Boomerang: Demand-Driven Flow- and Context-Sensitive Pointer Analysis for Java (Artifact)

Johannes Späth[1], Lisa Nguyen Quang Do[*2], Karim Ali[3], and Eric Bodden[†4]

1 **Fraunhofer SIT**
johannes.spaeth@sit.fraunhofer.de
2 **Universität Paderborn and Fraunhofer IEM**
lisa.nguyen@iem.fraunhofer.de
3 **Technische Universität Darmstadt**
karim.ali@cased.de
4 **Heinz Nixdorf Institut, Universität Paderborn and Fraunhofer IEM**
eric.bodden@uni-padeborn.de

Abstract

Evaluating pointer analyses with respect to soundness and precision has been a tedious task. Within this artifact we present PointerBench, the benchmark suite used in the paper to compare the pointer analysis Boomerang with two other demand-driven pointer analyses, SB [2] and DA [3]. We show PointerBench can be used to test different pointer analyses.

In addition to that, the artifact contains usage examples for Boomerang on simple test programs. The test programs and the input on these programs to Boomerang can be changed to experiment with the algorithm and its features.

Additionally, the artifact contains the integration of Boomerang, SB, and DA into FlowDroid, which can then be executed on arbitrary Android applications.

1998 ACM Subject Classification F.3.2 – Logics and Meanings of Programs – Semantics of Programming Languages—Program Analysis

Keywords and phrases Demand-Driven; Static Analysis; IFDS; Aliasing; Points-to Analysis

Digital Object Identifier 10.4230/DARTS.2.1.12

Related Article Johannes Späth, Lisa Nguyen Quang Do, Karim Ali, and Eric Bodden, "Boomerang: Demand-Driven Flow- and Context-Sensitive Pointer Analysis for Java", in Proceedings of the 30th European Conference on Object-Oriented Programming (ECOOP 2016), LIPIcs, Vol. 56, pp. 22:1–22:26, 2016.
http://dx.doi.org/10.4230/LIPIcs.ECOOP.2016.22

Related Conference 30th European Conference on Object-Oriented Programming (ECOOP 2016), July 18–22, 2016, Rome, Italy

1 Scope

The artifact is designed to support repeatability of the experiments performed in the companion paper. In particular, it can be used to evaluate points-to analyses on PointerBench. Additionally, the artifact enables running experiments in combination with the FlowDroid tool on arbitrary Android apps.

[*] Research was conducted while being at Fraunhofer SIT
[†] Research was conducted while being at Technische Universität Darmstadt and Fraunhofer SIT

2 Content

The artifact package includes six Eclipse projects:

1. PointerBench - a benchmark suite that can be used to evaluate precision and soundness of pointer analyses. It consists of micro benchmark programs that cover various test cases.
2. PointerBenchEvaluator - Parser and driver to execute and evaluate all queries of PointerBench with Boomerang, SB [2] and DA [3], as used in RQ1 of the paper.
3. SBandDA - Contains the implementation of the points-to analyses as referred to by SB [2] in the paper, and the alias analysis, noted by DA [3] in the paper.
4. soot-infoflow-alias - Adopted FlowDroid [1] integration for Boomerang, SB and DA.
5. soot-infoflow-android - FlowDroid's implementation to be able to run on Android.

The first three projects can be used to perform the experiments conducted in RQ1. That is, running all three pointer analyses, Boomerang, SB, and DA on PointerBench.

The last two projects include our adopted versions of FlowDroid that have been used to perform RQ2 and RQ3. Hence, by supplying Android applications, FlowDroid can be run with any of the three integrations.

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). PointerBench is also available under `https://github.com/secure-software-engineering/PointerBench`.

4 Tested platforms

The artifact was tested on linux platforms. RQ1 can be performed on a regular desktop computer using 4GB of RAM. The experiments with FlowDroid are tested on a 64 core machine with 32GB heap space. The bash scripts for this experiment can only be executed on linux.

5 License

GNU General Public License (GPL), Version 2.0 (`http://www.gnu.de/documents/gpl-2.0.de.html`)

6 MD5 sum of the artifact

17b19151cdb85b9493f5718976fbdffa

7 Size of the artifact

49.4mb

References

1 Steven Arzt, Siegfried Rasthofer, Christian Fritz, Eric Bodden, Alexandre Bartel, Jacques Klein, Yves Le Traon, Damien Octeau, and Patrick McDaniel. FlowDroid: precise context, flow, field, object-sensitive and lifecycle-aware taint analysis for Android apps. In *PLDI*, page 29, 2014.

2 Manu Sridharan and Rastislav Bodík. Refinement-based context-sensitive points-to analysis for Java. In *PLDI*, pages 387–400, 2006.

3 Dacong Yan, Guoqing (Harry) Xu, and Atanas Rountev. Demand-driven context-sensitive alias analysis for Java. In *ISSTA*, pages 155–165, 2011.