Internship proposal:
Grading LL-exponentials

Advisers. Flavien Breuvart and Damiano Mazza

Keywords. category theory, linear logic, denotational semantics, functional programming

Potential collaborations. Tarmo Uustalu

Prerequisite. A strong taste for category theory, linear logic and/or denotational semantics is required. Additional notion of advanced type systems and/or functional programming are also welcomed.

Localization. LIPN, University Paris 13

The dual notions of graded monads [3] and linear logic exponentials [4, 2] are emerging structures with both a potential for powerful static analysis, and an extremely clean semantics. The examples in the literature, however, are often quite simple and very similar. In order to analyze the expressive power of those gradation, a natural question is to ask what are the gradation refining a given monad/exponential.

This question was partially treated by Breuvart and Pagani [1] in order to create models of graded exponential in a semi-automatic way, but the approach somehow lacked universality a theorem. In a very recent (unpublished) result, Breuvart and Uustalu found a way to fully describe all possible gradations of a given monad under some specific assumptions.

This internship aims at performing the same study for the case of exponentials of linear logic, which is similar but with several additional subtleties coming from the monoidality and the contraction.

References

[1] Flavien Breuvart and Michele Pagani. Modelling Coeffects in the Relational Semantics of Linear Logic. In *CSL*, 2015.

[2] Aloïs Brunel, Marco Gaboardi, Damiano Mazza, and Steve Zdancewic. A core quantitative coeffect calculus. In Zhong Shao, editor, *23rd European Symposium on Programming, ESOP, Held as Part of ETAPS*, volume 8410 of *Lecture Notes in Computer Science*, pages 351–370. Springer, 2014.

[3] Soichiro Fujii, Shin-ya Katsumata, and Paul-André Melliès. Towards a formal theory of graded monads. In *International Conference on Foundations of Software Science and Computation Structures*, pages 513–530. Springer, 2016.

[4] Dan R. Ghica and Alex I. Smith. Geometry of synthesis III: resource management through type inference. In Thomas Ball and Mooly Sagiv, editors, *Proceedings of the 38th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL*, pages 345–356. ACM, 2011.