Strong Logic for Weak Memory: Reasoning About Release-Acquire Consistency in Iris (Artifact)*

Jan-Oliver Kaiser[1], Hoang-Hai Dang[2], Derek Dreyer[3], Ori Lahav[4], and Viktor Vafeiadis[5]

1 **MPI-SWS, Saarland Informatics Campus, Saarbrücken, Germany**
janno@mpi-sws.org
2 **MPI-SWS, Saarland Informatics Campus, Saarbrücken, Germany**
haidang@mpi-sws.org
3 **MPI-SWS, Saarland Informatics Campus, Saarbrücken, Germany**
dreyer@mpi-sws.org
4 **MPI-SWS, Saarland Informatics Campus, Saarbrücken, Germany**
orilahav@mpi-sws.org
5 **MPI-SWS, Saarland Informatics Campus, Saarbrücken, Germany**
viktor@mpi-sws.org

Abstract

This artifact provides the soundness proofs for the encodings in Iris the RSL and GPS logics, as well as the verification for all standard examples known to be verifiable in those logics. All of these proofs are formalized in Coq, which is the main content of this artifact. The formalization is provided in a virtual machine for the convenience of testing, but can also be built from source.

1998 ACM Subject Classification F.3.1 Specifying and Verifying and Reasoning about Programs; F.3.2 Semantics of Programming Languages

Keywords and phrases weak memory models, release-acquire, concurrency, separation logic

Digital Object Identifier 10.4230/DARTS.3.2.15

Related Article Jan-Oliver Kaiser, Hoang-Hai Dang, Derek Dreyer, Ori Lahav, and Viktor Vafeiadis, "Strong Logic for Weak Memory: Reasoning About Release-Acquire Consistency in Iris", in Proceedings of the 31st European Conference on Object-Oriented Programming (ECOOP 2017), LIPIcs, Vol. 74, pp. 17:1–17:29, 2017.
http://dx.doi.org/10.4230/LIPIcs.ECOOP.2017.17

Related Conference European Conference on Object-Oriented Programming (ECOOP 2017), June 18-23, 2017, Barcelona, Spain

1 Scope

The artifact is designed to support repeatability of all the proofs of the companion paper, allowing users to step through the proofs in their favorite Coq editor. The mapping between the paper and the Coq development is provided in the `README.md` file.

2 Content

The artifact package includes a VirtualBox-based Debian virtual machine, which contains a copy of the Coq development (revision d7f3799d9750df2754e9b181209cc1a092028724). The `README.md`

* This work was supported in part by an ERC Consolidator Grant for the project "RustBelt", funded under the European Union's Horizon 2020 Framework Programme (grant agreement no. 683289).

file contains a list of lemmas and theorems presented in the paper and where to find them in the Coq development. It also contains instructions to build the development from source, which requires a system with opam installed.

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). The latest version of the artifact is available at our project page: `http://plv.mpi-sws.org/igps/`.

4 Tested platforms

The artifact is known to work on any platform running Oracle VirtualBox version 5.1 with at least 4 GB of free space on disk.

Instructions:

1. (Optional) Increase the number of processors and the amount of memory assigned to the VM. This helps with the last step of the build process. (4 processors and 4GB RAM work reasonably well.)
2. The machine will boot into a minimal Debian installation. The user and password are "artefact", as is the password to execute su.
3. After logging in, open a terminal and navigate to `~/ra-gps` – this is a copy of the commit mentioned above.
4. Proceed to `~/ra-gps/coq/ra` and execute make build-dep. This will install all dependencies of our development. **Note:** this command can take between 30 and 90 minutes. (For reasons unknown to us, this step does not use all available processors.)
5. Finally, execute make, or make -jN where N is the number of processors assigned to the VM. This step will terminate in ca. 20 minutes with make -j4 or around 50 minutes with just make. It should finish without errors, indicating that all Coq files were compiled successfully.

5 License

BSD

6 MD5 sum of the artifact

dbdf8c863bb606643d8ec203012362e3

7 Size of the artifact

1.7 GB