

EvoSuite

Automatic Test Suite Generation for Java

Adrian Uffmann
Matrikelnummer: 12043921
`adrian.uffmann@campus.lmu.de`

09.02.2020

Masterseminar Fuzz Testing

Zusammenfassung

EvoSuite ist ein Programm, welches mithilfe eines genetischen Algorithmus automatisiert JUnit-Testfälle aus Java-Bytecode generieren kann. Auf der Internetseite des Projektes (<http://www.evosuite.org/publications/>) werden 54 Publikationen zu EvoSuite und dessen Vorgänger *μTest* aufgelistet. Diese Seminararbeit gibt einen Überblick über die Funktionsweise von EvoSuite.

1 Einleitung

Das Testen von Software ist ein wichtiger Teil der Softwareentwicklung, so schreiben Myers et al. [1]:

it [is] a well-known rule of thumb that in a typical programming project approximately 50 percent of the elapsed time and more than 50 percent of the total cost [are] expended in testing the program or system being developed.

Daher ist es kaum verwunderlich, dass es zahlreiche Werkzeuge gibt, die helfen sollen das Testen von Software zu automatisieren. EvoSuite geht noch einen Schritt weiter: hier wird nicht versucht das Testen der Software zu automatisieren, sondern das Erstellen von automatisierten Tests selbst durch Automatisierung zu vereinfachen. Der Fokus liegt dabei auf der Optimierung eines *test last* Ansatzes, d. h. getestet wird erst, nachdem das eigentliche Programm fertig implementiert ist. Bei diesem Ansatz würde sich ein Softwaretester zunächst Szenarien überlegen, mit denen er möglichst viele Pfade des Programms abdeckt und diese als Unittests ausformulieren. Mithilfe von sogenannten Orakeln wird dabei ein bestimmtes Verhalten des Programms

definiert. Verhält sich das Programm anders als von den Orakeln vorgegeben, so schlägt der Test fehl und es wurde ein Bug gefunden.

EvoSuite versucht nun diesen Prozess zu vereinfachen, indem es Unittests generiert, die das Verhalten des Programms möglichst gut beschreiben. Ein Softwaretester muss sich dann nicht mehr selbst Szenarien überlegen, sondern nur noch die generierten Testfälle auf unerwartetes Verhalten prüfen. Dazu nutzt EvoSuite einen genetischen Algorithmus mit dem zufällige Testfälle immer weiter verbessert werden, bis sie eine möglichst hohe Testabdeckung erreichen. Die Testfälle werden dann minimiert um die Arbeit des Softwaretesters zu vereinfachen und es werden mit sogenanntem *Mutation-Testing* möglichst repräsentative Orakel ausgewählt.

EvoSuite wurde seit 2010 immer weiter entwickelt und es sind verschiedene Algorithmen für das generieren von Testfällen implementiert, die über die Kommandozeile ein- und ausgeschaltet werden können. Dazu haben die Autoren von EvoSuite auf der Internetseite (<http://www.evosuite.org/publications/>) 54 Artikel über die Konzepte in und die Evaluation von EvoSuite und dessen Vorgänger $\mu Test$ veröffentlicht. Um den vorgegebenen Umfang einzuhalten, kann diese Seminararbeit nur Überblick über einen kleinen Teil von EvoSuite geben.

2 Genetische Algorithmen

3 Erzeugen von Testfällen durch Optimierung der Testabdeckung

4 Erzeugen von Orakeln mit Mutation-Testing

5 Search-Based Software Testing Competition

6 Praxistauglichkeit

7 Fazit

Literatur

- [1] Glenford J Myers, Tom Badgett, Todd M Thomas, and Corey Sandler. *The Art of Software Testing*. New Jersey: John Wiley & Sons, Inc., 2004.