Verified Compilation and Optimization of Floating-Point Programs in CakeML (Artifact)

Heiko Becker
MPI-SWS, Saarland Informatics Campus (SIC), Saarbrücken, Germany

Robert Rabe
TU München, Germany

Eva Darulova
Uppsala University, Sweden

Magnus O. Myreen
Chalmers University of Technology, Gothenburg, Sweden

Zachary Tatlock
University of Washington, Seattle, WA, USA

Ramana Kumar
DeepMind, London, UK

Yong Kiam Tan
Carnegie Mellon University, Pittsburgh, PA, USA

Anthony Fox
ARM Limited, Cambridge, UK

Abstract

Verified compilers such as CompCert and CakeML have become increasingly realistic over the last few years, but their support for floating-point arithmetic has thus far been limited. In particular, they lack the "fast-math-style" optimizations that unverified mainstream compilers perform. Supporting such optimizations in the setting of verified compilers is challenging because these optimizations, for the most part, *do not* preserve the IEEE-754 floating-point semantics. However, IEEE-754 floating-point numbers are finite approximations of the real numbers, and we argue that any compiler correctness result for fast-math optimizations should appeal to a real-valued semantics rather than the rigid IEEE-754 floating-point numbers.

This document describes the artifact for RealCake, an extension of CakeML that achieves end-to-end correctness results for fast-math-style optimized compilation of floating-point arithmetic. This result is achieved by giving CakeML a flexible floating-point semantics and integrating an external proof-producing accuracy analysis. RealCake's end-to-end theorems relate the I/O behavior of the original source program under real-number semantics to the observable I/O behavior of the compiler generated and fast-math-optimized machine code.

2012 ACM Subject Classification Software and its engineering → Formal software verification; Software and its engineering → Compilers; Software and its engineering → Software performance

Keywords and phrases compiler verification, compiler optimization, floating-point arithmetic

Digital Object Identifier 10.4230/DARTS.8.2.10

Funding *Eva Darulova*: Part of this work was done while the author was at MPI-SWS, Germany.
Magnus O. Myreen: Supported by the Swedish Foundation for Strategic Research.
Zachary Tatlock: Supported in part by the U.S. Department of Energy under Award Number DE-SC0022081 (ComPort), and by the National Science Foundation under Grant No. 1749570. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the DOE or NSF.

Related Article Heiko Becker, Robert Rabe, Eva Darulova, Magnus O. Myreen, Zachary Tatlock, Ramana Kumar, Yong Kiam Tan, and Anthony Fox, "Verified Compilation and Optimization of Floating-Point Programs in CakeML", in 36th European Conference on Object-Oriented Programming (ECOOP 2022), LIPIcs, Vol. 222, pp. 1:1–1:28, 2022.
`https://doi.org/10.4230/LIPIcs.ECOOP.2022.1`

Related Conference 36th European Conference on Object-Oriented Programming (ECOOP 2022), June 6–10, 2022, Berlin, Germany

Evaluation Policy The artifact has been evaluated as described in the ECOOP 2022 Call for Artifacts and the ACM Artifact Review and Badging Policy.

Dagstuhl Artifacts Series
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Scope

The artifact contains the CakeML extension, RealCake, described in the paper, the FloVer development with our extensions, as well as an installation of the HOL4 theorem prover. Our artifact also includes the scripts to reproduce the results from our evaluation. The formal developmentis is written as HOL4 input files and the evaluation scripts are provided as bash scripts.

2 Content

The artifact is provided as `zip` archive which includes:

- `ECOOP2022_AE_Submission_Document.md`, a README file explaining the details of the artifact and how to use it
- `ECOOP22 Artifact - RealCake.ova`, a VirtualBox image with the artifact installation

3 Getting the artifact

The artifact endorsed by the Artifact Evaluation Committee is available free of charge on the Dagstuhl Research Online Publication Server (DROPS).

4 Tested platforms

We provide the artifact as a VirtualBox OVA file to be imported as a virtual machine. The virtual machine comes installed with Xubuntu 20.04, and is configured to use 10GB of RAM. We recommend not to use less RAM as this may hinder compilation of HOL4 proofs. Providing less RAM will make some proofs fail checking with HOL4. We have obtained our results on a Raspberry Pi v3, running Raspbian via SSH access. The experiments can be run on an x86 processor too, or within the VM, but the timing measurements will differ from what we report in the paper. Other then that there are no specific hardware or software requirements for our artifact.

The username for the VM is ‘artifact‘ and the password is ‘artifact‘. The user also has the ability to run commands as root using ‘sudo‘.

5 License

6 MD5 sum of the artifact

fa4b2ffb32fb16c31dcf1f91a23e7567

7 Size of the artifact

4.3 GiB