Typing quantum superpositions and measurement

Alejandro Díaz-Caro
Universidad Nacional de Quilmes & CONICET
Bernal, Buenos Aires, Argentina

Gilles Dowek
Inria, LSV, ENS Paris-Saclay
Cachan, France

To appear in LNCS 10687 (TPNC 2017)

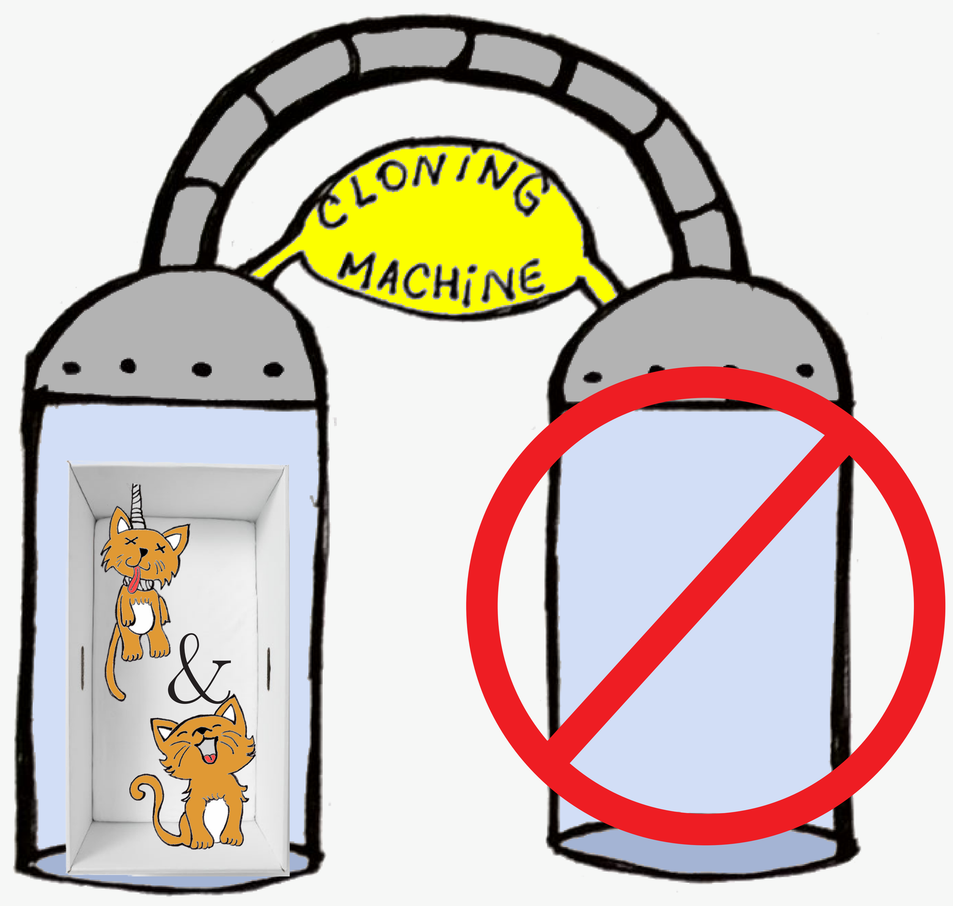
Abstract

We propose a way to unify two approaches of non-cloning in quantum lambda-calculi. The first approach is to forbid duplicating variables, while the second is to consider all lambda-terms as algebraic-linear functions. We illustrate this idea by defining a quantum extension of first-order simply-typed lambda-calculus, where the type is linear on superposition, while allows cloning base vectors. In addition, we provide an interpretation of the calculus where superposed types are interpreted as vector spaces and non-superposed types as their basis.

Forbidding duplication

Applying $\lambda x.(x,x)$ to t reduces to (t,t). But “cloning” any unknown t is forbidden in quantum computing in general.

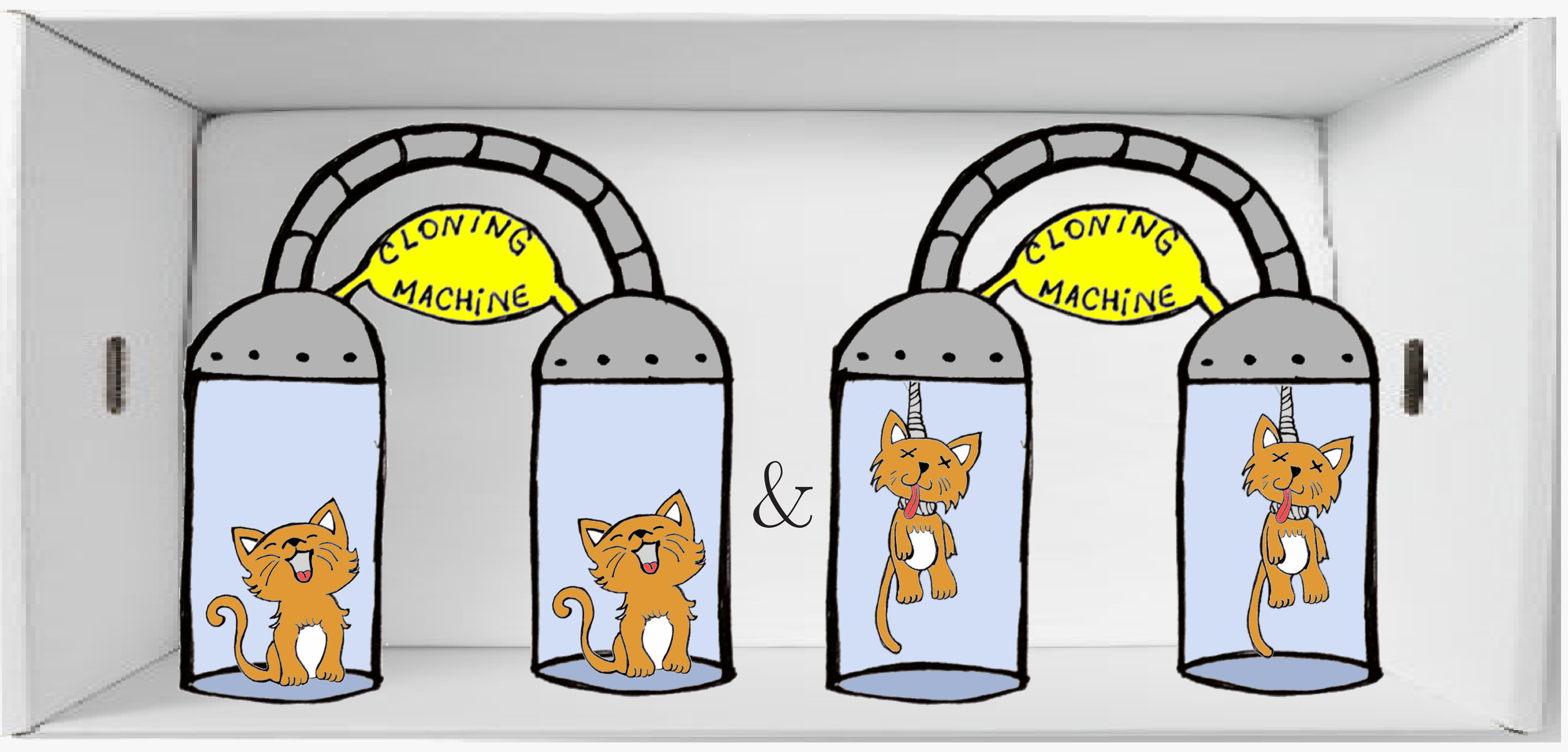
Linear-logic approach



The idea is to forbid abstractions from using its argument more than once. Which implies to forbid $\lambda x.(x,x)$. That is, to forbid the “cloning machine”.

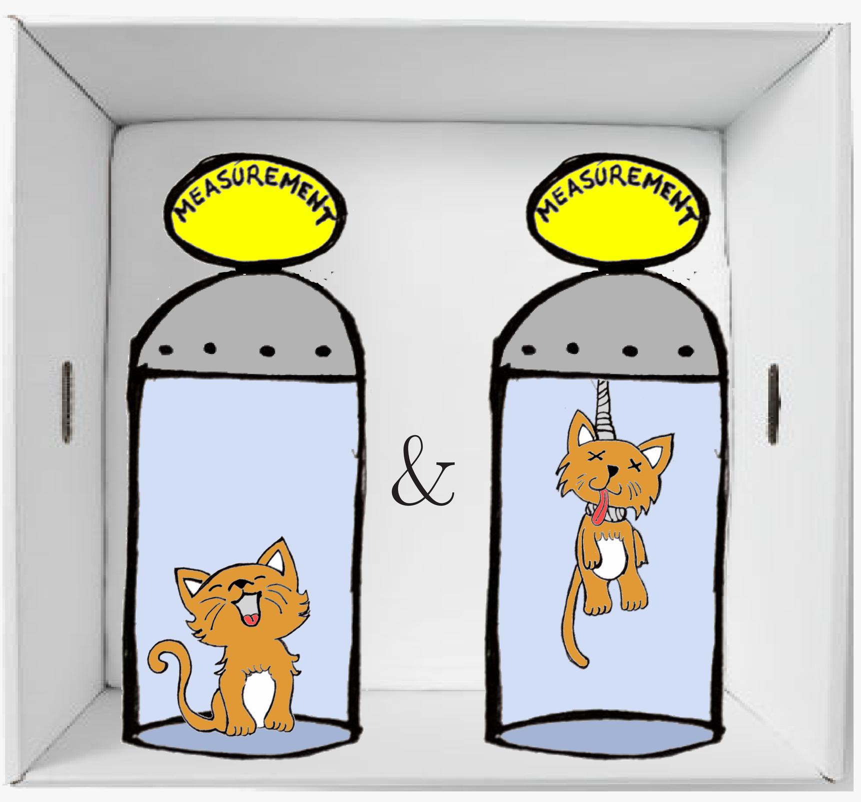
Problem: Forbidding $\lambda x.(x,x)$ may be too restrictive: cloning base states is not a problem in quantum computing.

Linear-algebra approach



The machine applied to a superposition enters the box and is applied to each elementary state.

and measurement



The linear-algebra approach does not make sense here...

...but the linear-logic one, does

Our approach: combining both

An abstraction measuring its argument, have a linear-type argument (a “superposition”).

$$(\lambda x^{S(\mathbb{B})}.\text{measure } x)(\alpha.|0\rangle + \beta.|1\rangle) \to \text{measure } (\alpha.|0\rangle + \beta.|1\rangle)$$

An abstraction not measuring its argument, have a non-linear type (a base type), and distributes linearly

$$(\lambda x^{\mathbb{B}}.(x,x))(\alpha.|0\rangle + \beta.|1\rangle) \to \alpha.(\lambda x^{\mathbb{B}}.(x,x))|0\rangle + \beta.(\lambda x^{\mathbb{B}}.(x,x))|1\rangle$$
$$\to^* \alpha.(|0\rangle,|0\rangle) + \beta.(|1\rangle,|1\rangle)$$

To each type we associate the subset of some vector space

$$[\![\mathbb{B}]\!] = \{\left(\begin{smallmatrix}1\\0\end{smallmatrix}\right), \left(\begin{smallmatrix}0\\1\end{smallmatrix}\right)\}, \text{ a subset of } \mathbb{C}^2$$
$$[\![\Psi \Rightarrow A]\!] = [\![\Psi]\!] \Rightarrow [\![A]\!]$$
$$[\![A \times B]\!] = [\![A]\!] \times [\![B]\!]$$
$$[\![S(A)]\!] = \mathcal{S}[\![A]\!], \text{ the span of the set}$$

Remark that $[\![S(\mathbb{B}\times\mathbb{B})]\!] = \mathcal{S}([\![\mathbb{B}]\!] \times [\![\mathbb{B}]\!]) = [\![\mathbb{B}]\!] \otimes [\![\mathbb{B}]\!]$.

Conclusion and future works

In this paper we have proposed a way to unify logic-linear and algebraic-linear quantum λ-calculi, by interpreting λ-terms as linear functions when they expect duplicable data and as non-linear ones when they do not, and illustrated this idea with the definition of a calculus.

Semantically, we are distinguishing between vectors in the computational basis from linear combinations of them. That information is carried on the type, even with multiple qubits, however, we added a mechanism to forget that information in certain situations.

Indeed, the type of $(|0\rangle, \alpha|0\rangle + \beta|1\rangle)$ would be $\mathbb{B} \times S(\mathbb{B})$, while the type of $\alpha.(|0\rangle,|0\rangle) + \beta.(|0\rangle,|1\rangle)$ can only be $S(\mathbb{B}\times\mathbb{B})$. Hence, the first cannot reduce to the latter but when an explicit type-casting is present.

Some ongoing continuations of this work are:

- With Octavio Malherbe (UdelaR) we are studying a categorical model for this calculus.
- With Ignacio Grima (UNR) and Pablo E. Martínez López (UNQ) we are working on an implementation of it.

Universidad Nacional de Quilmes · CONICET · Inria (informatiques mathématiques) · LSV · école normale supérieure paris-saclay