




Higher-Order Specifications for Deductive Synthesis of Programs with Pointers (Artifact)

David Young*   

University of Kansas, Lawrence, KS, USA

Ziyi Yang*   

National University of Singapore, Singapore

Ilya Sergey   

National University of Singapore, Singapore

Alex Potanin   

Australian National University, Canberra, Australia

Abstract

This artifact provides a translator from Pika code to SuSLik specifications. Additionally, it contains a test suite and benchmark suite. These suites are

fully automated using provided scripts. All source code is included.

2012 ACM Subject Classification Software and its engineering → General programming languages; Software and its engineering → Automatic programming; Software and its engineering → Compilers

Keywords and phrases Program Synthesis, Separation Logic, Functional Programming

Digital Object Identifier 10.4230/DARTS.10.2.25

Related Article David Young*, Ziyi Yang*, Ilya Sergey, and Alex Potanin, “Higher-Order Specifications for Deductive Synthesis of Programs with Pointers”, in 38th European Conference on Object-Oriented Programming (ECOOP 2024), LIPIcs, Vol. 313, pp. 45:1–45:26, 2024.

<https://doi.org/10.4230/LIPIcs.ECOOP.2024.45>

Related Conference 38th European Conference on Object-Oriented Programming (ECOOP 2024), September 16–20, 2024, Vienna, Austria

Evaluation Policy The artifact has been evaluated as described in the ECOOP 2024 Call for Artifacts and the ACM Artifact Review and Badging Policy.

1 Scope

We have developed a high-level functional language which translates into separation logic. This is used to improve the usability of a program synthesis tool, SuSLik, which is based on separation logic. [1] This artifact demonstrates the translation described in the paper. The benchmark and evaluation data is taken from running the included benchmark suite. The automation for this benchmark suite also automatically outputs these results in a easy-to-use form.

2 Content

The artifact package is provided in the form of a Docker image `Pika.tar.gz`. To use, first unzip. Do not extract the `.tar` file. Instead, run the command `docker load -i Pika.tar`. Then use the following command to use the container: `docker run -it [image ID from output of last command]`.

* Contributed equally to this work.



© David Young, Ziyi Yang, Ilya Sergey, and Alex Potanin;
licensed under Creative Commons License CC-BY 4.0
Dagstuhl Artifacts Series, Vol. 10, Issue 2, Artifact No. 25, pp. 25:1–25:2



DAGSTUHL
ARTIFACTS SERIES

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



25:2 Higher-Order Specifications for Deductive Synthesis of Programs with Pointers (Artifact)

Inside the image, there are the directories among others:

- `/Pika`: This is the main directory.
- `/Pika/run-tests.sh`: This runs the test suite.
- `/Pika/run-benchmarks.sh`: This runs the benchmark suite.
- `/Pika/src`: This is the location of the source code for the translator.
- `/Pika/tests`: This is the location of the tests and golden files.
- `/Pika/ECOOP_AE_Submission_Document.md`: There is further information here on which scripts generated the tables from the paper.

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://zenodo.org/doi/10.5281/zenodo.10547529>. The source code is available at <https://github.com/roboguy13/PikaC/>.

4 Tested platforms

This artifact is known to work on a 2021 MacBook Pro with 32 GB of RAM. However, it is very likely to work on computers with much fewer resources and much less compute power.

5 License

The artifact is available under the Creative Commons Attribution 4.0 International license.

6 MD5 sum of the artifact

e16fe2b6955f52307bbdadcaea9dfd2f

7 Size of the artifact

1.87 GB

References

- 1 Nadia Polikarpova and Ilya Sergey. Structuring the synthesis of heap-manipulating programs. *Proc. ACM Program. Lang.*, 3(POPL), January 2019. doi:10.1145/3290385.