Adaptating Amplified Unit Tests for Human Comprehension

Simon Bihel

simon.bihel@ens-rennes.fr

Thursday 8th March, 2018

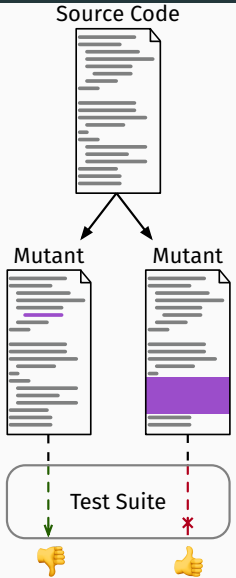
University of Rennes I
École Normale Supérieure de Rennes

Mutation Testing

Evaluating the quality of a test suite by injecting bugs

Example of *mutators*:

- change a > condition with <;
- delete the body of a method.



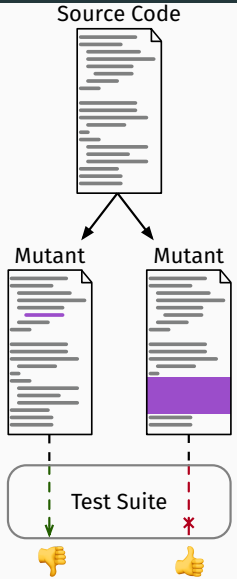
Mutation Testing

Evaluating the quality of a test suite by injecting bugs

Example of *mutators*:

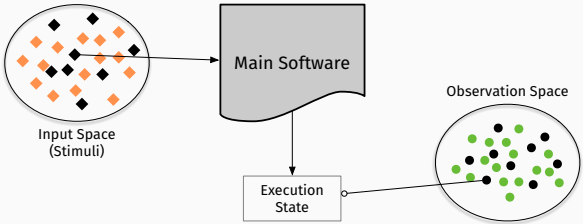
- change a > condition with <;
- delete the body of a method.

Goal Enhance test suite by detecting new mutants



DSpot[1]

Randomly modifies test cases:

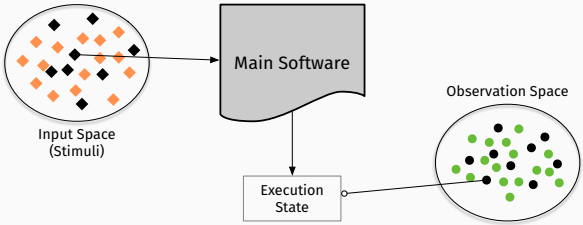


[1] https://github.com/STAMP-project/dspot

DSpot[1]

Randomly modifies test cases:

- new inputs to trigger new behaviors;

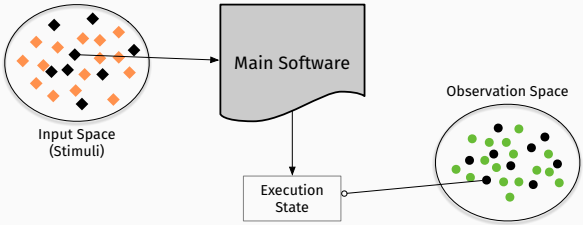


[1] https://github.com/STAMP-project/dspot

DSpot[1]

Randomly modifies test cases:

- new inputs to trigger new behaviors;
- new assertions for unchecked properties;

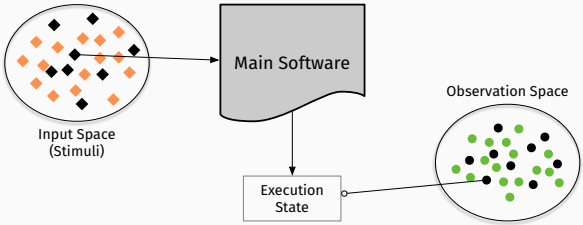


[1] https://github.com/STAMP-project/dspot

DSpot[1]

Randomly modifies test cases:

- new inputs to trigger new behaviors;
- new assertions for unchecked properties;
- targets regression.



Benjamin Danglot, INRIA Lille, France

[1] https://github.com/STAMP-project/dspot

Example[2]

```java
@Test
public void immutableGraph() {
  MutableGraph<String> mutableGraph = GraphBuilder.directed().build();
  mutableGraph.addNode("A");
  ImmutableGraph<String> immutableGraph = ImmutableGraph.copyOf(mutableGraph);

  assertThat(immutableGraph).isNotInstanceOf(MutableValueGraph.class);
  assertThat(immutableGraph).isEqualTo(mutableGraph);

  mutableGraph.addNode("B");
  assertThat(immutableGraph).isNotEqualTo(mutableGraph);
}
```

[2] https://github.com/google/guava/blob/master/guava-tests/test/com/google/common/graph/ImmutableGraphTest.java#L29-L40

Example of amplification

```java
@Test
public void immutableGraph() {
  MutableGraph<String> mutableGraph = GraphBuilder.directed().build();
  mutableGraph.addNode("A");
  mutableGraph.addNode("C");
  ImmutableGraph<String> immutableGraph = ImmutableGraph.copyOf(mutableGraph);

  assertThat(immutableGraph).isNotInstanceOf(MutableValueGraph.class);
  assertTrue(immutableGraph.nodes().contains("A"));
  assertThat(immutableGraph).isEqualTo(mutableGraph);

  mutableGraph.addNode("B");
  assertThat(immutableGraph).isNotEqualTo(mutableGraph);
}
```

Goal

- Human-friendly, high-level, natural language description

Minimization

- remove useless assertions

Slicing of Unit Tests

Simple static slicing

- Control-flow slicing
- Data-flow slicing

Java tool

T.J. Watson Libraries for Analysis (WALA)[3]

[3] https://github.com/wala/WALA

Natural Description

- Mutator categorization