Universität Stuttgart

INSTITUT FÜR
KOMMUNIKATIONSNETZE
UND RECHNERSYSTEME
Prof. Dr.-Ing. Andreas Kirstädter

Bachelor-Arbeit / Forschungsarbeit Nr. 1090

Automatisierte Testfallgenerierung zum Test eines Compilers für eine Variante der Programmiersprache Go

Methoden
Programmierung
Softwareentwurf

Themengebiete
Compilerbau
Testen

Hintergrund

Am Institut entsteht derzeit eine Systemsprache für ein experimentelles, objektbasiertes Rechnersystem. Es handelt sich dabei um eine Variante der Programmiersprache Go, die den Namen GoSUB trägt. Dazu entstehen auch prototypische Compiler: Ein Compiler, der Code für konventionelle Rechner erzeugt und das Bootstrapping-Problem für den anderen, in GoSUB implementierten Compiler löst, der Code für das auf einem FPGA realisierte Rechnersystem erzeugt.

Um die Qualität von Compilern zu verbessern, kommen Testfallgeneratoren wie "randprog", "csmith" und "yarpgen" zum Einsatz. Sie nutzen den sogenennten "Fuzzing-Ansatz" und versprechen, zahlreiche Compilerfehler aufzudecken. Ein solcher Generator nutzt intern eine Code-Zwischendarstellung, zum Beispiel einen abstrakten Syntaxbaum (AST). Dabei erzeugt der Generator Syntaxbaumknoten pseudo-zufällig unter Berücksichtigung von Sprachregeln, zum Beispiel zur Sichtbarkeit von Bezeichnern oder zur Zuweisbarkeit. Abschließend wandelt er die Zwischendarstellung in Quellcode um, compiliert diesen mit verschiedenen Compilern und vergleicht die Ausführungsergebnisse. Dazu muss das generierte Programm Werte globaler Variablen und Ergebnisse von Funktionsaufrufen geeignet ausgeben.

Aufagabenstellung

Im Rahmen dieser Arbeit entwerfen Sie einen Testfallgenerator in GoSUB, der automatisiert Testprogramme generiert, compiliert, ausführt und deren Ergebnisse auswertet. Die Arbeit gliedert sich in die folgenden Schritte:

- Einarbeitung in die Sprache GoSUB
- Literaturrecherche zu Testfallgeneratoren
- Entwurf und Implementierung eines Verfahrens zum Vergleich von Ausführungsergebnissen
- Implementierung der Quellcodeerzeugung unter Verwendung einer bestehenden Bibliothek zur Wandlung von ASTs in Quellcode
- Testfälle automatisiert ausführen und auswerten

Erworbene Kenntnisse und Fähigkeiten

Sie verstehen, wie ein Compiler funktioniert und vertiefen Ihr Verständnis von Programmiersprachkonzepten. Sie lernen neue Sprachmechanismen kennen und können Mechanismen in Programmiersprachen kritisch hinterfragen. Schließlich sammeln Sie Erfahrung mit der Implementierung und Verwendung moderner Testmethoden.

Voraussetzungen

Programmierkenntnisse

Kontakt

M.Sc. Tobias Schwientek
Raum 1.365 (ETI II), Telefon 685-69008, E-Mail tobias.schwientek@ikr.uni-stuttgart.de

M.Sc. Timo Madeheim
Raum 1.336 (ETI II), Telefon 685-69012, E-Mail timo.madeheim@ikr.uni-stuttgart.de