QuickChecking static analysis properties

QuickChecking static analysis properties

A static analysis can check programs for potential errors. A natural question that arises is therefore: who checks the checker? Researchers have given this question varying attention, ranging from basic testing techniques, informal monotonicity arguments, thorough pen-and-paper soundness proofs, to verified fixed point checking. In this paper, we demonstrate how quickchecking can be useful to test a range of static analysis properties with limited effort. We show how to check a range of algebraic lattice properties, to help ensure that an implementation follows the formal specification of a lattice. Moreover, we offer a number of generic, type-safe combinators to check transfer functions and operators on lattices, to help ensure that these are, eg, monotone, strict, or invariant. We substantiate our claims by quickchecking a type analysis for the Lua programming language.

General information
Publication status: Published
Organisations: Department of Applied Mathematics and Computer Science, Formal Methods, Aarhus University
Contributors: Midtgaard, J., Møller, A.
Number of pages: 23
Publication date: 2017
Peer-reviewed: Yes

Publication information
Journal: Software Testing, Verification and Reliability
Volume: 27
Issue number: 6
Article number: e1640
ISSN (Print): 0960-0833
Ratings:
BFI (2017): BFI-level 2
Scopus rating (2017): CiteScore 1.85 SJR 0.335 SNIP 1.593
Web of Science (2017): Impact factor 1.556
Web of Science (2017): Indexed yes
Original language: English
Keywords: Domain-specific languages, Monotonicity, Quickchecking, Static program analysis
Electronic versions:
Midtgaard_Moeller_STVR17.pdf
DOIs:
10.1002/stvr.1640
Research output: Contribution to journal › Journal article – Annual report year: 2017 › Research › peer-review