

Definite Reference Mutability (Artifact)*

Ana Milanova

Department of Computer Science, Rensselaer Polytechnic Institute
110 8th Street, Troy NY, USA
milanova@cs.rpi.edu

Wei Huang¹

Google
huangwe@google.com

Abstract

Related paper “Definite Reference Mutability” presents ReM (Re[ference] M[utability]), a type system that separates mutable references into (1) definitely mutable, and (2) maybe mutable, i.e., references whose mutability is due to inherent approximation. We have implemented ReM and applied it

on a large benchmark suite. Results show that $\approx 86\%$ of mutable references are definitely mutable.

This article describes the tool artifact from the related paper. The purpose of the article and artifact is to allow researchers to reproduce our results, as well as build new type systems upon our code.

2012 ACM Subject Classification Theory of computation - Abstraction
Software and its engineering - Object oriented languages

Keywords and phrases reference immutability, type inference, CFL-reachability

Digital Object Identifier 10.4230/DARTS.4.3.1

Related Article Ana Milanova, “Definite Reference Mutability”, in Proceedings of the 32nd European Conference on Object-Oriented Programming (ECOOP 2018), LIPIcs, Vol. 108, pp. 0:1–0:2, 2018.

<http://dx.doi.org/10.4230/LIPIcs.xxx.xxx.xxx>

Related Conference 32nd European Conference on Object-Oriented Programming (ECOOP 2018), July 19–21, 2018, Amsterdam, Netherlands

1 Scope

In previous work we developed a framework for inference and checking of pluggable types [?, ?]. Users instantiate the framework with certain parameters to define a type system. The framework takes as input a program (typically only partially annotated or not annotated at all), infers types for all variables and type checks the inferred types. We have instantiated the framework with known type systems and new ones. These include classical Ownership types [?, ?], Universe types [?, ?], ReIm reference immutability types [?], Information flow types for the detection of privacy leaks in Android apps [?], and AJ types for data centric synchronization [?, ?, ?].

The artifact builds upon this framework. Package `edu.rpi` is the heart of the framework: it includes type annotation utilities, visitors and a generic constraint solver. It is built on top of Soot [?]. Package `edu.rpi.reim` contains instantiations of ReIm and ReM. An instantiation introduces type-system-specific type qualifiers, initialization rules and typing rules, possibly overriding default rules defined in generic `InferenceTransformer` in package `edu.rpi`. For the majority of cases, ReIm and ReM reuse rules from the generic transformer.

The key purpose of this artifact is to reproduce and validate the claims of the related ECOOP paper. In addition, we invite researchers to build new type systems upon our framework.

* This work was partially supported by NSF grant 1319384.

¹ Work done while author was a PhD student at Rensselaer Polytechnic Institute.



2 Content

The artifact package includes:

- **bin** - directory contains compiled code
- **src** - directory contains all source code
- **lib** - directory contains all libraries: `soot-develop.jar` and `rt.jar` necessary to compile and run the code. We include the `rt.jar` from `jdk1.7.0_75` for MacOS. (It can be downloaded from the Oracle website: <http://www.oracle.com/technetwork/java/javase/downloads/java-archive-downloads-javase7-521261.html>.) The artifact requires a Java 7 `rt.jar`.
- **bench** - directory contains all benchmarks from the related paper
- **run-tests** - a script that automatically runs tool with benchmarks
- **README** - a description of artifact

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 available at: <http://www.cs.rpi.edu/~milanova/soot-reim-definite.zip>.

Source code for the framework, including all type systems, is available on GitHub: <https://github.com/proganalysis/type-inference>.

4 Tested platforms

1. Mac OS X El Capitan, 2.8 GHz Intel Core i7, 16 GB RAM. Java version 1.8.0_71.
2. Ubuntu 16.04.4 LTS, Intel(R) Xeon(R) CPU E5-2660 v3 @ 2.60GHz, 32 GB RAM. Java version 1.8.0_171.

The tool runs as is on these platforms using default maximal heap size.

5 License

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

Copyright 2018, Ana Milanova and Wei Huang.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE

56 COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
57 INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING,
58 BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS
59 OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
60 ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR
61 TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF
62 THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH
63 DAMAGE.

64 **6 MD5 sum of the artifact**

65 590b9f9b3160a342dcf2d71abb6c7585

66 **7 Size of the artifact**

67 120397581 B

68 **Acknowledgements.** We thank the ECOOP 2018 Artifact Evaluation committee and the ECOOP
69 2018 Program committee for valuable suggestions, and the National Science Foundation for
70 supporting our work under NSF grant 1319384.