Appendix

This appendix describes the files provided in the following directories:

1. *Checkers\_src/* : The weak and strong checkers for all Defects4J subjects
2. *PPA\_tool/* :The profiling tool supporting propagation path analysis (Section 4)
3. *Checkers\_stats/* : Collected statistics on the triggered checkers
4. *Supplement\_Section3.3/*

The appropriate library versions that were used are indicated below.

|  |  |
| --- | --- |
| **Library** | **Version** |
| JDK | 1.7 |
| Defects4J | 1.1.0 |
| ASM | 6.0 |

# Weak and Strong Checkers

The *Checkers\_src* directory contains the checkers that were manually inserted for all Defects4j defects. The content of the directory has the following structure: *<project>/<version>/*. The directory specified by the above path contains the java source code files that have been augmented by weak and strong checkers. In order to use them, simply copy the file(s) to the corresponding project and bug versions of the Defects4j library.

For instance, *Checkers\_src\Chart\v1\AbstractCategoryItemRenderer.java* should replace *source\org\jfree\chart\renderer\category\AbstractCategoryItemRenderer.java* in *Chart v1*’s buggy version directory.

# Propagation Path Analysis Tool

This section describes the program used to instrument and profile the subject programs to generate the results in section 4. The code required, which depends on the *ASM* library, is available under *PPA\_tool/*. After compiling the library, run one of the following instructions to instrument the program:

|  |
| --- |
| java com.findtargettests.Main <classfile> |

or

|  |
| --- |
| java com.findtargettests.Main <jarfile> |

To collect the statistics, run the Defects4J program, making sure that *com.findtargettests* is in the *classpath*.

# Statistics

This section describes the statistics related to the checkers. The structure of the directory is as follows:

|  |
| --- |
| <Project Name>/Strong/ : Directory containing statistics on strong checkers  <Project Name>\_output.csv : Statistics on triggered checkers for all unit tests  <Project Name>\_cleaned.csv : Same as above, but checkers’ names are cleaned  <Project Name>\_Fail.csv : Statistics on triggered checkers for failing test cases  <Project Name>\_failingtestData.csv : Failing unit tests per version  <Project Name>\_Fail\_Analysis.csv : Analysis on the collected profiles for failing test cases  <Project Name>\_Pass.csv : Stats on triggered checkers for passing unit tests  <Project Name>\_Pass\_Analysis.csv : Analysis on the collected stats for passing test cases  <Project Name>/Weak/ : Directory containing statistics on strong checkers  *(contains similar files to the above)* |

Each CSV file contains the following columns.

|  |
| --- |
| **ORACLE** : Checker triggered by test case  **NUM\_STAT** : Number of statements after the checker was triggered  **NUM\_EXECSTAT** : Number of statements executed after the checker was triggered  **NUM\_COND** : Number of conditional statements after the checker was triggered  **NUM\_EXECCOND** : Number of conditional statements executed after the checker was triggered  **NUM\_MODULO** : Number of modulo statements after the checker was triggered  **NUM\_EXECMODULO** : Number of modulo statements executed after the checker was triggered  **NUM\_MULT** : Number of multiplication instructions after the checker was triggered  **NUM\_EXECMULT** : Number of multiplication instructions executed after the checker was triggered  **NUM\_DIV** : Number of division instructions after the checker was triggered  **NUM\_EXECDIV** : Number of division instructions executed after the checker was triggered  **NUM\_INVOKE** : Number of method-invoke instructions after the checker was triggered  **NUM\_EXECINVOKE** : Number of method-invoke instructions executed after the checker was triggered  **COUNT** : Number of times the checker was triggered  **TESTNAME** : The unit test name  **PROJECT** : Project Name (e.g. Chart, Time)  **VERSION** : Version Number  **FAIL** : TRUE if the unit test failed |

The following is a sample from *Chart* statistics, in *Checkers\_stats\Chart\Strong\Chart\_cleaned.csv*.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ORACLE** | **NUM\_ STAT** | **NUM\_ EXECSTAT** | **NUM\_ COND** | **NUM\_ EXECCOND** | **NUM\_ INVOKE** | **NUM\_ EXECINVOKE** | **COUNT** | **TESTNAME** | **PROJECT** | **VERSION** |
| Chart Bug 6 - Strong Oracle | 218 | 479 | 38 | 86 | 21 | 38 | 1 | org.jfree.chart. renderer.xy.junit. HighLowRendererTests. testEquals | CHART | 6 |