Pin Wang (wangp5)

Mengxing Chen (chenm32)

Proposal for Automatic Test Assertion Generator

Motivation

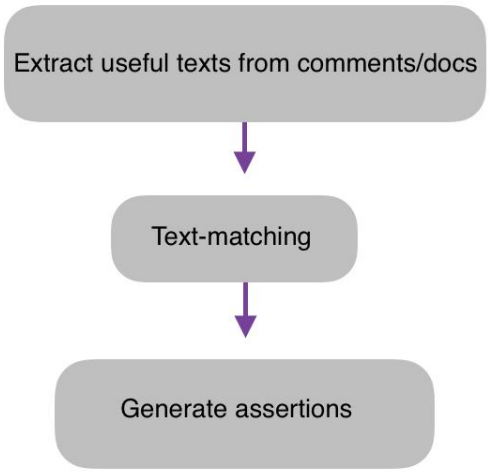
Test assertions indicate the expected behavior of a program. They are important, but programmers are reluctant to write them. They are also redundant with procedure documentation — programmers should not have to provide the same information multiple times. So we want to have a tool that efficiently generate test assertions from comments and documents.

Current Solution And Its Limitations

Toradocu is an existing tool that can match text in the Javadoc comment with method and variable names. However, there are some limitations of it: it can only identify descriptions tagged with @throws or @exception, and will miss untagged descriptions of exceptional behavior. It only looks at the text in the Javadoc comment, and doesn't consider the documentation of the method or variable. Nonetheless, Toradocu has other issues to be solved or improvements that can be made which are listed in its issue page on GitHub. (A. (n.d.). Albertogoffi/toradocu. Retrieved March 29, 2018, from https://github.com/albertogoffi/toradocu/issues)

Our Approach

We are going to analyze Toradocu's weaknesses and extend it to make it more effective. We will both look at the source code of Toradocu and run a huge bunch of tests to see the behavior of it. Also, we will let Toradocu read not only from Javadoc comments, but also from documentation of methods and variables using similar techniques Toradocu used. These methods include: using a Javadoc extractor that identifies all the comments related to exceptional behaviors for each method of a class under test (CUT); having a condition translator that uses a mix of natural language processing techniques and pattern-matching to translate Javadoc conditions into Java boolean expressions; and, finally, using an oracle generator to convert the output of the condition translator into a test oracle and injects the oracle into any test case that invokes the method. (Goffi, A., Gorla, A., Ernst, M. D., & Pezzè, M. (2016). Automatic generation of oracles for exceptional behaviors. Proceedings of the 25th International Symposium on Software Testing and Analysis - ISSTA 2016. doi:10.1145/2931037.2931061)



Our tool improves on the basis of Toradocu: it will look at the document of methods and variables in order to do text-matching. Also, for detection of exceptional behaviors, our tool focuses on not only tagged descriptions but also natural-language comments. These extensions will make our automatic assertion generator more accurate.

Suppose there is a method called checkAllocated is given to Toradocu, shown as following. Right now, Toradocu can not recognize the CannotWriteException, but after our extension, we would expect it to generate the following test:

```
/* check whether the block has already been allocated
   throw CannotWriteException if the block has not been allocated yet
*/
public void checkAllocated(Block b) {...}


public void test() {
    Block testB = new Block();
    if (!testB.isAllocated()) {
        try {
            testB.checkAllocated();
            error("CannotWriteException not thrown");
        } catch (CannotWriteException e) {
            // succesfully throw exception
        }
    } else {
        testB.checkAllocated();
        // should not throw exception here
    }
}
```

Challenges And Risks

One of the most serious challenges in extending Toradocu is the lacking of knowledge about natural language processing, which might be time consuming. Also, considering documentations of methods and variables may let Toradocu make unnecessary assertions or even unwanted assertions.