Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in Theoretical Computer Science 344 (2019) 83–100

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

A Concrete Categorical Semantics of Lambda-$\mathcal{S}$

Alejandro Díaz-Caro[a,b,1,3] Octavio Malherbe[c,d,2,4]

[a] *Universidad Nacional de Quilmes, Bernal, Buenos Aires, Argentina*

[b] *Instituto de Ciencias de la Computación (CONICET-Universidad de Buenos Aires), Buenos Aires, Argentina*

[c] *Departamento de Matemática y Afines, CURE, Universidad de la República, Maldonado, Uruguay*

[d] *IMERL, Facultad de Ingeniería, Universidad de la República, Montevideo, Uruguay*

Abstract

Lambda-$\mathcal{S}$ is an extension to first-order lambda calculus unifying two approaches of non-cloning in quantum lambda-calculi. One is to forbid duplication of variables, while the other is to consider all lambda-terms as algebraic linear functions. The type system of Lambda-$\mathcal{S}$ have a constructor S such that a type A is considered as the base of a vector space while $S(A)$ is its span. A first semantics of this calculus have been given when first presented, with such an interpretation: superposed types are interpreted as vectors spaces while non-superposed types as their basis. In this paper we give a concrete categorical semantics of Lambda-$\mathcal{S}$, showing that S is interpreted as the composition of two functors in an adjunction relation between the category of sets and the category of vector spaces over $\mathbb{C}$. The right adjoint is a forgetful functor U, which is hidden in the language, and plays a central role in the computational reasoning.

Keywords: Quantum computing, algebraic lambda-calculus, categorical semantics

1 Introduction

The non-cloning property of quantum computing has been treated in different ways in quantum programming languages. One way is to forbid duplication of variables with linear types [1, 12], and hence, a program taking a quantum argument will not duplicate it (e.g. [3, 14, 17, 20, 22]). Another way is to consider all lambda-terms as expressing linear functions (e.g. [4–6, 11]). The first approach forbids a term $\lambda x.(x \otimes x)$ (for some convenient definition of $\otimes$), while the second approach distributes $(\lambda x.(x \otimes x))(|0\rangle + |1\rangle)$ to $\lambda x.(x \otimes x)\,|0\rangle + \lambda x.(x \otimes x)\,|1\rangle$, mimicking the way that linear operations act on vectors in a vector space. However, adding a measurement operator to a calculus following the linear-algebraic approach need to also add linear

[1] Partially supported by PICT 2015 1208 and ECOS-Sud A17C03 QuCa.

[2] Partially supported by MIA CSIC UdelaR.

[3] Email: adiazcaro@icc.fcen.uba.ar

[4] Email: malherbe@fing.edu.uy

https://doi.org/10.1016/j.entcs.2019.07.006
1571-0661/© 2019 The Author(s). Published by Elsevier B.V.

types: indeed, if π represent a measurement operator, $(\lambda x.\pi x)(|0\rangle + |1\rangle)$ should not reduce to $(\lambda x.\pi x)\,|0\rangle + (\lambda x.\pi x)\,|1\rangle$ but to $\pi(|0\rangle + |1\rangle)$. Therefore, the functions taking a superposition have to be marked in some way and ensure that they will not use their arguments more than once (i.e. ensure linearity in the linear-logic sense).

The calculus Lambda-$\mathcal{S}$ has been introduced in [9] and slightly modified later in [18], as a first-order fragment of Lineal [6], extended with measurements. In linear logic we would write A the types of terms that cannot be duplicated while $!A$ types duplicable terms. In Lambda-$\mathcal{S}$ instead A are the types of the terms that cannot be superposed, while $S(A)$ are the terms that can be superposed, and since superposition forbids duplication, A means that we can duplicate, while $S(A)$ means that we cannot duplicate. So the S is not the same as the bang "!", but somehow the opposite. This can be explained by the fact that linear logic is focused on the possibility of duplication, while here we focus on the possibility of superposition, which implies the impossibility of duplication.

In [9] a first denotational semantics (in environment style) is given where the type $\mathbb{B}$ is interpreted as $\{|0\rangle, |1\rangle\}$ while $S(\mathbb{B})$ is interpreted as $\mathsf{Span}(\{|0\rangle, |1\rangle\}) = \mathbb{C}^2$, and, in general, a type A is interpreted as a basis while $S(A)$ is the vector space generated by such a basis. In this paper we go beyond and give a categorical interpretation of Lambda-$\mathcal{S}$ where S is a functor of an adjunction between the category **Set** and the category **Vec**. Explicitly, when we evaluate S we obtain formal finite linear combinations of elements of a set with complex numbers as coefficients and the other functor of the adjunction, U, allows us to forget the vectorial structure.

The main structural feature of our model is that it is expressive enough to describe the bridge between the quantum and the classical universes explicitly by controlling its interaction. This is achieved by providing a monoidal adjunction. In the literature, intuitionistic linear (as in linear-logic) models are obtained by a comonad determined by a monoidal adjunction $(S, m) \dashv (U, n)$, i.e. the bang ! is interpreted by the comonad SU (see [8]). In a different way, a crucial ingredient of our model is to consider the monad US for the interpretation of S determined by a similar monoidal adjunction. This implies that on the one hand we have a tight control of the Cartesian structure of the model (i.e. duplication, etc) and on the other hand the world of superpositions lives in some sense inside the classical world, i.e. determined externally by classical rules until we decide to explore it. This is given by the following composition of maps:

$$US(\mathbb{B}) \times US(\mathbb{B}) \xrightarrow{n} U(S(\mathbb{B}) \otimes S(\mathbb{B})) \xrightarrow{U(m)} US(\mathbb{B} \times \mathbb{B})$$

that allows us to operate in a monoidal structure representing the quantum world and then to return to the Cartesian product.

This is different from linear logic, where the classical world lives in some sense inside the quantum world i.e. $(!\mathbb{B}) \otimes (!\mathbb{B})$ is a product inside a monoidal category.

Another source of inspiration for our model has been the work of Selinger [19] and Abramsky and Coecke [2] where they captured the notion of scalars and inner product in a more abstract categorical setting, i.e. a category in which there is an abstract notion of a dagger functor. It is envisaged that this approach will provide the basis for an abstract model in future work.

The paper is structured as follows. In Section 2 we recall the definition of Lambda-$\mathcal{S}$ and give some examples, stating its main properties. Section 3 is divided in three subsections: first we define the categorical constructions needed to interpret the calculus, then we give the interpretation, and finally we prove its soundness and adequacy properties. Finally, we conclude in Section 4. An appendix with detailed proofs can be found in the pre-print in arXiv[5].

2 The calculus Lambda-$\mathcal{S}$

We give a slightly modified presentation of Lambda-$\mathcal{S}$, based on [18]. In particular, instead of giving a probabilistic rewrite system where $t \to_{p_k} r_k$ means that t reduces with probability p_k to r_k, we introduce the notation $t \longrightarrow p_1 r_1 \parallel \cdots \parallel p_n r_n$, this way, the rewrite system is deterministic and the probabilistic distribution is internalized in the syntax.

The syntax of terms and types is given in Figure 2. We write $\mathbb{B}^n$ for $\mathbb{B}\times\cdots\times\mathbb{B}$ n-times, with the convention that $\mathbb{B}^1 = \mathbb{B}$, and may write $\overset{n}{\underset{i=1}{\parallel}} p_i t_i$, for $p_1 t_1 \parallel \cdots \parallel p_n t_n$ with the convention that $\overset{1}{\underset{i=1}{\parallel}} 1t = t$. We use capital Latin letters $(A, B, C, \dots)$ for general types and the capital Greek letters Ψ, Φ, Ξ, and Υ for qubit types. $\mathcal{B} = \{\mathbb{B}^n \mid n \in \mathbb{N}\}$, $\mathcal{Q}$ is the set of qubit types, and $\mathcal{T}$ is the set of types ($\mathcal{B} \subsetneq \mathcal{Q} \subsetneq \mathcal{T}$). In the same way, Vars is the set of variables, B is the set of basis terms, V the set of values, Λ the set of terms, and D the set of probabilistic distributions on terms. We have $\mathsf{Vars} \subsetneq \mathsf{B} \subsetneq \mathsf{V} \subsetneq \Lambda \subsetneq \mathsf{D}$.

$$\begin{array}{rlr}
\Psi & := \mathbb{B}^n \mid S(\Psi) \mid \Psi \times \Psi & \text{Qubit types } (\mathcal{Q})\\
A & := \Psi \mid \Psi \Rightarrow A \mid S(A) & \text{Types } (\mathcal{T})\\
\\
b & := x \mid \lambda x{:}\Psi.t \mid |0\rangle \mid |1\rangle \mid b \times b & \text{Basis terms } (\mathsf{B})\\
v & := b \mid (v+v) \mid \mathbf{0}_{S(A)} \mid \alpha.v \mid v \times v & \text{Values } (\mathsf{V})\\
t & := v \mid tt \mid (t+t) \mid \pi_j t \mid ?t{\cdot}t \mid \alpha.t \mid t \times t \mid \mathit{head}\ t \mid \mathit{tail}\ t \mid \Uparrow_r t \mid \Uparrow_\ell t & \text{Terms } (\Lambda)\\
p & := p_1 t_1 \parallel \cdots \parallel p_n t_n & \text{Probabilistic distribution } (\mathsf{D})
\end{array}$$

$$\text{where } \alpha \in \mathbb{C} \text{ and } p_i \in [0,1] \subseteq \mathbb{R}.$$

Fig. 1: Syntax of types and terms of Lambda-$\mathcal{S}$.

The terms are considered modulo associativity and commutativity of the syntactic symbol $+$. On the other hand, the symbol $\parallel$ is used to represent a real probabilistic distribution of terms, not as a syntactic symbol, and so, it is not only associative and commutative, we also have that $pt \parallel qt$ is the same as $(p+q)t$ and $pt \parallel 0r = pt$[6].

[5] http://arxiv.org/abs/1806.09236.

[6] As a remark, notice that $\parallel$ can be seen as the $+$ symbol of the algebraic lambda calculus [], where the equality is confluent since scalars are positive, while our $+$ symbol coincides with the $+$ from Lineal [6]

There is one atomic type $\mathbb{B}$, for basis qubits $|0\rangle$ and $|1\rangle$, and three constructors: $\times$, for pairs, $\Rightarrow$, for first-order functions, and $S(\cdot)$ for superpositions.

The syntax of terms contains:

- The three basic terms for first-order lambda-calculus, namely, variables, abstractions and applications.
- Two basic terms $|0\rangle$ and $|1\rangle$ to represent qubits, and one test $?r{\cdot}s$ on them. We may write $t?r{\cdot}s$ for $(?r{\cdot}s)t$, see Example 2.1 for a clarification of why to choose this presentation.
- A product $\times$ to represent associative pairs (i.e. lists), with its destructors *head* and *tail*. We may use the notation $|b_1b_2\dots b_n\rangle$ for $|b_1\rangle \times |b_2\rangle \times \dots \times |b_n\rangle$.
- Constructors to write linear combinations of terms, namely $+$ (sum) and . (scalar multiplication), and its destructor π_j measuring the first j qubits written as linear combinations of lists of qubits, and one null vector $\mathbf{0}_{S(A)}$ for each type $S(A)$.
- Two casting functions $\Uparrow_r$ and $\Uparrow_\ell$ which allows us to consider lists of superpositions as superpositions of lists (see Example 2.2).

The rewrite system has not been given yet, however the next examples give some intuitions and clarify the $?r{\cdot}s$ and the casting functions.

Example 2.1 The term $?r{\cdot}s$ is meant to test whether the condition is $|1\rangle$ or $|0\rangle$. However, defining it as a function, allows us to use the algebraic linearity to implement the quantum-if [3]:

$$(?r{\cdot}s)(\alpha.\,|1\rangle + \beta.\,|0\rangle) = (\alpha.\,|1\rangle + \beta.\,|0\rangle)?r{\cdot}s \longrightarrow^* \alpha.|1\rangle?r{\cdot}s + \beta.|0\rangle?r{\cdot}s \longrightarrow^* \alpha.r + \beta.s$$

Example 2.2 The term $(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)) \times |0\rangle$ is the encoding of the qubit $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\otimes|0\rangle$. However, while the qubit $\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)\otimes|0\rangle$ is equal to $\frac{1}{\sqrt{2}}(|0\rangle\otimes|0\rangle+|1\rangle\otimes|0\rangle)$, the term will not rewrite to the encoding of it, unless a casting $\Uparrow_r$ is preceding the term:

$$\Uparrow_r (\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)) \times |0\rangle \longrightarrow^* \frac{1}{\sqrt{2}}(|0\rangle \times |0\rangle + |1\rangle \times |0\rangle)$$

The reason is that we want the term $(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)) \times |0\rangle$ to have type $S(\mathbb{B}) \times \mathbb{B}$, highlighting the fact that the second qubit is a basis qubit, i.e. duplicable, while the term $\frac{1}{\sqrt{2}}(|0\rangle \times |0\rangle + |1\rangle \times |0\rangle)$ will have type $S(\mathbb{B} \times \mathbb{B})$, showing that the full term is a superposition where no information can be extracted and hence, non-duplicable.

The rewrite system depends on types. Indeed, $\lambda x{:}S(\Psi).t$ follows a call-by-name strategy, while $\lambda x{:}\mathbb{B}.t$, which can duplicate its argument, must follow a call-by-base strategy [7], that is, not only the argument must be reduced first, but also it will distribute over linear combinations. Therefore, we give first the type system and then the rewrite system.

The typing relation is given in Figure 2. Contexts, identified by the capital Greek letters Γ, Δ, and Θ, are partial functions from Vars to $\mathcal{T}$. The contexts assigning

(see [7] for a more detailed discussion on different presentations of algebraic lambda calculi).

only types in $\mathcal{B}$ are identified with the super-index $\mathbb{B}$, e.g. $\Theta^{\mathbb{B}}$. Whenever more than one context appear in a typing rule, their domains are considered pair-wise disjoint. Observe that all types are linear (as in linear-logic) except on basis types $\mathbb{B}^n$, which can be weakened and contracted (expressed by the common contexts $\Theta^{\mathbb{B}}$).

$$\frac{}{\Theta^{\mathbb{B}}, x:\Psi \vdash x:\Psi}\ {\scriptstyle Ax} \qquad \frac{}{\Theta^{\mathbb{B}} \vdash \mathbf{0}_{S(A)} : S(A)}\ {\scriptstyle Ax_{\mathbf{0}}} \qquad \frac{}{\Theta^{\mathbb{B}} \vdash |0\rangle : \mathbb{B}}\ {\scriptstyle Ax_{|0\rangle}} \qquad \frac{}{\Theta^{\mathbb{B}} \vdash |1\rangle : \mathbb{B}}\ {\scriptstyle Ax_{|1\rangle}}$$

$$\frac{\Gamma \vdash t : S(A)}{\Gamma \vdash \alpha.t : S(A)}\ {\scriptstyle \alpha_I} \quad \frac{\Gamma, \Theta^{\mathbb{B}} \vdash t : S(A) \quad \Delta, \Theta^{\mathbb{B}} \vdash u : S(A)}{\Gamma, \Delta, \Theta^{\mathbb{B}} \vdash (t+u) : S(A)}\ {\scriptstyle +_I} \quad \frac{\Gamma \vdash t : A}{\Gamma \vdash t : S(A)}\ {\scriptstyle S_I} \quad \frac{\Gamma \vdash t : S^k(\mathbb{B}^n) \quad {\scriptstyle k>0}}{\Gamma \vdash \pi_j t : \mathbb{B}^j \times S(\mathbb{B}^{n-j})}\ {\scriptstyle S_E}$$

$$\frac{\Gamma \vdash t : A \quad \Gamma \vdash r : A}{\Gamma \vdash {?t{\cdot}r} : \mathbb{B} \Rightarrow A}\ {\scriptstyle If} \qquad \frac{\Gamma, x:\Psi \vdash t : A}{\Gamma \vdash \lambda x{:}\Psi.t : \Psi \Rightarrow A}\ {\scriptstyle \Rightarrow_I}$$

$$\frac{\Delta, \Theta^{\mathbb{B}} \vdash u : \Psi \quad \Gamma, \Theta^{\mathbb{B}} \vdash t : \Psi \Rightarrow A}{\Delta, \Gamma, \Theta^{\mathbb{B}} \vdash tu : A}\ {\scriptstyle \Rightarrow_E} \qquad \frac{\Delta, \Theta^{\mathbb{B}} \vdash u : S(\Psi) \quad \Gamma, \Theta^{\mathbb{B}} \vdash t : S(\Psi \Rightarrow A)}{\Delta, \Gamma, \Theta^{\mathbb{B}} \vdash tu : S(A)}\ {\scriptstyle \Rightarrow_{ES}}$$

$$\frac{\Gamma, \Theta^{\mathbb{B}} \vdash t : \Psi \quad \Delta, \Theta^{\mathbb{B}} \vdash u : \Phi}{\Gamma, \Delta, \Theta^{\mathbb{B}} \vdash t \times u : \Psi \times \Phi}\ {\scriptstyle \times_I} \qquad \frac{\Gamma \vdash t : \mathbb{B}^n \quad {\scriptstyle n>1}}{\Gamma \vdash \mathit{head}\ t : \mathbb{B}}\ {\scriptstyle \times_{Er}} \qquad \frac{\Gamma \vdash t : \mathbb{B}^n \quad {\scriptstyle n>1}}{\Gamma \vdash \mathit{tail}\ t : \mathbb{B}^{n-1}}\ {\scriptstyle \times_{El}}$$

$$\frac{\Gamma \vdash t : S(S(\Psi) \times \Phi)}{\Gamma \vdash \Uparrow_r t : S(\Psi \times \Phi)}\ {\scriptstyle \Uparrow_r} \qquad \frac{\Gamma \vdash t : S(\Psi \times S(\Phi))}{\Gamma \vdash \Uparrow_\ell t : S(\Psi \times \Phi)}\ {\scriptstyle \Uparrow_\ell} \qquad \frac{\Gamma \vdash t_i : A \quad \sum_i p_i = 1}{\Gamma \vdash p_1 t_1 \parallel \cdots \parallel p_n t_n : A}\ {\scriptstyle \parallel}$$

Fig. 2: Typing relation

The rewrite relation is given in Figures 3 to 10.

The two beta rules (Figure 3) are applied according to the type of the argument. If the abstraction expects an argument with a superposed type, then the reduction follows a call-by-name strategy (rule (β_{n})), while if the abstraction expects a basis type, the reduction is call-by-base (rule (β_{b})): it β-reduces only when its argument is a basis term. However, typing rules also allow to type an abstraction expecting an argument with basis type, applied to a term with superposed type (cf. Example 2.3). In this case, the beta reduction cannot occur and, instead, the application must distribute using the rules from Figure 4: the linear distribution rules.

Figure 5 gives the two rules for the conditional construction. Together with the linear distribution rules (cf. Figure 4), these rules implement the quantum-if

$$\text{If } b \text{ has type } \mathbb{B}^n \text{ and } b \in \mathsf{B},\ (\lambda x{:}\mathbb{B}^n.t)b \longrightarrow (b/x)t \quad (\beta_{\mathsf{b}})$$
$$\text{If } u \text{ has type } S(\Psi),\ (\lambda x{:}S(\Psi).t)u \longrightarrow (u/x)t \quad (\beta_{\mathsf{n}})$$

Fig. 3: Beta rules

$$\begin{aligned}
\text{If } t \text{ has type } \mathbb{B}^n \Rightarrow A,\ t(u+v) &\longrightarrow (tu+tv) && (\mathsf{lin}_{\mathsf{r}}^{+})\\
\text{If } t \text{ has type } \mathbb{B}^n \Rightarrow A,\ t(\alpha.u) &\longrightarrow \alpha.tu && (\mathsf{lin}_{\mathsf{r}}^{\alpha})\\
\text{If } t \text{ has type } \mathbb{B}^n \Rightarrow A,\ t\mathbf{0}_{S(\mathbb{B}^n)} &\longrightarrow \mathbf{0}_{S(A)} && (\mathsf{lin}_{\mathsf{r}}^{0})\\
(t+u)v &\longrightarrow (tv+uv) && (\mathsf{lin}_{\mathsf{l}}^{+})\\
(\alpha.t)u &\longrightarrow \alpha.tu && (\mathsf{lin}_{\mathsf{l}}^{\alpha})\\
\mathbf{0}_{S(\mathbb{B}^n \Rightarrow A)}t &\longrightarrow \mathbf{0}_{S(A)} && (\mathsf{lin}_{\mathsf{l}}^{0})
\end{aligned}$$

Fig. 4: Linear distribution rules

$$|1\rangle ?t{\cdot}r \longrightarrow t \quad (\mathsf{if}_1) \qquad\qquad |0\rangle ?t{\cdot}r \longrightarrow r \quad (\mathsf{if}_0)$$

Fig. 5: Rules of the conditional construction

$$\begin{array}{rr} \text{If } h \neq u \times v \text{ and } h \in \mathsf{B},\ \mathit{head}\ h \times t \longrightarrow h & (\mathsf{head}) \\ \text{If } h \neq u \times v \text{ and } h \in \mathsf{B},\ \mathit{tail}\ h \times t \longrightarrow t & (\mathsf{tail}) \end{array}$$

Fig. 6: Rules for lists

$$\begin{array}{rcll} (\mathbf{0}_{S(A)} + t) & \longrightarrow & t & (\mathsf{neutral}) \\ 1.t & \longrightarrow & t & (\mathsf{unit}) \\ \text{If } t \text{ has type } A,\ 0.t & \longrightarrow & \mathbf{0}_{S(A)} & (\mathsf{zero}_\alpha) \\ \alpha.\mathbf{0}_{S(A)} & \longrightarrow & \mathbf{0}_{S(A)} & (\mathsf{zero}) \\ \alpha.(\beta.t) & \longrightarrow & (\alpha\beta).t & (\mathsf{prod}) \\ \alpha.(t+u) & \longrightarrow & (\alpha.t + \alpha.u) & (\alpha\mathsf{dist}) \\ (\alpha.t + \beta.t) & \longrightarrow & (\alpha+\beta).t & (\mathsf{fact}) \\ (\alpha.t + t) & \longrightarrow & (\alpha+1).t & (\mathsf{fact}^1) \\ (t+t) & \longrightarrow & 2.t & (\mathsf{fact}^2) \end{array}$$

Fig. 7: Rules implementing the vector space axioms

(cf. Example 2.1).

Figure 6 gives the rules for lists, (head) and (tail).

Figure 7 deals with the vector space structure implementing a directed version of the vector space axioms. The direction is chosen in order to yield a canonical form [6].

Figure 8 are the rules to implement the castings. The idea is that $\times$ does not distribute with respect to $+$, unless a casting allows such a distribution. This way, the types $\mathbb{B} \times S(\mathbb{B})$ and $S(\mathbb{B} \times \mathbb{B})$ are different. Indeed, $|0\rangle \times (|0\rangle + |1\rangle)$ have the first type but not the second, while $|0\rangle \times |0\rangle + |0\rangle \times |1\rangle$ have the second type but not the first. This way, the first type give us the information that the state is separable, while the second type do not. We can choose to take the first state as a pair of qubits forgetting the separability information, by casting its type, in the same way as in certain programming languages an integer can be casted to a float (and so, forgetting the information that it was indeed an integer and not any float).

Figure 9 gives the rule for the projective measurement with respect to the basis $\{|0\rangle, |1\rangle\}$. In this rule, we use the following notations:

$[\alpha.]t$ may be either t or $\alpha.t$

$j \leq m$

$|k\rangle = |b_1\rangle \times \cdots \times |b_j\rangle$ where $b_1 \ldots b_j$ is the binary representation of k

$$\begin{array}{rll}
\Uparrow_r (r+s)\times u \longrightarrow (\Uparrow_r r\times u + \Uparrow_r s\times u) & & (\mathsf{dist}^+_\mathsf{r})\\
\Uparrow_\ell u\times(r+s) \longrightarrow (\Uparrow_\ell u\times r + \Uparrow_\ell u\times s) & & (\mathsf{dist}^+_\mathsf{l})\\
\Uparrow_r (\alpha.r)\times u \longrightarrow \alpha.\Uparrow_r r\times u & & (\mathsf{dist}^\alpha_\mathsf{r})\\
\Uparrow_\ell u\times(\alpha.r) \longrightarrow \alpha.\Uparrow_r u\times r & & (\mathsf{dist}^\alpha_\mathsf{l})\\
\text{If } u \text{ has type } \Psi,\ \Uparrow_r \mathbf{0}_{S(\Phi)}\times u \longrightarrow \mathbf{0}_{S(\Phi\times\Psi)} & & (\mathsf{dist}^0_\mathsf{r})\\
\text{If } u \text{ has type } \Psi,\ \Uparrow_\ell u\times\mathbf{0}_{S(\Phi)} \longrightarrow \mathbf{0}_{S(\Psi\times\Phi)} & & (\mathsf{dist}^0_\mathsf{l})\\
\Uparrow (t+u) \longrightarrow (\Uparrow t + \Uparrow u) & & (\mathsf{dist}^+_\Uparrow)\\
\Uparrow (\alpha.t) \longrightarrow \alpha.\Uparrow t & & (\mathsf{dist}^\alpha_\Uparrow)\\
\Uparrow_r \mathbf{0}_{S(S(S(\Psi))\times\Phi)} \longrightarrow \Uparrow_r \mathbf{0}_{S(S(\Psi)\times\Phi)} & & (\mathsf{dist}^0_{\Uparrow_\mathsf{r}})\\
\Uparrow_r \mathbf{0}_{S(S(\mathbb{B}^n)\times\Phi)} \longrightarrow \mathbf{0}_{S(\mathbb{B}^n\times\Phi)} & & (\mathsf{neut}^\Uparrow_{0\mathsf{r}})\\
\Uparrow_\ell \mathbf{0}_{S(\Psi\times S(S(\Phi)))} \longrightarrow \Uparrow_\ell \mathbf{0}_{S(\Psi\times S(\Phi))} & & (\mathsf{dist}^0_{\Uparrow_\ell})\\
\Uparrow_\ell \mathbf{0}_{S(\Psi\times S(\mathbb{B}^n))} \longrightarrow \mathbf{0}_{S(\Psi\times\mathbb{B}^n)} & & (\mathsf{neut}^\Uparrow_{0\ell})\\
\text{If } u\in \mathsf{B},\ \Uparrow_r u\times v \longrightarrow u\times v & & (\mathsf{neut}^\Uparrow_\mathsf{r})\\
\text{If } v\in \mathsf{B},\ \Uparrow_\ell u\times v \longrightarrow u\times v & & (\mathsf{neut}^\Uparrow_\mathsf{l})
\end{array}$$

Fig. 8: Rules for castings $\Uparrow_r$ and $\Uparrow_\ell$

$$\pi_j(\sum_{i=1}^{n}[\alpha_i.]\prod_{h=1}^{m}|b_{hi}\rangle) \longrightarrow \mathop{\Big|\Big|}_{k=0}^{2^j-1} p_k(|k\rangle\times|\phi_k\rangle) \quad (\mathsf{proj})$$

Fig. 9: Rule for the projection

$$|\phi_k\rangle = \sum_{i\in T_k}\left(\frac{\alpha_i}{\sqrt{\sum_{r\in T_k}|\alpha_r|^2}}\right)\prod_{h=j+1}^{m}|b_{hi}\rangle$$
$$p_k = \sum_{i\in T_k}\left(\frac{|\alpha_i|^2}{\sum_{r=1}^{n}|\alpha_r|^2}\right)$$
$$T_k = \{i\le n \mid |b_{1i}\rangle\times\cdots\times|b_{ji}\rangle = |k\rangle\}$$

This way, $p_k\,|k\rangle\times|\phi_k\rangle$ is the normalized k-th projection of the term.

Finally, Figure 10 give the contextual rules implementing the call-by-value and call-by-name strategies.

Example 2.3 The term $\lambda x{:}\mathbb{B}.x\times x$ does not represent a cloning machine, but a CNOT with an ancillary qubit $|0\rangle$. Indeed, $(\lambda x{:}\mathbb{B}.x\times x)\frac{1}{\sqrt{2}}.(|0\rangle+|1\rangle) \xrightarrow{(\mathsf{lin}^\alpha_\mathsf{r})} \frac{1}{\sqrt{2}}.(\lambda x{:}\mathbb{B}.x\times x)(|0\rangle+|1\rangle)$

$$\xrightarrow{(\mathsf{lin}^+_\mathsf{r})} \tfrac{1}{\sqrt{2}}.((\lambda x{:}\mathbb{B}.x\times x)\,|0\rangle + (\lambda x{:}\mathbb{B}.x\times x)\,|1\rangle)$$
$$\xrightarrow{\beta_b} \tfrac{1}{\sqrt{2}}.(|0\rangle\times|0\rangle + (\lambda x{:}\mathbb{B}.x\times x)\,|1\rangle)$$
$$\xrightarrow{\beta_b} \tfrac{1}{\sqrt{2}}.(|0\rangle\times|0\rangle + |1\rangle\times|1\rangle)$$

If $t \longrightarrow u$, then

$$\begin{array}{ccc}
tv \longrightarrow uv & (\lambda x^{\mathbb{B}^n}.v)t \longrightarrow (\lambda x^{\mathbb{B}^n}.v)u & (t+v) \longrightarrow (u+v) \\
\alpha.t \longrightarrow \alpha.u & \pi_j t \longrightarrow \pi_j u & t \times v \longrightarrow u \times v \\
v \times t \longrightarrow v \times u & \Uparrow_r t \longrightarrow \Uparrow_r u & \Uparrow_\ell t \longrightarrow \Uparrow_\ell u \\
\mathit{head}\ t \longrightarrow \mathit{head}\ u & \mathit{tail}\ t \longrightarrow \mathit{tail}\ u & t?r{\cdot}s \longrightarrow u?r{\cdot}s
\end{array}$$

$$(p_1 t_1 \parallel \cdots \parallel p_k t \parallel \cdots \parallel p_n t_n) \longrightarrow (p_1 t_1 \parallel \cdots \parallel p_k u \parallel \cdots \parallel p_n t_n)$$

Fig. 10: Contextual rules

The type derivation is as follows:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{x:\mathbb{B}\vdash x:\mathbb{B}}\,{\scriptstyle Ax} \quad \dfrac{}{x:\mathbb{B}\vdash x:\mathbb{B}}\,{\scriptstyle Ax}}{x:\mathbb{B}\vdash x\times x:\mathbb{B}^2}\,{\scriptstyle \times_I}}{\vdash \lambda x{:}\mathbb{B}.x\times x : \mathbb{B}\Rightarrow\mathbb{B}^2}\,{\scriptstyle \Rightarrow_I}}{\vdash \lambda x{:}\mathbb{B}.x\times x : S(\mathbb{B}\Rightarrow\mathbb{B}^2)}\,{\scriptstyle S_I} \quad \dfrac{\dfrac{\dfrac{\dfrac{}{\vdash |0\rangle:\mathbb{B}}\,{\scriptstyle Ax_{|0\rangle}}}{\vdash |0\rangle:S(\mathbb{B})}\,{\scriptstyle S_I} \quad \dfrac{\dfrac{}{\vdash |1\rangle:\mathbb{B}}\,{\scriptstyle Ax_{|1\rangle}}}{\vdash |1\rangle:S(\mathbb{B})}\,{\scriptstyle S_I}}{\vdash |0\rangle+|1\rangle:S(\mathbb{B})}\,{\scriptstyle +_I}}{\vdash \frac{1}{\sqrt{2}}.(|0\rangle+|1\rangle):S(\mathbb{B})}\,{\scriptstyle \alpha_I}}{\vdash (\lambda x{:}\mathbb{B}.x\times x)\frac{1}{\sqrt{2}}.(|0\rangle+|1\rangle):S(\mathbb{B}^2)}\,{\scriptstyle \Rightarrow_{ES}}$$

Example 2.4 The term π_2 measures the first two qubits of its argument (in Example 3.7 we give a more detailed explanation of its reduction):

$$\pi_2(|001\rangle + 2.\,|110\rangle + 3.\,|000\rangle) \xrightarrow{(\text{proj})} \tfrac{10}{14}\,|00\rangle \times (\tfrac{1}{\sqrt{10}}.\,|1\rangle + \tfrac{3}{\sqrt{10}}.\,|0\rangle) \parallel \tfrac{4}{14}\,|11\rangle \times (1.\,|0\rangle)$$

The typing derivation is the following:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash |0\rangle:\mathbb{B}} \quad \dfrac{}{\vdash |0\rangle:\mathbb{B}} \quad \dfrac{}{\vdash |1\rangle:\mathbb{B}}}{\vdash |001\rangle:\mathbb{B}^3}\,{\scriptstyle \times_I}}{\vdash |001\rangle:S(\mathbb{B}^3)}\,{\scriptstyle \alpha_I} \quad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash |1\rangle:\mathbb{B}} \quad \dfrac{}{\vdash |1\rangle:\mathbb{B}} \quad \dfrac{}{\vdash |0\rangle:\mathbb{B}}}{\vdash |110\rangle:\mathbb{B}^3}\,{\scriptstyle \times_I}}{\vdash |110\rangle:S(\mathbb{B}^3)}\,{\scriptstyle S_I}}{\vdash 2.\,|110\rangle:S(\mathbb{B}^3)}\,{\scriptstyle \alpha_I} \quad \dfrac{\dfrac{\dfrac{\dfrac{}{\vdash |0\rangle:\mathbb{B}} \quad \dfrac{}{\vdash |0\rangle:\mathbb{B}} \quad \dfrac{}{\vdash |0\rangle:\mathbb{B}}}{\vdash |000\rangle:\mathbb{B}^3}\,{\scriptstyle \times_I}}{\vdash |000\rangle:S(\mathbb{B}^3)}\,{\scriptstyle S_I}}{\vdash 3.\,|000\rangle:S(\mathbb{B}^3)}\,{\scriptstyle \alpha_I}}{\vdash 2.\,|110\rangle+3.\,|001\rangle:S(\mathbb{B}^3)}\,{\scriptstyle +_I}}{\vdash |001\rangle+2.\,|110\rangle+3.\,|001\rangle:S(\mathbb{B}^3)}\,{\scriptstyle +_I}}{\vdash \pi_2(|001\rangle+2.\,|110\rangle+3.\,|001\rangle):\mathbb{B}^2\times S(\mathbb{B})}\,{\scriptstyle S_E}$$

Example 2.5 A Hadamard gate can be implemented by $H = \lambda x : \mathbb{B}.x?|-\rangle{\cdot}|+\rangle$, where $|+\rangle = \frac{1}{\sqrt{2}}.\,|0\rangle + \frac{1}{\sqrt{2}}.\,|1\rangle$ and $|-\rangle = \frac{1}{\sqrt{2}}.\,|0\rangle - \frac{1}{\sqrt{2}}.\,|1\rangle$. Therefore, $H : \mathbb{B} \Rightarrow S(\mathbb{B})$ and we have $H\,|0\rangle \longrightarrow^* |+\rangle$ and $H\,|1\rangle \longrightarrow^* |-\rangle$.

Correctness has been established in previous works for slightly different versions of Lambda-$\mathcal{S}$, except for the case of confluence, which have only been proved for Lineal. Lineal can be seen as an untyped fragment without several constructions (in particular, without π_j). The proof of confluence for Lambda-$\mathcal{S}$ is delayed to future work, using the development of probabilistic confluence from [10]. The proof of Subject Reduction and Strong Normalization are straightforward modifications from the proofs of the different presentations of Lambda-$\mathcal{S}$.

Theorem 2.6 (Confluence of Lineal, [6, Thm. 7.25]) *Lineal, an untyped fragment of Lambda-$\mathcal{S}$, is confluent.* □

Theorem 2.7 (Subject reduction on closed terms, [9, Thm. 2]) *For any closed terms t and u and type A, if $t \longrightarrow \|_i\, p_i u_i$ and $\vdash t : A$, then $\vdash \|_i\, p_i u_i : A$.* □

Theorem 2.8 (Strong normalization, [18, Thm. 5.16]) *If $\Gamma \vdash t : A$ then t is strongly normalizing.* □

3 Denotational semantics

Even though the semantic of this article is about particular categories i.e. the category of sets and the category of vector spaces, from the start our approach is categorical in an abstract way. The idea is that the concrete situation exposed in this article will pave the way to a more abstract formulation, and that is why we give the constructions as abstract as general as possible. A more general treatment, using a monoidal adjunction between a Cartesian closed category and a monoidal category with some extra conditions, remains a topic for future publication.

Definition 3.1 A concrete categorical model for Lambda-$\mathcal{S}$ is given by the following data:

- A monoidal adjunction

$$\begin{tikzcd}
(\mathbf{Vec}, \otimes, I) \arrow[d, bend left, "(U{,}n)"] \arrow[d, phantom, "\dashv"] \\
(\mathbf{Set}, \times, 1) \arrow[u, bend left, "(S{,}m)"]
\end{tikzcd}$$

where
· **Set** is the category of sets with 1 as a terminal object.
· **Vec** is the category of vector spaces over $\mathbb{C}$, in which $I = \mathbb{C}$.
· S is the functor such that for each set A, $S(A)$ is the vector space whose vectors are the formal finite linear combinations of the elements of A with coefficients in $\mathbb{C}$, and given a function $f : A \to B$ we define $S(f) : S(A) \to S(B)$ by evaluating f in A.
· U is the forgetful functor such that for each vector space V, $U(V)$ is the underlying set of vectors in V and for each linear map f, $U(f)$ forgets of its linear property.
· m is a natural isomorphism defined by

$$m_{AB} : S(A) \otimes S(B) \to S(A \times B)$$
$$(\sum_{a\in A} \alpha_a a) \otimes (\sum_{b\in B} \beta_b b) \mapsto \sum_{(a,b)\in A\times B} \alpha_a \beta_b (a,b)$$

· n is a natural transformation defined by $n_{AB} : U(V) \times U(W) \to U(V \otimes W)$ such that $(v, w) \mapsto v \otimes w$.

- There is a subcategory of **Vec** such that for every morphism $f : V \to W$ one associates a morphism $f^\dagger : W \to V$, called the *dagger* of f, such that for all

$f : V \to W$ and $g : W \to U$ we have

$$\mathsf{Id}_V^\dagger = \mathsf{Id}_V \qquad (g \circ f)^\dagger = f^\dagger \circ g^\dagger \qquad f^{\dagger\dagger} = f$$

- A Kleisli category defined with the following monad, called the distribution monad, $(D, \hat{\eta}, \hat{\mu})$:

$$D : \mathbf{Set} \to \mathbf{Set} \qquad D(A) = \{\sum_{i=1}^{n} p_i \chi_{a_i} \mid \sum_{i=1}^{n} p_i = 1, a_i \in A, n \in \mathbb{N}\}$$

where χ_a is the characteristic function of a, and $\hat{\eta}$ and $\hat{\mu}$ are defined as follows:

$$\begin{aligned} \hat{\eta} : A &\to D(A) & \hat{\mu} : D(D(A)) &\to D(A) \\ a &\mapsto 1\chi_a & \sum_{i=1}^{n} p_i \chi_{(\sum_{j=1}^{m_i} q_{ij}\chi_{a_{ij}})} &\mapsto \sum_{i=1}^{n}\sum_{j=1}^{m_i} p_i q_{ij} \chi_{a_{ij}} \end{aligned}$$

Remark 3.2

- For dealing with the probabilistic effect of the measurement our semantics requires the notion of a distribution monad (see [13, 16]). In order to give a more abstract categorical description we consider the Kleisli category given by this monad where a morphism $f : A \to B$ in the Kleisli category is really a morphism $f : A \to D(B)$ in the category Set and corresponds to a computation of type B.
- There exists an object $\mathbb{B}$ and maps i_1, i_2 in **Set** such that for every $t : 1 \longrightarrow A$ and $r : 1 \longrightarrow A$, here exists a unique map $[t, r]$ such that the following diagram commutes

$$\begin{tikzcd} 1 \arrow[r, "i_1"] \arrow[dr, "t"'] & \mathbb{B} \arrow[d, "{[t,r]}"] & 1 \arrow[l, "i_2"'] \arrow[dl, "r"] \\ & A & \end{tikzcd}$$

This object $\mathbb{B}$ is the Boolean set, and such a map will allow us to interpret the *if* construction (Definition 3.5).

- There exists a map $+ : US(A) \times US(A) \to US(A)$ in **Set**, given by $(a, b) \mapsto a + b$ in which we use the sum defined in $S(A)$.
- To have an adjunction means that each function $g : A \to U(V)$ extends to a unique linear transformation $f : S(A) \to V$, given explicitly by $f(\sum_i \alpha_i x_i) = \sum_i \alpha_i g(x_i)$, that is, formal linear combinations in $S(A)$ to actual linear combinations in V (see [15] for details).
- For every $A \in |\mathbf{Set}|$, $\mathbf{Vec}(I, S(A))$ is an abelian group with the sum defined point-wise.
- **Set** is a Cartesian closed category where η^A is the unit and ε^A is the counit of $- \times A \dashv [A, -]$, from which we can define the curryfication (curry) and uncurryfication ($\mathsf{uncurry}$) of any map.
- The adjunction in Definition 3.1 gives rise to a monad (T, η, μ) in the category **Set**, where $T = US$, $\eta : \mathsf{Id} \to T$ is the unit of the adjunction, and using the counit ε, we obtain $\mu = U\varepsilon S : TT \to T$, satisfying unity and associativity laws (see [15]).

Definition 3.3 Types are interpreted in the category **Set**, as follows:

$$[\![\mathbb{B}]\!] = \mathbb{B} \qquad [\![\Psi \Rightarrow A]\!] = [\![\Psi]\!] \Rightarrow [\![A]\!] \qquad [\![S(A)]\!] = US\,[\![A]\!] \qquad [\![\Psi \times \Phi]\!] = [\![\Psi]\!] \times [\![\Phi]\!]$$

Remark 3.4 To avoid cumbersome notation, we will use the following convention: We write directly $US(A)$ for $[\![S(A)]\!] = US([\![A]\!])$ and A for $[\![A]\!]$, when there is no ambiguity.

Before giving the interpretation of typing derivation trees in the model, we need to define certain maps which will serve to implement some of the constructions in the language.

To implement the *if* construction we define the following map.

Definition 3.5 Given $t, r \in [\Gamma, A]$ there exists a map $\mathbb{B} \xrightarrow{f_{t,r}} [\Gamma, A]$ in **Set** defined by $f_{t,r} = [\hat{t}, \hat{r}]$ where $\hat{t} : 1 \to [\Gamma, A]$ and $\hat{r} : 1 \to [\Gamma, A]$ are given by $\hat{t} = \lambda x.t$ and $\hat{s} = \lambda x.s$. Concretely this means that $i_1(\star) \mapsto t$ and $i_2(\star) \mapsto r$.

Example 3.6 Consider $t = i_1$ and $r = i_2$, with $t, r \in [1, \mathbb{B}]$, where $\mathbb{B} = \{i_1(\star), i_2(\star)\}$. To make the example more clear, let us consider $i_1(\star) = |0\rangle$ and $i_2(\star) = |1\rangle$, hence $\mathbb{B} = \{|0\rangle, |1\rangle\}$. The map $\mathbb{B} \xrightarrow{f_{t,r}} [1, \mathbb{B}]$ in **Set** is defined by $f_{t,r} = [\lambda x.i_1, \lambda x.i_2]$, where $\lambda x.i_k : 1 \to [1, \mathbb{B}]$, for $k = 1, 2$. Therefore, we have the following commuting diagram

$$\begin{tikzcd}
1 \arrow[r, "i_1"] \arrow[dr, "\lambda x.i_1"'] & \mathbb{B} \arrow[d, "f_{t,r}"] & 1 \arrow[l, "i_2"'] \arrow[dl, "\lambda x.i_2"] \\
 & {[1, \mathbb{B}]} &
\end{tikzcd}$$

Hence, we have

$$f_{t,r}\,|0\rangle = f_{t,r}(i_i(\star)) = (i_i \circ f_{t,r})\star = (\lambda x.i_1)\star = i_1 = t$$
$$f_{t,r}\,|1\rangle = f_{t,r}(i_2(\star)) = (i_2 \circ f_{t,r})\star = (\lambda x.i_2)\star = i_2 = r$$

Therefore, $f_{t,r}$ is the map $|0\rangle \mapsto t$ and $|1\rangle \mapsto r$.

A projection π_{jk} acts in the following way: first it projects the first j components of its argument, an n-dimensional vector, to the basis vector $|k\rangle$ in the vector space of dimension j, then it renormalizes it, and finally it factorizes the first j components. Then, the projection π_j takes the probabilistic distribution between the 2^j projectors π_{jk}, each of these probabilities, calculates from the normalized vector to be projected.

Example 3.7 Let us analyse the Example 2.4:

$$\pi_2(|001\rangle + 2.\,|110\rangle + 3.\,|000\rangle) \xrightarrow{(\mathsf{proj})} \tfrac{10}{14}\,|00\rangle \times (\tfrac{1}{\sqrt{10}}.\,|1\rangle + \tfrac{3}{\sqrt{10}}.\,|0\rangle) \parallel \tfrac{4}{14}\,|11\rangle \times (1.\,|0\rangle)$$

We can divide this in four projectors (since $j = 2$, we have 2^2 projectors), which are taken in parallel (with the symbol $\parallel$). The four projectors are: $\pi_{2,00}$, $\pi_{2,01}$, $\pi_{2,10}$ and $\pi_{2,11}$. In this case, the probability for the projectors $\pi_{2,01}$ and $\pi_{2,10}$ are 0, and hence these do not appear in the final term.

The projector $\pi_{2,00}$ acts as described before: first it projects the first 2 components of $|\psi\rangle$ to the basis vector $|00\rangle$, obtaining $|001\rangle + 3.\,|000\rangle$. Then it renormalizes it, by dividing it by its norm, obtaining $\frac{1}{\sqrt{10}}.\,|001\rangle + \frac{3}{\sqrt{10}}.\,|000\rangle$. Finally, it factorizes the vector, obtaining $|00\rangle \times (\frac{1}{\sqrt{10}}.\,|1\rangle + \frac{3}{\sqrt{10}}.\,|0\rangle)$. Similarly, the projector $\pi_{2,11}$ gives $|11\rangle \times (1.\,|0\rangle)$.

Finally, the probabilities to assemble the final term are calculated by $p_0 = \frac{|1|^2+|3|^2}{|1|^2+|2|^2+|3|^2} = \frac{10}{14}$ and $p_1 = \frac{|2|^2}{|1|^2+|2|^2+|3|^2} = \frac{4}{14}$.

Categorically, we can describe the operator π_{jk} (Definition 3.11) by the composition of three arrows: a normalizing arrow Norm (Definition 3.8), a projector arrow to the $|k\rangle$ basis vector, and a factorizing arrow φ_j (Definition 3.9). Then, the projection π_j (Definition 3.14) maps a vector to the probabilistic distribution between the 2^j basis vectors $|k\rangle$, using a distribution map (Definition 3.12).

In the following definitions, if $|\psi\rangle$ is a vector of dimension n, we write $\overline{|\psi\rangle} : I \to S(\mathbb{B}^n)$ to the map $1 \mapsto |\psi\rangle$.

Definition 3.8 The normalizing arrow Norm is defined as follows:

$$\mathsf{Norm} : US(\mathbb{B}^n) \to US(\mathbb{B}^n) \qquad |\psi\rangle \mapsto \begin{cases} \frac{|\psi\rangle}{\sqrt{(\overline{|\psi\rangle}^\dagger \circ \overline{|\psi\rangle})(\star)}} & \text{if } |\psi\rangle \neq \mathbf{0} \\ |0\rangle & \text{otherwise} \end{cases}$$

Definition 3.9 The factorizing arrow φ_j is defined as any arrow making the following diagram commute:

$$\begin{tikzcd}
\mathbb{B}^j \times US(\mathbb{B}^{n-j}) \arrow[r, "\eta\times\mathsf{Id}"] \arrow[d, "\mathsf{Id}"] & US(\mathbb{B}^j) \times US(\mathbb{B}^{n-j}) \arrow[r, "n"] & U(S(\mathbb{B}^j) \otimes S(\mathbb{B}^{n-j})) \arrow[d, "U(m)"] \\
\mathbb{B}^j \times US(\mathbb{B}^{n-j}) & & US(\mathbb{B}^n) = US(\mathbb{B}^j \times \mathbb{B}^{n-j}) \arrow[ll, "\varphi_j"]
\end{tikzcd}$$

Example 3.10 For example, take φ_j as the following map:

$$\varphi_j : US(\mathbb{B}^n) \to \mathbb{B}^j \times US(\mathbb{B}^{j-n})$$
$$a \mapsto \begin{cases} \prod_{h=1}^{j} |b_h\rangle \times \sum_{i=1}^{n} \alpha_i.\left(\prod_{h=j+1}^{n} |b_{ih}\rangle\right) & \text{if } a = \sum_{i=1}^{n} \alpha_i.\left(\prod_{h=1}^{j} |b_h\rangle \times \prod_{h=j+1}^{n} |b_{ih}\rangle\right) \\ |0\rangle^n & \text{otherwise} \end{cases}$$

Definition 3.11 For each $k = 0, \dots, 2^j - 1$, the projection to the $|k\rangle$ basis vector, π_{jk}, is defined as any arrow making the following diagram commute:

$$\begin{tikzcd}
US(\mathbb{B}^n) \cong U(S(\mathbb{B})^{\otimes n}) \arrow[r, "U((\overline{|k\rangle} \circ \overline{|k\rangle}^\dagger) \otimes I)"] \arrow[d, "\pi_{jk}"] & U(S(\mathbb{B})^{\otimes n}) \cong US(\mathbb{B}^n) \arrow[d, "\mathsf{Norm}"] \\
\mathbb{B}^j \times US(\mathbb{B}^{n-j}) & US(\mathbb{B}^n) \arrow[l, "\varphi_j"]
\end{tikzcd}$$

where the isomorphism $US(\mathbb{B}^n) \cong U(S(\mathbb{B})^{\otimes n})$ is obtained by composing $n-1$ times the mediating arrow m and then applying the functor U.

The following distribution map will allow to assemble the final distribution of projections in Definition 3.14.

Definition 3.12 Let $\{p_i\}_{i=1}^n$ be a set with $p_i \in [0,1]$, and $\sum_{i=1}^n p_i = 1$. Then, we define $d_{\{p_i\}_i}$ as the arrow $d_{\{p_i\}_i} : A^n \to D(A)$ such that $(a_1,\dots,a_n) \mapsto \sum_{i=1}^n p_i\chi_{a_i}$.

Example 3.13 Consider $d_{\{\frac{1}{2},\frac{1}{3},\frac{1}{6}\}} : \mathbb{B}^3 \to D(\mathbb{B}^3)$ defined by $d_{\{\frac{1}{2},\frac{1}{3},\frac{1}{6}\}}(b_1 \times b_2 \times b_3) = \frac{1}{2}\chi_{b_1} + \frac{1}{3}\chi_{b_2} + \frac{1}{6}\chi_{b_3}$.
Then, for example, $d_{\{\frac{1}{2},\frac{1}{3},\frac{1}{6}\}}\,|101\rangle = \frac{1}{2}\chi_{|1\rangle} + \frac{1}{3}\chi_{|0\rangle} + \frac{1}{6}\chi_{|1\rangle}$.

Definition 3.14 π_j is the arrow $\pi_j : US(\mathbb{B}^n) \to D(\mathbb{B}^j \times US(\mathbb{B}^{n-j}))$ such that $|\psi\rangle \mapsto \sum_{k=0}^{2^j-1} p_k\chi_{\pi_{jk}|\psi\rangle}$, where $p_k = \overline{\mathsf{Norm}(|\psi\rangle)}^\dagger \circ P_k \circ \overline{\mathsf{Norm}(|\psi\rangle)}$ with $P_k = (\overline{|k\rangle} \circ \overline{|k\rangle}^\dagger) \otimes \mathsf{Id}$ and π_{jk} is the arrow given in Definition 3.11.

Example 3.15 Consider the set $\mathbb{B}^2$ and the vector space $S(\mathbb{B}^2)$. We can describe the projection π_1 as the map $\pi_1 : US(\mathbb{B}^2) \to D(\mathbb{B} \times US(\mathbb{B}))$ such that $|\psi\rangle \mapsto p_0\chi_{\pi_{10}|\psi\rangle} + p_1\chi_{\pi_{11}|\psi\rangle}$, where, if $|\psi\rangle = \alpha_1.|00\rangle + \alpha_2.|01\rangle + \alpha_3.|10\rangle + \alpha_4.|11\rangle$, then $p_0 = \frac{|\alpha_1|^2+|\alpha_2|^2}{\sqrt{\sum_{i=1}^4 |\alpha_i|^2}}$ and $p_1 = \frac{|\alpha_3|^2+|\alpha_4|^2}{\sqrt{\sum_{i=1}^4 |\alpha_i|^2}}$.

The Norm arrow is the arrow $\mathsf{Norm} : US(\mathbb{B}^2) \to US(\mathbb{B}^2)$ such that

$$\alpha_1.|00\rangle+\alpha_2.|01\rangle+\alpha_3.|10\rangle+\alpha_4.|11\rangle \mapsto \frac{\alpha_1}{\sqrt{\sum_{i=1}^4|\alpha_i|^2}}.|00\rangle+\frac{\alpha_2}{\sqrt{\sum_{i=1}^4|\alpha_i|^2}}.|01\rangle+\frac{\alpha_3}{\sqrt{\sum_{i=1}^4|\alpha_i|^2}}.|10\rangle+\frac{\alpha_4}{\sqrt{\sum_{i=1}^4|\alpha_i|^2}}.|11\rangle$$

The factorisation arrow is the arrow $\varphi_1 : US(\mathbb{B}^2) \to \mathbb{B} \times US(\mathbb{B})$ such that

$$\alpha_1.|00\rangle + \alpha_2.|01\rangle + \alpha_3.|10\rangle + \alpha_4.|11\rangle \mapsto \begin{cases} |0\rangle \times (\alpha_1.|0\rangle + \alpha_2.|1\rangle) & \text{if } \alpha_3 = \alpha_4 = 0 \\ |1\rangle \times (\alpha_3.|0\rangle + \alpha_4.|1\rangle) & \text{if } \alpha_1 = \alpha_2 = 0 \\ |00\rangle & \text{otherwise} \end{cases}$$

Finally, π_{10} and π_{11} are defined as $\pi_{10} : US(\mathbb{B}^2) \to \mathbb{B} \times US(\mathbb{B})$ and $\pi_{11} : US(\mathbb{B}^2) \to \mathbb{B} \times US(\mathbb{B})$ such that

$$\pi_{10} = \varphi_1 \circ \mathsf{Norm} \circ U(\overline{|0\rangle} \circ \overline{|0\rangle}^\dagger \otimes Id) \qquad\qquad \pi_{11} = \varphi_1 \circ \mathsf{Norm} \circ U(\overline{|1\rangle} \circ \overline{|1\rangle}^\dagger \otimes Id)$$

We write $(US)^m(A)$ for $US(\dots US(A))$, where $m > 0$ and $A \neq US(B)$. The arrow sum on $(US)^m(A)$ will use the underlying sum in the vector space $S(A)$. Therefore, in order to implement this sum, we need the following map.

Definition 3.16 The map $g_k : (US)^{k+1}(A) \times (US)^{k+1}(A) \to (US)^k(US(A) \times US(A))$ is defined by

$$g_k = (US)^{k-1}U(m) \circ (US)^{k-1}(n) \circ (US)^{k-2}U(m) \circ (US)^{k-2}(n) \circ \dots \circ U(m) \circ n$$

Example 3.17 We can define the sum on $(US)^3(A) \times (US)^3(A)$ by using the sum on $S(A)$ as $g_2 \circ (US)^2(+)$, where $g_2 = USU(m) \circ US(n) \circ U(m) \circ n$. This gives the following diagram

$$\begin{tikzcd}
USUSUS(A) \times USUSUS(A) \arrow[r, "n"] \arrow[d, "\mathrm{sum}"] & U(SUSUS(A) \otimes SUSUS(A)) \arrow[r, "U(m)"] & US(USUS(A) \times USUS(A)) \arrow[d, "US(n)"] \\
USUSUS(A) & USUS(US(A) \times US(A)) \arrow[l, "USUS(+)"'] & USU(SUS(A) \otimes SUS(A)) \arrow[l, "USU(m)"']
\end{tikzcd}$$

Using all the previous definitions, we can finally give the interpretation of a type derivation tree in our model. If $\Gamma \vdash t : A$ with a derivation T, we write it generically $\llbracket T \rrbracket$ as $\Gamma \xrightarrow{t} A$. On the following definition, we write $S^m(A)$ for $S(\dots S(A))$, where $m > 0$ and $A \neq S(B)$.

Definition 3.18 If T is a type derivation tree, we define $\llbracket T \rrbracket$ inductively as follows,

$$\left\llbracket \frac{}{\Gamma^{\mathbb{B}}, x:\Psi \vdash x:\Psi}\ \mathsf{Ax} \right\rrbracket = \Gamma^{\mathbb{B}} \times \Psi \xrightarrow{!\times\mathsf{Id}} 1 \times \Psi \approx \Psi \quad \text{where } \mathsf{Id} \text{ is the identity in } \mathbf{Set}$$

$$\left\llbracket \frac{}{\Gamma^{\mathbb{B}} \vdash \mathbf{0}_{S(A)} : S(A)}\ Ax_{\mathbf{0}} \right\rrbracket = \Gamma^{\mathbb{B}} \xrightarrow{!} 1 \xrightarrow{\lambda x.\mathbf{0}} US(A)$$

$$\left\llbracket \frac{}{\Gamma^{\mathbb{B}} \vdash |0\rangle : \mathbb{B}}\ Ax_{|0\rangle} \right\rrbracket = \Gamma^{\mathbb{B}} \xrightarrow{!} 1 \xrightarrow{\lambda x.|0\rangle} \mathbb{B}$$

$$\left\llbracket \frac{}{\Gamma^{\mathbb{B}} \vdash |1\rangle : \mathbb{B}}\ Ax_{|1\rangle} \right\rrbracket = \Gamma^{\mathbb{B}} \xrightarrow{!} 1 \xrightarrow{\lambda x.|1\rangle} \mathbb{B}$$

$$\left\llbracket \frac{\Gamma \vdash t : S^m(A)}{\Gamma \vdash \alpha.t : S^m(A)}\ \alpha_I \right\rrbracket = \Gamma \xrightarrow{t} (US)^m(A) \xrightarrow{(US)^{m-1}U(\lambda)} (US)^{m-1}U(S(A)\otimes I)$$
$$\xrightarrow{(US)^{m-1}U(\mathsf{Id}\otimes\alpha)} (US)^{m-1}U(S(A)\otimes I) \xrightarrow{(US)^{m-1}U(\lambda^{-1})} (US)^m(A)$$

$$\left\llbracket \frac{\Gamma, \Xi^{\mathbb{B}} \vdash t : S^m(A) \quad \Delta, \Xi^{\mathbb{B}} \vdash r : S^m(A)}{\Gamma, \Delta, \Xi^{\mathbb{B}} \vdash t + r : S^m(A)}\ +_I \right\rrbracket = \Gamma \times \Delta \times \Xi^{\mathbb{B}} \xrightarrow{\mathsf{Id}\times\delta} \Gamma \times \Delta \times \Xi^{\mathbb{B}} \times \Xi^{\mathbb{B}} \xrightarrow{\mathsf{Id}\times\sigma\times\mathsf{Id}} \Gamma \times \Xi^{\mathbb{B}} \times \Delta \times \Xi^{\mathbb{B}}$$
$$\xrightarrow{t\times r} (US)^m(A) \times (US)^m(A) \xrightarrow{g_{m-1}} (US)^{m-1}(US(A) \times US(A))$$
$$\xrightarrow{(US)^{m-1}(+)} (US)^m(A)$$

$$\left\llbracket \frac{\Gamma \vdash t : A}{\Gamma \vdash t : S(A)}\ S_I \right\rrbracket = \Gamma \xrightarrow{t} A \xrightarrow{\eta} US(A)$$

$$\left\llbracket \frac{\Gamma \vdash t : S^k(\mathbb{B}^n)}{\Gamma \vdash \pi_j t : \mathbb{B}^j \times S(\mathbb{B}^{n-j})}\ S_E \right\rrbracket = \Gamma \xrightarrow{t} (US)^k(\mathbb{B}^n) \xrightarrow{\mu^{k-1}} US(\mathbb{B}^n) \xrightarrow{\pi_j} D(\mathbb{B}^j \times S(\mathbb{B}^{n-j}))$$

$$\left\llbracket \frac{\Gamma \vdash t : A \quad \Gamma \vdash r : A}{\Gamma \vdash ?t{\cdot}r : \mathbb{B} \Rightarrow A}\ \mathsf{If} \right\rrbracket = \Gamma \xrightarrow{\mathsf{curry}(\mathsf{uncurry}(f_{t,r}) \circ \mathsf{swap})} [\mathbb{B}, A]$$

$$\left\llbracket \frac{\Gamma, x:\Psi \vdash t : A}{\Gamma \vdash \lambda x{:}\Psi.t : \Psi \Rightarrow A}\ \Rightarrow_I \right\rrbracket = \Gamma \xrightarrow{\eta^{\Psi}} [\Psi, \Gamma \times \Psi] \xrightarrow{[\mathsf{Id}, t]} [\Psi, A]$$

$$\left\llbracket \frac{\Delta, \Xi^{\mathbb{B}} \vdash u : \Psi \quad \Gamma, \Xi^{\mathbb{B}} \vdash t : \Psi \Rightarrow A}{\Delta, \Gamma, \Xi^{\mathbb{B}} \vdash tu : A}\ \Rightarrow_E \right\rrbracket = \Delta \times \Gamma \times \Xi^{\mathbb{B}} \xrightarrow{\mathsf{Id}\times\delta} \Delta \times \Gamma \times \Xi^{\mathbb{B}} \times \Xi^{\mathbb{B}} \xrightarrow{\mathsf{Id}\times\sigma\times\mathsf{Id}} \Delta \times \Xi^{\mathbb{B}} \times \Gamma \times \Xi^{\mathbb{B}}$$
$$\xrightarrow{u\times t} \Psi \times [\Psi, A] \xrightarrow{\varepsilon^{\Psi}} A$$

$$\left\llbracket \frac{\Delta, \Xi^{\mathbb{B}} \vdash u : S(\Psi) \quad \Gamma, \Xi^{\mathbb{B}} \vdash t : S(\Psi \Rightarrow A)}{\Delta, \Gamma, \Xi^{\mathbb{B}} \vdash tu : S(A)}\ \Rightarrow_{ES} \right\rrbracket = \Delta \times \Gamma \times \Xi^{\mathbb{B}} \xrightarrow{\mathsf{Id}\times\delta} \Delta \times \Gamma \times \Xi^{\mathbb{B}} \times \Xi^{\mathbb{B}} \xrightarrow{\mathsf{Id}\times\sigma\times\mathsf{Id}} \Delta\times\Xi^{\mathbb{B}}\times\Gamma\times\Xi^{\mathbb{B}}$$
$$\xrightarrow{u\times t} US(\Psi) \times US([\Psi, A]) \xrightarrow{n} U(S(\Psi) \otimes S([\Psi, A]))$$
$$\xrightarrow{U(m)} US(\Psi \times [\Psi, A]) \xrightarrow{US(\varepsilon^{\Psi})} US(A)$$

$$\left\llbracket \frac{\Gamma, \Xi^{\mathbb{B}} \vdash t : \Psi \quad \Delta, \Xi^{\mathbb{B}} \vdash u : \Phi}{\Gamma, \Delta, \Xi^{\mathbb{B}} \vdash t \times u : \Psi \times \Phi}\ \times_I \right\rrbracket = \Gamma \times \Delta \times \Xi^{\mathbb{B}} \xrightarrow{\mathsf{Id}\times\delta} \Gamma \times \Delta \times \Xi^{\mathbb{B}} \times \Xi^{\mathbb{B}} \xrightarrow{\mathsf{Id}\times\sigma\times\mathsf{Id}} \Gamma \times \Xi^{\mathbb{B}} \times \Delta \times \Xi^{\mathbb{B}} \xrightarrow{t\times u} \Psi \times \Phi$$

$$\left\llbracket \frac{\Gamma \vdash t : \mathbb{B}^n}{\Gamma \vdash \mathit{head}\ t : \mathbb{B}}\ \times_{Er} \right\rrbracket = \Gamma \xrightarrow{t} \mathbb{B}^n \xrightarrow{\mathit{head}} \mathbb{B} \quad \text{where } \mathit{head} \text{ is the projector of the first component in } \mathbf{Set}$$

$$\left\llbracket \frac{\Gamma \vdash t : \mathbb{B}^n}{\Gamma \vdash \mathit{tail}\ t : \mathbb{B}^{n-1}}\ \times_{El} \right\rrbracket = \Gamma \xrightarrow{t} \mathbb{B}^n \xrightarrow{\mathit{tail}} \mathbb{B}^{n-1} \quad \text{where } \mathit{tail} \text{ is the projector of the } n-1 \text{ last components}$$

$$\left\llbracket \frac{\Gamma \vdash t : S(S(\Psi) \times \Phi)}{\Gamma \vdash \Uparrow_r t : S(\Psi \times \Phi)}\ \Uparrow_r \right\rrbracket = \Gamma \xrightarrow{t} US(US(\Psi) \times \Phi) \xrightarrow{U(\mathsf{Id}\times\eta)} US(US(\Psi) \times US(\Phi)) \xrightarrow{US(n)} US(U(S(\Psi) \otimes S(\Phi)))$$
$$\xrightarrow{USU(m)} USUS(\Psi \times \Phi) \xrightarrow{\mu} US(\Psi \times \Phi)$$

$$\left\llbracket \frac{\Gamma \vdash t : S(\Psi \times S(\Phi))}{\Gamma \vdash \Uparrow_\ell t : S(\Psi \times \Phi)}\ \Uparrow_l \right\rrbracket = \Gamma \xrightarrow{t} US(\Psi \times US(\Phi)) \xrightarrow{U(\eta\times\mathsf{Id})} US(US(\Psi) \times US(\Phi)) \xrightarrow{US(n)} US(U(S(\Psi) \otimes S(\Phi)))$$
$$\xrightarrow{USU(m)} USUS(\Psi \times \Phi) \xrightarrow{\mu} US(\Psi \times \Phi)$$

$$\left\llbracket \frac{\Gamma \vdash t_i : A \quad \sum_i p_i = 1}{\Gamma \vdash p_1 t_1 \parallel \cdots \parallel p_n t_n : A}\ \parallel \right\rrbracket = \Gamma \xrightarrow{\delta} \Gamma^n \xrightarrow{t_1\times\cdots\times t_n} A^n \xrightarrow{d_{\{p_i\}_i}} D(A)$$

Proposition 3.19 (Independence of derivation) *If $\Gamma \vdash t : A$ can be derived with two different derivations T and T', then $\llbracket T \rrbracket = \llbracket T' \rrbracket$*

Proof. Without taking into account rules $\Rightarrow_E$, $\Rightarrow_{ES}$ and S_I, the typing system is syntax directed. In the case of the application (rules $\Rightarrow_E$ and $\Rightarrow_{ES}$), they can be interchanged only in few specific cases.

Hence, we give a rewrite system on trees such that each time a rule S_I can be applied before or after another rule, we chose a direction to rewrite the three to one of these forms. Similarly, we chose a direction for rules $\Rightarrow_E$ and $\Rightarrow_{ES}$. Then we prove that every rule preserves the semantics of the tree. This rewrite system is clearly confluent and normalizing, hence for each tree T we can take the semantics of its normal form, and so every sequent will have one way to calculate its semantics: as the semantics of the normal tree. □

Remark 3.20 Proposition 3.19 allows us to write the semantics of a sequent, independently of its derivation. Hence, from now on, we will use $[\![\Gamma \vdash t : A]\!]$, without ambiguity.

Lemma 3.21 (Substitution) *If* $\Gamma', x : \Psi, \Gamma \vdash t : A$ *and* $\vdash r : \Psi$*, then the following diagram commutes:*

$$\begin{tikzcd}
\Gamma' \times \Gamma \arrow[r, "(r/x)t"] \arrow[d, dashed, "\approx"] & A \\
\Gamma' \times 1 \times \Gamma \arrow[r, "\mathsf{Id}\times r\times\mathsf{Id}"] & \Gamma' \times \Psi \times \Gamma \arrow[u, "t"]
\end{tikzcd}$$

That is, $[\![\Gamma', \Gamma \vdash (r/x)t : A]\!] = [\![\Gamma', x : \Psi, \Gamma \vdash t : A]\!] \circ ([\![\vdash r : \Psi]\!] \times \mathsf{Id})$.

Proof. By induction on the derivation of $\Gamma', x : \Psi, \Gamma \vdash t : A$. □

Theorem 3.22 (Soundness) *If* $\vdash t : A$*, and* $t \longrightarrow r$*, then* $[\![\vdash t : A]\!] = [\![\vdash r : A]\!]$.

Proof. By induction on the rewrite relation, using the first derivable type for each term. □

In order to prove adequacy (Theorem 3.26), we use an adaptation to Lambda-$\mathcal{S}$ of Tait's proof for strong normalization.

Definition 3.23 Let $\mathfrak{A}, \mathfrak{B}$ be sets of closed terms. We define the following operators on them:

- *Closure by antireduction:* $\overline{\mathfrak{A}} = \{t \mid t \longrightarrow^* r, \text{ with } r \in \mathfrak{A} \text{ and } FV(t) = \emptyset\}$.
- *Closure by parallelism:* $\mathfrak{A}^{\|} = \{\|_i p_i t_i \mid t_i \in \mathfrak{A} \text{ and } \sum_i p_i = 1\}$
- *Product:* $\mathfrak{A} \times \mathfrak{B} = \{t \times u \mid t \in \mathfrak{A} \text{ and } u \in \mathfrak{B}\}$.
- *Arrow:* $\mathfrak{A} \Rightarrow \mathfrak{B} = \{t \mid \forall u \in \mathfrak{A}, tu \in \mathfrak{B}\}$.
- *Span:* $S\mathfrak{A} = \{\sum_i \alpha_i r_i \mid r_i \in \mathfrak{A}\}$ where αr is a notation for $\alpha.r$ when $\alpha \neq 1$, or $1.r$ or just r when $\alpha = 1$. Also, we use the convention that $\sum_{i=1}^{1} \alpha_i r_i = \alpha_i r_i$.
- *Error:* $E\mathfrak{A} = \mathfrak{A} \cup \{\mathsf{error}\}$, where error is any term containing a subterm $\pi_j \mathbf{0}_{S(\mathbb{B}^n)}$.

The set of computational closed terms of type A (denoted $\mathfrak{C}_A$), is defined by

$$\mathfrak{C}_{\mathbb{B}} = \overline{\{|0\rangle, |1\rangle, \mathsf{error}\}}^{\parallel} \qquad \mathfrak{C}_{\Psi\Rightarrow A} = \overline{E(\mathfrak{C}_\Psi \Rightarrow \mathfrak{C}_A)}^{\parallel}$$
$$\mathfrak{C}_{A\times B} = \overline{E(\mathfrak{C}_A \times \mathfrak{C}_B)}^{\parallel} \qquad \mathfrak{C}_{S(A)} = \overline{ES\mathfrak{C}_A \cup \{\mathbf{0}_{S(B)} \mid S(B) \preceq S(A)\}}^{\parallel}$$

Where $\preceq$ is defined as $S(S(A)) \preceq S(A)$ and $A \preceq S(A)$.

A substitution σ is valid with respect to a context Γ (notation $\sigma \vDash \Gamma$) if for each $x : A \in \Gamma$, $\sigma x \in \mathfrak{C}_A$.

Lemma 3.24 *For any type A, we have* $\mathsf{error} \in \mathfrak{C}_A$.

Proof. By induction on A. □

Lemma 3.25 *If $\Gamma \vdash t : A$ and $\sigma \vDash \Gamma$, then $\sigma t \in \mathfrak{C}_A$.*

Proof. By induction on the derivation of $\Gamma \vdash t : A$. □

Theorem 3.26 (Adequacy) *If $[\![\vdash t : \mathbb{B}]\!] = [\![\vdash v : \mathbb{B}]\!]$, where $v \in \{|0\rangle, |1\rangle\}$, then either $t \longrightarrow^* v$ or $t \longrightarrow^*$* error.

Proof. By Lemma 3.25, $t \in \mathfrak{C}_{\mathbb{B}} = \overline{\{|0\rangle, |1\rangle, \mathsf{error}\}}^{\parallel}$, therefore, $t \longrightarrow^* \|_{i=1}^{n} p_i r_i$ where $r_i \in \{|0\rangle, |1\rangle, \mathsf{error}\}$.

Since $[\![\vdash t : \mathbb{B}]\!] = [\![\vdash v : \mathbb{B}]\!] = \lambda x.\,|0\rangle$ or $\lambda x.\,|1\rangle$, we have that $n = p_1 = 1$ and so $t \longrightarrow^* v$, or $t \longrightarrow^* \mathsf{error}$. □

4 Conclusion

In this paper we have given a concrete categorical semantics of Lambda-$\mathcal{S}$ and proved that it is sound (Theorem 3.22) and adequate (Theorem 3.26). Such a semantics highlights the dynamics of the calculus: The algebraic rewriting (linear distribution, vector space axioms, and typing casts rules) emphasize the standard behaviour of vector spaces, and the natural transformation n takes these arrows from the Cartesian category **Set** to the tensorial category **Vec**, where such a behaviour occur naturally, and then are taken back to the Cartesian realm with the natural transformation m. This way, rules such as $(\mathsf{lin}_{\mathsf{r}}^{+})$: $t(u+v) \longrightarrow tu + tv$, are simply considered as $U(m) \circ n$ producing $(u+v, t) \mapsto (u,t) + (v,t)$ in two steps: $(u+v,t) \mapsto (u+v)\otimes t = u\otimes t + v \otimes t \mapsto (u,t)+(v,t)$, using the fact that $(u+v)\otimes t = u \otimes t + v \otimes t$ in **Vec**.

We have constructed a concrete mathematical semantic model of Lambda-$\mathcal{S}$ based on a monoidal adjunction with some extra conditions. However, the construction depends crucially on inherent properties of the categories of set and vector spaces. In a future work we will study the semantics from a more abstract point of view. Our approach will be based on recasting the concrete model at a more abstract categorical level of monoidal categories with some axiomatic properties that are now veiled in the concrete model. Some of these properties, such as to consider an abstract dagger instead of an inner product, were introduced in the concrete model from the very beginning, but others are described in Remark 3.2 and Definitions 3.5, 3.8, 3.9, 3.11, 3.12, and 3.14. Another question we hope to address in future work is the exact categorical relationship between the notion of amplitude and probability in the context of the abstract semantics. While some research has been done in this topic (see, for example, [2, 19]) it differs from our point of view in some important aspects: for example to consider a notion of abstract normalization as primitive.

References

[1] Abramsky, S., *Computational interpretations of linear logic*, Theoretical Computer Science **111** (1993), pp. 3–57.

[2] Abramsky, S. and B. Coecke, *A categorical semantics of quantum protocols*, in: *Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science (LICS)* (2004), pp. 415–425.

[3] Altenkirch, T. and J. Grattage, *A functional quantum programming language*, in: *Proceedings of the 20th Annual IEEE Symposium on Logic in Computer Science (LICS)* (2005), pp. 249–258.

[4] Arrighi, P. and A. Díaz-Caro, *A System F accounting for scalars*, Logical Methods in Computer Science **8(1:11)** (2012).

[5] Arrighi, P., A. Díaz-Caro and B. Valiron, *The vectorial lambda-calculus*, Information and Computation **254** (2017), pp. 105–139.

[6] Arrighi, P. and G. Dowek, *Lineal: a linear-algebraic lambda-calculus*, Logical Methods in Computer Science **13(1:8)** (2017).

[7] Assaf, A., A. Díaz-Caro, S. Perdrix, C. Tasson and B. Valiron, *Call-by-value, call-by-name and the vectorial behaviour of the algebraic λ-calculus*, Logical Methods in Computer Science **10(4:8)** (2014).

[8] Benton, N., *A mixed linear and non-linear logic: Proofs, terms and models*, in: L. Pacholski and J. Tiuryn, editors, *Computer Science Logic (CSL 1994)*, Lecture Notes in Computer Science **933** (1994), pp. 121–135.

[9] Díaz-Caro, A. and G. Dowek, *Typing quantum superpositions and measurement*, in: *Theory and Practice of Natural Computing (TPNC 2017)*, Lecture Notes in Computer Science **10687** (2017), pp. 281–293.

[10] Díaz-Caro, A. and G. Martínez, *Confluence in probabilistic rewriting*, Preproceedings of LSFA 2017. To appear in ENTCS. Preprint at `arXiv:1708.03536`. (2017).

[11] Díaz-Caro, A. and B. Petit, *Linearity in the non-deterministic call-by-value setting*, in: L. Ong and R. de Queiroz, editors, *Logic, Language, Information and Computation*, Lecture Notes in Computer Science **7456** (2012), pp. 216–231.

[12] Girard, J.-Y., *Linear logic*, Theoretical Compututer Science **50** (1987), pp. 1–102.

[13] Giry, M., *A categorical approach to probability theory*, in: *Categorical Aspects of Topology and Analysis*, Lecture Notes in Mathematics **915** (1982), pp. 68–85.

[14] Green, A. S., P. L. Lumsdaine, N. J. Ross, P. Selinger and B. Valiron, *Quipper: a scalable quantum programming language*, ACM SIGPLAN Notices (PLDI'13) **48** (2013), pp. 333–342.

[15] Lane, S. M., "Categories for the Working Mathematician," Springer, 1998, 2 edition.

[16] Moggi, E., *Computational lambda-calculus and monads*, Technical Report ECS-LFCS-88-66, Lab. for Foundations of Computer Science, University of Edinburgh (1988).

[17] Pagani, M., P. Selinger and B. Valiron, *Applying quantitative semantics to higher-order quantum computing*, ACM SIGPLAN Notices (POPL'14) **49** (2014), pp. 647–658.

[18] Rinaldi, J. P., "Demostrando normalización fuerte sobre una extensión cuántica del lambda cálculo," Master's thesis, Universidad Nacional de Rosario (2018).

[19] Selinger, P., *Dagger compact closed categories and completely positive maps*, in: *3rd International Workshop on Quantum Programming Languages (QPL 2005)*, Electronic Notes in Theoretical Computer Science **170**, 2007, pp. 139–163.

[20] Selinger, P. and B. Valiron, *A lambda calculus for quantum computation with classical control*, Mathematical Structures in Computer Science **16** (2006), pp. 527–552.

[21] Vaux, L., *The algebraic lambda calculus*, Mathematical Structures in Computer Science **19** (2009), pp. 1029–1059.

[22] Zorzi, M., *On quantum lambda calculi: a foundational perspective*, Mathematical Structures in Computer Science **26** (2016), pp. 1107–1195.