Towards Quantitative Logics in Rocq

Janis Bailitis[1], Reynald Affeldt[3], Alessandro Bruni[4], Alessio Coltellacci[4], Ekaterina Komendantskaya[1,2], and Kathrin Stark[1]

[1] Heriot-Watt University, Edinburgh, United Kingdom
[2] University of Southampton, Southampton, United Kingdom
[3] National Institute of Advanced Industrial Science and Technology (AIST), Tokyo, Japan
[4] IT University of Copenhagen, Copenhagen, Denmark

Quantitative Logics (QLs) are logics whose truth values go beyond 0 and 1. They were first studied by Łukasiewicz [33] and Gödel [16, 20, 26], and have recently gained importance due to their applicability to safer machine learning [14]. A well-known and long-established important instance of QLs are fuzzy logics [11, 32] dealing with inherently vague propositions like "x is much larger than 10", valued in the interval $[0, \infty]$. Much more recently, Bacci et al. [5–7] present propositional QLs, for example to reason about inequalities between rational functions. They are valued in structures based on the Lawvere quantale [23], with $\wedge$ and $\vee$ interpreted by max and min. Bacci and Møgelberg [8] introduce a higher-order QL interpreted in $[0, 1]$, newly allowing for unified reasoning on, but not limited to, program logics for probabilistic programming languages, bisimilarity on probabilistic processes, and properties of machine learning. Lastly, Capucci et al. [10] propose a QL parametric in $p \in (0, \infty]$ with quantitative proof calculus, called *p-hard quantitative linear logic* (pQLL).

Quantitative logics are becoming increasingly important, yet they lack a general formal theory. Therefore, we are interested in establishing a solid formal basis for QLs, envisaging a general and reusable library for these logics mechanised in Rocq [28]. As a key ingredient to many QLs are intervals of (extended) real numbers, our implementation should allow for extensive reasoning about these numbers at its core. We will be building on top of an exploration of selected traditional and recent propositional QLs in Rocq by Affeldt et al. [3, 4], also taking inspiration from a position paper by Marulanda-Giraldo et al. [25]. As a case study, we work towards mechanising pQLL.

In this abstract, we discuss

1. the ongoing effort to build a theory of nonnegative extended real numbers in Rocq, including connectives often used in QLs, relying on MathComp [24] and MathComp-analysis [1],
2. key features of pQLL and their mechanisation, alongside work towards a novel completeness result, and
3. improvements made and being made to MathComp(-analysis) as part of the process.

Our mechanisation can be accessed on `https://github.com/Janis-Bai/QLLib/releases/tag/Rocqshop2026`. The links in the document point to the corresponding Rocq code.

Extended real numbers in Rocq. Many QLs are based on intervals of real numbers such as $[0, \infty]$ used in pQLL. Besides standard operations such as multiplication $\otimes$, we require the notions of *comultiplication* $\otimes^*$ and *p-sums* $\oplus^p$. Comultiplication $a \otimes^* b := (a^{-1} \otimes b^{-1})^{-1}$ only differs from multiplication for $a = 0$ and $b = \infty$ (and vice versa), and p-sums are a soft variant of max and min, explained below.

For machine learning applications, it is desirable to have operations that are differentiable and componentwise strictly increasing (*smooth monotone*) [29, Property 4], also referred to as *soft operations*. The motivation behind this stems from learning algorithms such as gradient descent, which solve a minimisation problem to find optimal network parameters. For a soft operation, a small change in one of the parameters will result in a change in the overall value, improving the algorithm's effectiveness. This is contrary to more traditional non-soft operations such as max and min, which are referred to as *hard operations*.

Hence, due to being soft, p-sums $a \oplus^p b := (a^p + b^p)^{1/p}$ and harmonic p-sums $a \oplus^{-p} b := (a^{-1} \oplus^p b^{-1})^{-1}$, originally applied to QLs in Yager logic [31], have recently gained importance and are used, for example, in pQLL. A key relation we are currently mechanising is that $\oplus^p$ converges to the binary maximum function as $p \to \infty$.

As of now, we mechanised all the operations above, along with a few fundamental facts such as associativity of $\oplus^p$.

Quantitative linear logic. In pQLL, there are soft connectives $\vee$ and $\wedge$ interpreted by $\oplus^p$ and $\oplus^{-p}$. This approach sacrifices idempotence (since $1 \oplus^p 1 = 2^{1/p} \neq 1$), which is related to a general result about logical connectives: connectives cannot be associative, idempotent, and smooth monotone at the same time [29, Proposition 1]. Also, the algebraic and proof-theoretic properties of QLs with soft connectives are not well-studied, unlike those with hard connectives, which have an extensive theory since the 1990s, e.g. via residuated lattices [19, 30].

A key novelty of pQLL is its *quantitative proof calculus*, which is based on sequent calculus: deduction rules are annotated by operations on or constants in the extended reals. For example, the axiom rule and the left rule for additive soft disjunction are

$$\frac{1}{A \vdash A}\ \text{AX} \qquad\qquad \frac{\Gamma \vdash A, \Delta \quad \oplus^p \quad \Gamma \vdash B, \Delta}{\Gamma \vdash A \vee B, \Delta}\ \vee_R.$$

Given a proof, one can compute a unique extended real number by traversing the proof tree, called the *validity* of the proof. For instance, the proof of $A \vdash A \vee A$ using $\vee_R$ and then AX twice has validity $1 \oplus^p 1 = 2^{1/p}$. This notion can be extended to sequents $\Gamma \vdash \Delta$, called *provability* and denoted $|\Gamma \vdash \Delta|$, defined as $\sup\{\mathsf{validity}(P) \mid P \text{ is a proof of } \Gamma \vdash \Delta\}$, noting that $\sup \varnothing = 0$. The mechanisation of these definitions is standard and outlined below.

Mechanisation of pQLL's syntax and deduction system. Formulas in pQLL are given by the following grammar, where V is a type of propositional variables:

$$A, B : \mathcal{F}_V ::= a \mid a^* \mid \mathbf{1} \mid A \otimes B \mid A \otimes^* B \mid \bot \mid \top \mid A \vee B \mid A \wedge B \qquad\qquad a \in V$$

We mechanised sequents using an inductive type with type constructor $\vdash : \mathsf{List}(\mathcal{F}_V) \to \mathsf{List}(\mathcal{F}_V) \to \mathsf{Type}$ and one value constructor per deduction rule, ignoring the annotations referring to a proof's validity. To compute a proof's validity, we implemented a fixpoint of type $\forall \Gamma, \Delta. \Gamma \vdash \Delta \to [0, \infty]$ recursing on the derivation of $\Gamma \vdash \Delta$. Provability $|\Gamma \vdash \Delta|$ is mechanised by forming the set S of all proofs of $\Gamma \vdash \Delta$, and then taking the supremum of $\{\mathsf{validity}(p) \mid p \in S\}$.

Towards completeness. Semantics for pQLL is defined via *softales* [10, Definition 5.2], a preorder-like structure where the order relation is quantitative, i.e. $a \leq b$ is not a binary truth value, but an extended real number. Using this notion, Capucci et al. prove that pQLL is sound [10, Proposition 5.9].

We are currently working towards a novel completeness result for 1QLL without propositional variables, and currently conjecture the following lemma, which we believe follows by induction on A.

Lemma 1. *For $A \in \mathcal{F}_\varnothing$, we have $[\![A]\!] \leq |\vdash A|$, where $[\![\cdot]\!] : \mathcal{F}_\varnothing \to [0, \infty]$ is the interpretation of formulas.*

From this lemma, we believe that the completeness result follows readily once softales are mechanised. We expect the latter to be straightforward using the Hierarchy Builder [13].

Implementation details. In Rocq, operations such as $\otimes^*, \oplus^p$ operate on the type `{nonneg \bar R}`, morally a Σ-type of an extended real number x and a certificate that $0 \leq x$.

The key challenge in defining these operations is to verify that they preserve nonnegativity. While MathComp has automation to infer these and other preservation properties, known as interval inference [2], this automation was not fully implemented and tested yet for extended reals. For example, during our mechanisation we found that the + notation for addition on extended reals was not recognised by the inference automation. Further, we proved the required inference lemmas for inversion and powers.

We defined p-sums $a \oplus^p b$ as p-integral $||f||_p := (\int_\Omega |f|^p \, \mathrm{d}\mu)^{-p}$, where μ is the counting measure on $\Omega := \mathbb{N}$ and $f(0) = a$, $f(1) = b$, and $f(n + 2) = 0$. This approach may seem cumbersome at first, but it has the advantage of allowing results about p-integrals to be directly applied to p-sums. Additionally, some properties of p-sums can be mechanised in greater generality.

The original definition of pQLL uses finite multisets to define sequents, while in our mechanisation of the calculus, we employ the well-established type of lists because of its ease of automation and speed. As a consequence, exchange rules were added in Rocq which are absent in the original definition of the logic. In the future, we may facilitate an implementation of finite multisets based on MathComp [12], inspired by Affeldt et al. [4], who point out that multisets may allow for easier proof automation in the absence of exchange rules.

Future work. Cut elimination, which is admitted by pQLL [10, Theorem 4.1], enables proof search and facilitates the consistency proof for pQLL. Motivated by this, we are currently mechanising cut elimination, having already completed the core argument for a one-sided variant of pQLL subject to a purely algebraic lemma. We conjecture the single-sided calculus to be equivalent to pQLL. The cut elimination proof extends Laurent's mechanised proof [22] for linear logic, following their high-level structure as well as the use of a one-sided calculus, and adding reasoning about proof validities as required by pQLL.

Adding quantification to pQLL is an ongoing effort [9, 10] motivated by its applications to machine learning [17], where specifications are often first-order. We are interested in mechanising this, which would require dealing with object-level binders. The Rocq Libraries for First-Order Logic [21] and Undecidability [18] might provide a good framework as their type of formulas is quantified over arbitrary logical atoms, connectives, and quantifiers and as these libraries provide extensive support for binders using de Bruijn syntax [15], inspired by Autosubst 2 [27].

Acknowledgements. The authors would like to thank Matteo Capucci for insightful discussions as well as helpful suggestions and advice.

References

[1] Reynald Affeldt, Yves Bertot, Alessandro Bruni, Cyril Cohen, Marie Kerjean, Assia Mahboubi, Damien Rouhling, Pierre Roux, Kazuhiko Sakaguchi, Zachary Stone, Pierre-Yves Strub, and Laurent Théry. MathComp-Analysis: Mathematical Components compliant analysis library, 2017. Last stable version: 1.16.0 (2026). URL: https://github.com/math-comp/analysis.

[2] Reynald Affeldt, Alessandro Bruni, Cyril Cohen, Pierre Roux, and Takafumi Saikawa. Formalizing Concentration Inequalities in Rocq: Infrastructure and Automation. In Yannick Forster and Chantal Keller, editors, *16th International Conference on Interactive Theorem Proving, ITP 2025, Reykjavik, Iceland, September 28 - October 1, 2025*, LIPIcs, pages 21:1–21:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2025. doi:10.4230/LIPICS.ITP.2025.21.

[3] Reynald Affeldt, Alessandro Bruni, Ekaterina Komendantskaya, Natalia Slusarz, and Kathrin Stark. Taming Differentiable Logics with Coq Formalisation. In Yves Bertot, Temur Kutsia, and Michael Norrish, editors, *15th International Conference on Interactive Theorem Proving, ITP 2024, Tbilisi, Georgia, September 9-14, 2024*, LIPIcs, pages 4:1–4:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPICS.ITP.2024.4.

[4] Reynald Affeldt, Alessandro Bruni, Ekaterina Komendantskaya, Natalia Ślusarz, and Kathrin Stark. A Foundation for Differentiable Logics using Dependent Type Theory, 2026. arXiv:2602.23878, doi:10.48550/arXiv.2602.23878.

[5] Giorgio Bacci, Radu Mardare, Prakash Panangaden, and Gordon D. Plotkin. Propositional Logics for the Lawvere Quantale. In Marie Kerjean and Paul Blain Levy, editors, *Proceedings of the 39th Conference on the Mathematical Foundations of Programming Semantics, MFPS XXXIX, Indiana University, Bloomington, IN, USA, June 21-23, 2023*, Electronic Notes in Theoretical Informatics and Computer Science. EpiSciences, 2023. doi:10.46298/ENTICS.12292.

[6] Giorgio Bacci, Radu Mardare, Prakash Panangaden, and Gordon D. Plotkin. Polynomial Lawvere Logic. *CoRR*, abs/2402.03543, 2024. arXiv:2402.03543, doi:10.48550/ARXIV.2402.03543.

[7] Giorgio Bacci, Radu Mardare, Prakash Panangaden, and Gordon D. Plotkin. Rational Lawvere Logic. In Stefano Guerrini and Barbara König, editors, *34th EACSL Annual Conference on Computer Science Logic, CSL 2026, Paris, France, February 23-28, 2026*, LIPIcs, pages 3:1–3:21. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2026. Invited Paper. doi:10.4230/LIPICS.CSL.2026.3.

[8] Giorgio Bacci and Rasmus Ejlers Møgelberg. Induction and Recursion Principles in a Higher-Order Quantitative Logic for Probability. In *41st Annual Symposium on Logic in Computer Science, LICS 2026, Lisbon, July 20-23, 2026*, 2026. To appear. Preprint available on arXiv with doi 10.48550/arXiv.2501.18275.

[9] Matteo Capucci. On Quantifiers for Quantitative Reasoning. *CoRR*, abs/2406.04936, 2024. arXiv:2406.04936, doi:10.48550/ARXIV.2406.04936.

[10] Matteo Capucci, Robert Atkey, Charles Grellois, and Ekaterina Komendantskaya. Quantitative Linear Logic. *CoRR*, abs/2605.13348, 2026. arXiv:2605.13348, doi:10.48550/arXiv.2605.13348.

[11] Petr Cintula, Petr Hájek, and Carles Noguera. *Handbook of Mathematical Fuzzy Logic*, volume 1 of *Studies in Logic*. College Publications, London, UK, 2011.

[12] Cyril Cohen and Kazuhiko Sakaguchi. MathComp-finmap: Finite Maps. Last stable version: Finmap 2.2.3 (2026). URL: https://github.com/math-comp/finmap.

[13] Cyril Cohen, Kazuhiko Sakaguchi, and Enrico Tassi. Hierarchy Builder: Algebraic hierarchies Made Easy in Coq with Elpi (System Description). In Zena M. Ariola, editor, *5th International Conference on Formal Structures for Computation and Deduction, FSCD 2020, Paris, France (Virtual Conference), June 29 - July 6, 2020*, LIPIcs, pages 34:1–34:21. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPICS.FSCD.2020.34.

[14] David “davidad” Dalrymple. Safeguarded AI: constructing guaranteed safety, 2024. Programme thesis. Version v1.2. URL: https://aria.org.uk/media/3nhijno4/aria-safeguarded-ai-programme-thesis-v1.pdf.

[15] Nicolaas G. de Bruijn. Lambda calculus notation with nameless dummies, a tool for automatic formula manipulation, with application to the Church-Rosser theorem. In *Indagationes mathematicae (proceedings)*, volume 75, pages 381–392. Elsevier, 1972. doi:10.1016/1385-7258(72)90034-0.

[16] Michael Dummett. A Propositional Calculus with Denumerable Matrix. *The Journal of Symbolic Logic*, 24(2):97–106, 1959. doi:10.2307/2964753.

[17] Thomas Flinkow, Ekaterina Komendantskaya, Matteo Capucci, and Rosemary Monahan. Quantitative Linear Logic for Neuro-Symbolic Learning and Verification. *CoRR*, abs/2605.13845, 2026. arXiv:2605.13845, doi:10.48550/arXiv.2605.13845.

[18] Yannick Forster, Dominique Larchey-Wendling, Andrej Dudenhefner, Edith Heiter, Dominik Kirst, Fabian Kunze, Gert Smolka, Simon Spies, Dominik Wehr, and Maximilian Wuttke. A Coq library of undecidable problems. *CoqPL 2020: The Sixth International Workshop on Coq for Programming Languages*, 2020.

[19] Nikolaos Galatos, Peter Jipsen, Tomasz Kowalski, and Hiroakira Ono. *Residuated Lattices: An Algebraic Glimpse at Substructural Logics*, volume 151 of *Studies in Logic and the Foundations of Mathematics*. Elsevier Science, San Diego, CA, USA, 1st edition, 2007.

[20] Kurt Gödel. Zum intuitionistischen Aussagenkalkül (in German). *Anzeiger Akademie der Wissenschaften Wien, mathematisch-naturwiss. Klasse*, 32:65–66, 1932.

[21] Dominik Kirst, Johannes Hostert, Andrej Dudenhefner, Yannick Forster, Marc Hermes, Mark Koch, Dominique Larchey-Wendling, Niklas Mück, Benjamin Peters, Gert Smolka, and Dominik Wehr. A Coq Library for Mechanised First-Order Logic. *The Coq Workshop*, 2022.

[22] Olivier Laurent. YALLA: Yet Another Deep Embedding of Linear Logic in Rocq. *Journal of Automated Reasoning*, 70(1):9, 2026. doi:10.1007/S10817-026-09755-Y.

[23] F. William Lawvere. Metric spaces, generalized logic, and closed categories. *Rendiconti del Seminario Matematico e Fisico di Milano*, 43(1):135–166, 1973. doi:10.1007/BF02924844.

[24] Assia Mahboubi and Enrico Tassi. Mathematical Components, 2022. doi:10.5281/zenodo.3999478.

[25] Jairo Miguel Marulanda-Giraldo, Ekaterina Komendantskaya, Alessandro Bruni, Reynald Affeldt, Matteo Capucci, and Enrico Marchioni. Quantifiers for Differentiable Logics in Rocq (Extended Abstract). In Mirco Giacobbe and Anna Lukina, editors, *AI Verification - Second International Symposium, SAIV 2025, Zagreb, Croatia, July 21-22, 2025, Proceedings*, Lecture Notes in Computer Science, pages 227–237. Springer, 2025. doi:10.1007/978-3-031-99991-8_12.

[26] George Metcalfe, Nicola Olivetti, and Dov M. Gabbay. *Proof Theory for Fuzzy Logics*. Applied Logic Series. Springer Dordrecht, 1st edition, 2008. doi:10.1007/978-1-4020-9409-5.

[27] Kathrin Stark, Steven Schäfer, and Jonas Kaiser. Autosubst 2: reasoning with multi-sorted de Bruijn terms and vector substitutions. In Assia Mahboubi and Magnus O. Myreen, editors, *Proceedings of the 8th ACM SIGPLAN International Conference on Certified Programs and Proofs, CPP 2019, Cascais, Portugal, January 14-15, 2019*, pages 166–180. ACM, 2019. doi:10.1145/3293880.3294101.

[28] The Rocq Development Team. The Rocq Prover, September 2025. doi:10.5281/zenodo.17473943.

[29] Péter Várnai and Dimos V. Dimarogonas. On Robustness Metrics for Learning STL Tasks. In *2020 American Control Conference, ACC 2020, Denver, CO, USA, July 1-3, 2020*, pages 5394–5399. IEEE, 2020. doi:10.23919/ACC45564.2020.9147692.

[30] Morgan Ward and R. P. Dilworth. Residuated Lattices. *Proceedings of the National Academy of Sciences of the United States of America*, 24(3):162–164, 1938. doi:10.2307/1990008.

[31] Ronald R. Yager. On a general class of fuzzy connectives. *Fuzzy Sets and Systems*, 4(3):235–242, 1980. doi:10.1016/0165-0114(80)90013-5.

[32] Lotfi Asker Zadeh. Fuzzy Sets. *Information and Control*, 8(3):338–353, 1965. doi:10.1016/S0019-9958(65)90241-X.

[33] Jan Łukasiewicz. *O logice trójwartościowej (in Polish). English translation: On Three-Valued Logic, in Borkowski, L.(ed.) 1970. Jan Łukasiewicz: Selected Works, Amsterdam: North Holland*. Ruch Filozoficzny, 1920.