How to set-up the experiment ---- details

# There are three main steps to do the experiment: prepare the subject, conduct the experiment, and collect the result.

Prepare the subject.

Main steps:

1. Exhaustive run the test sets

For each possible configuration, run the test case under the configuration and collect the outcomes. Then , obtain the mapping collection: test configuration 🡪 outcome.

For example, for the HSQLDB 2.rc8, the result should be collected as follows (Note that we test 8 options on this version, the test model is 3×2×2×2×2×2×2×2×4×3×2×2):

0 0 0 0 0 0 0 0 0 0 0 0 🡪 pass …...

1 0 0 0 1 0 1 0 0 0 0 1 🡪 exception1

……

1 0 1 0 1 0 1 0 2 0 0 1 🡪 exception2

…...

2 1 1 1 1 1 1 1 3 2 1 1 🡪 pass

Specifically, for different subject, this preparation is as follows:

* HSQLDB: Import the test 2rc8, test2.25, and test 2.29, into the eclipse as a project. Each subject has one test case which are, *test2rc8 – scr/org.hsqldb.test/ScrollAndLongString.java, test2.25 -- scr/org.hsqldb.test/IncompatibleDataAndUpdateRow.java, and test2.29 -- scr/org.hsqldb.test/TestJoinAndDelete.java*, respectively. Run the main testing function (*test2rc8 – scr/org.hsqldb.test/TestScrollAndLong.java, test2.25 -- scr/org.hsqldb.test/TestIncompatibleDataAndUpdateRow.java, and test2.29 -- scr/org.hsqldb.test/JoinAndDeleteRows.java*), which test the test case under each possible configuration of that subject. The results will show into file “resultNew.txt”, and note that we use different number to represent different fault a configuration triggers, the mapping of number to specific exception information is listed in file “bugInfoNew.txt”.
* JFlex: Import the Jflex1.4.1, Jflex1.4.2 into the eclipse as a project. The test case of each subject is based on the text files under each subject (For Jflex1.4.1, they are State.jflex, State\_options.jflex, State\_normal.jflex, State\_normal\_options.jflex, Bug.jflex, Bug\_options.jflex, Bug\_add.jflex, and Bug\_add\_options.jflex , while for Jflex 1.4.2, they are testF.jflex, testF\_options.jflex,remove\_type.jflex,remove\_type\_options.jflex, Bug2.jflex, Bug2\_remove.jflex, Bug2\_remove\_options.jflex, and Bug2\_options.jflex ). Run the main testing function (*Jflex1.4.1– scr/org.jflex.test/TestJFlex.java, Jflex1.4.2-- scr/org.jflex.test/TestJFlex.java*). The results will show into file "result\_of\_testCase.txt", and the mapping of number to specific exception information is listed in file “bugInfo.txt”.
* Grep:
* Synthetic: Import. For each synthetic subject, the test case is just a function that maps the test configuration to an outcome. Run the, and get the result.

1. Get the real MFS of each subject.

Through inspecting the source code and the bug tracker information (which is given in the readme.txt), we get the MFS of each subject. Specifically, they are as following:

• HSQLDB : 2rc8 --- () () , 2.25 --- (),(). 2.29 (),()

• JFlex : 1.4.1 --- () () . 1.4.2--- ()()

• Grep:

• Synthetic:

Conduct the experiment

Main steps:

For each failing test case, we run three subject: FIC\_BS with two strategies and Replacement strategy with ILP and random approach.

Collect the result

Available tools

Each of them has a ReadMe to show how to run it.

* Tools that can implemented our approach.
* Tools that implemented the FDA-CIT.