Internship proposal:
Toward graded arrows and graded profunctors

Advisors. Flavien Breuvart and Thomas Seiller

Keywords. category theory, denotational semantics, functional programming, (co)monads, quantitative, static analysis

Potential collaborations. Exequiel Rivas

Prerequisite. A strong taste for denotational semantics is required. Additional notion of Linear Logic, advanced type systems, functional programming, category and/or static analysis are also welcomed.

Localisation. LIPN, University Paris 13

Principle

The dual notions of parameterised monads [3] and comonads [5, 2] are emerging structures with both a potential for poweful static annalysis, and an extremly clean semantics. However, this apparent "duality" is less obvious when inspected in details [4]. In this intership, we aim at clarifying a bit this duality.

For this, we are interested in arrows [6], and their categorical interpretations as profunctors [1]. Arrows (and profunctors) can be seen as generalisation of both (non-graded) monads and comonads, giving one of the numerous way to expose their duality. In this internship, we are looking for a graded version which could subsums both graded monads and graded comonads.

References

[1] Kazuyuki Asada and Ichiro Hasuo. Categorifying computations into components via arrows as profunctors. *Electr. Notes Theor. Comput. Sci.*, 264(2):25–45, 2010.

[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] Marco Gaboardi, Shin-ya Katsumata, Dominic A. Orchard, Flavien Breuvart, and Tarmo Uustalu. Combining effects and coeffects via grading. In *ICFP*, 2016.

[5] 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.

[6] John Hughes. Generalising monads to arrows. *Sci. Comput. Program.*, 37(1-3):67–111, 2000.

CoGITARe Project and fully funded PhD

This internship can be followed by a PhD fully funded by the ANR CoGITARe on the thematic.

Type systems are used to automatically check security properties of large programs. CoGITARe's goal is to extend typing methodology to a large panel of properties currently unreachable by state-of-the-art techniques, enabling in particular the analysis of quantitative properties of programs.

We will develop a way to keep track of the extensional information inside types in order to perform the whole static analysis at the level of types. For this purpose, we will combine two (re)emerging type systems, namely graded types and intersection type systems, with the well established techniques from the field of abstract interpretation such as widening.

Graded type systems formally embed a first order structure within types, while intersection types will help to circumvent the unconditional non-compositionality of fine grained resource analyses. This is how we plan to tackle the long running problem of applying abstract interpretation result in functional programming.