Semesterarbeit

Transformation eines Korrektheitsbeweises für ein Java-Programm auf die Bytecode-Ebene

Grundidee von *Proof-Carrying Code (PCC)* ist es, Beweise für Programmeigenschaften in Komponenten der Zielsprache einzubetten. Diese können von Nutzern der Komponenten einfach extrahiert und vollautomatisch geprüft werden. Dadurch ergibt sich für die Nutzer eine effiziente Möglichkeit sich zu vergewissern, dass die Komponente diese Programmeigenschaften tatsächlich besitzt.

Existierende Arbeiten zu PCC behandeln relativ einfache Eigenschaften wie Typkorrektheit. Für solche Eigenschaften können sogenannte *Certifying Compiler* die Korrektheitsbeweise automatisch erstellen und einbetten. Komplexere Eigenschaften, wie z.B. das Erfüllen einer formalen Schnittstellenspezifikation, sind jedoch im Allgemeinen nicht entscheidbar. Für solche Eigenschaften ist es daher notwendig, dass der Beweis manuell erstellt wird, was üblicherweise auf der Ebene der Quellsprache geschieht. Ein Certifying Compiler könnte einen solchen Beweis dann automatisch auf die Zielsprache übertragen. Diese Semesterarbeit ist eine Vorstudie über solche Beweistransformationen.

Aufgabenstellung dieser Semesterarbeit Im Rahmen dieser Semesterarbeit soll ein sehr kleines Java Programm spezifiziert und verifiziert werden. Sowohl die Spezifikation als auch der Beweis sind dann manuell auf das entsprechende Bytecode Programm zu übertragen. Dabei sollen Muster identifiziert und beschrieben werden, nach denen ein Certifying Compiler die Transformation automatisch durchführen könnte.

Aufgabensteller
Prof. Peter Müller
`peter.mueller@inf.ethz.ch`