

Oral Communication: Generating Network Data for Automated Unit Test Generation

Andrea Arcuri¹, Gordon Fraser², and Juan P. Galeotti³

¹ University of Luxembourg, Luxembourg
aa@scienta.no

² University of Sheffield, UK
Gordon.Fraser@sheffield.ac.uk

³ University of Buenos Aires, Argentina
jgaleotti@dc.uba.ar

Abstract. Although automated unit test generation techniques can in principle generate test suites that achieve high code coverage, in practice this is often inhibited by the dependence of the code under test on external resources. In particular, a common problem in modern programming languages is posed by code that involves networking (e.g., opening a TCP listening port). In order to generate tests for such code, we describe an approach where we mock (simulate) the networking interfaces of the Java standard library, such that a search-based test generator can treat the network as part of the test input space. This not only has the benefit that it overcomes many limitations of testing networking code (e.g., different tests binding to the same local ports, and deterministic resolution of hostnames and ephemeral ports), it also substantially increases code coverage. An evaluation on 23,886 classes from 110 open source projects, totalling more than 6.6 million lines of Java code, reveals that network access happens in 2,642 classes (11%). Our implementation of the proposed technique as part of the EvoSuite testing tool addresses the networking code contained in 1,672 (63%) of these classes, and leads to an increase of the average line coverage from 29.1% to 50.8%. On a manual selection of 42 Java classes heavily depending on networking, line coverage with EvoSuite more than doubled with the use of network mocking, increasing from 31.8% to 76.6%.

This oral communication will present the results of our extension to the EvoSuite tool that were published at the ESEC/FSE conference held at Bergamo, Italy, on September 2015.