

Abschlusspräsentation

Birgit Pohl, Philipp Badenhoop

Tim Sikatzki, Daniel Bucher

Software Engineering II

Humboldt-Universität zu Berlin, Institut für Informatik



Verantwortlicher: S. Heiden

Ergebnisse

Idee

Erster Ansatz

- ▶ Suche nach einem Opensourceprogramm
- ▶ Über www.codetriage.com
 - ▷ Zur Vermittlung von Opensourceprojekten an Entwickler
- ▶ Auswahl von Astor als Debuggingtool

libgdx

- ▶ Ist ein Java-Framework
- ▶ Für plattformunabhängige Spielentwicklung
- ▶ Unterstützt Windows, Mac, Linux, Android, iOS & Blackberry
- ▶ Unter Apache-2-Lizenz freigegeben

Astor

- ▶ Automatic Software Transformations fOr program Repair
- ▶ Für automatische Reparatur von Java Programmen
- ▶ Ursprüngliche implementaiton in C, jetzt in Java
- ▶ Besteht aus 3 Programmteilen
 - ▷ jGenProg2
 - ▷ jKali
 - ▷ jMutRepair

Probleme

- ▶ Installationsprobleme von libgdx
 - ▷ Benötigt spezielle Bibliotheken
- ▶ Probleme bei Astor
 - ▷ Kann Fehler nicht fixen, welche GenProg können soll
- ▶ Neuer Fokus auf Astor

Astor - GenProg

Astor

- ▶ Wendet eins der drei Modi an
 - ▷ jGenProg2
 - ▷ jKali
 - ▷ jMutRepair
- ▶ Unser Fokus wurde auf GenProg gelegt

Kali

- ▶ Zielt auf schwache Testsuits
- ▶ Vorgehen bei der Reperatur":
 - ▷ löschen von Zeilen
 - ▷ überspringen von Zeilen

MutRepair

- ▶ Mutiert die Konditionen von if-Statements
- ▶ Hat drei Arten der Mutation:
 - ▷ Relations Operationen
 - ▷ Logische Operationen
 - ▷ Negation

GenProg

- ▶ Idee: Reparatur durch Evolution
 - ▷ Nutzung von generischer Programmierung
 - ▷ gezielte, zufällige Mutation

Vorgehen

- ▶ Fehlerlokalisierung
 - ▷ Erstellung eines abstrakten Syntax Baums
 - ▷ Durchlaufen der Testfälle
 - ▷ Fehlerbestimmung anhand von Pfaden mit negativen Testfällen
- ▶ Patch-Generierung
 - ▷ Mutation von Crossovervarianten
 - ▷ Solange, bis ein optimaler Kandidat gefunden wurde
- ▶ Validierung
 - ▷ Prüfen des Testsuits

Problemstellung



- ▶ Philipp baut hier noch Folien



Ergebnisse

Die Testmenge

- ▶ Als Testmenge wurden die Fehler von defects4j verwendet.
- ▶ 395 Test stehen zur Verfügung in folgenden Bereichen
 - ▷ JFreechart (26)
 - ▷ Closure compiler (133)
 - ▷ Apache commons-lang (65)
 - ▷ Apache commons-math (106)
 - ▷ Mockito (38)
 - ▷ Joda-Time (27)

Project	jGenProg	jKali	jMutRepair
Chart	C1, C3, C5, C7, C13, C15, C25,	C1, C5, C13, C15, C25, C26	C1, C7, C25, C26
	$\sum = 7$	$\sum = 6$	$\sum = 4$
Lang	0	0	L27
	$\sum = 0$	$\sum = 0$	$\sum = 1$
Math	M2, M5, M7, M8, M28, M40, M49, M50, M53, M60, M70, M71, M73, M78, M80, M81, M82, M84, M85, M95	M2, M8, M28, M32, M40, M49, M50, M78, M80, M81, M82, M84, M85, M95	M2, 28, 40, 50, 57, 58, 81, 82, 84, 85, 88
	$\sum = 20$	$\sum = 14$	$\sum = 11$
Time	T4, T11	T4, T11	T11
	$\sum = 2$	$\sum = 2$	$\sum = 1$
Total	29	22	17

Entnommen aus [MM16]

Erste Überlegung

- ▶ Anwendung auf das Paket Lang
- ▶ Höchste Wahrscheinlichkeit einen Fix zu finden
- ▶ Leider nicht erfolgreich gewesen

Nächste Überlegung

- ▶ Durchsuchen aller Fehlerhaften Programme
- ▶ Überprüfung der fehlerhaften Testfälle
- ▶ Kein Testfall gefunden, welcher den Patch validieren würde

Validierung

- ▶ Entwicklung eines eigenen Testprogramms
- ▶ Welches den Patch validieren kann

```
1 public Language getLanguage(String lang) {  
2     if (lang.equals("C"))  
3         return Language.C;  
4     else if (lang.equals("CPP"))  
5         return Language.CPP;  
6     else  
7         return Language.JAVA;  
8 }
```

```
1 public Language getLanguageWorking(String lang) {  
2     if (lang.equals("C"))  
3         return Language.C;  
4     else if (lang.equals("CPP"))  
5         return Language.CPP;  
6     else if (lang.equals("JAVA"))  
7         return Language.JAVA;  
8     else  
9         return Language.PYTHON;  
10 }
```

Von Astor generierter Patch

```
1    public Language getLanguage(String lang) {  
2        if (lang.equals("C"))  
3            return Language.C;  
4        else if (lang.equals("CPP"))  
5            return Language.CPP;  
6        else if (lang.equals("JAVA"))  
7            return Language.JAVA;  
8        else  
9            return Language.PYTHON;  
10    }
```

Quellen



R. Just, D. Jalali, and M. D. Ernst.

Defects4j: A database of existing faults to enable controlled testing studies for java programs, 2014.



M. Martines and M. Monperrus.

ASTOR: A Program Repair Library for Java.
2016.