Karlsruher Institut für Technologie

Institut für Theoretische Informatik (ITI)
Anwendungsorientierte Formale Verifikation
**Prof. Dr. Bernhard Beckert**

Bachelorarbeit

# „Proof Slicing“ von KeY-Beweisen

**Hintergrund.** Ziel bei der deduktiven Programmverifikation ist es, mithilfe formaler Methoden zu beweisen, dass Software die für sie spezifizierten Eigenschaften erfüllt. Obwohl das im Allgemeinen ein unentscheidbares Problem ist, können in der Praxis oftmals mithilfe von Heuristiken sogar automatisch Beweise gefunden werden.

KeY ist ein am Lehrstuhl entwickeltes Tool zur deduktiven Verifikation von Java-Programmen. Es unterstützt sowohl automatische als auch interaktive Beweissuche. Ist ein Beweis gefunden, kann er abgespeichert werden, um ein prüfbares Zertifikat für die Korrektheit des Programms zu erhalten.

Ein solcher Beweis enthält allerdings oftmals eine große Anzahl an Regelanwendungen, die zwar von der Heuristik oder dem Benutzer ausgewählt und angewandt wurden, aber für den eigentlichen Beweis nicht nötig wären. Dadurch nehmen die Beweise beim Abspeichern dann unnötig viel Platz ein, sind für den Benutzer schwieriger zu verstehen und ein automatischer Proof-Checker braucht unnötig lange, um den Beweis zu prüfen.

Slicing ist eine Technik, um bei Programmcode diejenigen Programmzeilen bzw. Statements zu finden, die Einfluss auf eine bestimmte Eigenschaft haben. In dieser Arbeit möchten wir Slicing auf KeY-Beweise anwenden.

**Aufgabe.**

Ziel der Arbeit ist es, eine neue Slicing-Technik für KeY-Beweise zu entwickeln und zu implementieren, mit der existierende Beweise verkleinert werden können. Diese Technik soll anschließend an existierenden Beweisen evaluiert werden (z.B. am Beweis einer Quicksort-Implementierung).

**Ihr Profil.** Sie sollten solide Java-Kenntnisse besitzen sowie das Modul *Formale Systeme* erfolgreich abgeschlossen haben.

**Kontakt**
Wolfram Pfeifer wolfram.pfeifer@kit.edu 50.34 R228