TestR: generating unit tests for R internals

Roman Tsegelskyi[⋆]**, Jan Vitek**

Purdue University, West Lafayette, IN, USA
[⋆]Contact author: rtsegels@purdue.edu

Keywords: testing, R implementation

When implementation a programming language like *R*, one of the biggest challenges is ensuring correctness of the many runtime functions that are part of Rs environment. For example, most of the extensively used operations like arithmetics are implemented in *C*. Overall current *GNU R* relies on 695 internal functions implemented in *C*. Constructing test cases for every function separately does not seem practical. We attack this problem with an automated method using the existing *GNU R* test suite. This test suite emerged with the evolution of *R*, but two main problems arise when trying to use it for testing a new implementation of *R* such as *FastR*. Firstly, it is not possible to tests functions separately, thus running the whole suite requires full implementation of all internal functions. Another problem is that errors in this test suite are hard to interpret. Where did the error come from exactly? How do we localize it?

This project presents an approach to generating a concise test suite that can be used for unit testing Rs internals. Our goal was to create a test suite for in internal functions that can be used to test each function separately, while maintaining the same code coverage level as regression test suite does. Our approach is based on instrumenting the *R* VM to capture calls to built in/special functions and generates test cases based on captured information. As a lot of those calls are redundant in terms of code coverage, we are also filtering them based, on the impact on code coverage.

We will explain the status of the project, and current results compared to full *R* test suite. Currently, we were able to generate tests that cover more that 80% of what *R* test suite covers while shrinking test suite size to only 3000 function calls. We will provide some thoughts about how this can be used for creating test suites for *R* packages and where to go from there.