Categorical operational semantics

Research area: programming language semantics

Supervisor: Tom Hirschowitz
tom.hirschowitz@univ-smb.fr
https://www.lama.univ-savoie.fr/pagesmembres/hirschowitz

Université Savoie Mont Blanc
Laboratoire de mathématiques (LAMA)
Team: Logique, Informatique et Mathématiques discrètes

CONTEXT

In most research on programming languages, ideas and results are presented on one example language, even though they are often rather widely applicable and the experts roughly agree on their scopes. This is problematic, because it means that even in closely related situations, ideas always need to be adapted, and results reproved.

In the past decades, several lines of research have tried to remedy this situation, with different application areas. But any general theory of programming languages should account for many different aspects:

• First of all, it should account for the interaction of syntax with dynamics. General theories of syntax and dynamics (e.g., as labelled transition systems) exist, but are not easily reconciled.
• Furthermore, a general theory of programming languages should be flexible enough to enable the study of subjects as different as behavioural equivalences (e.g., congruence of bisimilarity) and type soundness (e.g., absence of certain errors during execution).
• Finally, it should include some theory of denotational semantics, i.e., it should say what the models of a given language are.

Most importantly, a successful theory of programming languages will probably provide high-level tools for designing and reasoning about programming languages. E.g., working with format theory [5] feels a bit like programming in some assembly language. One particular approach seemed to meet all these criteria for some time, namely the bialgebraic semantics of Turi and Plotkin [3, 6]. It is a very abstract approach based on category theory [4], which models

• syntax by algebras for a monad T,
• dynamics by coalgebras for a functor B,
• the interaction of syntax and dynamics by a distributive law between T and B.

However, it now appears that certain types of languages like, e.g., functional languages, are very difficult, if not impossible to model in this setting.

The internship is about an alternative approach that has emerged in recent years, and demonstrated its ability to cover an important result about functional languages [2]. Very roughly, this approach avoids coalgebras completely, and views the dynamics as a monad ‘on top of’ the monad for syntax, in a suitable sense. The result in question shows that if the considered language satisfies a certain well-formedness hypothesis, then Abramsky’s applicative bisimilarity [1] is a congruence, i.e., is closed under context.

Work in progress attempts

• to generalise this congruence result, notably by targeting higher-order, concurrent languages,
• to prove congruence results for other behavioural equivalences, notably Lassen’s normal-form bisimilarity,
• to investigate simple type soundness results.

GOAL
The goal of the internship is to contribute to the development of the new framework. E.g., this could take one of the following forms:

1. contribute to one of the above directions,
2. investigate congruence for other behavioural equivalences, e.g., weak or environmental bisimilarity,
3. generalise the framework to cover more languages, e.g., ones with dependent types.

EXPECTED ABILITIES
The student is expected to have some minimal background in theoretical computer science, and some appetite for abstract mathematics.

REFERENCES