Reading and understanding a real test suite

Validation and Verification
University of Rennes 1

José A. Galindo(jagalindo@us.es)
Simon Allier (simon.allier@irisa.fr)

Last update October 14, 2015

The objective of this session is not to write tests, but to explore and understand some parts of one of the biggest open test suite. You will have to answer questions on multiple aspects of the the suite, and finally you will have to use Spoon to analyze the tests.

1 Background: Apache Commons Math

Apache Commons Math[1] is a famous library of mathematics components for the Java programming language. It contains functions as simple as means computing, to much more complex such as equation solving or function optimization. Figure 1 describes all the packages of commons math, as given on the official website.

Installation Go to `https://github.com/apache/commons-math` and download the zip of the current version of the repository. Open it as a Maven project with IntelliJ.

2 Analysis of the commons math test suite

In this part you must read some parts of the test suite and answer some questions.

2.1 JUnit customization and random data generation

Question 1 *Find the org.apache.commons.math3.RetryRunner class. What class does it inherit from? Explain its purpose and how it works.*

Question 2 *How is RetryRunner tested?*

[1] `https://commons.apache.org/proper/commons-math/`

org.apache.commons.math3.stat - statistics, statistical tests

org.apache.commons.math3.analysis - rootfinding, integration, interpolation, polynomials

org.apache.commons.math3.random - random numbers, strings and data generation

org.apache.commons.math3.special - special functions (Gamma, Beta)

org.apache.commons.math3.linear - matrices, solving linear systems

org.apache.commons.math3.util - common math/stat functions extending java.lang.Math

org.apache.commons.math3.complex - complex numbers

org.apache.commons.math3.distribution - probability distributions

org.apache.commons.math3.fraction - rational numbers

org.apache.commons.math3.transform - transform methods (Fast Fourier)

org.apache.commons.math3.geometry - geometry (Euclidean spaces and Binary Space Partitioning)

org.apache.commons.math3.optimization - function maximization or minimization

org.apache.commons.math3.ode - Ordinary Differential Equations integration

org.apache.commons.math3.genetics - Genetic Algorithms

Figure 1: List of the commons math packages, from the commons math webpage

Question 3 *Analyze the two aspects of testing random data generation with classes* org.apache.commons.math3.random.Well512aTest *and* org.apache.commons.math3.random.RandomDataGeneratorTest.

Question 4 *Look in the* org.apache.commons.math3.TestUtils *class. Why are new assertion methods defined?*

Question 5 *Look in the* org.apache.commons.math3.primes.PrimesTest *class. How are exceptions tested? Can we do better?*

2.2 "Exotic" tests

Question 6 *Look into the* org.apache.commons.math3.optim.nonlinear.scalar.noderiv.BOBYQAOptimizerTest *class. What does it test? How? What are the test data? You might want to have a look here:* `https://en.wikipedia.org/wiki/Test_functions_for_optimization`

3 Dynamic analysis

Programming 1 *Write a spoon processor that counts the number of defined tests and and the number of defined assertions of the test suite.*

Programming 2 *Write a spoon processor that instruments the test suite in order to count the number of executed tests and the number of executed assertions.*

Question 7 *Compare the number of defined tests/assertions with the number of executed tests/assertions.*

4 What to produce

You have to produce:

- a report with answers to all the questions.
- a Maven project with your Spoon processors.