Towards a Computational Quantum Logic: An Overview of an Ongoing Research Program*

Alejandro Díaz-Caro

[1] Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France
[2] Universidad Nacional de Quilmes, Bernal, Buenos Aires, Argentina

Abstract

This extended abstract provides an overview of a long-term collaboration aimed at extending the Curry-Howard-Lambek correspondence to the realm of quantum computation. I will introduce the Lambda-S calculus, a (partial) dual to intuitionistic linear logic whose proof terms serve as a foundation for a quantum programming language. Additionally, I will discuss the $\mathcal{L}^{\mathcal{S}}$ calculus, a proof language for intuitionistic linear logic, which also enables the construction of quantum programming languages. These frameworks offer a logical and computational foundation for reasoning about quantum programs and provide a glimpse into the structure of a potential quantum logic as the dual of linear logic.

Introduction. Quantum logic is a formal system inspired by the structure of quantum theory, originally developed by Garrett Birkhoff and John von Neumann [BvN36]. Unlike classical logic, which is based on Boolean algebra, quantum logic weakens the distributive law, leading to an orthocomplemented lattice structure. This formulation aligns with the mathematical properties of quantum mechanics, where propositions correspond to projections on a Hilbert space. However, while quantum logic has been explored as a foundational system for reasoning about quantum mechanics, its connection to computation has been less explored. Indeed, the connection between intuitionistic logic, typed lambda calculus, and Cartesian closed categories has been a fruitful area of research, with the Curry-Howard-Lambek correspondence [SU06, Cro93] providing a deep connection between these areas. If we are to extend this correspondence to quantum computation, we need to start from a logical foundation that captures the structure of quantum computation. In this extended abstract, I will provide an overview of a research program aimed at developing a *computational* quantum logic, which will serve as a foundation for quantum programming languages. This program is based on two main frameworks: the Lambda-S calculus, which is an extension of the lambda calculus to quantum computing, and the $\mathcal{L}^{\mathcal{S}}$ calculus, which is a proof language for intuitionistic linear logic whose proof terms can be used to construct quantum programs. The idea was to start from computing to logic (the Lambda-S calculus), and then from logic to computing (the $\mathcal{L}^{\mathcal{S}}$ calculus), in order to meet in the middle.

The Lambda-S Calculus. From Computing to Logic. Lambda-S [DCDR19] and Lambda-S_1 [DCGMV19] are quantum lambda calculi designed to handle quantum superpositions and control while preserving computational properties such as strong normalization and subject reduction. Both calculi extend the lambda calculus with algebraic linearity and type-based constraints to enforce quantum mechanics principles, particularly the no-cloning theorem.

Lambda-S is an extension of simply typed lambda calculus with linear combinations of terms, it incorporates a type constructor $S(A)$ where a simple type A denotes a set of basis vectors

*A longer version of this abstract will appear at [DC25].

(terms) and $S(A)$ its span. This system ensures that terms in A are duplicable, whereas terms in $S(A)$ are not, reflecting the impossibility of cloning unknown quantum states. Lambda-S has been given a categorical semantics through an adjunction between Cartesian and additive symmetric monoidal categories, where S is a functor transforming sets into vector spaces, and its adjoint forgets the vectorial structure [DCM23, DCM20].

Lambda-S_1 extends Lambda-S by enforcing norm-preserving constraints on superpositions, making it a stricter model suitable for representing unitary transformations explicitly. It has been obtained via a realizability model in [DCGMV19]. The categorical semantics of Lambda-S_1 [DCM22] is structured around adjunctions but diverges from Lambda-S by ensuring that all terms maintain a unitary norm, addressing the long-standing issue of preserving quantum unitarity in quantum control lambda calculi.

Both calculi serve as foundations for quantum programming languages and categorical models of quantum computation, bridging classical and quantum computational paradigms through rigorous mathematical structures.

One of the most notable results of this *side* of the research program, is the fact that we proved the model of Lambda-S to be a (partial) dual of known models for intuitionistic linear logic (ILL). Partial, because it favors a computational basis, but some preliminary results suggest that it can be extended to a proper dual [Mon25]. Usually, ILL is interpreted in a monoidal closed category, using an adjunction with a Cartesian closed category to interpret duplicable data. Lambda-S, on the other hand, is interpreted in a Cartesian closed category, using the same adjunction, but the other way around, to interpret non-duplicable data. This duality is a strong indication that the structure of quantum computation is the dual of linear logic, and that the Curry-Howard-Lambek correspondence can be extended to quantum computation, with this linear logic dual as the logical side and Lambda-S as the computational side.

The $\mathcal{L}^{\mathcal{S}}$ Calculus. From Logic to Computing. The $\mathcal{L}^{\mathcal{S}}$ calculus extends intuitionistic multiplicative additive linear logic (IMALL) with algebraic structures, incorporating sum and scalar multiplication within proof terms. It builds on the $\odot$-calculus [DCD23], which introduced an algebraic connective for quantum superpositions, and refines this idea within a fully linear framework [DCD24]. In a recent draft, we replace the $\odot$ connective with an alternative rule for disjunction introduction, enabling a structured representation of the quantum measurement [DCD25]. Its categorical semantics formalizes this approach, aligning it with monoidal structures that preserve linearity [DCM24]. Moreover, the calculus has been extended [DCDIM24] with !, that is, from IMALL to ILL, and polymorphism, ensuring expressive power suitable for quantum programming languages.

The $\mathcal{L}^{\mathcal{S}}$ calculus provides a logical foundation for quantum programming languages, enabling the construction of quantum circuits and algorithms through proof terms. Its algebraic structure aligns with quantum mechanics principles, allowing for a direct representation of quantum superpositions and measurements. The calculus serves as a bridge between linear logic and quantum computation, providing a formal framework for reasoning about quantum programs.

The fact that both languages, Lambda-S and $\mathcal{L}^{\mathcal{S}}$, use the same adjunction

$$\begin{tikzcd} \text{Lambda-S} \arrow[r, dashed] & (\mathcal{S},\times,I) \arrow[r, bend left, "{(F,m)}"] \arrow[r, phantom, "\perp"] & (\mathcal{V},\otimes,1) \arrow[l, bend left, "{(G,n)}"] & \mathcal{L}^{\mathcal{S}} \arrow[l, dashed] \end{tikzcd}$$

where $\mathcal{S}$ is a Cartesian closed category, $\mathcal{V}$ is a monoidal closed category, but with Lambda-S being interpreted in $\mathcal{S}$, using the monad GF to interpret the non-duplicable terms, and $\mathcal{L}^{\mathcal{S}}$

being interpreted in $\mathcal{V}$, using the comonad FG to interpret the duplicable terms, is a strong indication that the structure of quantum computation can be seen as the dual of linear logic, and that the Curry-Howard-Lambek correspondence can be extended to quantum computation following this path.

Several open challenges remain. On the one hand, while preliminary results suggest a duality between quantum computation and linear logic, a complete characterisation of this duality, both syntactically and categorically, is still an open question. On the other hand, extending the Lambda-S and $\mathcal{L}^{\mathcal{S}}$ calculi to fully capture quantum measurement processes, particularly in the presence of multiple bases and dependent types, demands further development. Additionally, establishing a formal correspondence between the algebraic structures of the calculi and graphical languages such as the ZX-calculus, and exploiting this link for program verification and optimisation, represents another promising but challenging direction.

This ongoing research program is advancing steadily, and the promising results so far suggest that computational quantum logic is a viable and interesting direction.

Acknowledgments This work is supported by the European Union through the MSCA SE project QCOMICAL (Grant Agreement ID: 101182520), by the Plan France 2030 through the PEPR integrated project EPiQ (ANR-22-PETQ-0007), and by the Uruguayan CSIC grant 22520220100073UD.

References

[BvN36] Garrett Birkhoff and John von Neumann. The logic of quantum mechanics. *Annals of Mathematics*, 37(4):823–843, 1936. doi:10.2307/1968621.

[Cro93] Roy Crole. *Categories for Types*. Cambridge University Press, 1993. doi:10.1017/CBO9781139172707.

[DC25] Alejandro Díaz-Caro. Towards a computational quantum logic: An overview of an ongoing research program. Invited talk at the Quantum Computing session at CiE 2025: Computability in Europe. To appear at LNCS. Preprint at arXiv:2504.07609, 2025.

[DCD23] Alejandro Díaz-Caro and Gilles Dowek. A new connective in natural deduction, and its application to quantum computing. *Theoretical Computer Science*, 957:113840, 2023. doi:10.1016/j.tcs.2023.113840.

[DCD24] Alejandro Díaz-Caro and Gilles Dowek. A linear linear lambda-calculus. *Mathematical Structures in Computer Science*, 34:1103–1137, 2024. doi:10.1017/S0960129524000197.

[DCD25] Alejandro Díaz-Caro and Gilles Dowek. A new introduction rule for disjunction. arXiv:2502.19172, 2025.

[DCDIM24] Alejandro Díaz-Caro, Gilles Dowek, Malena Ivnisky, and Octavio Malherbe. A linear proof language for second-order intuitionistic linear logic. In George Metcalfe, Thomas Studer, and Ruy de Queiroz, editors, *Logic, Language, Information and Computation (WoLLIC 2024)*, volume 14672 of *Lecture Notes in Computer Science*, pages 18–35. Springer, 2024. doi:10.1007/978-3-031-62687-6_2.

[DCDR19] Alejandro Díaz-Caro, Gilles Dowek, and Juan Pablo Rinaldi. Two linearities for quantum computing in the lambda calculus. *BioSystems*, 186:104012, 2019. Postproceedings of TPNC 2017. doi:10.1016/j.biosystems.2019.104012.

[DCGMV19] Alejandro Díaz-Caro, Mauricio Guillermo, Alexandre Miquel, and Benoît 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. doi:10.1109/LICS.2019.8785834.

[DCM20] Alejandro Díaz-Caro and Octavio Malherbe. A categorical construction for the computational definition of vector spaces. *Applied Categorical Structures*, 28(5):807–844, 2020. doi:10.1007/s10485-020-09598-7.

[DCM22] Alejandro Díaz-Caro and Octavio Malherbe. Quantum control in the unitary sphere: Lambda-S_1 and its categorical model. *Logical Methods in Computer Science*, 18(3:32), 2022. doi:10.46298/lmcs-18(3:32)2022.

[DCM23] Alejandro Díaz-Caro and Octavio Malherbe. A concrete model for a linear algebraic lambda calculus. *Mathematical Structures in Computer Science*, 34(1):1–44, 2023. doi:10.1017/s0960129523000361.

[DCM24] Alejandro Díaz-Caro and Octavio Malherbe. The sup connective in IMALL: A categorical semantics. arXiv:2405.02142, 2024.

[Mon25] Nicolás Monzón. Extension of Lambda-S for different measurement bases. Master's thesis, Universidad Argentina de la Empresa, March 2025.

[SU06] Morten Heine Sørensen and Paweł Urzyczyn. *Lectures on the Curry-Howard Isomorphism*. Elsevier, 2006. doi:10.1016/S0049-237X(06)80499-4.