A Complete Program Logic for Compositional Linearizability (Artifact)

Eashan Hatti
Yale University, New Haven, CT, USA

Arthur Oliveira Vale
Yale University, New Haven, CT, USA

Zhongye Wang
Yale University, New Haven, CT, USA

Yueyang Feng
Yale University, New Haven, CT, USA

Zhong Shao
Yale University, New Haven, CT, USA

Abstract

We present Linearizability Hoare Logic (LHL), the first *mechanized*, *sound*, and *complete* program logic for atomic, set, and interval linearizability. We achieve this by showing soundness and completeness of LHL w.r.t. a more general criterion, *compositional linearizability*, which subsumes all three criteria. We showcase the expressivity of LHL by verifying an exchanger with a set linearizable specification, the elimination-backoff stack built above the exchanger, a lock with an atomic linearized specification, and a write-snapshot object with an interval linearizable specification.

Together with LHL we formalize a modular verification framework for concurrent components based on the theory of compositional linearizability. This allows us to specify components at a high level of abstraction and granularity, and then assemble them into large systems that are correct by construction. As a showcase, we verify the elimination-backoff stack modularly by verifying each of its sub-components against their linearized specifications and then linking them together.

2012 ACM Subject Classification Theory of computation → Program verification; Computing methodologies → Concurrent computing methodologies; Theory of computation → Programming logic

Keywords and phrases Program Logic, Rely-Guarantee, Linearizability, Compositional Verification, Concurrency

Digital Object Identifier 10.4230/DARTS.12.1.5

Funding This work is supported in part by NSF grants 2019285, 2313433, 2442888, and by the Defense Advanced Research Projects Agency (DARPA) under Agreement No. HR00112590130. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies.

Related Article Eashan Hatti, Arthur Oliveira Vale, Zhongye Wang, Yueyang Feng, and Zhong Shao, "A Complete Program Logic for Compositional Linearizability", in 40th European Conference on Object-Oriented Programming (ECOOP 2026), LIPIcs, Vol. 372, pp. 11:1–11:28, 2026.
`https://doi.org/10.4230/LIPIcs.ECOOP.2026.11`

Related Conference 40th European Conference on Object-Oriented Programming (ECOOP 2026), June 29–July 3, 2026, Brussels, Belgium

Evaluation Policy The artifact has been evaluated as described in the ECOOP 2026 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

All theorems and examples in the paper are mechanized in the artifact. Key elements are the compositionality of our definition of linearizability (see `Core/LinFacts.v`), the soundness and completeness of our logic (see `Core/LogicFacts.v`), and all examples (see the `Examples` folder).

2 Content

The `Examples` folder contains mechanizations of all examples in the paper. They depend on the `Core` folder, which contains the mechanization of the program logic and linearizability. The following describes its most important files.

- `RefinesFacts.v`: Contains refinement theorems about vertical composition, necessary to prove observational refinement.
- `TensorFacts.v`: Contains refinement theorems about horizontal composition along with the interaction of horizontal composition with vertical composition, necessary to prove locality.
- `LinFacts.v`: Contains the proofs of observational refinement and locality, showing our definition of linearizability is indeed compositional.
- `Logic.v`: Contains the definition of the program logic.
- `LogicFacts.v`: Contains the proofs of soundness and completeness of our program logic.
- `ProgramRules.v`: Contains essential derived rules of the program logic, necessary for practical proofs using it.

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). In addition, the artifact is also available at: `https://github.com/ehatti/LHL`.

4 Tested platforms

Building the artifact has been tested on Windows, Mac, and Linux, and requires minimal computational resources (building takes approximately 2 minutes on a low-end Macbook Pro). A docker image and build script are provided to automate the build process, instructions may be found in the artifact's `README.md`.

5 License

The artifact is available under the BSD 3-Clause license.

6 MD5 sum of the artifact

d24ff88eb94f21812201c3acaa166d0a

7 Size of the artifact

3.3 MB