CS304 Software Engineering - Lab 6

Date: 23, April
Due: 23:59pm, 2 May, 2018

Instruction

In this lab, we will use a tool implemented by Kevin Bi@UW to see fault-localization with Tarantula in action. Before we start, you should have installed `maven` and `ant` which are the buiding management tools for `Java`. You should also have a working `JDK`. Linux or Mac OSX with Homebrew is recommended for this Lab.

Preparation

1. Install `Git` to clone the code repo or download the code:
   `https://git.cra.moe/11310380/fault-localization`

2. Change directory to the repo:

   ```
   cd fault-localization
   ```

Install Dependencies

1. Test and install primitive hamcrest onto the local repo (needed for Tarantula)

   - cd primitive-hamcrest
   - mvn test
   - mvn install

2. Compile tacoco (needed for Tarantula)

   - cd tacoco
   - mvn compile

3. Build triangle (for testing demo)

   - cd src/triangle
   - mvn clean package

Build and Run

1. Enter the tacoco-scripts dir, all the scripts to run tacoco are held here:

   `cd scripts/tacoco-scripts`

2. Run Tacoco: The run-tacoco script runs tacoco and creates cp.txt files in both the system-under-test dir and the tacoco dir the script requires the absolute paths of the dir to be evaluated and the tacoco directory, see the script for more information

   `./run-tacoco /absoluteolute/path/to/repo/triangle /absolute/path/to/repo/tacoco`

3. Run Jacoco.exec Analyzer: The run-jacoco script creates the jacoco.exec file and the .json files in the tacoco dir, the script will have to be run inside the tacoco dir so you can copy it into the tacoco directory, see the script for more information

```
cp -i run-jacoco ../../tacoco
cd ../../tacoco/
./run-jacoco /absolute/path/to/repo/triangle triangle /absolute/path/to/repo/tacoco
./run-jacoco /absolute/path/to/repo/triangle  /absolute/path/to/repo/tacoco
```

4. Compile the Tarantula classes (**in base dir**): `ant compile`

5. Run Tarantula's Main: Main requires two args(*see the file for more details*), the first is the absolute path to the cov-matrix.json file, the second is the name of the Test class, if the test program belongs to a package make sure to specify the package, ex: Triangle.TestSuite. Input the two arguments to the ant target run.tarantula

   `ant -Darg0=/abs/path/to/tacoco-compact-cov-matrix.json -Darg1=triangle.TestSuite run.tarantula`

   You should see output like this:

```
compile:
    [javac] Compiling 2 source files to /Users/KellyZhang/Documents/J
avaCode/fault-localization/bin

run.tarantula:
     [java] For program: triangle/Triangle.java
     [java] line 3: suspiciousness: -1.000000, confidence: -1.000000
     [java] line 4: suspiciousness: -1.000000, confidence: -1.000000
     [java] line 5: suspiciousness: 0.000000, confidence: 0.034483
     [java] line 6: suspiciousness: 0.000000, confidence: 0.034483
     [java] line 7: suspiciousness: -1.000000, confidence: -1.000000
     [java] line 8: suspiciousness: -1.000000, confidence: -1.000000
     [java] line 9: suspiciousness: -1.000000, confidence: -1.000000
     [java] line 10: suspiciousness: -1.000000, confidence: -1.000000
     [java] line 11: suspiciousness: 0.500000, confidence: 1.000000
```

The output means you have run a test suite using Tarantula successfully.

6. (optinal) Clean the tacoco dir: Run clean-tacoco in order to clean the dir so it can be reused, this means tacoco will have to be recompiled

```
cd scripts/tacoco-scripts
./clean-tacoco
mvn clean package (in tacoco dir)
ant clean (in base dir)
```

Assignment

1. **Implement the coloful visualization** of Tarantula ouput as described in *section 2.1 and 2.2* of this paper using Python3 sample code provided(*required to use Matplotlib, please try to make your code work*). You could change the original code to save the results to a file and read the file with Python. To draw the colormap, checkout tutorial here.

2. Submit a **report** with screenshots of results for every step (3 in *Install Dependencies*, 5 in *Build and Run*, total 8) and your code implemented in Assignment 1.

Note: plagiarism is strictly prohibited.