Informatik 11
Embedded Software

RWTH AACHEN UNIVERSITY

Bachelor-/Masterarbeit

Policy iteration zur statischen Analyse von SPS-Programmen

Problemstellung

Funktionale Sicherheit ist ein wichtiges Ziel moderner Softwareentwicklung. Neben spezialisierten Entwicklungstechniken und Projektmanagementstrategien (V-Modell, Scrum, etc.) sind daher das automatisierte Testen und die automatisierte Verifikation zentrale Hilfsmittel zur Sicherung der erforderlichen Qualitätsstandards. Zur Programmanalyse von Speicherprogrammierbaren Steuerungen (SPSen) wird daher am Lehrstuhl i11 das Codeanalysewerkzeug ARCADE (Aachen Rigourous Code Analysis & Debugging Environment) entwickelt, das verschiedene Analysetechniken bereitstellt.

Ein wichtiger Analysebereich ist die statische Analyse, bei der Wertebereiche von Variablen und andere Informationen an Programmstellen bestimmt werden, ohne das Programm explizit auszuführen. Diese Analysen suchen im Prinzip einen möglichst kleinen Fixpunkt der Programmtransitionsfunktion (d.h. der Transition, die einen Schritt der Programmausführung kodiert). Dieser Fixpunkt wird üblicherweise über eine simple Fixpunktiteration berechnet, mit optionalen Techniken wie *Widening* um Konvergenz (auf Kosten der Genauigkeit) zu garantieren.

Da die Fixpunktiteration oft viele Schritte benötigt, sind solche Techniken von großer Bedeutung, um brauchbare Ergebnisse zu bekommen. Aus diesem Grund wurde eine Technik namens *policy iteration*[1] entwickelt, die solche Fixpunkte schneller zu finden vermag. Die Grundidee hier ist, die Funktion ***f***, deren Fixpunkt es zu finden gilt, als Infimum einer Menge ***G*** einfacherer Funktionen darzustellen. Damit ergibt sich ein Fixpunkt von ***f*** als das Infimum der Fixpunkte von ***G***.

Aufgabenstellung

Ziel dieser Abschlussarbeit ist, policy iteration als Algorithmus in ARCADE zu implementieren. Dies muss in jedem Fall für die im Quellpaper vorgestellte Intervalldomäne geschehen, optional auch für andere in ARCADE verwendete Domänen (Bitvektoren, k-Sets, Oktagone, etc.). Danach soll die Performance der so optimierten Analysen mit der Performance der bisherigen Analysen verglichen werden. Hierbei ist es wichtig, Heuristiken für die Auswahl der richtigen Policies zu finden. Auch soll evaluiert werden, welche SPS-spezifischen Optimierungen helfen können, die Laufzeit noch weiter zu verbessern.

Vorkenntnisse

Vorkenntnisse im Bereich der statischen Analyse sind hilfreich, aber nicht vorausgesetzt. Da ARCADE in Java geschrieben ist, sind fundierte Javakenntnisse von Vorteil.

Ansprechpartner

Marcus Völker, M. Sc. RWTH
voelker@embedded.rwth-aachen.de

[1] Costan, A.; Gaubert, S.; Goubault, E.; Martel, M. & Putot, S. (2005), A Policy Iteration Algorithm for Computing Fixed Points in Static Analysis of Programs., *in* Kousha Etessami & Sriram K. Rajamani, ed., 'CAV' , Springer, pp. 462-475 .