Functional pearl: zero-knowledge testing for module interfaces

Thomas Braibant   Jacques-Henri Jourdan   Jonathan Protzenko   Gabriel Scherer
INRIA
http://gallium.inria.fr/blog/

Abstract

In spite of recent advances in full program certification, testing remains a widely-used component of the software development cycle. Various flavors of testing exist: popular ones include *unit testing*, which consists in manually crafting test cases for specific parts of the code base, as well as quickcheck-style testing, where instances of a type are automatically generated to serve as test inputs.

These classical methods of testing can be thought of as *internal* testing: the test routines access the internal representation of whatever module should be checked. We propose a new method of *external* testing where test code checks an *abstract* data structure. Our new testing method takes a description of a *module signature*, then builds sequences of function calls that generate elements of the abstract type just like any other client code. Our testing framework thus builds elements that respect the invariants hidden by the module signature.

Categories and Subject Descriptors CR-number [*subcategory*]: third-level

Keywords functional programming, testing, quickcheck

ICFP '14, September 1–3, 2014, Copenhagen, Denmark.
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-nnnn-nnnn-n/yy/mm. . . $15.00.
http://dx.doi.org/10.1145/nnnnnnn.nnnnnnn