Formally Verifying a Compiler: What Does It Mean, Exactly?

Xavier Leroy

INRIA, Paris, France
`xavier.leroy@inria.fr`

Abstract

Compilers, and especially optimizing compilers, are complicated programs. Bugs in compilers happen, and can lead to miscompilation: the production of wrong executable code from a correct source program. Miscompilation is documented in the literature and a concern for high-assurance software, as it endangers the guarantees obtained by source-level formal verification of programs.

Compiler verification is a radical solution to the miscompilation problem: by applying program proof to the compiler itself, we can obtain mathematically strong guarantees that the generated executable code is faithful to the semantics of the source program. The state of the art in this line of research is arguably the CompCert verified compiler. This talk will give an overview of this optimizing C compiler and of its formal verification, conducted with the Coq proof assistant.

A formal verification is as good as the specifications it uses. In other words, verification reduces the problem of trusting a large implementation to that of ensuring that its formal specification enforce the intended correctness properties. In the case of CompCert, the correctness statement that is proved is rather complex, as it involves large operational semantics (for the C language and for the assembly languages of the target architectures) and simulations between these semantics that support both choice refinement and behavior refinement. The talk will review and discuss these elements of the specification, along with some of the accompanying proof principles.

1998 ACM Subject Classification D.2.4 Software/Program Verification

Keywords and phrases Compilers, Compiler Optimization, Compiler Verification

Digital Object Identifier 10.4230/LIPIcs.ICALP.2016.2

Category Invited Talk

43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016).
Editors: Ioannis Chatzigiannakis, Michael Mitzenmacher, Yuval Rabani, and Davide Sangiorgi;
Article No. 2; pp. 2:1–2:1
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany