Internship

# Formalization of the executable semantics of the ANSI/ISO C Specification Language

## Location

CEA LIST, Lab. de Sûreté du Logiciel
Nano-Innov
Centre de Saclay
91191 Gif-sur-Yvette cedex

## Supervisors of the Internship

Julien Signoles (CEA LIST, Lab. de Sûreté du Logiciel)
Tél: 01 69 08 00 18
Email : Julien.Signoles@cea.fr

Claude Marché (LRI & Inria-Saclay, équipe Toccata)
Tél: 01 72 92 59 69
Email : Claude.Marche@inria.fr

## Context of the Internship

Frama-C is a suite of tools dedicated to the analysis of the source code of software written in C (http://frama-c.com [3]). It is developed jointly by CEA LIST and Inria-Saclay. It has an extensible architecture. A kernel provides common features: parsing of C files, typing, etc. Various analyzers can be implemented as plug-ins on top of this kernel. The kernel also implements a specification language called ACSL [1]. Specifications are given as annotations in source files.

Executable ACSL (E-ACSL [4]) is a subset of ACSL in which annotations can be executed. A plug-in translates annotated programs into an equivalent C code, where annotations are checked at run-time. Any violated annotations is reported to the user.

Operational semantics of C programs is complicated. Developing a C compiler without any bug is even more complicated. Nevertheless, as C programs are used in many safety-critical contexts, it is desirable to guarantee the soundness of the compilation. This is the goal of the CompCert project: developing a C compiler whose soundness is formally proved. This was done using the Coq proof assistant [2].

## Work Description

The goal of this internship is to provide some guarantees that the translation of E-ACSL annotations is sound. More precisely, it should focus on the translation of integer arithmetic expressions. In ACSL,

integer arithmetic is the standard mathematical one, on unbounded integers. To be sound, the translation does not use machine integers (which could overflow) but the GMP library (http://gmplib.org/).

A first step of the internship should be to survey some previous work: E-ACSL, CompCert, but also a formalization of ACSL into Coq [6].

A second step is the formalization of the translation process. An issue that must be taken into account is that translating an arithmetic expression into C code generates calls to GMP library and temporary memory allocations and de-allocations. This formalization should be done on paper as a first step, and then could be implemented into some computer tool: in Coq, or in an another environment for program verification called Why3 [5].

## Prerequisites

Basic knowledge in logic and in semantics of programming languages. Additional knowledge in compilation techniques or in proof environments would be appreciated.

## References


[1] Patrick Baudin, Jean-Christophe Filliâtre, Claude Marché, Benjamin Monate, Yannick Moy, and Virgile Prevosto. *ACSL: ANSI/ISO C Specification Language, version 1.7*, 2013. http://frama-c.com/acsl.html.

[2] Sandrine Blazy and Xavier Leroy. Mechanized semantics for the Clight subset of the C language. *Journal of Automated Reasoning*, 43(3):263–288, 2009.

[3] Pascal Cuoq, Florent Kirchner, Nikolaï Kosmatov, Virgile Prevosto, Julien Signoles, and Boris Yakobowski. Frama-c: a software analysis perspective. In *International Conference on Software Engineering and Formal Methods (SEFM'12)*, pages 233–247. Springer, October 2012.

[4] Mickaël Delahaye, Nikolaï Kosmatov, and Julien Signoles. Common specification language for static and dynamic analysis of C programs. In *Symposium on Applied Computing (SAC'13)*, pages 1230–1235. ACM, March 2013.

[5] Jean-Christophe Filliâtre and Andrei Paskevich. Why3 — where programs meet provers. In Matthias Felleisen and Philippa Gardner, editors, *Proceedings of the 22nd European Symposium on Programming*, volume 7792 of *Lecture Notes in Computer Science*, pages 125–128. Springer, March 2013.

[6] Paolo Herms. *Certification of a Tool Chain for Deductive Program Verification*. Thèse de doctorat, Université Paris-Sud, January 2013.