Project proposal
Master Informatique

Modular reasoning about graph transformations

Keywords: Logic; Semantics of programming languages; Program verification; Interactive proof assistants

Scientific Context

Within the Climt project[1], we are working on graph transformations, and in particular the verification of graph transformations.

Graph transformations have wide-spread applications, such as pointer-manipulating programs and model transformations in languages such as UML. They are also interesting for graph databases, such as those represented with RDF schemas, and the link structure of the World-Wide Web.

We are currently developing an imperative language for graph transformations, and a family of Hoare-style logics for reasoning about programs of this language. The logics are variants of Description Logics. A distinctive feature of our work is that both the programming language and the logic have been formalized in the proof assistant Isabelle[2].

Planned work

The aim of the proposed project is to make our approach more modular, as follows:

1. *by introducing a notion of procedures:* our transformation language is so far only composed of instructions (loops, conditionals) without the notion of procedure. The project will introduce a notion of procedure and define a corresponding assertional language.

2. *by allowing to reason about regions of a graph,* in order to localize the impact of a graph transformation, for example for the purpose of optimizing or parallelizing transformations.

These apparently different topics have a common core, because they require the definition of a language of contracts. We propose to start with (1) and then to extend the investigations to (2).

The work will consist of a *conceptual phase*:

- Study of the existing graph transformation language and its semantics.

- Study of contracts for traditional procedural languages.

[1] www.irit.fr/~Martin.Strecker/CLIMT/
[2] http://isabelle.informatik.tu-muenchen.de/

- Study of specialized logics for reasoning about locality, for example separation logics and "shapes" in abstract interpretation.
- Proposal of a language of contracts.
- Derivation of proof rules for this language.

This phase will be followed by an *implementation phase* (in a large sense) which can consist of the following, depending on previous knowledge and interests:

- either a formal development in a proof assistant:
  - coding of extensions of the programming language's syntax and semantics and of proof rules in Isabelle;
  - formally verifying the adequacy of the proof rules *wrt.* the programming language semantics;
- or an implementation in a programming language (Ocaml or Scala):
  - extensions of the programming language
  - proof obligation generator
  - communication with specialized Description Logic provers

Prerequisites and interests

From the participating students, we expect

- interest in formal semantics of programming languages, logics, methods of correctness proofs for imperative programs (*required*)
- experience with functional (Ocaml, Haskell) or OO-functional (Scala) languages (*required*)
- experience with interactive proof assistants (such as Coq, Isabelle, PVS) (*desirable*)

Administrative Context

Interested? Then please contact us for further inquiries and for applying for the project. The internship can be carried out at either of the sites of the Climt project:

- LIG (Grenoble), contact Rachid Echahed[3]
- IRIT (Toulouse), contact Martin Strecker[4]

Funding will be provided for the duration of the project. The project is an ideal starting point for a PhD thesis.

[3] http://membres-lig.imag.fr/echahed/
[4] http://www.irit.fr/~Martin.Strecker/