INRIA — Département d'Informatique de l'École Normale Supérieure

Abstract Interpretation-based Static Analysis to Infer Smoothness Properties of Imperative Programs or (optionally) Probabilistic Programs

Internship location: École Normale Supérieure ; 45, rue d'Ulm ; 75 230, PARIS.

Team: Équipe Sémantique et Interprétation Abstraite / Équipe-Projet "ANTIQUE".

Advisor & Contact : Xavier RIVAL (*e-mail* : `rival@di.ens.fr`, tél : 01 44 32 21 50, fax : 01 44 32 21 51)

Internship topic:

Common smoothness properties include, for instance, differentiability, continuity, or lipschitz continuity. Their study is very useful in various areas of computer science. For instance, they play a great role in the study of the behavior of hybrid systems. Similarly, machine learning applications make heavy use of optimisation algorithms such as gradient descent, which typically require some specific smoothness properties to hold (otherwise no guarantee over the optimisation results may be ensured).

A first approach to establish smoothness relies on a generalisation of dependence analysis [2]. However, it is known to fall short whenever programs have multiple execution paths (due to conditions or loops) and when smoothness still holds. Part of the difficulty comes from the need to simulatenously reason over multiple program executions.

In this internship, we propose to develop abstract interpretation techniques [1] to infer smoothness properties and that can deal with non trivial program branching. In particular, we will consider relational abstractions to establish similarities across them.

The work plan involves the following steps:

1. **Design and formalization of relational abstractions for smoothness properties**. This task consists in identifying adapted forms of logical predicates that would be used in paper proof and constructing abstract domains that are able to make such reasoning systematic.
2. **Static analysis and proof of soundness**. The design of the analysis will follow by adapting standard abstract interpretation static analysis derivation and proof techniques to a more challenging setup. At this stage, a very simple imperative language manipulating arithmetic expressions over real numbers will be considered (although the adaptation of the smoothness analysis to assess whether optimisation algorithms behave correctly in the presence of floating point arithmetic is a very interesting problem it is also much harder and would only be considered as a possible continuation of this internship).
3. **[Optional] Adaptation to probabilistic programs**. This step aims at generalising the analysis designed previously to a more complex programming paradigm. Probabilistic programming language are one of the emerging and quickly evolving topics and have applications in many fields like the modelling of physical systems or machine learning. However, adapting a novel smoothness analysis to this set up should be rather systematic, it is expected to yield interesting applications. This task is optional and may be considered as a continuation of the rest of the work.
4. **Implementation and evaluation**. The final step consists in the implementation and evaluation of the analysis. A static analysis tool will be developed (we suggest to use a functional language such as OCaml –for which static analysis infrastructure would be provided). Another subtask is to come up with interesting case studies to evaluate the analysis.

Pre-requisite:

For this internship, it would be preferable that the student is familiar enough with semantics and abstract interpretation, but a candidate interested in the foundations of programming languages would also be able to acquire the required knowledge. Prior knowledge in probabilistic programming is useful but not required (the topic may be considered from a non probabilistic point of view as well).

References

[1] Patrick Cousot, Radhia Cousot. Systematic Design of Program Analysis Frameworks. In POPL'79, pages 269-282, 1979.

[2] Wonyeol Lee, Hongseok Yang, and Xavier Rival. Smoothness Analysis for Probabilistic Programs with Application to Optimised Variational Inference In POPL'23 (To appear).