of view. Every state vector is used exactly once. If we want to speak about continuations, then they are also linear. Vectors cannot be discarded in the middle of computation.
- From the physical perspective the model shall represent a *changing, unique world*, not a sequence of state vectors which exist independently from each other.

This encourages us to think that linear continuations have strong potential to model typical quantum processes. A *direct* model of an operator is not immediate, since it should act on a function, finally its argument should act on *something*. If we begin with a qubit function, say **`const B0`**, or **`cnot x = if x==B0 then B1 else B0`**, then the lifting of such a function **`cop`** to an operator acting on kets is

```
(lift cop) psi = \ax -> psi (\alpha -> ax (cop alpha))
```

This is not a Plotkin-style functional, but the type of the object surely is a typed "continued function". If the lifting of a value x is $\overline{x} = \lambda k.k\,x$, the appropriate lifting of $f = \lambda x.f\,x$, acting on lifted values, will be $\overline{f} = \lambda\overline{x}\,k.\overline{x}(\lambda x.k(f\,x))$. This is what we get, the formalism produces a linear "tail call" chain. More general operators are constructed through linear superpositions. We note that the algebraic structure implies also the computational linearity: continuations are linear, used once. The operators which act on kets can be uniquely mapped to their adjoints, which are *continuation transformers*. They are also obviously linear.

4 Tensor states and operators

Quantum structures interesting for computations describe compound systems. For two non-interacting subsystems the state is given by the tensor product, $|\phi\rangle|\psi\rangle$ (noted simply as $|\phi\rangle \otimes |\psi\rangle$, or $|\phi\psi\rangle$)which in functional spaces has a straightforward implementation; if **`psi = \ax -> psi ax`**, etc., then

```
psip = phi <*> psi = \ax ay -> phi ax * psi ay
```

We must deal with n-ary functions, and so, with "compound continuations". The associativity is obvious, and the twist (argument flip) isomorphism as well. This is not a Cartesian product, there are no projections to individual components. Even without interactions we cannot extract individual states from it. We can of course superpose them. The duals of compound states may be constructed through

```
(dual_2 psip)  = \alpha beta ->
   conjugate (psip (axis alpha) (axis beta))
```

This is a one object, a 2-axis. It cannot be directly considered as the continuation of a 2-ket, which needs 2 arguments, and the connection between duals and continuations should be re-analyzed, but all the chaining properties sketched above, hold.

Operators acting independently on subsets of lines are also products. Tensor operators acting on kets are constructed from the *adjoints* acting on axes, by a multilinear lifting. For example,

```
lift_2 ao1 ao2 psip  =
  \ax1 ax2 -> psip (ao1 ax1) (ao2 ax2)
```

which can be generalized to any number of arguments, and linearly combined into non-factorized operators. The drawing at the right is an example of such a composite operator. From the adjoint perspective the construction **`lift_2 ao1 ao2`** might be seen as something which *produces* two outputs, implemented as an object which *consumes the consumer* of two inputs.

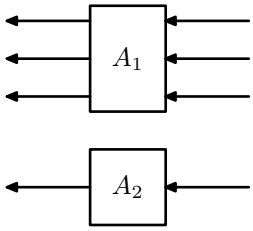

Figure 5: Compound operators

Finally, we see that already the (simple) single-ket operators may be considered as tensor products, the projectors $\hat{P} = |\alpha\rangle\langle\beta|$. The Dirac notation is handy on paper, such forms may act either on vectors or on co-vectors ("from the right", as adjoints), and yield the appropriate scalar products, $\hat{P}|\psi\rangle = \langle\beta|\psi\rangle|\alpha\rangle$, and $\langle\phi|\hat{P} = \langle\phi|\alpha\rangle\langle\beta|$, but the implementation must specialize the variant. If we want to construct an operator acting on axes from **`ktb`** representing $|\beta\rangle$, and **`axa`** denoting $\langle\alpha|$, then the construction is

```
ax_P = \ax -> (ktb ax) *> axa
```

The dual variant is **`kt_P=\psi -> (ket axa psi)*>ktb`**. In both cases they are tensor products. This is analogous to the known equivalence in linear logic, restricted here to $A \multimap A \cong A^{\perp} \otimes A$, where A is either the type of axes or of kets. The tensor structure of our formalism and its relation to linear logic needs further work; according to Barr, the equivalence above holds in finitely-dimensional spaces (compact categories) only. This might be not related directly to continuations...

5 Conclusions

From a certain distance, the dominant role of continuations in our model of quantum processes is obvious, and may be traced back to Church. After all, if basic entities in the framework are functions, then — as in the lambda-calculus — the only way to *do something to them* is to *apply them*. Strong relation between the linearity of operations in vector spaces and the computational linearity is known. From the control point of view the quantum circuits are rather simple, complications arise when we have to deal with the probabilistic nature of measurements, but this part of the simulation has been omitted from these notes, although it deserves our full attention.

We exposed just a bridge which seems quite intuitive, the continuations appear *naturally*. From our personal perspective it is interesting that formalisms developed within the theory of computation, may find some usage in the practice of quantum calculi, and we hope that one day the functional programming will influence the teaching of quantum mechanics as well.