DISS. ETH NO. 22477

Automatic Fixing of Programs with Contracts

A dissertation submitted to attain the degree of
DOCTOR OF SCIENCES of ETH ZURICH

presented by
YU PEI
Ph.D. in Science, Nanjing University, Nanjing, China
Bachelor of Engineering, Nanjing University, Nanjing, China

born on
December 24th, 1977

citizen of
China

accepted on the recommendation of
Prof. Dr. Bertrand Meyer, examiner
Prof. Dr. Andreas Zeller, co-examiner
Prof. Dr. Harald Gall, co-examiner
Dr. Manuel Oriol, co-examiner

2015

ABSTRACT

Debugging—the activity of finding and correcting errors in programs—is so everyday in every programmer's job that any improvement at automating even parts of it has the potential for a significant impact on productivity and software quality. While automation remains formidably difficult in general, the last few years have seen the first successful attempts at automatically generating fixes to errors in some situations. This thesis aims at advancing the techniques and tools for the automatic fixing of errors in object-oriented programs with contracts (a.k.a. assertions).

To this purpose, the thesis has developed techniques and a supporting tool, collectively called AutoFix, that programmers can use in their everyday development to generate fixes to errors in an automatic fashion. AutoFix relies on the presence of simple specification elements in the form of contracts (such as pre- and post-conditions) to provide high-quality fix suggestions and to enable automation of the whole debugging process: using contracts enhances the precision of dynamic analysis techniques for fault detection and localization, and for validating fixes.

AutoFix consists of three major parts: the ImpleFix technique which generates fixes to the program implementation, the SpeciFix technique which suggests fixes to the contracts, and the AutoFix tool which implements both ImpleFix and SpeciFix.

Both ImpleFix and SpeciFix are driven by a set of test cases exercising the routine where the fault occurs. ImpleFix employs various program analysis techniques like dynamic invariant inference, simple static analysis, and fault localization to produce a collection of candidate fixes that change the implementation; while SpeciFix infers dynamic invariants in passing tests to summarize abstract program behavior, and synthesizes weakening and strengthening changes to the contracts to avoid failing behaviors. The generated fixes are validated against a regression test suite and ranked by preferring the ones that are more relevant to the failure or lead to less restrictive contracts. In the experiments conducted to evaluate the techniques, ImpleFix and SpeciFix generated fixes that are genuine corrections of quality comparable to those competent programmers would write to 25% of the subject faults.

The AutoFix tool is integrated into the Eiffel Verification Environment and functions like a recommendation system that is capable of automatically finding bugs and suggesting fixes in the form of source-code patches: it exploits the AutoTest random testing framework to detect errors and prepare test cases, and applies the ImpleFix and SpeciFix techniques to generate candidate fixes to the errors.

In conclusion, this thesis provides an automatic and integrated solution to the fixing of errors in object-oriented programs with contracts, which can greatly reduce the programmer's debugging effort in many cases.

ZUSAMMENFASSUNG

Debugging — das Finden und Korrigieren von Fehlern in Programmen — ist für Programmierer so alltäglich, dass sich Verbesserungen und Automatisierungen dieses Prozesses sehr positiv auf die Produktivität und Softwarequalität auswirken können. Obwohl Automatisierung in diesem Gebiet schwierig ist, wurden in den letzten Jahren erste erfolgreiche Ansätze zum automatischen Erzeugung von Fixes für Programmfehler vorgestellt. Diese Dissertation versucht die Techniken und Werkzeuge für das automatische Korrigieren von Fehlern in objekt-orientierten Programmen, welche mit Contracts versehen sind, zu verbessern.

Wir präsentieren in dieser Dissertation sowohl eine Technik, als auch ein dazugehöriges Werkzeug, die gemeinsam als AutoFix bezeichnet werden. Programmierer können AutoFix in ihrer täglichen Arbeit benutzen, um Korrekturen von Fehlern automatisch zu erzeugen. AutoFix stützt sich auf einfache Spezifikationselemente in Form von Contracts (z.B. Vor- und Nachbedingungen), um qualitativ hochwertige Vorschläge für Korrekturen zu erzeugen und eine Automatisierung des gesamten Debugging-Prozesses zu ermöglichen: Contracts erlauben dabei die Präzision der dynamischen Analysetechniken zur Erkennung und Lokalisierung von Fehlern zu erhöhen und der automatische Bewertung generierter Fixes zu verbessern.

AutoFix besteht aus drei Hauptteilen: der ImpleFix Technik, welche Korrekturen für Programmcode erzeugt; der SpeciFix Technik, welche Korrekturen von Contracts vorschlägt; und dem AutoFix Werkzeug, das sowohl ImpleFix als auch SpeciFix implementiert.

Sowohl ImpleFix als auch SpeciFix nutzen eine Gruppe von Tests, welche den Fehler einer Routine aufzeigen. ImpleFix benutzt verschiedene Analysetechniken wie dynamische Inferenz von Invarianten, einfache statische Analyse und Fehlerlokalisierung, um eine Sammlung von Kandidaten-Fixes zu produzieren, welche die Implementierung der Routine verändern. SpeciFix leitet aus erfolgreichen Tests dynamische Invarianten ab, um abstraktes Programmverhalten zusammenzufassen, und erzeugt abschwächende oder verstärkende Änderungen der Contracts, um fehlerhaftes Verhalten zu vermeiden. Die erzeugten Fixes werden mithilfe einer Regressionstestsuite validiert und gewichtet, wobei die Fixes, wel-

che relevanter für den Programmfehler oder zu weniger einschränkenden Contracts führen, bevorzugt werden. In den Experimenten, welche zur Evaluation der Techniken durchgeführt wurden, haben ImpleFix und SpeciFix in 25% der Fälle Fixes generiert, welche vergleichbar mit Fixes von erfahrenen Programmieren sind.

Das AutoFix Werkzeug ist in die Eiffel Verifikationsumgebung integriert und funktioniert wie ein Empfehlungssystem, das automatisch Fehler findet und Fixes in Form von Quellcode-Patches vorschlägt: es benutzt das AutoTest Test-Framework um Fehler zu erkennen und Tests vorzubereiten, und verwendet die ImpleFix und SpeciFix Techniken um Kandidaten-Fixes für Fehler zu erzeugen.

Zusammengefasst bietet diese Arbeit eine automatische und integrierte Lösung für das Korrigieren von Fehlern in objekt-orientierten Programmen mit Contracts, die den Debuggingaufwand für Programmierer erheblich reduzieren kann.