A New Connective in Natural Deduction, and its Application to Quantum Computing*

Alejandro Díaz-Caro
ICC, CONICET–Universidad de Buenos Aires, Argentina
DCyT, Universidad Nacional de Quilmes, Argentina

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

We investigate an unsuspected connection between non harmonious logical connectives, such as Prior's *tonk*, and quantum computing. We defend the idea that non harmonious connectives model the information erasure, the non-reversibility, and the non-determinism that occur, among other places, in quantum measurement. More concretely, we introduce a propositional logic with a non harmonious connective $\odot$ (read: "sup", for "superposition"), prove cut elimination for this logic, and show that its proof language forms the core of a quantum programming language.

Insufficient, harmonious, and excessive connectives In natural deduction, to prove a proposition C, the elimination rule of a connective $\triangle$ requires a proof of $A \triangle B$ and a proof of C using, as extra hypotheses, exactly the premises needed to prove the proposition $A \triangle B$, with the introduction rules of the connective $\triangle$. This principle of inversion, or of harmony, has been introduced by Gentzen [5] and developed, among others, by Prawitz [9] and Dummett [4] in natural deduction, by Miller and Pimentel [6] in sequent calculus, and by Read [11–13] for the rules of equality.

For example, to prove the proposition $A \wedge B$, the introduction rule of the conjunction requires a proof of A and a proof of B, hence, to prove a proposition C, the generalized elimination rule of the conjunction [7, 8, 14] requires, a proof of $A \wedge B$ and a proof of C, using, as extra hypotheses, the propositions A and B

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

This principle of inversion permits to define a cut elimination process where the proof

$$\frac{\dfrac{\dfrac{\pi_1}{\Gamma \vdash A} \quad \dfrac{\pi_2}{\Gamma \vdash B}}{\Gamma \vdash A \wedge B}\ \wedge\text{-i} \quad \dfrac{\pi_3}{\Gamma, A, B \vdash C}}{\Gamma \vdash C}\ \wedge\text{-e}$$

reduces to $(\pi_1/A, \pi_2/B)\pi_3$.

In the same way, to prove the proposition $A \vee B$, the introduction rules of the disjunction require a proof of A or a proof of B, hence, to prove a proposition C, the elimination rule of the disjunction requires a proof of $A \vee B$ and two proofs of C, one using, as extra hypothesis, the proposition A and the other the proposition B

$$\frac{\Gamma \vdash A}{\Gamma \vdash A \vee B}\ \vee\text{-i1} \qquad \frac{\Gamma \vdash B}{\Gamma \vdash A \vee B}\ \vee\text{-i2} \qquad \frac{\Gamma \vdash A \vee B \quad \Gamma, A \vdash C \quad \Gamma, B \vdash C}{\Gamma \vdash C}\ \vee\text{-e}$$

and a cut elimination process can be defined in a similar way.

*The full paper can be found at arXiv:2012.08994.

Submitted to:
QPL 2021

We also can imagine connectives that do not verify this inversion principle, because the introduction rules require an insufficient amount of information with respect to what the elimination rule provides, as extra hypotheses, in the required proof of C. An example of such an *insufficient* connective is Prior's *tonk* [10], with the introduction and elimination rules as follows

$$\frac{\Gamma \vdash A}{\Gamma \vdash A \textit{ tonk } B}\ \textit{tonk}\text{-i} \qquad \frac{\Gamma \vdash A \textit{ tonk } B \quad \Gamma, B \vdash C}{\Gamma \vdash C}\ \textit{tonk}\text{-e}$$

where the elimination rule requires a proof of A *tonk* B and a proof of C, using the extra hypothesis B, that is not required in the proof of A *tonk* B, with the introduction rule. For such connectives, cuts *tonk*-i / *tonk*-e cannot be reduced.

But, it is also possible that a connective does not verify the inversion principle because the introduction rules require an excessive amount of information. An example of such an *excessive* connective is the connective $\odot$ that has the introduction rule of the conjunction and the elimination rule of the disjunction

$$\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \odot B}\ \odot\text{-i} \qquad \frac{\Gamma \vdash A \odot B \quad \Gamma, A \vdash C \quad \Gamma, B \vdash C}{\Gamma \vdash C}\ \odot\text{-e}$$

In this case,cuts can be eliminated. Moreover, several cut elimination processes can be defined, exploiting, in different ways, the excess of the connective. For example, the $\odot$-cut

$$\frac{\dfrac{\dfrac{\pi_1}{\Gamma \vdash A} \quad \dfrac{\pi_2}{\Gamma \vdash B}}{\Gamma \vdash A \odot B}\ \odot\text{-i} \quad \dfrac{\pi_3}{\Gamma, A \vdash C} \quad \dfrac{\pi_4}{\Gamma, B \vdash C}}{\Gamma \vdash C}\ \odot\text{-e}$$

can be reduced to $(\pi_1/A)\pi_3$, it can be reduced to $(\pi_2/A)\pi_4$, it also can be reduced, non deterministically, either to $(\pi_1/A)\pi_3$ or to $(\pi_2/A)\pi_4$. Finally, to keep both proofs, we can add a structural rule

$$\frac{\Gamma \vdash A \quad \Gamma \vdash A}{\Gamma \vdash A}\ \text{parallel} \qquad \text{and reduce it to} \qquad \frac{\dfrac{(\pi_1/A)\pi_3}{\Gamma \vdash C} \quad \dfrac{(\pi_2/B)\pi_4}{\Gamma \vdash C}}{\Gamma \vdash C}\ \text{parallel}$$

Information loss With harmonious connectives, when a proof is built with an introduction rule, the information contained in the proofs of the premises of this rule is preserved. For example, the information contained in the proof π_1 is *present* in the proof π

$$\frac{\dfrac{\pi_1}{\Gamma \vdash A} \quad \dfrac{\pi_2}{\Gamma \vdash B}}{\Gamma \vdash A \wedge B}\ \wedge\text{-i}$$

in the sense that π_1 is a subtree of π. But it is moreover *accessible*, in the sense that, for all π_1, putting the proof π in the right context yields a proof that reduces to π_1. And the same holds for the proof π_2.

The situation is different with an excessive connective: the excess of information, required by the introduction rule, and not returned by the elimination rule in the form of an extra hypothesis, in the required proof of C, is lost. For example, the information contained in the proofs π_1 and π_2 is present in the proof

$$\frac{\dfrac{\pi_1}{\Gamma \vdash A} \quad \dfrac{\pi_2}{\Gamma \vdash B}}{\Gamma \vdash A \odot B}\ \odot\text{-i}$$

but its accessibility depends on the way we decide to reduce the cut

$$\dfrac{\dfrac{\dfrac{\pi_1}{\Gamma \vdash A} \quad \dfrac{\pi_2}{\Gamma \vdash B}}{\Gamma \vdash A \odot B}\ \odot\text{-i} \quad \dfrac{\pi_3}{\Gamma, A \vdash C} \quad \dfrac{\pi_4}{\Gamma, B \vdash C}}{\Gamma \vdash C}\ \odot\text{-e}$$

If we reduce it systematically to $(\pi_1/A)\pi_3$, then the information contained in π_1 is accessible, but that contained in π_2 is not. If we reduce it systematically to $(\pi_2/A)\pi_4$, then the information contained in π_2 is accessible, but not that contained in π_1. If we reduce it not deterministically to $(\pi_1/A)\pi_3$ or to $(\pi_2/A)\pi_4$, then the information contained in both π_1 and π_2 is accessible but non deterministically. If we reduce it with parallel, then the information contained in both π_1 and π_2 is inaccessible.

So, while harmonious connectives, that verify the inversion principle, model information preservation, reversibility, and determinism, these excessive connectives, that do not verify the inversion principle, model information erasure, non-reversibility, and non-determinism. Such information erasure, non-reversibility, and non-determinism, occur, for example, in quantum physics, where the measurement of the superposition of two states does not yield both states back.

Quantum physics and quantum languages Several programming languages have been designed to express quantum algorithms. Among them, Lineal [1] is an untyped λ-calculus extended with linear combinations of terms, expressing superpositions, and Lambda-$\mathcal{S}$ [3] is a typed version of the first-order fragment of Lineal, extended with a measurement operator π and a rule reducing $\pi(t+u)$ non deterministically to t or to u.

The superposition $t+u$ can be considered as the pair (t,u). Hence, it should have the type $A \wedge A$. In other words, it is a proof-term of the proposition $A \wedge A$. In System I [2], various type-isomorphisms have been introduced, in particular the commutativity isomorphism $A \wedge B \equiv B \wedge A$, hence $t+u \equiv u+t$. In such a system, where $A \wedge B$ and $B \wedge A$ are identical, it is not possible to define the two elimination rules, as the two usual projections rules π_1 and π_2 of the λ-calculus. They were replaced with a single projection parametrized with a proposition A: π_A, such that if $t : A$ and $u : B$ then $\pi_A(t+u)$ reduces to t and $\pi_B(t+u)$ to u. When $A = B$, hence t and u both have type A, the proof-term $\pi_A(t+u)$ reduces, non deterministically, to t or to u. Thus, this modified elimination rule behaves like a measurement operator.

These works on Lambda-$\mathcal{S}$ and System I brought to light the fact that the pair superposition / measurement, in a quantum programming language, behaves like a pair introduction / elimination, for some connective, in a proof language, as the succession of a superposition and a measurement yields a term that can be reduced. In System I, the assumption was made that this connective was a commutative conjunction, with a modified elimination rule, leading to a non deterministic reduction.

But, as the measurement of the superposition of two states does not yield both states back, this connective should probably be excessive. Moreover, as, to build the superposition $a.|0\rangle + b.|1\rangle$, we need both $|0\rangle$ and $|1\rangle$ and the measurement, in the basis $|0\rangle, |1\rangle$, yields either $|0\rangle$ or $|1\rangle$, this connective should have the introduction rule of the conjunction, and the elimination rule of the disjunction, that is that it should be the connective $\odot$.

Outline of the paper In this paper, we present a propositional logic with the connective $\odot$, a language of proof-terms, the $\odot$-calculus (read: "the sup-calculus"), for this logic, and we prove a cut elimination theorem. We then extend this calculus, introducing scalars to quantify the propensity of a proof to reduce to another and show that its proof language forms the core of a quantum programming language.

References

[1] Pablo Arrighi & Gilles Dowek (2017): *Lineal: A linear-algebraic λ-calculus*. Logical Methods in Computer Science 13(1:8), pp. 1–33.

[2] Alejandro Díaz-Caro & Gilles Dowek (2019): *Proof Normalisation in a Logic Identifying Isomorphic Propositions*. In Herman Geuvers, editor: 4th International Conference on Formal Structures for Computation and Deduction (FSCD 2019), Leibniz International Proceedings in Informatics (LIPIcs) 131, pp. 14:1–14:23.

[3] Alejandro Díaz-Caro, Gilles Dowek & Juan Pablo Rinaldi (2019): *Two linearities for quantum computing in the lambda calculus*. BioSystems 186, p. 104012. Postproceedings of TPNC 2017.

[4] M. Dummett (1991): *The Logical basis of metaphysics*. Duckworth.

[5] G. Gentzen (1969): *Untersuchungen über das logische Schliessen*. In M. Szabo, editor: The Collected Papers of Gerhard Gentzen, North-Holland, pp. 68–131.

[6] D. Miller & E. Pimentel (2013): *A formal framework for specifying sequent calculus proof systems*. Theoretical Computer Science 474, pp. 98–116.

[7] S. Negri & J. von Plato (2008): *Structural Proof Theory*. Cambridge University Press.

[8] M. Parigot (1991): *Free Deduction: an Analysis of Computations in Classical Logic*. In: Proceedings of Russian Conference on Logic Programming, Lecture Notes in Computer Science, Springer, pp. 361–380.

[9] D. Prawitz (1965): *Natural deduction. A proof-theoretical study*. Almqvist & Wiksell.

[10] A.N. Prior (1960): *The Runabout Inference-Ticket*. Analysis 21(2), pp. 38–39.

[11] S. Read (2004): *Identity and harmony*. Analysis 64, pp. 113–119.

[12] S. Read (2010): *General-elimination harmony and the meaning of the logical constants*. Journal of Philosophical Logic 39, pp. 557–576.

[13] S. Read (2014): *Identity and Harmony revisited*. `https://www.st-andrews.ac.uk/~slr/identity_revisited.pdf`.

[14] P. Schroeder-Heister (1984): *A natural extension of Natural deduction*. The Journal of Symbolic Logic 49(4), pp. 1284–1300.