HAL open science

Linear Lambda-Calculus is Linear

Alejandro Díaz-Caro, Gilles Dowek

To cite this version:

Alejandro Díaz-Caro, Gilles Dowek. Linear Lambda-Calculus is Linear. FSCD 2022 - 7th International Conference on Formal Structures for Computation and Deduction, Tel Aviv University, Aug 2022, Haifa, Israel. 10.4230/LIPIcs.FSCD.2022.21 . hal-03959343

HAL Id: hal-03959343

https://inria.hal.science/hal-03959343v1

Submitted on 27 Jan 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Linear Lambda-Calculus is Linear

Alejandro Díaz-Caro
Departamento de Ciencia y Tecnología, Universidad Nacional de Quilmes, Bernal, Buenos Aires, Argentina
Instituto de Ciencias de la Computación, CONICET / Universidad de Buenos Aires, Buenos Aires, Argentina

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

Abstract

We prove a linearity theorem for an extension of linear logic with addition and multiplication by a scalar: the proofs of some propositions in this logic are linear in the algebraic sense. This work is part of a wider research program that aims at defining a logic whose proof language is a quantum programming language.

2012 ACM Subject Classification Theory of computation → Proof theory; Theory of computation → Lambda calculus; Theory of computation → Linear logic; Theory of computation → Quantum computation theory

Keywords and phrases Proof theory, Lambda calculus, Linear logic, Quantum computing

Digital Object Identifier 10.4230/LIPIcs.FSCD.2022.21

Related Version *Full Version*: `arXiv:2201.11221`

Funding STIC-AmSud 21STIC10, ECOS-Sud A17C03, and the French-Argentinian IRP SINFIN. *Alejandro Díaz-Caro*: PIP 11220200100368CO and PICT-2019-1272.

Acknowledgements The authors want to thank Thomas Ehrhard, Jean-Baptiste Joinet, Jean-Pierre Jouannaud, Dale Miller, Alberto Naibo, Simon Perdrix, Alex Tsokurov, and Lionel Vaux for useful discussions.

1 Introduction

The name of linear logic [10] suggests that this logic has some relation with the algebraic notion of linearity. A common account of this relation is that a proof of a linear implication between two propositions A and B should not be any function mapping proofs of A to proofs of B, but a linear one. This idea has been fruitfully exploited to build models of linear logic (for example [3, 9, 11]), but it seems difficult to even formulate it within the proof language itself. Indeed, expressing the properties $f(u+v) = f(u) + f(v)$ and $f(a.u) = a.f(u)$ requires an addition and a multiplication by a scalar, that are usually not present in proof languages.

The situation has changed with quantum programming languages [1, 2, 4, 6, 8, 12, 14] and the algebraic λ-calculus [13], that mix some usual constructions of programming languages with algebraic operations. More specifically, several extensions of the lambda-calculus, or of a language of proof-terms, with addition and multiplication by a scalar have been proposed [2, 5, 13].

In this paper, we investigate an extension of linear logic with addition and multiplication by a scalar, the $\mathcal{L}^{\mathcal{S}}$-logic (where $\mathcal{S}$ denotes the field of scalars used), and we prove a linearity theorem: if f is a proof of an implication between two propositions of some specific form, then $f(u+v) = f(u) + f(v)$ and $f(a.u) = a.f(u)$.

© Alejandro Díaz-Caro and Gilles Dowek;
licensed under Creative Commons License CC-BY 4.0
7th International Conference on Formal Structures for Computation and Deduction (FSCD 2022).
Editor: Amy P. Felty; Article No. 21; pp. 21:1–21:17
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

This work is part of a wider research program that aims at determining in which way propositional logic must be extended or restricted, so that its proof language becomes a quantum programming language. There are two main issues in the design of a quantum programming language: the first is to take into account the linearity of the unitary operators and, for instance, avoid cloning, and the second is to express the information-erasure, non-reversibility, and non-determinism of the measurement. In [5], we addressed the question of the measurement. In this paper, we address that of linearity.

1.1 Interstitial rules

To extend linear logic with addition and multiplication by a scalar, we proceed, like in [5, long version], in two steps: we first add interstitial rules and then scalars.

An interstitial rule is a deduction rule whose premises are identical to its conclusion. In the $\mathcal{L}^{\mathcal{S}}$-logic, we consider two such rules

$$\dfrac{\Gamma \vdash A \quad \Gamma \vdash A}{\Gamma \vdash A}\,\text{sum} \qquad\qquad \dfrac{\Gamma \vdash A}{\Gamma \vdash A}\,\text{prod}$$

Adding these rules permits to build proofs that cannot be reduced, because the introduction rule of some connective and its elimination rule are separated by an interstitial rule, for example

$$\dfrac{\dfrac{\dfrac{\dfrac{\pi_1}{\Gamma \vdash A} \quad \dfrac{\pi_2}{\Gamma \vdash B}}{\Gamma \vdash A \wedge B}\,\wedge\text{-i} \quad \dfrac{\dfrac{\pi_3}{\Gamma \vdash A} \quad \dfrac{\pi_4}{\Gamma \vdash B}}{\Gamma \vdash A \wedge B}\,\wedge\text{-i}}{\Gamma \vdash A \wedge B}\,\text{sum} \quad \dfrac{\pi_5}{\Gamma, A \vdash C}}{\Gamma \vdash C}\,\wedge\text{-e1}$$

Reducing such a proof, sometimes called a commuting cut, requires reduction rules to commute the rule sum either with the elimination rule below or with the introduction rules above.

As the commutation with the introduction rules above is not always possible, for example in the proof

$$\dfrac{\dfrac{\dfrac{\pi_1}{\Gamma \vdash A}}{\Gamma \vdash A \vee B}\,\vee\text{-i1} \quad \dfrac{\dfrac{\pi_2}{\Gamma \vdash B}}{\Gamma \vdash A \vee B}\,\vee\text{-i2}}{\Gamma \vdash A \vee B}\,\text{sum}$$

the commutation with the elimination rule below is often preferred. In this paper, we favour the commutation of the interstitial rules with the introduction rules, rather than with the elimination rules, whenever it is possible, that is for all connectives except the disjunction. For example, the proof

$$\dfrac{\dfrac{\dfrac{\pi_1}{\Gamma \vdash A} \quad \dfrac{\pi_2}{\Gamma \vdash B}}{\Gamma \vdash A \wedge B}\,\wedge\text{-i} \quad \dfrac{\dfrac{\pi_3}{\Gamma \vdash A} \quad \dfrac{\pi_4}{\Gamma \vdash B}}{\Gamma \vdash A \wedge B}\,\wedge\text{-i}}{\Gamma \vdash A \wedge B}\,\text{sum}$$

reduces to

$$\dfrac{\dfrac{\dfrac{\pi_1}{\Gamma \vdash A} \quad \dfrac{\pi_3}{\Gamma \vdash A}}{\Gamma \vdash A}\,\text{sum} \quad \dfrac{\dfrac{\pi_2}{\Gamma \vdash B} \quad \dfrac{\pi_4}{\Gamma \vdash B}}{\Gamma \vdash B}\,\text{sum}}{\Gamma \vdash A \wedge B}\,\wedge\text{-i}$$

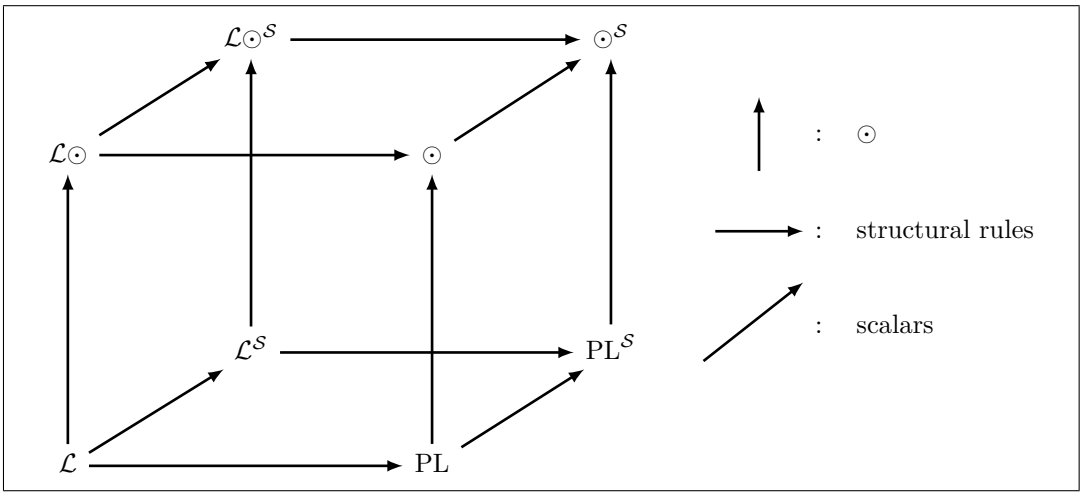

Figure 1 Eight logics.

Such a commutation yields a stronger introduction property for the considered connective.

For coherence, we commute both rules sum and prod with the elimination rule of the disjunction, rather that with its introduction rules. But, for the rule prod, both choices are possible.

1.2 Scalars

We then consider a field $\mathcal{S}$ of scalars and replace the introduction rule of the connective $\top$ with a family of rules $\top$-i(a), one for each scalar, and the rule prod with a family of rules prod(a), also one for each scalar

$$\frac{}{\Gamma \vdash \top}\ \top\text{-i}(a) \qquad\qquad \frac{\Gamma \vdash A}{\Gamma \vdash A}\ \text{prod}(a)$$

1.3 The connective $\odot$

Besides interstitial rules and scalars, we have introduced, in [5, long version], a new connective $\odot$ (read "sup" for "superposition"), that has an introduction rule $\odot$-i similar to that of the conjunction, two elimination rules $\odot$-e1 and $\odot$-e2 similar to those of the conjunction, but also a third elimination rule $\odot$-e similar to that of the disjunction.

The elimination rules $\odot$-e1 and $\odot$-e2 are used to express the information-preserving, reversible, and deterministic operations, such as the unitary transformations of quantum computing. The elimination rule $\odot$-e is used to express the information-erasing, non-reversible, and non-deterministic operations, such as quantum measurement. We will come back to this full system at Section 6.1 (the $\odot$ rules are listed at Figure 4).

Starting from propositional logic with the interstitial rules sum and prod, we can thus either add scalars, or the connective $\odot$, or both. This yields the four logics on the right face of the cube of Figure 1: PL is propositional logic with the interstitial rules sum and prod, $\mathrm{PL}^{\mathcal{S}}$ is propositional logic with the interstitial rules and scalars, $\odot$ is propositional logic with the interstitial rules and the connective $\odot$, and $\odot^{\mathcal{S}}$ is propositional logic with the interstitial rules, the connective $\odot$, and scalars.

1.4 Linearity

The proof language of the $\odot^{\mathcal{S}}$-logic is a quantum programming language, as quantum algorithms can be expressed in it. However, this language addresses the question of quantum measurement, but not the that of linearity, and non-linear functions, such as cloning operators, can also be expressed in it.

This leads to introduce, in this paper, a linear variant of the $\odot^{\mathcal{S}}$-logic, and prove a linearity theorem for it.

More generally, we can introduce, on the left face of the cube of Figure 1, a linear variant for each of the four logics of the right face: $\mathcal{L}$ is linear logic with the interstitial rules sum and prod, $\mathcal{L}^{\mathcal{S}}$ is linear logic with the interstitial rules and scalars, $\mathcal{L}\odot$ is linear logic with the interstitial rules and the connective $\odot$, and $\mathcal{L}\odot^{\mathcal{S}}$ is linear logic with the interstitial rules, the connective $\odot$, and scalars.

Our goal is to prove a linearity theorem for the proof language of the $\mathcal{L}\odot^{\mathcal{S}}$-logic. But such a theorem does not hold for the full $\mathcal{L}\odot^{\mathcal{S}}$-logic, that contains the rule $\odot$-e, that enables to express measurement operators, which are not linear. Thus, our linearity theorem should concern the fragment of the $\mathcal{L}\odot^{\mathcal{S}}$-logic without this rule. But, if $\odot$-e rule is excluded, the connective $\odot$ is just the conjunction, and this fragment of the $\mathcal{L}\odot^{\mathcal{S}}$-logic is the $\mathcal{L}^{\mathcal{S}}$-logic.

So, for a greater generality, we prove our linearity theorem for the $\mathcal{L}^{\mathcal{S}}$-logic: linear logic with the interstitial rules and scalars, but without the $\odot$ connective, and discuss, at the end of the paper, how this result extends to the $\mathcal{L}\odot^{\mathcal{S}}$-logic.

1.5 Linear connectives

In the $\mathcal{L}^{\mathcal{S}}$-logic, we have to make a choice of connectives.

In intuitionistic linear logic, there is no multiplicative falsehood, no additive implication, and no multiplicative disjunction. Thus, we have two possible truths and two possible conjunctions, but only one possible falsehood, implication, and disjunction.

In the $\mathcal{L}^{\mathcal{S}}$-logic, we have chosen a multiplicative truth, an additive falsehood, a multiplicative implication, an additive conjunction, and an additive disjunction. The rule sum also is additive. The reasons for this choice of connectives will be justified in Remarks 2.1, 5.1, and 5.2.

These symbols are often written 1, 0, $\multimap$, &, and $\oplus$. As we use only one conjunction, one disjunction, etc., to make our paper more accessible to readers who are not familiar with linear logic (and also because we have several zeros, for scalars, vectors, etc.), we use the usual symbols $\top$, $\bot$, $\Rightarrow$, $\wedge$, and $\vee$ instead. Of course, notations are arbitrary and the notations of linear logic can also be used.

The introduction rule for the additive conjunction is the same as that in usual natural deduction

$$\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \wedge B}\ \wedge\text{-i}$$

In particular, the proofs of A and B are in the same context Γ. But, in the elimination rule

$$\frac{\Gamma \vdash A \wedge B \quad \Delta, A \vdash C}{\Gamma, \Delta \vdash C}\ \wedge\text{-e1}$$

the proof of $A \wedge B$ and that of C must be in contexts Γ and Δ, A.

$$\frac{}{x : A \vdash x : A}\ \text{ax} \qquad \frac{\Gamma \vdash t : A \quad \Gamma \vdash u : A}{\Gamma \vdash t \boldsymbol{+} u : A}\ \text{sum} \qquad \frac{\Gamma \vdash t : A}{\Gamma \vdash a \bullet t : A}\ \text{prod}(a)$$

$$\frac{}{\vdash a.\star : \top}\ \top\text{-i}(a) \qquad \frac{\Gamma \vdash t : \top \quad \Delta \vdash u : A}{\Gamma, \Delta \vdash \delta_\top(t, u) : A}\ \top\text{-e} \qquad \frac{\Gamma \vdash t : \bot}{\Gamma, \Delta \vdash \delta_\bot(t) : C}\ \bot\text{-e}$$

$$\frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x.t : A \Rightarrow B}\ \Rightarrow\text{-i} \qquad \frac{\Gamma \vdash t : A \Rightarrow B \quad \Delta \vdash u : A}{\Gamma, \Delta \vdash t\ u : B}\ \Rightarrow\text{-e}$$

$$\frac{\Gamma \vdash t : A \quad \Gamma \vdash u : B}{\Gamma \vdash \langle t, u\rangle : A \wedge B}\ \wedge\text{-i}$$

$$\frac{\Gamma \vdash t : A \wedge B \quad \Delta, x : A \vdash u : C}{\Gamma, \Delta \vdash \delta^1_\wedge(t, x.u) : C}\ \wedge\text{-e1} \qquad \frac{\Gamma \vdash t : A \wedge B \quad \Delta, x : B \vdash u : C}{\Gamma, \Delta \vdash \delta^2_\wedge(t, x.u) : C}\ \wedge\text{-e2}$$

$$\frac{\Gamma \vdash t : A}{\Gamma \vdash \mathit{inl}(t) : A \vee B}\ \vee\text{-i1} \qquad \frac{\Gamma \vdash t : B}{\Gamma \vdash \mathit{inr}(t) : A \vee B}\ \vee\text{-i2}$$

$$\frac{\Gamma \vdash t : A \vee B \quad \Delta, x : A \vdash u : C \quad \Delta, y : B \vdash v : C}{\Gamma, \Delta \vdash \delta_\vee(t, x.u, y.v) : C}\ \vee\text{-e}$$

Figure 2 The deduction rules of the $\mathcal{L}^{\mathcal{S}}$-calculus.

In this paper, we first define the $\mathcal{L}^{\mathcal{S}}$-logic and its proof-language: the $\mathcal{L}^{\mathcal{S}}$-calculus, and prove that it verifies the subject reduction, confluence, termination, and introduction properties (Section 2). We then show how the vectors of $\mathcal{S}^n$ can be expressed in this calculus and how the irreducible closed proofs of some propositions are equipped with a structure of vector space (Section 3). We prove that all linear functions from $\mathcal{S}^m$ to $\mathcal{S}^n$ can be expressed as proofs of an implication between such propositions (Section 4). We then prove the main result of this paper: that, conversely, all the proofs of implications between such propositions are linear (Section 5). Finally, we show how this result extents to the proof language of the $\mathcal{L}\odot^{\mathcal{S}}$-logic and how this language is a quantum programming language (Section 6).

Most proofs are omitted from this conference paper, they can be found in its long version.

2 The $\mathcal{L}^{\mathcal{S}}$-calculus

Let $\mathcal{S}$ be a field of *scalars*, for instance $\mathbb{Q}$, $\mathbb{R}$, or $\mathbb{C}$.

The propositions of the $\mathcal{L}^{\mathcal{S}}$-logic are those of propositional logic

$$A = \top \mid \bot \mid A \Rightarrow A \mid A \wedge A \mid A \vee A$$

The proof-terms of this logic are

$$\begin{aligned} t = {} & x \mid t \boldsymbol{+} u \mid a \bullet t \mid a.\star \mid \delta_\top(t, u) \mid \delta_\bot(t) \\ & \mid \lambda x.t \mid (t\ u) \mid \langle t, u\rangle \mid \delta^1_\wedge(t, x.u) \mid \delta^2_\wedge(t, x.u) \\ & \mid \mathit{inl}(t) \mid \mathit{inr}(t) \mid \delta_\vee(t, x.u, y.v) \end{aligned}$$

where a is a scalar.

The variables x express the proofs built with the rule axiom, the terms $t \boldsymbol{+} u$ those built with the rule sum, the terms $a \bullet t$ those built with the family of rules prod(a), the terms $a.\star$ those built with the family of rules $\top$-i(a), the terms $\delta_\top(t, u)$ those built with the rule $\top$-e, the terms $\delta_\bot(t)$ those built with the rule $\bot$-e, the terms $\lambda x.t$ those built with the rule $\Rightarrow$-i, the terms $t\ u$ those built with the rule $\Rightarrow$-e, the terms $\langle t, u\rangle$ those built with the rule $\wedge$-i,

$$
\begin{aligned}
\delta_\top(a.\star, t) &\longrightarrow a \bullet t \\
(\lambda x.t)\ u &\longrightarrow (u/x)t \\
\delta^1_\wedge(\langle t, u\rangle, x.v) &\longrightarrow (t/x)v \\
\delta^2_\wedge(\langle t, u\rangle, x.v) &\longrightarrow (u/x)v \\
\delta_\vee(\mathit{inl}(t), x.v, y.w) &\longrightarrow (t/x)v \\
\delta_\vee(\mathit{inr}(u), x.v, y.w) &\longrightarrow (u/y)w \\
\\
a.\star + b.\star &\longrightarrow (a+b).\star \\
(\lambda x.t) + (\lambda x.u) &\longrightarrow \lambda x.(t + u) \\
\langle t, u\rangle + \langle v, w\rangle &\longrightarrow \langle t + v, u + w\rangle \\
\delta_\vee(t + u, x.v, y.w) &\longrightarrow \delta_\vee(t, x.v, y.w) + \delta_\vee(u, x.v, y.w) \\
\\
a \bullet b.\star &\longrightarrow (a \times b).\star \\
a \bullet \lambda x.t &\longrightarrow \lambda x.a \bullet t \\
a \bullet \langle t, u\rangle &\longrightarrow \langle a \bullet t, a \bullet u\rangle \\
\delta_\vee(a \bullet t, x.v, y.w) &\longrightarrow a \bullet \delta_\vee(t, x.v, y.w)
\end{aligned}
$$

Figure 3 The reduction rules of the $\mathcal{L}^{\mathcal{S}}$-calculus.

the terms $\delta^1_\wedge(t, x.u)$ and $\delta^2_\wedge(t, x.u)$ those built with the rules $\wedge$-e1 and $\wedge$-e2, the terms $\mathit{inl}(t)$ and $\mathit{inr}(t)$ those built with the rules $\vee$-i1 and $\vee$-i2, and the terms $\delta_\vee(t, x.u, y.v)$ those built with the rule $\vee$-e.

The proofs of the form $\star$, $\lambda x.t$, $\langle t, u\rangle$, $\mathit{inl}(t)$, and $\mathit{inr}(t)$ are called *introductions*, and those of the form $\delta_\top(t, u)$, $\delta_\bot(t)$, $t\ u$, $\delta^1_\wedge(t, x.u)$, $\delta^2_\wedge(t, x.u)$, and $\delta_\vee(t, x.u, y.v)$ *eliminations*. The variables and the proofs of the form $t + u$ and $a \bullet t$ are neither introductions nor eliminations.

The α-equivalence relation and the free and bound variables of a proof-term are defined as usual. Proof-terms are defined modulo α-equivalence. A proof-term is closed if it contains no free variables. We write $(u/x)t$ for the substitution of u for x in t and if $FV(t) \subseteq \{x\}$, we also use the notation $t\{u\}$.

The typing rules are those of Figure 2. These typing rules are exactly deduction rules of linear natural deduction for the multiplicative truth, the additive falsehood, the multiplicative implication, the additive conjunction, and the additive disjunction, with proof-terms, with two differences: the interstitial rules and the scalars.

The reduction rules are those of Figure 3. As usual, the reduction relation is written $\longrightarrow$, its inverse $\longleftarrow$, its reflexive-transitive closure $\longrightarrow^*$, the reflexive-transitive closure of its inverse ${}^*\!\!\longleftarrow$, and its reflexive-symmetric-transitive closure $\equiv$. The first six rules correspond to the reduction of cuts on the connectives $\top$, $\Rightarrow$, $\wedge$, and $\vee$. The eight others enable to commute the interstitial rules sum and prod with the introduction rules of the connectives $\top$, $\Rightarrow$, and $\wedge$, and with the elimination rule of the connective $\vee$. For instance, the rule

$$\langle t, u\rangle + \langle v, w\rangle \longrightarrow \langle t + v, u + w\rangle$$

pushes the symbol $+$ inside the pair. In a calculus without scalars we would have the zero-ary commutation rules

$$\star + \star \longrightarrow \star \qquad\qquad \bullet\star \longrightarrow \star$$

In the rules with scalars the scalars are added in the first case and multiplied in the second

$$a.\star + b.\star \longrightarrow (a+b).\star \qquad\qquad a \bullet b.\star \longrightarrow (a \times b).\star$$

▶ Remark 2.1. The rule $\langle t, u\rangle + \langle v, w\rangle \longrightarrow \langle t + v, u + w\rangle$ is possible because the conjunction is additive. If it were multiplicative, from $\Gamma \vdash \langle t, u\rangle : A \wedge B$ and $\Gamma \vdash \langle v, w\rangle : A \wedge B$, we could deduce that there exist Γ_1, Γ_2, Γ'_1, Γ'_2 such that $\Gamma_1 \vdash t : A$, $\Gamma_2 \vdash u : B$, $\Gamma'_1 \vdash v : A$, $\Gamma'_2 \vdash w : B$, and $\Gamma_1, \Gamma_2 = \Gamma'_1, \Gamma'_2 = \Gamma$, but Γ_1 and Γ'_1 could be different, and we would not be able to type $t + v$. This is the justification for the choice of the additive disjunction in the $\mathcal{L}^{\mathcal{S}}$-calculus and the exclusion of the multiplicative one. This remark is also key in the subject reduction proof below.

The $\mathcal{L}^{\mathcal{S}}$-calculus has the subject reduction, confluence, termination, and introduction properties. The subject reduction property is non trivial as shown by the remark above, but its proof uses standard methods. The termination property and the introduction properties are consequences of the termination and of the introduction properties of the $\odot^{\mathcal{S}}$-calculus. The full proofs are given in the long version of the paper.

▶ **Theorem 2.2** (Subject reduction). *If $\Gamma \vdash t : A$ and $t \longrightarrow u$, then $\Gamma \vdash u : A$.*

▶ **Theorem 2.3** (Confluence). *The $\mathcal{L}^{\mathcal{S}}$-calculus is confluent.*

▶ **Theorem 2.4** (Termination). *The $\mathcal{L}^{\mathcal{S}}$-calculus is strongly terminating.*

▶ **Theorem 2.5** (Introduction). *Let t be a closed irreducible proof of A.*

- *If A has the form $\top$, then t has the form $a.\star$.*
- *The proposition A is not $\bot$.*
- *If A has the form $B \Rightarrow C$, then t has the form $\lambda x.u$.*
- *If A has the form $B \wedge C$, then t has the form $\langle u, v\rangle$.*
- *If A has the form $B \vee C$, then t has the form $inl(u)$, $inr(u)$, $u + v$, or $a \bullet u$.*

3 Vectors

As there is one rule $\top$-i for each scalar a, there is one closed irreducible proof $a.\star$ for each scalar a. Thus, the closed irreducible proofs $a.\star$ of $\top$ are in one-to-one correspondence with the elements of $\mathcal{S}$. Therefore, the proofs $\langle a.\star, b.\star\rangle$ of $\top \wedge \top$ are in one-to-one with the elements of $\mathcal{S}^2$, the proofs $\langle\langle a.\star, b.\star\rangle, c.\star\rangle$ of $(\top \wedge \top) \wedge \top$, and also the proofs $\langle a.\star, \langle b.\star, c.\star\rangle\rangle$ of $\top \wedge (\top \wedge \top)$, are in one-to-one correspondence with the elements of $\mathcal{S}^3$, etc.

Thus, as any vector space of finite dimension n is isomorphic to $\mathcal{S}^n$, we have a way to express the vectors of any $\mathcal{S}$-vector space of finite dimension. Yet, choosing an isomorphism between a vector space and $\mathcal{S}^n$ amounts to choosing a basis in this vector space, thus the expression of a vector depends on the choice of a basis. This situation is analogous to that of matrix formalisms. Matrices can represent vectors and linear functions, but the matrix representation is restricted to finite dimensional vector spaces, and the representation of a vector depends on the choice of a basis. A change of basis in the vector space is reflected by the use of a transformation matrix.

▶ **Definition 3.1** (The set $\mathcal{V}$). *The set $\mathcal{V}$ is inductively defined as follows: $\top \in \mathcal{V}$, and if A and B are in $\mathcal{V}$, then so is $A \wedge B$.*

We now show that if $A \in \mathcal{V}$, then the set of closed irreducible proofs of A has a vector space structure.

▶ **Definition 3.2** (Zero vector). *If $A \in \mathcal{V}$, we define the proof 0_A of A by induction on A. If $A = \top$, then $0_A = 0.\star$. If $A = A_1 \wedge A_2$, then $0_A = \langle 0_{A_1}, 0_{A_2}\rangle$.*

▶ **Definition 3.3** (Additive inverse). *If $A \in \mathcal{V}$, and t is a proof of A, we define the proof $-t$ of A by induction on A. If $A = \top$, then t reduces to $a.\star$, we let $-t = (-a).\star$. If $A = A_1 \wedge A_2$, t reduces to $\langle t_1, t_2\rangle$ where t_1 is a proof of A_1 and t_2 of A_1. We let $-t = \langle -t_1, -t_2\rangle$.*

▶ **Lemma 3.4.** *If $A \in \mathcal{V}$ and t, t_1, t_2, and t_3 are closed proofs of A, then*

1. $(t_1 \boldsymbol{+} t_2) \boldsymbol{+} t_3 \equiv t_1 \boldsymbol{+} (t_2 \boldsymbol{+} t_3)$
2. $t_1 \boldsymbol{+} t_2 \equiv t_2 \boldsymbol{+} t_1$
3. $t \boldsymbol{+} 0_A \equiv t$
4. $t \boldsymbol{+} -t \equiv 0_A$
5. $a \bullet b \bullet t \equiv (a \times b) \bullet t$
6. $1 \bullet t \equiv t$
7. $a \bullet (t_1 \boldsymbol{+} t_2) \equiv a \bullet t_1 \boldsymbol{+} a \bullet t_2$
8. $(a + b) \bullet t \equiv a \bullet t \boldsymbol{+} b \bullet t$

▶ **Definition 3.5** (Dimension of a propositon in $\mathcal{V}$). *To each proposition $A \in \mathcal{V}$, we associate a positive natural number $d(A)$, which is the number of occurrences of the symbol $\top$ in A: $d(\top) = 1$ and $d(B \wedge C) = d(B) + d(C)$.*

If $A \in \mathcal{V}$ and $d(A) = n$, then the closed normal proofs of A and the vectors of $\mathcal{S}^n$ are in one-to-one correspondence: to each closed irreducible proof t of A, we associate a vector $\underline{t}$ of $\mathcal{S}^n$ and to each vector $\mathbf{u}$ of $\mathcal{S}^n$, we associate a closed irreducible proof $\overline{\mathbf{u}}^A$ of A.

▶ **Definition 3.6** (One-to-one correspondance). *Let $A \in \mathcal{V}$ with $d(A) = n$. To each closed irreducible proof t of A, we associate a vector $\underline{t}$ of $\mathcal{S}^n$ as follows.*

- *If $A = \top$, then $t = a.\star$. We let $\underline{t} = \left(\begin{smallmatrix} a \end{smallmatrix}\right)$.*
- *If $A = A_1 \wedge A_2$, then $t = \langle u, v\rangle$. We let $\underline{t}$ be the vector with two blocks $\underline{u}$ and $\underline{v}$: $\underline{t} = \left(\begin{smallmatrix} \underline{u} \\ \underline{v} \end{smallmatrix}\right)$.*

To each vector $\mathbf{u}$ of $\mathcal{S}^n$, we associate a closed irreducible proof $\overline{\mathbf{u}}^A$ of A.

- *If $n = 1$, then $\mathbf{u} = \left(\begin{smallmatrix} a \end{smallmatrix}\right)$. We let $\overline{\mathbf{u}}^A = a.\star$.*
- *If $n > 1$, then $A = A_1 \wedge A_2$, let n_1 and n_2 be the dimensions of A_1 and A_2. Let $\mathbf{u}_1$ and $\mathbf{u}_2$ be the two blocks of $\mathbf{u}$ of n_1 and n_2 lines, so $\mathbf{u} = \left(\begin{smallmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{smallmatrix}\right)$. We let $\overline{\mathbf{u}}^A = \langle \overline{\mathbf{u}_1}^{A_1}, \overline{\mathbf{u}_2}^{A_2}\rangle$.*

We extend the definition of $\underline{t}$ to any closed proof of A, $\underline{t}$ is by definition $\underline{t'}$ where t' is the irreducible form of t.

The next lemmas show that the symbol $\boldsymbol{+}$ expresses the sum of vectors and the symbol $\bullet$, the product of a vector by a scalar.

▶ **Lemma 3.7** (Sum of two vectors). *Let $A \in \mathcal{V}$, and u and v be two closed proofs of A. Then, $\underline{u \boldsymbol{+} v} = \underline{u} + \underline{v}$.*

▶ **Lemma 3.8** (Product of a vector by a scalar). *Let $A \in \mathcal{V}$ and u be a closed proof of A. Then $\underline{a \bullet u} = a\underline{u}$.*

▶ Remark 3.9. We have seen that the rules

$$a.\star \boldsymbol{+} b.\star \longrightarrow (a + b).\star \qquad a \bullet b.\star \longrightarrow (a \times b).\star$$
$$\langle t, u\rangle \boldsymbol{+} \langle v, w\rangle \longrightarrow \langle t \boldsymbol{+} v, u \boldsymbol{+} w\rangle \qquad a \bullet \langle t, u\rangle \longrightarrow \langle a \bullet t, a \bullet u\rangle$$

come from the rules of a calculus without scalars

$$\star \boldsymbol{+} \star \longrightarrow \star \qquad \bullet\star \longrightarrow \star$$
$$\langle t, u\rangle \boldsymbol{+} \langle v, w\rangle \longrightarrow \langle t \boldsymbol{+} v, u \boldsymbol{+} w\rangle \qquad \bullet\langle t, u\rangle \longrightarrow \langle \bullet t, \bullet u\rangle$$

that are commutation rules between the interstitial rules, sum and prod, and introduction rules $\top$-i and $\wedge$-i.

Now, these rules appear to be also vector calculation rules.

4 Matrices

We now want to prove that if $A, B \in \mathcal{V}$ with $d(A) = m$ and $d(B) = n$, and F is a linear function from $\mathcal{S}^m$ to $\mathcal{S}^n$, then there exists a closed proof f of $A \Rightarrow B$ such that, for all vectors $\mathbf{u} \in \mathcal{S}^m$, $\underline{f\ \overline{\mathbf{u}}^A} = F(\mathbf{u})$. This can equivalently be formulated as the fact that if M is a matrix with m columns and n lines, then there exists a closed proof f of $A \Rightarrow B$ such that for all vectors $\mathbf{u} \in \mathcal{S}^m$, $\underline{f\ \overline{\mathbf{u}}^A} = M\mathbf{u}$.

This theorem has been proved for the $\odot^{\mathcal{S}}$-calculus in [5, long version]. The proof of the following theorem is just a check that the construction given there verifies the linearity constraints of the $\mathcal{L}^{\mathcal{S}}$-calculus.

▶ **Theorem 4.1** (Matrices). *Let $A, B \in \mathcal{V}$ with $d(A) = m$ and $d(B) = n$ and let M be a matrix with m columns and n lines, then there exists a closed proof t of $A \Rightarrow B$ such that, for all vectors $\mathbf{u} \in \mathcal{S}^m$, $\underline{t\ \overline{\mathbf{u}}^A} = M\mathbf{u}$.*

Proof. By induction on A.

- If $A = \top$, then M is a matrix of one column and n lines. Hence, it is also a vector of n lines. We take

  $$t = \lambda x.\delta_\top(x, \overline{M}^B)$$

  Let $\mathbf{u} \in \mathcal{S}^1$, $\mathbf{u}$ has the form $\left(\begin{smallmatrix} a \end{smallmatrix}\right)$ and $\overline{\mathbf{u}}^A = a.\star$.
  Then, using Lemma 3.8, we have $\underline{t\ \overline{\mathbf{u}}^A} = \underline{\delta_\top(\overline{\mathbf{u}}^A, \overline{M}^B)} = \underline{\delta_\top(a.\star, \overline{M}^B)} = \underline{a \bullet \overline{M}^B} = a\underline{\overline{M}^B} = aM = M\left(\begin{smallmatrix} a \end{smallmatrix}\right) = M\mathbf{u}$.
- If $A = A_1 \wedge A_2$, then let $d(A_1) = m_1$ and $d(A_2) = m_2$. Let M_1 and M_2 be the two blocks of M of m_1 and m_2 columns, so $M = \left(\begin{smallmatrix} M_1 & M_2 \end{smallmatrix}\right)$.
  By induction hypothesis, there exist closed proofs t_1 and t_2 of the propositions $A_1 \Rightarrow B$ and $A_2 \Rightarrow B$ such that, for all vectors $\mathbf{u}_1 \in \mathcal{S}^{m_1}$ and $\mathbf{u}_2 \in \mathcal{S}^{m_2}$, we have $\underline{t_1\ \overline{\mathbf{u}_1}^{A_1}} = M_1\mathbf{u}_1$ and $\underline{t_2\ \overline{\mathbf{u}_2}^{A_2}} = M_2\mathbf{u}_2$. We take

  $$t = \lambda x.(\delta^1_\wedge(x, y.(t_1\ y)) \boldsymbol{+} \delta^2_\wedge(x, z.(t_2\ z)))$$

  Let $\mathbf{u} \in \mathcal{S}^m$, and $\mathbf{u}_1$ and $\mathbf{u}_2$ be the two blocks of m_1 and m_2 lines of $\mathbf{u}$, so $\mathbf{u} = \left(\begin{smallmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{smallmatrix}\right)$, and $\overline{\mathbf{u}}^A = \langle \overline{\mathbf{u}_1}^{A_1}, \overline{\mathbf{u}_2}^{A_2} \rangle$.
  Then, using Lemma 3.7, $\underline{t\ \overline{\mathbf{u}}^A} = \underline{\delta^1_\wedge(\langle \overline{\mathbf{u}_1}^{A_1}, \overline{\mathbf{u}_2}^{A_2} \rangle, y.(t_1\ y)) \boldsymbol{+} \delta^2_\wedge(\langle \overline{\mathbf{u}_1}^{A_1}, \overline{\mathbf{u}_2}^{A_2} \rangle, z.(t_2\ z))} = \underline{(t_1\ \overline{\mathbf{u}_1}^{A_1}) \boldsymbol{+} (t_2\ \overline{\mathbf{u}_2}^{A_2})} = \underline{t_1\ \overline{\mathbf{u}_1}^{A_1}} + \underline{t_2\ \overline{\mathbf{u}_2}^{A_2}} = M_1\mathbf{u}_1 + M_2\mathbf{u}_2 = \left(\begin{smallmatrix} M_1 & M_2 \end{smallmatrix}\right)\left(\begin{smallmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{smallmatrix}\right) = M\mathbf{u}$. ◀

▶ Remark 4.2. In the proofs $\delta_\top(x, \overline{M}^B)$, $\delta^1_\wedge(x, y.(t_1\ y)$, and $\delta^2_\wedge(x, z.(t_2\ z))$, the variable x occurs in one argument of the symbols $\delta_\top$, $\delta^1_\wedge$, and $\delta^2_\wedge$, but not in the other. In contrast, in the proof $\delta^1_\wedge(x, y.(t_1\ y)) \boldsymbol{+} \delta^2_\wedge(x, z.(t_2\ z))$, it occurs in both arguments of the symbol $\boldsymbol{+}$. Thus, these proofs are well-typed in the system of Figure 2.

▶ Remark 4.3. The rules

$$\begin{array}{rcl} \delta_\top(a.\star, t) \longrightarrow a \bullet t & \qquad & \delta^1_\wedge(\langle t, u \rangle, x.v) \longrightarrow (t/x)v \\ (\lambda x.t)\ u \longrightarrow (u/x)t & & \delta^2_\wedge(\langle t, u \rangle, x.v) \longrightarrow (u/x)v \end{array}$$

were introduced as cut reduction rules.

Now, these rules appear to be also matrix calculation rules.

▶ **Example 4.4** (Matrices with two colums and two lines). The matrix $\left(\begin{smallmatrix} a & c \\ b & d \end{smallmatrix}\right)$ is expressed as the proof

$$t = \lambda x.(\delta^1_\wedge(x, y.\delta_\top(y, \langle a.\star, b.\star \rangle)) \boldsymbol{+} \delta^2_\wedge(x, z.\delta_\top(z, \langle c.\star, d.\star \rangle)))$$

And applying the rules of Figure 3, we get

$$t\ \langle e.\star, f.\star\rangle \longrightarrow^* \langle (a \times e + c \times f).\star, (b \times e + d \times f).\star\rangle$$

5 Linearity

We now prove the converse: if $A, B \in \mathcal{V}$, then each proof t of $A \Rightarrow B$ expresses a linear function, that is

$$t\ (u \mathbin{+} v) \equiv (t\ u) \mathbin{+} (t\ v) \qquad \text{and} \qquad t\ (a \bullet u) \equiv a \bullet (t\ u)$$

A first idea could be to generalize this statement and prove that these properties hold for all closed proofs t, whatever their type. But this generalization is too strong. For example, if $A = \top$ and $B = (\top \Rightarrow \top) \Rightarrow \top$, $t = \lambda x.\lambda y.(y\ x)$ is a proof of $A \Rightarrow B$, but

$$t\ (1.\star \mathbin{+} 2.\star) \longrightarrow^* \lambda y.(y\ 3.\star) \qquad \text{and} \qquad t\ 1.\star \mathbin{+} t\ 2.\star \longrightarrow^* \lambda y.((y\ 1.\star) \mathbin{+} (y\ 2.\star))$$

and these two irreducible proofs are different. So we will prove that these properties hold when A is arbitrary and $B \in \mathcal{V}$.

▶ Remark 5.1. The fact that we want all proofs of $\top \Rightarrow \top$ to be linear functions from $\mathcal{S}$ to $\mathcal{S}$ explains why the symbol $\top$ must be multiplicative. If it were additive, the proposition $\top \Rightarrow \top$ would have the proof $f = \lambda x.\ (1.\star)$ that is not linear as $f\ (1.\star \mathbin{+} 1.\star) \longrightarrow^* 1.\star \not\equiv 2.\star\ {}^*\!\!\longleftarrow (f\ 1.\star) \mathbin{+} (f\ 1.\star)$.

▶ Remark 5.2. The fact that we want all proofs of $\top \Rightarrow \top$ to be linear functions from $\mathcal{S}$ to $\mathcal{S}$ explains why the rule sum must be additive. If it were multiplicative, the proposition $\top \Rightarrow \top$ would have the proof $g = \lambda x.\ (x \mathbin{+} 1.\star)$ that is not linear as $g\ (1.\star \mathbin{+} 1.\star) \longrightarrow^* 3.\star \not\equiv 4.\star\ {}^*\!\!\longleftarrow (g\ 1.\star) \mathbin{+} (g\ 1.\star)$.

5.1 Size of a proof

The proof of the linearity theorem proceeds by induction on the size of the proof, and the first part of this proof is the definition of such a size function μ. Our goal could be to build a size function such that if t is proof of B in a context $\Gamma, x : A$ and u is a proof of A, then $\mu((u/x)t) = \mu(t) + \mu(u)$. This would be the case, for the usual notion of size, if x had exactly one occurrence in t. But, due to additive connectives, the variable x may have zero, one, or several occurrences in t.

First, as the rule $\bot$-e is additive, it may happen that $\delta_\bot(t)$ is a proof in the context $\Gamma, x : A$, and x has no occurrence in t. Thus, we lower our expectations to $\mu((u/x)t) \leq \mu(t) + \mu(u)$, which is sufficient for the linearity theorem.

Then, as the rules $\mathbin{+}$, $\wedge$-i, and $\vee$-e rules are additive, if $u \mathbin{+} v$ is proof of B in a context $\Gamma, x : A$, x may occur both in u and in v. And the same holds for the proofs $\langle u, v\rangle$, and $\delta_\vee(t, x.u, y.v)$. In these cases, we modify the definition of the size function and take $\mu(t \mathbin{+} u) = 1 + \max(\mu(t), \mu(u))$, instead of $\mu(t \mathbin{+} u) = 1 + \mu(t) + \mu(u)$, etc. making the function μ a mix between a size function and a depth function. Note that the depth function itself cannot be used, as Lemma 5.8 does not hold for the depth function.

This leads to the following definition.

▶ **Definition 5.3** (Size of a proof).
- $\mu(x) = 0$
- $\mu(t \mathbin{+} u) = 1 + \max(\mu(t), \mu(u))$
- $\mu(a \bullet t) = 1 + \mu(t)$

- $\mu(a.\star) = 1$
- $\mu(\delta_\top(t,u)) = 1 + \mu(t) + \mu(u)$,
- $\mu(\delta_\bot(t)) = 1 + \mu(t)$
- $\mu(\lambda x.t) = 1 + \mu(t)$
- $\mu(t\ u) = 1 + \mu(t) + \mu(u)$
- $\mu(\langle t,u\rangle) = 1 + \max(\mu(t),\mu(u))$
- $\mu(\delta^1_\wedge(t,y.u)) = 1 + \mu(t) + \mu(u)$
- $\mu(\delta^2_\wedge(t,y.u)) = 1 + \mu(t) + \mu(u)$
- $\mu(inl(t)) = 1 + \mu(t)$
- $\mu(inr(t)) = 1 + \mu(t)$
- $\mu(\delta_\vee(t,y.u,z.v)) = 1 + \mu(t) + \max(\mu(u),\mu(v))$

▶ **Lemma 5.4.** *If $\Gamma, x : A \vdash t : B$ and $\Delta \vdash u : B$ then $\mu((u/x)t) \leq \mu(t) + \mu(u)$.*

▶ **Example 5.5.** Let $t = \delta_\bot(y)$ and $u = 1.\star$. We have $y : \bot, x : \top \vdash t : C$, $\mu(t) = 1$, $\mu(u) = 1$ and $\mu((u/x)t) = 1$. Thus $\mu((u/x)t) \leq \mu(t) + \mu(u)$.

As a corollary, we get a similar size preservation theorem for reduction.

▶ **Lemma 5.6.** *If $t \longrightarrow u$, then $\mu(t) \geq \mu(u)$.*

5.2 Elimination contexts

The second part of the proof is a standard generalization of the notion of head variable. In the λ-calculus, we can decompose a term t as a sequence of applications $t = u\ v_1\ \ldots\ v_n$, with terms $v_1, \ldots, v_n$ and a term u, which is not an application. Then u may either be a variable, in which case it is the head variable of the term, or an abstraction.

In a similar way, any proof in the $\mathcal{L}^\mathcal{S}$-calculus can be decomposed into a sequence of elimination rules, forming an elimination context, and a proof u that is either a variable, an introduction, a sum, or a product.

▶ **Definition 5.7** (Elimination context)**.** *An elimination context is a proof with a single free variable, written _, that is in the language*

$$K = _ \mid \delta_\top(K,u) \mid \delta_\bot(K) \mid K\ t \mid \delta^1_\wedge(K,x.r) \mid \delta^2_\wedge(K,x.r) \mid \delta_\vee(K,x.r,y.s)$$

where u is a closed proof, $FV(r) \subseteq \{x\}$, and $FV(s) \subseteq \{y\}$.

In the case of elimination contexts, Lemma 5.4 can be strengthened.

▶ **Lemma 5.8.** $\mu(K\{t\}) = \mu(K) + \mu(t)$

Note that in Example 5.5, $(_/x)t$ is not a context as _ does not occur in it.

▶ **Lemma 5.9** (Decomposition of a proof)**.** *If t is an irreducible proof such that $x : C \vdash t : A$, then there exist an elimination context K, a proof u, and a proposition B such that $_ : B \vdash K : A$, $x : C \vdash u : B$, u is either the variable x, an introduction, a sum, or a product, and $t = K\{u\}$.*

A final lemma shows that, in the same way we can always decompose a non-empty list into a smaller list and its last element, we can always decompose an elimination context K different from _ into an elimination context K_1 and a last elimination rule K_2.

▶ **Lemma 5.10** (Decomposition of an elimination context). *If K is a elimination context such that $_ : A \vdash K : B$ and $K \neq _$, then K has the form $K_1\{K_2\}$, and*

- *if $A = \top$, then K_2 has the form $\delta_\top(_, t)$,*
- *if $A = \bot$, then K_2 has the form $\delta_\bot(_)$,*
- *if $A = B \Rightarrow C$, then K_2 has the form $_\ t$,*
- *if $A = B \wedge C$, then K_2 has the form $\delta^1_\wedge(_, x.t)$ or $\delta^2_\wedge(_, x.t)$,*
- *if $A = B \vee C$, then K_2 has the form $\delta_\vee(_, x_1.t_1, x_2.t_2)$.*

5.3 Linearity

We now have the tools to prove the linearity theorem. Instead of proving the theorem for a closed proof t of $A \Rightarrow B$, it is more convenient to prove it for a proof t of B in the context $x : A$. The result for the proofs of $A \Rightarrow B$ is Corollary 5.12.

▶ **Theorem 5.11** (Linearity). *For every proposition A, proposition $B \in \mathcal{V}$, proofs t, u_1, and u_2, such that $x : A \vdash t : B$, t is irreducible, $\vdash u_1 : A$, and $\vdash u_2 : A$. Then*

$$t\{u_1 \mathbin{+} u_2\} \equiv t\{u_1\} \mathbin{+} t\{u_2\} \qquad \textit{and} \qquad t\{a \bullet u_1\} \equiv a \bullet t\{u_1\}$$

Proof (Sketch). The proof proceeds by induction on the size $\mu(t)$ of the proof t, but the organization of the cases is complex. We first consider the different forms for t: it can be a variable, a sum, a product, an introduction, or an elimination. If it is an introduction, as $B \in \mathcal{V}$, it must be a pair. The key point here is that taking $B \in \mathcal{V}$, we avoid the case of the abstraction that would lead to a failure, as show by the counter-example above.

The case where t is an elimination leads to a second case analysis. We first use Lemma 5.9 to decompose the proof t into an elimination context K and a proof v. Then we consider the different possible forms of v: it can be neither an introduction nor an elimination, hence it is a variable, a sum, or a product.

Finally, in the case it is a variable, we have a third case analysis: we use Lemma 5.10 to decompose K into a context K' and a last elimination rule and we consider the different possible cases for this last elimination rule.

In all these cases, we use various cases of the Lemma 3.4 to prove the convertibility of the proofs, and the Lemmas 5.4, 5.6, and 5.8 to show that the induction hypothesis applies to smaller proofs. ◀

▶ **Corollary 5.12.** *Let A be a proposition and $B \in \mathcal{V}$. Let t be a closed proof of $A \Rightarrow B$ and u and v be closed proofs of A. Then*

$$t\ (u \mathbin{+} v) \equiv (t\ u) \mathbin{+} (t\ v) \qquad \textit{and} \qquad t\ (a \bullet u) \equiv a \bullet (t\ u)$$

▶ Remark 5.13. As we have seen, Corollary 5.12 does not generalize when $B \notin \mathcal{V}$. For example, $t = \lambda x.\lambda y.(y\ x)$ is a closed irreducible form of $\top \Rightarrow (\top \Rightarrow \top) \Rightarrow \top$, but the proofs $t\ (1.\star \mathbin{+} 2.\star)$ and $t\ 1.\star \mathbin{+} t\ 2.\star$ are not convertible. Indeed

$$t\ (1.\star \mathbin{+} 2.\star) \longrightarrow^* \lambda y.(y\ 3.\star) \qquad \text{and} \qquad t\ 1.\star \mathbin{+} t\ 2.\star \longrightarrow^* \lambda y.((y\ 1.\star) \mathbin{+} (y\ 2.\star))$$

and the two irreducible proofs $\lambda y.(y\ 3.\star)$ and $\lambda y.((y\ 1.\star) \mathbin{+} (y\ 2.\star))$ are different.

Yet, these two proofs are observationally equivalent: if $B \in \mathcal{V}$ and s is a closed proof of $((\top \Rightarrow \top) \Rightarrow \top) \Rightarrow B$

$$s\ (t\ (1.\star \mathbin{+} 2.\star)) \equiv s\ (t\ 1.\star \mathbin{+} t\ 2.\star)$$

$$\frac{\Gamma \vdash t : A \quad \Gamma \vdash u : B}{\Gamma \vdash [t,u] : A \odot B}\ \odot\text{-i}$$

$$\frac{\Gamma \vdash t : A \odot B \quad \Delta, x : A \vdash u : C}{\Gamma, \Delta \vdash \delta^1_\odot(t, x.u) : C}\ \odot\text{-e1}$$

$$\frac{\Gamma \vdash t : A \odot B \quad \Delta, x : B \vdash u : C}{\Gamma, \Delta \vdash \delta^2_\odot(t, x.u) : C}\ \odot\text{-e2}$$

$$\frac{\Gamma \vdash t : A \odot B \quad \Delta, x : A \vdash u : C \quad \Delta, y : B \vdash v : C}{\Gamma, \Delta \vdash \delta_\odot(t, x.u, y.v) : C}\ \odot\text{-e}$$

Figure 4 The deduction rules of the $\mathcal{L}\odot^{\mathcal{S}}$-calculus.

Indeed, applying Corollary 5.12 with the proof $\lambda x\ (s\ (t\ x))$, we obtain that the first proof is convertible with $(s\ (t\ 1.\star)) + (s\ (t\ 2.\star))$ and applying it to s we obtain that the second is convertible with this same proof.

▶ **Corollary 5.14.** *Let $A, B \in \mathcal{V}$, such that $d(A) = m$ and $d(B) = n$, and t be a closed proof of $A \Rightarrow B$. Then the function F from $\mathcal{S}^m$ to $\mathcal{S}^n$, defined as $F(\mathbf{u}) = \underline{t\ \overline{\mathbf{u}}^A}$ is linear.*

5.4 No-cloning

In the PL$^{\mathcal{S}}$-calculus, that is in the proof language of propositional logic extended with interstitial rules and scalars, the cloning function from $\mathcal{S}^2$ to $\mathcal{S}^4$, mapping $\left(\begin{smallmatrix} a \\ b \end{smallmatrix}\right)$ to $\left(\begin{smallmatrix} a^2 \\ ab \\ ab \\ b^2 \end{smallmatrix}\right)$ can be expressed with the proof of $(\top \wedge \top) \Rightarrow ((\top \wedge \top) \wedge (\top \wedge \top))$

$$\begin{aligned}&\lambda x.\delta^1_\wedge(x,y.\delta^1_\wedge(x, y_1.\langle\langle\delta_\top(y, y_1), 0.\star\rangle, \langle 0.\star, 0.\star\rangle\rangle) + \delta^2_\wedge(x, z_1.\langle\langle 0.\star, \delta_\top(y, z_1)\rangle, \langle 0.\star, 0.\star\rangle\rangle))\\ &\quad +\\ &\quad \delta^2_\wedge(x,z.\delta^1_\wedge(x, y_2.\langle\langle 0.\star, 0.\star\rangle, \langle\delta_\top(z, y_2), 0.\star\rangle\rangle) + \delta^2_\wedge(x, z_2.\langle\langle 0.\star, 0.\star\rangle, \langle 0.\star, \delta_\top(z, z_2)\rangle\rangle))\end{aligned}$$

This proof cannot be expressed in $\mathcal{L}^{\mathcal{S}}$-calculus, as it uses twice an elimination symbol of the conjunction with the variable x occurring in both arguments.

Moreover, by Corollary 5.14, this function cannot be expressed as a proof of the proposition $(\top \wedge \top) \Rightarrow ((\top \wedge \top) \wedge (\top \wedge \top))$ in the $\mathcal{L}^{\mathcal{S}}$-calculus, because it is not linear.

6 The $\mathcal{L}\odot^{\mathcal{S}}$-calculus and its application to quantum computing

6.1 The $\mathcal{L}\odot^{\mathcal{S}}$-calculus

The $\mathcal{L}\odot^{\mathcal{S}}$-calculus is obtained by adding the symbols $[,]$, $\delta^1_\odot$, $\delta^2_\odot$, $\delta_\odot$, the deduction rules of Figure 4, and the reduction rules of Figure 5, to the $\mathcal{L}^{\mathcal{S}}$-calculus. It is similar to the $\odot^{\mathcal{S}}$-calculus [5, long version] except that its typing rules are linear.

6.2 Quantum computing

Like the $\odot^{\mathbb{C}}$-calculus, the $\mathcal{L}\odot^{\mathbb{C}}$-calculus, with a reduction strategy restricting the reduction of $\delta_\odot([t,u], x.v, y.w)$ to the cases where t and u are closed irreducible proofs, can be used to express quantum algorithms. The following reproduces the Section 4 of [5, long version], focusing on the differences due to linearity.

$$\begin{aligned}
\delta^1_\odot([t,u],x.v) &\longrightarrow (t/x)v \\
\delta^2_\odot([t,u],x.v) &\longrightarrow (u/x)v \\
\delta_\odot([t,u],x.v,y.w) &\longrightarrow (t/x)v \\
\delta_\odot([t,u],x.v,y.w) &\longrightarrow (u/y)w \\
\\
[t,u] \boldsymbol{+} [v,w] &\longrightarrow [t \boldsymbol{+} v, u \boldsymbol{+} w] \\
a \bullet [t,u] &\longrightarrow [a \bullet t, a \bullet u]
\end{aligned}$$

Figure 5 The reduction rules of the $\mathcal{L}\odot^S$-calculus.

Bits can be expressed as proofs of the proposition $\top \vee \top$: $\mathbf{0} = \mathit{inl}(1.\star)$ and $\mathbf{1} = \mathit{inr}(1.\star)$. The test operation was defined in [5, long version] as

$$\mathit{if}(t,u,v) = \delta_\vee(t, x.u, y.v)$$

where x and y are variables not occurring in u and v. But this proof is not linear, so we define it as

$$\mathit{if}(t,u,v) = \delta_\vee(t, x.\delta_\top(x,u), y.\delta_\top(y,v))$$

Note that $\mathit{if}(\mathbf{0},u,v) \longrightarrow 1 \bullet u$ and $\mathit{if}(\mathbf{1},u,v) \longrightarrow 1 \bullet v$.

Then, we express the vectors, like in Section 3, except that we use the connective $\odot$ instead of $\wedge$. For instance, the vector $\binom{a}{b}$ is not expressed as the proof $\langle a.\star, b.\star\rangle$ but as the proof $[a.\star, b.\star]$, etc. In particular n-qubit, for $n \geq 1$, are expressed, in the basis $|0\ldots00\rangle, |0\ldots01\rangle, \ldots |1\ldots11\rangle$, as elements of $\mathbb{C}^{2^n}$, that is as proofs of the proposition $\mathcal{Q}_n$ defined by induction on n as follows: $\mathcal{Q}_0 = \top$ and $\mathcal{Q}_{n+1} = \mathcal{Q}_n \odot \mathcal{Q}_n$.

If t is a closed irreducible proof of $\mathcal{Q}_n$, we define the square of the norm $\|t\|^2$ of t by induction on n.

- If $n = 0$, then $t = a.\star$ and we take $\|t\|^2 = |a|^2$.
- If $n = n' + 1$, then $t = [u_1, u_2]$ and we take $\|t\|^2 = \|u_1\|^2 + \|u_2\|^2$.

We take the convention that any closed irreducible proof u of $\mathcal{Q}_n$, expressing a non-zero vector $\underline{u} \in \mathbb{C}^{2^n}$, is an alternative expression of the n-qubit $\frac{\underline{u}}{\|\underline{u}\|}$. For example, the qubit $\frac{1}{\sqrt{2}}.|0\rangle + \frac{1}{\sqrt{2}}.|1\rangle$ is expressed as the proof $[\frac{1}{\sqrt{2}}.\star, \frac{1}{\sqrt{2}}.\star]$, but also as the proof $[1.\star, 1.\star]$.

Matrices are expressed as in Section 4.

Like in [5, long version], thanks to the reduction strategy, probabilities can be assigned to the non-deterministic reductions of closed proofs of the form $\delta_\odot(u, x.v, y.w)$, that is proofs of the form $\delta_\odot([u_1, u_2], x.v, y.w)$.

If u_1 and u_2 are closed irreducible proofs of of $\mathcal{Q}_n$ and $\|u_1\|^2$ and $\|u_2\|^2$ are not both 0, then we assign the probability $\frac{\|u_1\|^2}{\|u_1\|^2+\|u_2\|^2}$ to the reduction

$$\delta_\odot([u_1,u_2], x.v, y.w) \longrightarrow (u_1/x)v$$

and the probability $\frac{\|u_2\|^2}{\|u_1\|^2+\|u_2\|^2}$ to the reduction

$$\delta_\odot([u_1,u_2], x.v, y.w) \longrightarrow (u_2/y)w$$

If $\|u_1\|^2 = \|u_2\|^2 = 0$, or u_1 and u_2 are proofs of propositions of a different form, we assign any probability, for example $\frac{1}{2}$, to both reductions.

If n is a non-zero natural number, we can define the measurement operator π_n, measuring the first qubit of an n-qubit, as the proof

$$\pi_n = \lambda x.\delta_\odot(x, y.[y, 0_{\mathcal{Q}_{n-1}}], z.[0_{\mathcal{Q}_{n-1}}, z])$$

of the proposition $\mathcal{Q}_n \Rightarrow \mathcal{Q}_n$.

Indeed, if t is a closed irreducible proof of $\mathcal{Q}_n$ of the form $[u_1, u_2]$, such that $\|t\|^2 = \|u_1\|^2 + \|u_2\|^2 \neq 0$, expressing the state of an n-qubit, then the proof $\pi_n\ t$ of the proposition $\mathcal{Q}_n$ reduces, with probabilities $\frac{\|u_1\|^2}{\|u_1\|^2+\|u_2\|^2}$ and $\frac{\|u_2\|^2}{\|u_1\|^2+\|u_2\|^2}$ to $[u_1, 0_{\mathcal{Q}_{n-1}}]$ and to $[0_{\mathcal{Q}_{n-1}}, u_2]$, that are the states of the n-qubit, after the partial measure of the first qubit.

Note that, as the $\mathcal{L}\odot^{\mathcal{S}}$-calculus is purely linear, it cannot express the measurement operator $\lambda x.\delta_\odot(x, y.\mathbf{0}, z.\mathbf{1})$ that returns the "classical" result of the measure and that could be expressed in the $\odot^{\mathcal{S}}$-calculus. Typing this measurement operator would require to extend the type system to express that, in the premises $\Delta, A \vdash C$ and $\Delta, B \vdash C$ of the rule $\odot$-e, the hypotheses A and B may be weakened.

Instead, our measurement operators return the state of the full system after the measure. In the first case, this state is a linear combination of the first 2^{n-1} vectors $|00\ldots0\rangle \ldots |01\ldots1\rangle$ of the basis: those starting with a 0, in the second, this state is a linear combination of the last 2^{n-1} vectors $|10\ldots0\rangle \ldots |11\ldots1\rangle$ of the basis: those starting with a 1. In the first case, the result of the measurement is $|0\rangle$, in the second it is $|1\rangle$.

As we have a representation of linear functions and measurement operators, we can express in the $\mathcal{L}\odot^{\mathcal{S}}$-calculus, all quantum algorithms, for instance Deutsch's algorithm.

6.3 Linearity

The main motivation for introducing this linear variant of the $\odot^{\mathbb{C}}$-calculus was to prove a linearity theorem for this calculus. But, the $\mathcal{L}\odot^{\mathbb{C}}$-calculus contains the $\delta_\odot$ symbol, that enables to express measurement operators, which are not linear.

Thus, our linearity theorem should be that using the $\delta_\odot$ symbol is the only way to construct a non-linear function. In other words, that, in the fragment of the $\mathcal{L}\odot^{\mathcal{S}}$-calculus excluding the $\delta_\odot$ symbol, only linear functions can be expressed. But, if $\odot$-e rule is excluded, $\odot$ is just another conjunction, and this fragment of the $\mathcal{L}\odot^{\mathcal{S}}$-logic is the $\mathcal{L}^{\mathcal{S}}$-logic with two copies of the conjunction. As a corollary of the Corollary 5.14, only linear functions can be expressed in this calculus and cloning cannot.

7 Conclusion

We can now attempt a possible answer to the question stated in the introduction: in which way must propositional logic be extended or restricted, so that its proof language becomes a quantum programming language. This answer is in four parts: we need to extend it with interstitial rules, scalars, and the connective $\odot$, and we need to restrict it by making it linear.

We obtain this way the $\mathcal{L}\odot^{\mathcal{S}}$-logic that addresses both the question of linearity and, for instance, avoids cloning, and that of the information-erasure, non-reversibility, and non-determinism of the measurement.

Another issue is to restrict the logic further so that linear functions are unitary. We can either enforce unitarity, following the methods of [1, 6, 7], or let these unitarity constraints as properties of the program that must be proved for each program, rather than enforced by the type system.

We may also wish to make this quantum representation more compositional, by considering some form of tensor product. Indeed, for example in Lineal [2] the tensor product is just the standard encoding of pairs, since in Lineal pairs are, by construction, bilinear, so a pair of superpositions (which are constructed with the symbol + in that language) such as $\langle \alpha_1.|0\rangle + \alpha_2.|1\rangle, \beta_1.|0\rangle + \beta_2.|1\rangle\rangle$ would reduce to $\alpha_1\beta_1.\langle|0\rangle,|0\rangle\rangle + \alpha_1\beta_2.\langle|0\rangle,|1\rangle\rangle + \alpha_2\beta_1.\langle|1\rangle,|0\rangle\rangle + \alpha_2\beta_2.\langle|1\rangle,|1\rangle\rangle$ which represents the four-dimensional vector obtained by the tensor product. It is not the case in the $\mathcal{L}\odot^{\mathcal{S}}$-calculus, since the pair does not commute with the sup, so $\langle[\alpha_1.\star,\alpha_2.\star],[\beta_1.\star,\beta_2.\star]\rangle$ is in normal form. However, we could encode a tensor product $\otimes_{n,m}$ as a proof term of $Q_n \wedge Q_m \Rightarrow Q_{n\times m}$, whose size depends on n and m, or, to be more in line with Lineal, where there is no dependency on the size, just introduce a new proof term for a new rule

$$\frac{\Gamma \vdash t : \top^n \quad \Delta \vdash r : \top^m}{\Gamma\Delta \vdash t \otimes r : \top^{n\times m}}\ \text{tens}$$

with the following reduction rules:

$$[t,r] \otimes s \longrightarrow [t\otimes s, r\otimes s] \qquad \text{and} \qquad \alpha.\star \otimes t \longrightarrow \alpha \bullet t$$

This way, $[\alpha_1.\star,\alpha_2.\star] \otimes [\beta_1.\star,\beta_2.\star] \longrightarrow^* [[\alpha_1\beta_1.\star,\alpha_1\beta_2.\star],[\alpha_2\beta_1.\star,\alpha_2\beta_2.\star]]$.

Also, it may be interesting to study if a tensor connective could be added, with some notion of equivalence where $Q_n \otimes Q_m \equiv Q_{n\times m}$, relating the $\odot$ connective with the $\otimes$ connective. We left this study for future work.

References

1. T. Altenkirch and J. Grattage. A functional quantum programming language. In *Proceedings of LICS 2005*, pages 249–258. IEEE, 2005.
2. P. Arrighi and G. Dowek. Lineal: A linear-algebraic lambda-calculus. *Logical Methods in Computer Science*, 13(1), 2017.
3. R. Blute. Hopf algebras and linear logic. *Mathematical Structures in Computer Science*, 6(2):189–217, 1996.
4. B. Coecke and A. Kissinger. *Picturing Quantum Processes: A First Course in Quantum Theory and Diagrammatic Reasoning*. Cambridge University Press, 2017. doi:10.1017/9781316219317.
5. A. Díaz-Caro and G. Dowek. A new connective in natural deduction, and its application to quantum computing. In A. Cerone and P. Csaba Ölveczky, editors, *Prooceedings of the International Colloquium on Theoretical Aspects of Computing*, volume 12819 of *Lecture Notes in Computer Science*, pages 175–193. Springer, 2021. Long version accessible at arXiv:2012.08994.
6. A. Díaz-Caro, M. Guillermo, A. Miquel, and B. Valiron. Realizability in the unitary sphere. In *Proceedings of the 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS 2019)*, pages 1–13, 2019.
7. A. Díaz-Caro and O. Malherbe. Quantum control in the unitary sphere: Lambda-S_1 and its categorical model. Draft at arXiv:2012.05887, 2020.
8. A. Díaz-Caro, G. Dowek, and J.P. Rinaldi. Two linearities for quantum computing in the lambda calculus. *Biosystems*, 2019.
9. Th. Ehrhard. On Köthe sequence spaces and linear logic. *Mathematical Structures in Computer Science*, 12(5):579–623, 2002.
10. J.-Y. Girard. Linear logic. *Theoreoretical Computer Science*, 50:1–102, 1987.
11. J.-Y. Girard. Coherent banach spaces: A continuous denotational semantics. *Theoretical Computer Science*, 227(1-2):275–297, 1999.

12 P. Selinger and B. Valiron. A lambda calculus for quantum computation with classical control. *Mathematical Structures in Computer Science*, 16(3):527–552, 2006.

13 L. Vaux. The algebraic lambda calculus. *Mathematical Structures in Computer Science*, 19(5):1029–1059, 2009.

14 M. Zorzi. On quantum lambda calculi: a foundational perspective. *Mathematical Structures in Computer Science*, 26(7):1107–1195, 2016.