Grafter: Automated Transplantation and Differential Testing for Code Clones

Tianyi Zhang, Miryung Kim
University of California, Los Angeles

**Problem Statement**

- Code clones are common in modern software systems.
- It is difficult to examine and contrast the runtime behavior of code clones due to a lack of test cases.

```java
public class Copy extends Task{
    private IncludePatternSet includes;
    public void setIncludes(String patterns){
        String[] tokens = StringUitls.split(patterns, ".");
        for(String tok : tokens){
            includes.addPattern(tok);
        }
    }

    public void testCopy()
    {
        testCopy();
    }
}
```

**Approach**

- Grafter uses de-facto analysis to expose the de-facto interfaces (i.e., input & output parameters) of clones, grafts one clone in place of its counterpart, and runs the same test on the grafted clones.
- Grafter illuminates and exposes behavioral differences of clones at a find-grained level by pinpointing which variables’ states differ in which test.

```java
public class Delete extends Task{
    private ExcludePatternSet excludes;
    public void setExcludes(String patterns){
        String[] tokens = StringUitls.split(patterns, ".");
        for(String tok : tokens){
            excludes.addPattern(tok);
        }
    }

    public void testDelete()
    {
        testDelete();
    }
}
```

**Implementation & Tool**

- Grafter supports detecting clones using existing clone detectors such as Deckard and Simian or loading clones manually.
- Grafter allows users to manually inspect clones in a side-by-side diff view and further update the loaded clones.
- Grafter automatically locates corresponding test cases of each clone using test coverage analysis.
- Grafter allows users to experiment the clone transplantation process.

**Evaluation**

**Grafting Capability**

- Among 52 pairs of non-identical clones from 3 open-source projects, Grafter successfully transplants 49 of them without inducing any compilation errors.

**Behavioral Comparison Capability**

- Grafter detects up to 2X more injected faults (i.e., mutants) than a static approach [Jiang et al., FSE 2007].
- Grafter is less biased to different types of mutants than Jiang et al.

**Future work:**

- We will investigate new techniques to transplant test cases as opposed to transplant clones for test reuse.
- We will design a new differential testing technique to assess behavioral similarity of Java bytecode clones and to also increase the confidence of behavioral preservation after automated clone removal.

This work is supported by AFRL grant FA8750-15-2-0075, and NSF grants CCF-1527923 and CCF-1460325