

Mailbox Abstractions for Static Analysis of Actor Programs (Artifact)*

Quentin Stiévenart¹, Jens Nicolay², Wolfgang De Meuter³, and Coen De Roover⁴

1 Software Languages Lab, Vrije Universiteit Brussel, Belgium
qstieven@vub.ac.be

2 Software Languages Lab, Vrije Universiteit Brussel, Belgium
jnicolay@vub.ac.be

3 Software Languages Lab, Vrije Universiteit Brussel, Belgium
wdmeuter@vub.ac.be

4 Software Languages Lab, Vrije Universiteit Brussel, Belgium
cderoove@vub.ac.be

Abstract

This artifact is based on SCALA-AM, a static analysis framework relying on the Abstracting Abstract Machines approach. This version of the framework is extended to support actor-based programs, written in a variant of Scheme. The sound static analysis is performed in order to verify the absence of errors in actor-based program, and to compute

upper bounds on actor's mailboxes. We developed several mailbox abstractions with which the static analysis can be run, and evaluate the precision of the technique with these mailbox abstractions. This artifact contains documentation on how to use analysis and on how to reproduce the results presented in the companion paper.

1998 ACM Subject Classification F.3.2 Semantics of Programming Languages – Program Analysis

Keywords and phrases static analysis, abstraction, abstract interpretation, actors, mailbox

Digital Object Identifier 10.4230/DARTS.3.2.11

Related Article Quentin Stiévenart, Jens Nicolay, Coen De Roover, and Wolfgang De Meuter, “Mailbox Abstractions for Static Analysis of Actor Programs”, in Proceedings of the 31st European Conference on Object-Oriented Programming (ECOOP 2017), LIPIcs, Vol. 74, pp. 25:1–25:30, 2017.

<http://dx.doi.org/10.4230/LIPIcs.ECOOP.2017.25>

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

1 Scope

This artifact aims to provide the necessary material to reproduce the experiments in the companion paper. It is an extended version of SCALA-AM, a static analysis framework, and perform static analysis on actor-based programs written in a variant of Scheme. The artifact implements the analysis described in the companion paper, as well as the different mailbox abstractions presented.

* Quentin Stiévenart is funded by the GRAVE project of the “Fonds voor Wetenschappelijk Onderzoek” (FWO Flanders), and Jens Nicolay is funded by the SeCloud project sponsored by Innoviris, the Brussels Institute for Research and Innovation.



© Quentin Stiévenart, Jens Nicolay, Wolfgang De Meuter, Coen De Roover;
licensed under Creative Commons Attribution 3.0 Germany (CC BY 3.0 DE)

Dagstuhl Artifacts Series, Vol. 3, Issue 2, Artifact No. 11, pp. 11:1–11:2



DAGSTUHL
ARTIFACTS SERIES

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

11:2 Mailbox Abstractions for Static Analysis of Actor Programs (Artifact)

2 Content

This artifact package includes:

- a VirtualBox image containing:
 - the modified version of SCALA-AM with all the dependencies needed,
 - the mechanized proofs of soundness with all the dependencies needed,
- detailed instruction for reproducing the experiments conducted in the companion paper ([artifact.pdf](#)).

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 artifact endorsed by the Artifact Evaluation Committee is available free of charge on the Dagstuhl Research Online Publication Server (DROPS). The source code artifact is also accessible at the following addresses:

- <https://github.com/acieroid/scala-am/tree/ecoop2017actors> for the extended version of SCALA-AM,
- <https://github.com/acieroid/mailbox-abstraction-proofs> for the Coq scripts of the proofs of soundness.

Moreover, the detailed instructions for reproducing the experiments conducted in the companion paper are accessible at <https://soft.vub.ac.be/~qstieven/ecoop2017/artifact.pdf>.

4 Tested platforms

The artifact can be installed on any platform running the Java Virtual Machine, version 6 or more recent. The provided VirtualBox virtual machine image (.vdi) requires around 7 GB of free space on disk and 2 GB of free RAM.

5 License

MIT License (<https://opensource.org/licenses/MIT>).

6 MD5 sum of the artifact

cccc1c14ec19366c8f99f0a341fbc139

7 Size of the artifact

1.8 GB