Cloning and Deleting Quantum Information from a Linear Logical Point of View

Anderson Beraldo-de-Araújo

August 1, 2022

Cloning and Deleting Quantum Information from a Linear Logical Point of View*

Anderson Beraldo-de-Araujo
Center for Natural and Human Sciences (CCNH)
Federal University of ABC (UFABC)
Center of Mathematics (CMAT)
University of Minho (UMinho)
anderson.araujo@ufabc.edu.br

This paper displays a linear sequent calculus in accordance with the no-cloning and no-deleting theorems of quantum computing. The calculus represents typing operations on matrices in terms of linear sequent rules, satisfing admissibility of cut. It is possible to define a strict monoidal categorical semantics for it using categories generated by finite vector spaces extended with a Kronecker product, which can be viewed as the dual of an approach proposed by Abramsky-Coecke.

1 Introduction

It is recorrent in linear logical approaches to quantum computing, to mention the no-cloning theorem as a motivation for the use of linear logic in this context (Cf. [4, 6]). It says that there is no unitary operation which makes perfect copies of an unknown (pure) quantum state [8, 20]. In linear logic this can be represented by the fact that $A \nvdash A \otimes A$. Nonetheless, there is also a no-deleting theorem [17], and thus a correct linear logical approach to quantum computing should give us $A \otimes A \nvdash A$ too, which is not the case in the usual systems for multiplicative linear logic wherein $\otimes$ is a kind of conjunction.

Abramsky [1] studied this problem from the point of view of categorical quantum mechanics based on linear logic. He proved that there is no universal cloning morphism for any non-trivial dagger compact categories (the same holds for deleting morphisms). In this categorical context, cloning is associated with diagonals and deleting to projections. By considering that a cartesian structure is a monoidal structure plus natural diagonals, and with the tensor unit a terminal object, Abramsky realized that the categorical versions of cloning and deleting correspond to Joyal's lemma [12], which states that any cartesian closed category with a dualizing object is a preorder (hence trivial as a semantics for proofs). The same holds for the categorical semantics of the multiplicative linear logic via $*$-autonomous category [5] - their cartesian monoidal structure makes them a preorder. Hence, Abramsky posed the problem: Are there non-trivial examples of $*$-autonomous categories with uniform cloning operations?

In the present article we propose to consider a kind of dual of Abramsky's problem: Is there a linear logic system with a non-trivial categorical semantics in which non-deleting holds? We believe this is the really interesting question as far as the relation between linear logic and quantum cloning/deleting is concerned, due to two main reasons. First, no-cloning and no-deleting are quantum mechanics facts. Why should we care about the opposite? Second, linear logic already satisfies a version of no-cloning, namely, $A \nvdash A \otimes A$, and so does its semantics in terms of $*$-autonomous categories. What can we do to get $A \otimes A \nvdash A$ in linear logic? That is our problem, inspired by Abramsky's work.

We display a linear sequent calculus $\mathscr{M}$ for matrices in which no-cloning and no-deleting are immediate properties of tensors. $\mathscr{M}$ is a linear calculus which considers linear operations on matrices in

*This research was supported by Fapesp, Grant 10/51038-0.

Submitted to:
LINEARITY 2022

terms of linear sequent calculus rules (Section 2). The admissibility of the cut rule can be proved using the usual methods (Cf. [15]), but $\mathscr{M}$ is quite different from the approaches that use linear logic to understand the linear algebra underlying quantum computing (Section 3). This perspective follows MacLane's [14] idea of typing matrices as morphisms, introducing biproducts. This line of research has been explored by Macedo & Oliveira in the pursuit of avoiding the cumbersome indexed-based operations of matrices [13]. More recently, Fong & Spivak [9] and Jacobs & Sprunger [11] have also developed it in the context of machine learning. In contrast, we do not employ biproducts as Macedo & Oliveira nor work on monoidal categories as Fong & Spivak and Jacobs & Sprunger. We provide a categorical semantics for $\mathscr{M}$ using categories generated by finite vector spaces extended with a Kronecker product (Section 4), something more closed to the standard practice of linear algebra used in quantum computing (Cf. [16]) than the monoidal categorical approach initiated by Abramsky and Coecke [2].

2 A linear matrix calculus

In this section, we display the linear sequent calculus $\mathscr{M}$. This calculus defines rules for sums, scalar product, multiplication and Kronecker product of matrices. It makes an internal representation of the linear operations, and not an external one, as usual in the linear logic literature, in which tensors are viewed as operations on vector spaces.

We assume an infinite, but enumerable, set of variables $\mathscr{S}$.

Definition 2.1. *The* scalars *are given by the syntax:*

$$0 \mid 1 \mid \zeta + \xi \mid \zeta \cdot \xi \mid -\zeta \mid \zeta^{-1}$$

We assume an infinite, but enumerable, set of variables $\mathscr{V}$.

Definition 2.2. *The* vectors *are given by the syntax:*

$$\mathbb{O} \mid \mathbb{I} \mid \zeta * \rho \mid \eta \oplus \rho \mid \eta \circ \rho \mid \eta \otimes \rho$$

where ζ is a scalar.

From now on, $\mathbb{N}^\star$ is the set of natural numbers without the number zero.

Definition 2.3. *The* types *are given by the syntax:*

$$n \multimap m \mid n \cdot p \multimap m \cdot q$$

where $n, m, p, q \in \mathbb{N}^\star$.

Our main thesis is this: matrices are typed vectors. Everything follows from this ground.

Definition 2.4. *The* matrices *are given by the syntax:*

$$\sigma : \tau$$

where σ is a vector and τ is a type.

In the system $\mathscr{M}$, sequent are kind of lists. When the order matters, we use semicolons, when it does not, we use commas. We call these mixtures of order and unorder lists by *aggregates*, and assume in the present work an intuitive understanding about them. This expedient is necessary since matrix multiplication and Kronecker product are not commutative.

Definition 2.5. *The* rules *of the system $\mathscr{M}$ are the following:*

$$\frac{}{A : n \multimap m \vdash A : n \multimap m}\ Ax$$

$$\frac{\Gamma, \rho : \pi, \eta : \tau \vdash \alpha}{\Gamma, \eta : \tau, \rho : \pi \vdash \alpha}\ Ex$$

$$\frac{\Gamma, \eta : \tau \vdash \alpha}{\Gamma, 1 * \eta : \tau \vdash \alpha}\ 1_L \qquad \frac{\Gamma \vdash \eta : \tau}{\Gamma \vdash 1 * \eta : \tau}\ 1_R$$

$$\frac{\Gamma, a : 1 \multimap 1, \eta : \tau \vdash \alpha}{\Gamma, a * \eta : \tau \vdash \alpha}\ *_L \qquad \frac{\Gamma, \vdash a : 1 \multimap 1 \quad \Delta, \vdash \eta : \tau}{\Gamma, \Delta \vdash a * \eta : \tau}\ *_R$$

$$\frac{\Gamma, \eta : \tau \vdash \alpha}{\Gamma, \eta \oplus \mathbb{O} : \tau \vdash \alpha}\ \mathbb{O}_{L_0} \qquad \frac{\Gamma, \eta : \tau \vdash \alpha}{\Gamma, \mathbb{O} \oplus \eta : \tau \vdash \alpha}\ \mathbb{O}_{L_1}$$

$$\frac{\Gamma, \mathbb{O} : \tau \vdash \alpha}{\Gamma, (-1 * \eta) \oplus \eta : \tau \vdash \alpha}\ -_L \qquad \frac{\Gamma \vdash \mathbb{O} : \tau}{\Gamma \vdash (-1 * \eta) \oplus \eta : \tau}\ -_R$$

$$\frac{\Gamma, a * \eta : \tau \vdash \alpha \quad \Gamma, b * \eta : \tau \vdash \alpha}{\Gamma, (a+b) * \eta : \tau \vdash \alpha}\ +_L \qquad \frac{\Gamma \vdash a * \eta : \tau}{\Gamma \vdash (a+b) * \eta : \tau}\ +_{R_0} \qquad \frac{\Gamma \vdash b * \eta : \tau}{\Gamma \vdash (a+b) * \eta : \tau}\ +_{R_1}$$

$$\frac{\Gamma, \eta : \tau \vdash \alpha \quad \Gamma, \rho : \tau \vdash \alpha}{\Gamma, \eta \oplus \rho : \tau \vdash \alpha}\ \oplus_L \qquad \frac{\Gamma \vdash \eta : \tau}{\Gamma \vdash \eta \oplus \rho : \tau}\ \oplus_{R_0} \qquad \frac{\Gamma \vdash \rho : \tau}{\Gamma \vdash \eta \oplus \rho : \tau}\ \oplus_{R_1}$$

$$\frac{\Gamma, \eta : u \multimap m; \rho : n \multimap u \vdash \alpha}{\Gamma, \eta \circ \rho : n \multimap m \vdash \alpha}\ \circ_L \qquad \frac{\Gamma \vdash \eta : u \multimap m \quad \Delta \vdash \rho : n \multimap u}{\Gamma, \Delta \vdash \eta \circ \rho : n \multimap m}\ \circ_R$$

$$\frac{\Gamma, \eta : \tau \vdash \alpha}{\Gamma, 1 \otimes \eta : \tau \vdash \alpha}\ 1^{\otimes}_{L_0} \qquad \frac{\Gamma, \eta : \tau \vdash \alpha}{\Gamma, \eta \otimes 1 : \tau \vdash \alpha}\ 1^{\otimes}_{L_1} \qquad \frac{\Gamma \vdash \eta : \tau}{\Gamma \vdash 1 \otimes \eta : \tau}\ 1^{\otimes}_{R_0} \qquad \frac{\Gamma \vdash \eta : \tau}{\Gamma \vdash \eta \otimes 1 : \tau}\ 1^{\otimes}_{R_1}$$

$$\frac{\Gamma, \eta : n \multimap m; \rho : q \multimap p \vdash \alpha}{\Gamma, \eta \otimes \rho : n \cdot q \multimap m \cdot p \vdash \alpha}\ \otimes_L \qquad \frac{\Gamma \vdash \eta : n \multimap m \quad \Delta \vdash \rho : q \multimap p}{\Gamma, \Delta \vdash \eta \otimes \rho : n \cdot q \multimap m \cdot p}\ \otimes_R$$

We assume all proof-theoretical concepts such as they are defined in [15]. Due to the restrictions about the axioms, we can verify that no-cloning and no-deleting are true for the system $\mathscr{M}$.

Theorem 2.1. *For every vector ρ of the language of $\mathscr{M}$ different from 1, $\rho : n \multimap m \nvdash \rho \otimes \rho : n \cdot n \multimap m \cdot m$ and $\rho \otimes \rho : n \cdot n \multimap m \cdot m \nvdash \rho : n \multimap m$ in $\mathscr{M}$.*

Proof. The proof is by induction on the complexity of ρ with a subinduction on the height of proofs. For ρ an atomic vector A, the smallest possible proof segments are these:

$$\frac{A : n \multimap m \vdash A : n \multimap m \quad A : n \multimap m \vdash A : n \multimap m}{A : n \multimap m, A : n \multimap m \vdash A \otimes A : n \cdot n \multimap m \cdot m}\ \otimes_R \qquad \frac{A : n \multimap m; A : n \multimap m \vdash A : n \multimap m}{A \otimes A : n \cdot n \multimap m \cdot m \vdash A : n \multimap m}\ \otimes_L$$

The segment to the left can be transformed into a proof with just the rule Ax, but the right one cannot. Anyway, it is true that $A : n \multimap m, A : n \multimap m \nvdash A \otimes A : n \cdot n \multimap m \cdot m$ and $A \otimes A : n \cdot n \multimap m \cdot m \nvdash A : n \multimap m$. For ρ in general, the last rule applied to get deleting may have the following forms:

$$\frac{\rho : n \multimap m; \rho : n \multimap m \vdash \rho : n \multimap m}{\rho \otimes \rho : n \cdot n \multimap m \cdot m \vdash \rho : n \multimap m} \otimes_L$$

$$\frac{\rho \otimes \rho : n \cdot n \multimap m \cdot m \vdash \eta : n \multimap m}{\rho \otimes \rho : n \cdot n \multimap m \cdot m \vdash \rho : n \multimap m} Rule_{R_0} \qquad \frac{\vdash \kappa : w \multimap u \qquad \rho \otimes \rho : n \cdot n \multimap m \cdot m \vdash \lambda : q \multimap p}{\rho \otimes \rho : n \cdot n \multimap m \cdot m \vdash \rho : n \multimap m} Rule_{R_1}$$

We can show on the induction on the height of these segments that there is no proof of $\rho : n \multimap m; \rho : n \multimap m \vdash \rho : n \multimap m$, and there is no proof of $\rho \otimes \rho : n \cdot n \multimap m \cdot m \vdash \eta : n \multimap m$ for η a subformula of ρ and neither of $\vdash \kappa : w \multimap u$. The last one is straightforward because there is no axiom of the form $\vdash A : w \multimap u$, and it is sufficient to use the induction hypothesis. For $\rho \otimes \rho : n \cdot n \multimap m \cdot m \vdash \eta : n \multimap m$ for η, if there were a proof of such a sequent, we could change it to a proof of $\eta \otimes \eta : n \cdot n \multimap m \cdot m \vdash \eta : n \multimap m$, which would contradict the induction hypothesis on the complexity of ρ. Finally, for $\rho : n \multimap m; \rho : n \multimap m \vdash \rho : n \multimap m$ the base is equal to the case for ρ an atom, which we already analysed. For the induction step, it does matter which rule was applied to get a proof P of $\rho : n \multimap m; \rho : n \multimap m \vdash \rho : n \multimap m$, this rule was applied to a subformula ρ_0 of ρ. Thus, we could take this proof P and uniformly substitute ρ_0 for ρ in it, possibility changing types in accordance to the type of ρ_0, to obtain a proof of $\rho_0 : s \multimap r; \rho_0 : s \multimap r \vdash \rho_0 : s \multimap r$. This would imply the existence of the following proof:

$$\frac{\dfrac{\begin{array}{c} P \\ \vdots \end{array}}{\rho_0 : s \multimap r; \rho_0 : s \multimap r \vdash \rho_0 : s \multimap r}}{\rho_0 \otimes \rho_0 : s \cdot s \multimap r \cdot r \vdash \rho_0 : s \multimap r} \otimes_L$$

This would contradict the assumption that there is no such a proof of deleting for any subformula of ρ. Of course, the proof for cloning is dual. □

3 Cut admissibility

In this section we analyse the relation between the system $\mathscr{M}$ and linear logic in general. We do this in the context of the cut admissibility, which can be proved for $\mathscr{M}$ using the usual method of reduction of the cut degree except by a subtle detail.

Theorem 3.1. *The following cut rules is, jointely, admissible in the system $\mathscr{M}$:*

$$\frac{\Gamma, \Delta \vdash \eta : \tau \qquad \Gamma, \eta : \tau, \Lambda \vdash \alpha}{\Gamma, \Delta, \Lambda \vdash \alpha} Cut_0,$$

$$\frac{\Gamma, \Delta \vdash \eta : \tau \qquad \Gamma, \eta : \tau; \Lambda \vdash \alpha}{\Gamma, \Delta, \Lambda \vdash \alpha} Cut_1,$$

where $\Gamma \cap \Delta \cap \Lambda = \varnothing$.

Proof. Since there is no novelty in the proof, we just show the case of the interaction application of the cut rule for matrix multiplication, because this case illustrate the main difference to the regular cut rule. In this case, we have:

$$\dfrac{\dfrac{\Gamma \vdash A : u \multimap m \quad \Delta \vdash B : n \multimap u}{\Gamma \vdash A \circ B : n \multimap m} \quad \dfrac{\Gamma, \Lambda, A : u \multimap m; B : n \multimap u \vdash}{\Gamma, \Lambda, A \circ B : n \multimap m \vdash}}{\Gamma, \Delta, \Lambda \vdash}\ Cut_0$$

We replace that cut by the following cuts, wherein the first has cut-degree smaller than the one above and the second is applied to a formula of smaller complexity:

$$\dfrac{\Delta \vdash B : n \multimap u \quad \dfrac{\Gamma \vdash A : u \multimap m \quad \Gamma, \Lambda, A : u \multimap m; B : n \multimap u \vdash}{\Gamma, \Lambda, B : n \multimap u \vdash}\ Cut_1}{\Gamma, \Delta, \Lambda \vdash}\ Cut_0$$

In this reduction, we have assumed the cut admissibility for both Cut_0 and Cut_1 together in the system $\mathscr{M}$. □

Beyond the fact that the cut rules Cut_0 and Cut_1 operate on aggregates, the crucial aspect of them relatively to the cut rule in additive, multiplicative or full linear logics is that Cut_0 and Cut_1 are neither additive or multiplicative. As we can see in the proof above neither additive nor multiplicative cuts will be admissible. This is not a restriction associated with the way we have defined the rules for the multiplication $\circ$ or the tensor $\otimes$. Such a restriction arises from the no-deleting property of the tensor, namely: $A \otimes A \nvdash A$. To keep this property the system cannot allow general axioms of the form $\Gamma, A \nvdash A$, which forces, by its turn, the rules for $\circ$ and $\otimes$ to be multiplicative.

In contrast, the unacceptability of general axioms $\Gamma, A \nvdash A$ forces the addition $\oplus$ to be additive. If one looks at the proof of the distributive property $(\eta \circ \rho) \oplus (\eta \circ \gamma) : n \multimap m \equiv \eta \circ (\rho \oplus \gamma) : n \multimap m$ displayed at Theorem 4.1, they will see that such a proof would not work for an additive rule for $\oplus$. Another curiosity about vector addition is that its unity $\mathbb{O}$ requires special rules. In the usual systems of linear logic there is no need for a rule for $\mathbb{O}$. Indeed, the computational interpretation $\oplus$ is that a proof of $A \oplus B$ is either a proof of A or a proof of B, so the set of proofs of $A \oplus B$ is mostly the disjoint union of the set of proofs of A and the set of proofs of B. But if $A \oplus \mathbb{O}$ and A must be equivalent, this means that the set of proofs of $A \oplus \mathbb{O}$ and the set of proofs of A are essentially the same, therefore there must be no proof of $\mathbb{O}$. This is actually the content of the rules $\mathbb{O}_{L_0}$ and $\mathbb{O}_{L_1}$, but in the context of linear algebra they must be stated to ensure the existence of the neutral for $\oplus$.

Of course, one could ask yourself: Does the cut rule for systems where the objects are matrices make any sense at all? A proof with cuts of a sequent like $A, B \vdash C$ is understandable for A, B and C as propositions, but what does it mean to say that a matrix C is deducible from matrices A and B? One possible interpretation is that C is decomposable in terms of A and B. In this reading $(\eta \circ \rho) \oplus (\eta \circ \gamma) : n \multimap m \equiv \eta \circ (\rho \oplus \gamma) : n \multimap m$ expresses that the three operations from the left can be decomposed into the two from the right, and vice-versa. In general, cut elimination is equivalent to the subformula property and thus it says that the decomposition of linear operators only involves the linear transformations present in their arguments.

4 Categorical semantics

In this section, we define a categorical semantics for the system $\mathscr{M}$. We begin with the definition of a structure $\mathscr{A}$ that interprets the vectors of the language of $\mathscr{M}$.

Definition 4.1. *Let $\mathbb{F} = (F, 0, 1, +, \cdot)$ be a field and $n, m \in \mathbb{N}^\star$ natural numbers. An $m \times n$-matrix M is a function $M : m \times n \to F$. Addition $M \oplus N$ of $m \times n$-matrices M and N is given by $(M \oplus N)(i, j) = M(i, j) + N(i, j)$. Multiplication of the $m \times u$-matrix M and $u \times n$-matrix N gives an $m \times n$-matrix $M \circ N$ defined by $(M \circ N)(i, j) = \sum_{v \leq u} M(i, v) \cdot N(v, j)$. Intra-product of the 1×1-matrix a and the $m \times n$-matrix M gives an $m \times n$-matrix $a * M$ defined by $(a * M)(i, j) = a(1, 1) \cdot M(i, j)$. Kronecker product of the $m \times n$-matrix M and $p \times q$-matrix N gives an $mp \times nq$-matrix $M \otimes N$ defined by $(M \otimes N)(i, j) = M(\lceil i/p \rceil, \lceil j/q \rceil) \circ N((i-1)\,\$\,p + 1, (j-1)\,\$\,q + 1)$, where $\lceil\ \rceil$ is the ceil and $\$$ is the remainder. The matrix $\mathbb{O}_{m\times n}$ is such that $\mathbb{O}_{m\times n}(i, j) = 0$ for all $(i, j) \in m \times n$. The matrix $\mathbb{I}_k$ is such that $\mathbb{I}_k(i, i) = 1$ for all $i \leq k$.*

The definition above only states the usual definitions of the matrix operations, with some subtle differences. Instead of using the scalar product, we defined the intra-product, which acts precisely like the scalar multiplication but is defined on matrices. Another detail is the pointwise definition of the Kronecker product, in contrast with the block-wise regular approach. Of course, these distinctions do not change the meaning of the matrix operations. We now adapt MacLane's category of matrix to our framework.

Lemma 4.1. *The structure $(\mathbb{N}^*, \mathbb{M}(\mathbb{F}), \{\mathbb{I}_k\}_{k\in\mathbb{N}^*}, \circ, \otimes)$, where $\mathbb{M}(\mathbb{F})$ is the set of $m \times n$-matrices over a field $\mathbb{F}$ for each $n, m \in \mathbb{N}^\star$, generates a strict monoidal category of matrices $\mathbb{M}$.*

Proof. The category $\mathbb{M}$ defined over $(\mathbb{N}^*, \mathbb{M}(\mathbb{F}), \{\mathbb{I}_k\}_{k\in\mathbb{N}^*}, \circ)$ has as objects the natural numbers $\mathbb{N}^*$ and, for each $m \times n$-matrix $M \in Mat(\mathbb{F})$, $\mathbb{M}$ has a morphism $f_M : n \to m$. For each identity matrix $\mathbb{I}_k$, $\mathbb{M}$ has a morphism $1_k : k \to k$ which are identities in $\mathbb{M}$. Given morphisms $f_M : n \to u$ and $g_N : u \to m$ of $\mathbb{M}$, the composition $g_N \circ f_M$ is the map $h_{N\circ M} : m \to n$, where $N \circ M$ is the multiplication of the matrices N and M, that is, the following diagram commutes in $\mathbb{M}$:

$$\begin{tikzcd} n \arrow[r, "f_M"] \arrow[dr, "h_{N\circ M}"'] & u \arrow[d, "g_N"] \\ & m \end{tikzcd}$$

Hence, unitality and associativity follows, respectively, from the properties of the identity matrices and the multiplication of matrices. The monoidal unit is the morphim $1_{[1]} : 1 \to 1$. Given morphisms $f_M : n \to m$ and $g_N : q \to p$ of $\mathbb{M}$, the monoidal functor $g_N \otimes f_M$ is the map $h_{N\otimes M} : q \cdot n \to p \cdot m$, where $N \otimes M$ is the Kronecker product of the matrices N and M, that is, the following diagram commutes in $\mathbb{M}$:

$$\begin{tikzcd} (n,q) \arrow[r, "\cdot"] \arrow[d, "(f_M, g_N)"'] & n \cdot q \arrow[d, "h_{N\otimes M}"] \\ (m,p) \arrow[r, "\cdot"] & m \cdot p \end{tikzcd}$$

Since $[1]\otimes A = A\otimes[1] = A$ and $A\otimes(B\otimes C) = (A\otimes B)\otimes C$ are properties of the Kronecker product, $\otimes_{N\times M}$ satisfies the strict coherence conditions. Therefore, $\mathbb{M}$ is actually a strict monoidal category of matrices. $\square$

Definition 4.2. *A* vectorial category *is a monoidal category with two partial functors ϕ and ψ such that ϕ is a monoidal product and ψ is an invertible monoidal product.*

Lemma 4.2. *The monoidal category of matrices $\mathbb{M}$ plus the matrix operations $\oplus$ and $*$ forms a vectorial category.*

Proof. We extend $\mathbb{M}$ in such a way that if $a : 1 \to 1$ and $h_P : k \to u$ are in $\mathbb{M}$, then $a * h_P$ in $\mathbb{M}$ is the map $r_{a*P} : k \to u$ as well as if $f_M : n \to m$ and $g_N : n \to m$ are in $\mathbb{V}$, then $g_N \oplus f_M$ in $\mathbb{M}$ is the map $s_{N\oplus M} : n \to m$. These conditions mean that the following diagrams commute:

$$
\begin{tikzcd}
(1,k) \arrow[r, "\pi_R"] \arrow[d, "(a{,}h_P)"'] & k \arrow[d, "r_{a*P}"] \\
(1,u) \arrow[r, "\pi_R"] & u
\end{tikzcd}
\qquad
\begin{tikzcd}
(n,n) \arrow[r, "\pi_R"] \arrow[d, "(f_M{,}g_N)"'] & n \arrow[d, "s_{N\oplus M}"] \\
(m,m) \arrow[r, "\pi_R"] & m
\end{tikzcd}
$$

$\square$

Definition 4.3. *The symbol $\eta : \tau \equiv \rho : \tau$ means $\eta : \tau \vdash \rho : \tau$ and $\rho : \tau \vdash \eta : \tau$ in the system $\mathscr{M}$.*

To finish, we prove the fundamental result of the present work.

Theorem 4.1. *There is a vectorial category that induces a structure $\mathbb{V}$ over the functional signature of $\mathscr{M}$ such that, for every matrices $\eta : \tau$ and $\rho : \tau$, if $\mathbb{V} \models \eta = \rho$, then $\eta : \tau \equiv \rho : \tau$.*

Proof. Let $\mathbb{M}$ be the vectorial category built in lemma 4.2. From it, we can define the structure $\mathbb{V} = (\mathbb{M}(\mathbb{F}), \{\mathbb{O}_{m\times n}, \mathbb{I}_k\}_{k,m,n\in\mathbb{N}^*}, \circ, \otimes, \oplus, *)$, which is a first-order structure with an infinite number of designated elements. We add to the language of $\mathbb{V}$ the identity $=$ and define the interpretation of the functional signature of $\mathscr{M}$ in $\mathbb{V}$ in the regular way. Hence, $\mathbb{V} \models \eta = \rho$ means that $\mathbb{V}(\eta) = \mathbb{V}(\rho)$ is true in $\mathbb{V}$, wherein is pressuposed that η and ρ have the same type to the equality to be true.

Suppose that $\mathbb{V} \models \eta = \rho$ for arbitrary matrices $\eta : \tau$ and $\rho : \tau$. We know that $\mathbb{V}$ is a vector space. This means that $\eta = \rho$ either is a vector space axiom or it follows from the vector space axioms via reflexivity, symmetry and transitivity of identity, because vector spaces are given by equational theories [3]. We can prove by induction of the height of proofs that $\eta : \tau \equiv \eta : \tau$, because of the axioms $A : n \multimap m \vdash A : n \multimap m$, and that $\eta : \tau \equiv \rho : \tau$ implies $\rho : \tau \equiv \eta : \tau$, due to the symmetry of the rules of $\mathscr{M}$ to left and to the right. The transitivity follows from the cut admissibility (Theorem 3.1). Hence, it is sufficient to prove that $\eta : \tau \equiv \rho : \tau$ when $\eta = \rho$ is a vector space axiom. This can be straightforwardly done, consider, for instance, the distributivity axioms bellow:

$$
\dfrac{
\dfrac{
\dfrac{
\dfrac{}{\eta : u \multimap m \vdash \eta : u \multimap m}\,Ax
\quad
\dfrac{}{\rho : n \multimap u \vdash \rho : n \multimap u}\,Ax
}{\eta : u \multimap m; \rho : n \multimap u \vdash \eta\circ\rho : n \multimap m}\,\circ_R
}{\eta : u \multimap m; \rho : n \multimap u \vdash (\eta\circ\rho)\oplus(\eta\circ\gamma) : n \multimap m}\,\oplus_R
\quad
\dfrac{
\dfrac{
\dfrac{}{\eta : u \multimap m \vdash \eta : u \multimap m}\,Ax
\quad
\dfrac{}{\gamma : n \multimap u \vdash \gamma : n \multimap u}\,Ax
}{\eta : u \multimap m; \gamma : n \multimap u \vdash \eta\circ\gamma : n \multimap m}\,\circ_R
}{\eta : u \multimap m; \gamma : n \multimap u \vdash (\eta\circ\rho)\oplus(\eta\circ\gamma) : n \multimap m}\,\oplus_R
}{
\dfrac{\eta : u \multimap m; \rho\oplus\gamma : n \multimap u \vdash (\eta\circ\rho)\oplus(\eta\circ\gamma) : n \multimap m}{\eta\circ(\rho\oplus\gamma) : n \multimap m \vdash (\eta\circ\rho)\oplus(\eta\circ\gamma) : n \multimap m}\,\circ_L
}\,\oplus_L
$$

$$
\begin{prooftree}
\AxiomC{}
\RightLabel{Ax}
\UnaryInfC{$\eta : u \multimap m \vdash \eta : u \multimap m$}
\AxiomC{}
\RightLabel{Ax}
\UnaryInfC{$\rho : n \multimap u \vdash \rho : n \multimap u$}
\RightLabel{$\oplus_R$}
\UnaryInfC{$\rho : n \multimap u \vdash \rho \oplus \gamma : n \multimap u$}
\RightLabel{$\circ_R$}
\BinaryInfC{$\eta : u \multimap m; \rho : n \multimap u \vdash \eta \circ (\rho \oplus \gamma) : n \multimap m$}
\RightLabel{$\circ_L$}
\UnaryInfC{$\eta \circ \rho : n \multimap m \vdash \eta \circ (\rho \oplus \gamma) : n \multimap m$}
\AxiomC{}
\RightLabel{Ax}
\UnaryInfC{$\eta : u \multimap m \vdash \eta : u \multimap m$}
\AxiomC{}
\RightLabel{Ax}
\UnaryInfC{$\gamma : n \multimap u \vdash \gamma : n \multimap u$}
\RightLabel{$\oplus_R$}
\UnaryInfC{$\gamma : n \multimap u \vdash \rho \oplus \gamma : n \multimap u$}
\RightLabel{$\circ_L$}
\BinaryInfC{$\eta : u \multimap m; \gamma : n \multimap u \vdash \eta \circ (\rho \oplus \gamma) : n \multimap m$}
\RightLabel{$\circ_L$}
\UnaryInfC{$\eta \circ \gamma : n \multimap m \vdash \eta \circ (\rho \oplus \gamma) : n \multimap m$}
\RightLabel{$\oplus_L$}
\BinaryInfC{$(\eta \circ \rho) \oplus (\eta \circ \gamma) : n \multimap m \vdash \eta \circ (\rho \oplus \gamma) : n \multimap m$}
\end{prooftree}
$$

□

5 Conclusion

We have shown how an approach to no-cloning and no-deleting in quantum computing can be carried out in a linear logical framework, allowing to define a categorical semantics associated to vector spaces of matrices with Kronecker product. This opens two main perspectives.

From a semantic point of view, the present work indicates how to make linear algebra in terms of strict monoidal categories and, consequently, opens the possibility to a categorical perspective to quantum computing with some advantages relatively to the Abramsky and Coecke approach. Notably, we avoid the pitfalls of defining dagger categories [19] and do not need indirect representations of quantum processes using string diagrams [2]. The categorical semantics presented in this article has as morphism the matrices actually used in quantum computing, and so the categorical abstractions can be used to understand the patterns in quantum processes, they are not used to represent them. Quantum computing will require, however, the ability to represent the matrix operations on their entries, not only on their types as made in the present article. This leads us to the second perspective.

From a syntactic point of view, the sequent calculus displayed shows that a linear logical system with a mixture of additive and multiplicative rules can represent vector spaces. We find this an amazing phenomenon, because since Girard's work [10] additivity is associated to intuitionistic logic and multiplicativity to classical logic, but here they are put together to make linear algebra with logic. Moreover, we can extend this sequent calculus via the introduction of term reduction rules similar to the one used in quantum lambda calculus (Cf. [7, 18]). In that way, it will be possible to use the very concrete methods from automated reasoning to perform matrix operations, but running over a semantics with the power of abstraction from category theory.

6 Acknowledgement

The author is thankful to the comments of the anonymous referees, from which the following are highlighted. The first referee asked for a more clear presentation of the category of matrices, the second one requested more details about the proof of categorical representability, and the third called for the relation between the proposed sequent calculus and additive vs multiplicative linear logics. Further improvements will be made until the version for publication, but hopefully the present one is now better than the initial version and so it may generate some interesting discussion at TLLA-Linearity.

References

[1] S. Abramsky. No-Cloning in categorical quantum mechanics. In: I. Mackie and S. Gay (eds): Semantic Techniques for Quantum Computation, Cambridge University Press, 2008.

[2] S. Abramsky; B. Coecke. A categorical semantics of quantum protocols. In: Proceedings of 19th IEEE conference on Logic in Computer Science, pages 415-425. IEEE Press, 2004

[3] S. Axler. *Linear Algebra Done Right*. London: Springer, 2015.

[4] J. Baez; M. Stay. Physics, Topology, Logic and Computation: A Rosetta Stone. New Structures for Physics. Berlin: Springer. pp. 95-172, 2010.

[5] M. Barr. $*$-Autonomous Categories and linear logic. Mathematical Structures in Computer Science, Vol. 1(02), pp 159-178, 1991.

[6] Coecke, Bob (2009). Quantum Picturalism. Contemporary Physics. 51: 59-83.

[7] U. Dal Lago, A. Masini, M. Zorzi (2009): On a Measurement-Free Quantum Lambda Calculus with Classical Control. Math. Structures Comput. Sci. 19(2), pp. 297-335.

[8] D. G. B. J. Dieks. Communication by EPR devices. Physics Letters A 92, 271-272, 1982.

[9] B. Fong, D. Spivak, R. Tuyéras. Backprop as Functor: A compositional perspective on supervised learning. *34th Symposium on Logic in Computer Science (LICS)*: 2019.

[10] J.-Y. Girard. Linear logic. Theoretical Computer Science 50: 1-102, 1987.

[11] B. Jacobs, D. Sprunger. Neural Nets via Forward State Transformation and Backward Loss Transformation. *Electronic Notes in Theoretical Computer Science* 347 (2019) 161-177.

[12] J. Lambek and P. J. Scott. Introduction to Higher-Order Categorical Logic. Cambridge University Press, 1986.

[13] H.D. Macedo; J.N. Oliveira. Typing linear algebra: A biproduct-oriented approach. Science of Computer Programming 78.11 (2013): 2160-2191.

[14] S. MacLane. Categories for the Working Mathematician. Graduate Texts in Mathematics. Vol. 5 (Second ed.): Springer, 1998.

[15] S. Negri and J. Von Plato. *Structural Proof Theory*. Cambridge: Cambridge University Press, 2001.

[16] M. A. Nielsen; I.L. Chuang: Quantum Computation and Quantum Information, Cambridge University Press, 2000.

[17] A. K. Pati and S. L. Braunstein. Impossibility of deleting an unknown quantum state. Nature 404, 164-165, 2000.

[18] P. Selinger, B. Valiron (2006): A lambda calculus for quantum computation with classical control. Math. Structures Comput. Sci. 16(3), pp. 527-552.

[19] P. Selinger. Dagger compact categories and completely positive maps. Electronic Notes in Theoretical Computer Science 170, 139-163, 2007.

[20] W. K. Wootters and W. Zurek. A single quantum cannot be cloned. Nature 299, 802-803, 1982.