Typestates specification and verification in Frama-C

Keywords: Frama-C, Program analysis, Verification

Institution

The French Alternative Energies and Atomic Energy Commission (CEA) is a key player in research, development, and innovation. Drawing on the widely acknowledged expertise gained by its 16,000 staff spanned over 9 research centers with a budget of 4.1 billion Euros, CEA actively participates in more than 400 European collaborative projects with a large number of academic (notably as a member of Paris-Saclay University) and industrial partners. Within the CEA Technological Research Division, the CEA List institute addresses the challenges coming from smart digital systems.

Among other activities, CEA List's Software Safety and Security Laboratory (LSL) research teams design and implement automated analysis in order to make software systems more trustworthy, to exhaustively detect their vulnerabilities, to guarantee conformity to their specifications, and to accelerate their certification. In particular, the Frama-C platform is dedicated to perform a wide range of analyses over C programs (with an experimental C++ front-end).

Objectives

Within Frama-C, the ACSL language is the primary mean to specify which properties are supposed to hold at a given point of the program under analysis. In particular, ACSL annotations are handled by the major analysis plug-ins (Eva, WP and E-ACSL). However, ACSL is heavily biased towards the verification of functional properties expressible as function contracts or assertions evaluated at a specific program points. Not every property of interest is easily amenable to such setting. To allow users to express such different kinds of properties more conveniently, some plugins introduce a Domain-Specific Language (DSL), in order to generate the corresponding set of ACSL annotations needed for verifying the original property. For instance, MetAcsl is used to specify a property that must be checked at many program points (for instance at every write access) scattered throughout the code. Similarly, Aoraï is dedicated to specify constraints on the sequences of calls that may occur during the execution.

The goal of this internship is to design and implement a new Frama-C plug-in for specifying and verifying *typestates* (Strom and Yemini 1986; Aldrich et al. 2009). Briefly speaking, typestates are used to constrain the set of operations that can be applied to a given datastructure depending on its current state. A typical exemple is the representation of a `FILE`, which, after a call to `fopen` can be in typestate `r`, `w` or `rw`. Then, the file can be an argument of `read` only if it is either in `r` or `rw`, and conversely an argument of `write` only if it is either in `w` or `rw`. In any case, the typestate of the `FILE` stays the same. Finally, at some point, the function `fclose` gets called on the file, which enters the typestate `closed` (if it wasn't already in that state, since we can't close the file twice). More generally, the lifecycle of an object `o` of a given type `t` is described as a finite automaton, and each time a function gets (a pointer to) `o` as argument, we have to verify whether the current state of the automaton allows the call, and potentially update the state.

The intern will first propose a DSL for describing such automata, based on simple examples in C. They will then develop a Frama-C plug-in for parsing this DSL and instrument the code under analysis in order to let other plug-ins (primarily the Eva and/or E-ACSL plug-ins, but, depending on the interests of the intern, WP might also be considered).

Qualifications

- **Minimal**
- master 2 student in Computer Science
- knowledge of program analysis
- knowledge of OCaml
- ability to work in a team
- **Preferred**

- familiarity with the Frama-C platform
- some knowledge in C

Characteristics

- **Duration:** 6 months from early 2022
- **Location:** CEA Nano-INNOV, Paris-Saclay Campus, France
- **Compensation:**
  - €700 to €1300 monthly stipend (determined by CEA compensation grids)
  - maximum €229 housing and travel expense monthly allowance (in case a relocation is needed)
  - CEA buses in Paris region and 75% refund of transit pass
  - subsidized lunches

Application

If you are interested in this internship, please send to the contact persons an application containing:

- your resume;
- a cover letter indicating how your curriculum and experience match the qualifications expected and how you would plan to contribute to the project;
- your bachelor and master 1 transcripts;
- the contact details of two persons (at least one academic) who can be contacted to provide references.

Applications are welcomed until the position is filled. Please note that the administrative processing may take up to 3 months.

Contact persons

For further information or details about the internship before applying, please contact:

- Virgile Prevosto (virgile.prevosto@cea.fr)

Bibliography

Aldrich, Jonathan, Joshua Sunshine, Darpan Saini, and Zachary Sparks. 2009. "Typestate-Oriented Programming." In *Proceedings of the 24th ACM SIGPLAN Conference Companion on Object Oriented Programming Systems Languages and Applications*, 1015–22. OOPSLA '09. New York, NY, USA: Association for Computing Machinery. doi:10.1145/1639950.1640073.

Strom, Robert E., and Shaula Yemini. 1986. "Typestate: A Programming Language Concept for Enhancing Software Relability." *IEEE Transactions on Software Engineering* 12 (1).