Informatics 115 – Fall 2018
Professor Iftekhar Ahmed
Homework 4
Assigned: Wednesday, November 14th, 2018
Due: Wednesday, November 21st, 2018 at 11:59PM

Applying static analysis on real world projects

Defects4J is a collection of reproducible bugs and a supporting infrastructure containing 395 bugs from the following open-source projects:

Identifier	Project name	Number of bugs
Chart	JFreeChart	26
Closure	Closure compiler	133
Lang	Apache commons-lang	65
Math	Apache commons-math	106
Mockito	Mockito	38
Time	Joda-Time	27

Follow the steps to install Defects4J (Taken from https://github.com/rjust/defects4j):
1. Clone Defects4J:

```
git clone https://github.com/rjust/defects4j
```

2. Initialize Defects4J (download the project repositories and external libraries, which are not included in the git repository for size purposes and to avoid redundancies):

```
cd defects4j
./init.sh
```

3.Add Defects4J's executables to your PATH:

```
export PATH=$PATH:"yourpath2defects4j"/framework/bin
```

4. Check installation:

```
defects4j info -p Lang
```

Example commands for using Defects4J (Taken from https://github.com/rjust/defects4j):
1. Get information for a specific project (commons lang):

```
defects4j info -p Lang
```

2. Get information for a specific bug (commons lang, bug 1):

```
defects4j info -p Lang -b 1
```

3. Checkout a buggy source code version (commons lang, bug 1, buggy version):

```
defects4j checkout -p Lang -v 1b -w /tmp/lang_1_buggy
```

4. Change to the working directory, compile sources and tests, and run tests:

```
cd /tmp/lang_1_buggy
defects4j compile
defects4j test
```

1. Everyone is assigned a specific bug and a project which you will find in the Bug assignment.csv file.
2. Checkout the buggy source code version and fixed source code version of the project and bug assigned to you.
3. Download and install **any** one of these 3 tools (Infer from https://fbinfer.com/) , PMD(https://pmd.github.io/), FindBugs(http://findbugs.sourceforge.net/)
4. Run **any one of these 3 tools** on buggy version.
5. Provide a csv file with the following information:
    a. Project, bugid, filename, line number, Full warning text
    b. Name the file using the structure Projectname_bugnumber_buggy.csv. For example, if you are working on Lang and bug 1 then the file name will be: Lang_1_buggy.csv
6. Run **the same tool used in step 4** on fixed version.
7. Provide a csv file with the following information by parsing the raw output from infer:
    a. Project, bugid, filename, line number, Full warning text
    b. Name the file using the structure Projectname_bugnumber_fixed.csv. For example, if you are working on Lang and bug 1 then the file name will be: Lang_1_fixed.csv
8. Summarize the information for the specific bug assigned to you and write them in Summary.txt
    a. Comparing the fixed version with buggy version (i.e. compare the buggy version of the file with fixed version).
    b. From the summary provided by the defects4j tool.
    c. From the triggering tests.
    d. Comparing the **tools** outputs (which statements are flagged by the used tool in both version, only in fixed version, only in buggy version, did the tool flag the actual buggy lines etc.)
9. The summarization categories listed above are not the only ways to think about a bug. **What is your intuition regarding the likelihood of finding the fault using information such as source code, a static analysis tool warnings and multiple test cases**. Write the intuition in Summary.txt with a separate header *Intuition*.
10. Make sure to mention the name of the tool in Summary.txt.

Note: We will be grading based on the successful completion of steps. In addition to that, for the written summary, we will be checking for the attempt of inference, rather than simply copy pasting the output from the infer outputs.

Upload Instructions

Turn in one zip file that contains the following files:

1. Script used for executing infer
2. Script used for parsing the infer output
3. Raw output from infer named: **infer_"Projectname"_"bugnumber"_fixed.csv**
4. Parsed output from infer named: **Projectname_bugnumber_fixed.csv**
5. Raw output from infer named: **infer_"Projectname"_"bugnumber"_buggy.csv**
6. Parsed output from infer named: **"Projectname"_"bugnumber"_buggy.csv**
7. Summary.txt
8. Scripts used for executing step 8.d. If you decide to do step 8.d manually, write down the steps with enough details so that someone reading them will be able to replicate them.