Identifying transformation issues in refactoring engines

Melina Mongiovì

melina@computacao.ufcg.edu.br

Federal University of Campina Grande

It's difficult to implement refactorings!

Transformation Issues

- Compile
- Preserve the program behavior
- Don't follow the refactoring definitions
How to detect transformation issues?

Differential Testing Oracle (DT)

Identifying transformation issues

8 refactoring implementations
18 transformation issues detected

Identifying transformation issues in refactoring engines
Eclipse's output program has a class that JRRT's output program does not have:

```java
public class A {}
```

```java
class B extends A {
    public int m() {
        return 1;
    }
}
```

### Changes considered by our program comparator

<table>
<thead>
<tr>
<th>Level of Change</th>
<th>Real of Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>Package</td>
<td>Set of packages</td>
</tr>
<tr>
<td>Class/Interface</td>
<td>Set of methods</td>
</tr>
<tr>
<td>Field</td>
<td>Set of modifiers</td>
</tr>
<tr>
<td>Method</td>
<td>Type</td>
</tr>
</tbody>
</table>

### Structural Change Analysis Oracle (SCA)

#### Check Refactoring Definitions

<table>
<thead>
<tr>
<th>Refactoring Type</th>
<th>Must do</th>
<th>Must not do</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add method</td>
<td>In the superclass</td>
<td>In the subclasses of A that have the same signature</td>
</tr>
<tr>
<td>Remove method</td>
<td>In the subclasses of A</td>
<td>In the same class</td>
</tr>
</tbody>
</table>

#### Push Down Method A.m()

**Must do:**
- Update method calls
- Add the method in the subclasses

**Must not do:**
- Remove the method in the subclasses

#### Pull Up Method B.m()

**Must do:**
- Add the method in the subclass

**Must not do:**
- Pull up method that are in the subclasses of A
Pull Up Method B.m()

Must not do:

There is another method with the same signature

There is no superclass

Encapsulate Field A.f

Must do:

Update field accesses

Create getter methods

Create setter methods

A

public int f = 1;

public void setF(int f) {
    return f;
}

public int getF() {
    return f;
}

public static long getF() {
    return f;
}

public static long setF(long f) {
    return f;
}

public int test(int x) {
    return x;
}

private int f = 1;

private void getF() {
    return f;
}

private void setF(int f) {
    return f = f;
}

private static long getF() {
    return f;
}

private static long setF(long f) {
    return f;
}

Encapsulate Field A.f

Must not do:

The field is private

There is a get or set method with the same signature as the method to be created
For all transformations

**Must not do:**

Remove a class, method or field of the program
Add a class, method or field in the program
Make abstract a class or method
Reduce accessibility of a class, method or field

For all transformations

**Must not do:**

Remove a class, method or field of the program
Add a class, method or field in the program
Make abstract a class or method
Reduce accessibility of a class, method or field

How to classify failures into distinct issues?

Issue categorizer for the DT oracle outputs

Issue categorizer for the SCA oracle outputs (for each kind of refactoring)

Evaluation

8 refactoring implementations
2,189,705 Alloy instances
87,587 generated programs
18 transformation issues
Refactoring Scope Alloy Programs (P-C-M-F) Compilable (GP)
Encapsulate Field 2-3-1-1 350,550 13,622 100
Pull Up Field 2-3-2-2 1,051,608 42,064 99.96
Pull Up Method 2-3-3-0 332,370 13,294 97.24
Push Down Method 2-3-4-0 255,177 10,107 99.47
Total/Average 2,189,705 87,587 99.50

Can the proposed technique using the Differential Testing oracle detect transformation issues?

Can the proposed technique using the Structural Change Analysis oracle detect transformation issues?

Can the proposed technique using the Differential Testing oracle detect transformation issues?
Can the proposed technique using the Structural Change Analysis oracle detect transformation issues?

Identifying transformation issues in refactoring engines

Melina Mongiovi
melina@computacao.ufcg.edu.br
Federal University of Campina Grande