PhD proposal:
Type systems for resource analysis

Advisors. Flavien Breuvart and Damiano Mazza

Localisation. LIPN, University Paris 13,

Keywords. Type systems, abstract interpretation, functional programming, algebraic types, subtyping, resource analysis, denotational semantics, category theory, domain theory, type inference, graded types, intersection types.

Laboratory (LIPN)

LIPN (http://www-lipn.univ-paris13.fr) plays a major role in research in computer science within the northern Paris area. LIPN is situated in Villetaneuse, in the northern suburbs of Paris, within the campus of the University Paris 13 (about 30 minutes from the city center by public transportation).

It is affiliated both with the university and with CNRS. Its members conduct research in several areas: combinatorics, combinatorial optimization, algorithmic, logics, software engineering, natural languages, and machine learning. The department is structured into the following five teams, each on one or several of these research areas :
Team A3: Machine learning,
Team AOC: Combinatorial Optimization,
Team CALIN: Combinatorics,
Team LoVe: Logics and Verification,
Team RCLN: Natural languages.

An activity report for 2012-2017 is available here.

Team (LoVe)

The LoVe stands for Logic or Verification, with the or being a logical OR : some of us are working on logic, some on verification, some on both. This is a big team in the lab with 21 permanent member among with 4 CNRS researchers. We are structured in two axes, one rather focused on logic, the other on verification. This thesis will be held in the "logical" axis called "Types, models and theory of programming", nonetheless, as we will see later on, verification is still of major importance for the project.

The common ground for this axis research activities is the use of tools and techniques of logical nature for the analysis, the development and the formalization of programs. Of particular importance is the foundational study of quantitative aspects of programs, such as computational complexity (time, space), probabilistic behaviours and the formalization of real analysis, as well as modelling non-determinism and concurrency. Our research also covers natural language modelling and temporal reasoning. Our research can be grouped into two principal themes: "Proof theory and lambda-calculus" and "logic and models of computational resources".

CoGITARe project

The PhD is founded by CoGITARe, an ANR starting grant (JCJC) that have started in March.

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.

More details available in the proposal and on the dedicated website.

PhD subject

The project cover a large variety of thematic from very theoretic ones to practical ones; as such we have the flexibility to adapt the PhD subject to different kinds of candidate profiles. For the sake of clarity, we are giving one possible subject, but do not hesitate to apply or to contact Flavien Breuvart (breuvart@lipn.univ-paris13.fr) if you think you may be more interested by a subject following an other orientation of the project.

The PhD student may/will work on combining types systems with graded monads[1] with polymorphic subtyping *a la* MLSub.[2] The first systems allow the analysis of precise properties such that exception tracking, but is not expressive enough, the main reason being that it is too rigid and do not provide polymorphism, lacking any kind of principal type property. The second type system is an elegant solution to combine polymorphic and subtyping, with surprisingly rich results.

In addition to the simple join of those two theory, the successful candidate is expected to study the interaction between those systems. In particular, a substitution operator is expected in the resource algebra of the graded monad in order to deal with polymorphism. Under certain conditions, this operator allows to recover unexpected properties. Indeed, the parametric study[3] of those polymorphic resources gives access to new kind of information such as the sequenciality of the execution. For example, we can write, and infer, a type like $(\alpha|\texttt{err}).\tau \rightarrow \beta\tau \rightarrow$

Personal requirements

- Master degree in computer science or applied mathematics, preferably a research oriented one.
- Knowledge of, and taste for at least one of the major theories involved (type theory, abstract interpretation and denotational semantics).
- Skills in functional programming are appreciated.

Application

We are happy to provide more details to potential applicants. Requests for additional information regarding the projects content/motivation, other informal inquiries, and formal applications can be sent to Flavien Breuvart (breuvart@lipn.univ-paris13.fr). Formal applications must include:

- A motivation letter
- A cv, including a list of courses with grades
- A copy of the masters thesis (if already available)
- The name of at least one scientist able and willing to provide a reference

Deadline for applications: August 15th 2019

[1] Soichiro Fujii, Shin-ya Katsumata, Paul-Andr Melliès: “Towards a Formal Theory of Graded Monads” FoSSaCS 2016: 513-530

[2] Stephen Dolan, Alan Mycroft: “Polymorphism, subtyping, and type inference in MLsub”. POPL 2017: 60-72

[3] Philip Wadler: “Theorems for Free!” FPCA 1989: 347-359