

Konzeption und Entwicklung eines Systemabhängigkeitsgraphen für Java- Applikationen auf Basis des Quelltextes

Hinweis

Je nach Art der Abschlussarbeit (Bachelor oder Master) wird der Umfang, Ausrichtung und die Details der Aufgabenstellung unterschiedlich ausgestaltet sein.

Kontext

Ein Systemabhängigkeitsgraph (SDG) bildet die statische Struktur eines Programms in Form von Knoten und Kanten ab. Insbesondere werden Kontroll- und Datenabhängigkeiten zwischen Anweisungen durch Kanten dargestellt und für komplexe Programmanalysen verwendet.

Zum Beispiel baut ein Compiler intern ein Systemabhängigkeitsgraphen auf, um Code-Restrukturierungen (Refactoring) und Constant Propagation durchzuführen.

In dieser Arbeit soll ein Konzept und eine Implementierung für hierarchische, objektorientierte SDGs entwickelt werden. Daraus resultierende Ergebnisse sollen später als Grundlage für einen semi-automatischen Ansatz zur Einführung von Parallelität in Legacy-Systeme dienen. Dieser Ansatz beschreibt einerseits sogenannte Candidate Patterns, um Stellen im SDG zu identifizieren, die ein hohes Parallelisierungspotential aufweisen. Andererseits dienen sogenannte Parallelization Patterns als Zielvorlage für eine Transformation ausgehend von Candidate Patterns.

Aufgabenstellung

- Konzeption einer Transformation von Quelltext zu einem Systemabhängigkeitsgraph (inkl. rudimentärer Kontroll- und Datenflussanalyse)
- Umsetzung dieser Transformation für Java-Quelltext in Java

Links und Literatur

- GraphML (<http://graphml.graphdrawing.org/>)
- „The Java System Dependence Graph“, Walkinshaw et. al., (<http://dx.doi.org/10.1109/SCAM.2003.1238031>)
- „Speeding up context-, object- and field-sensitive SDG generation“, Jürgen Graf, (<http://dx.doi.org/10.1109/SCAM.2010.9>)

Kontakt

Bei Interesse oder Fragen zu dieser Abschlussarbeit wenden Sie sich bitte an:

Christian Wulf
Christian-Albrechts-Platz 4
24098 Kiel
Raum 1213
Tel.: 0431-880 2776
E-Mail: chw (at) informatik.uni-kiel.de