A Loop-summary-based Heuristic for Policy Improvement in Value Set Analysis using Policy Iteration

(Masterarbeit)

ROBIN MROSS

Motivation

Viele Laufzeitfehler von Programmen werden durch Variablen verursacht, die unerwartete Werte annehmen. Daher ist es hilfreich, bereits vor dem Ausführen der Applikation ermitteln zu können, wie sich Variablen im Programmverlauf verhalten, das heißt, welche Werte zu welchen Zeitpunkten möglich sind.

Im Rahmen der Datenflussanalyse in der statischen Programmanalyse beschäftigt sich Wertebereichsanalyse mit der Berechnung von möglichen Werten solcher Variablen, zum Beispiel durch Angabe von (Integer-)Intervallen.

Stand der Technik

In der Wertebereichsanalyse können mithilfe der *Kleene Iteration* kleinste Fixpunkte für mögliche (Interval-)Werte von Variablen im Programm berechnet werden. Damit Terminierung garantiert und Präzision verbessert werden kann, wird der Algorithmus in Kombination mit sogenannten *Widening* und *Narrowing* Operatoren verwendet.

Eine Alternative zu diesem Verfahren bietet *Policy Iteration*. Hier wird ein Fixpunkt k einer Selbstabbildung f auf einem Verband berechnet, indem Fixpunkte von in gewissem Sinne einfacheren Abbildungen g aus einer Menge G ermittelt werden. Dabei entspricht k einem Fixpunkt einer der Abbildungen g in G.

Die Arbeit, auf der hier aufgebaut werden soll, betrachtet Verbesserungen von *Policy Iteration* durch Berücksichtigung der Veränderungsrichtung von Variablen in Schleifen, d.h. ob eine Variable konstant bleibt, größer oder kleiner (oder beides) wird.

Zielsetzung

Ein wesentlicher Schritt des *Policy Iteration* Algorithmus ist *Policy Improvement*: wenn der Fixpunkt der ausgewählten Abbildung g aus G nicht dem gesuchten Fixpunkt für f entspricht, muss eine andere Funktion g' aus G gewählt und die Berechnung wiederholt werden. Dabei kann die Wahl von g' entscheidend für die Laufzeit des Algorithmus sein. Ziel der Arbeit ist es, diesen Schritt zu verbessern, indem die oben beschriebenen Information über Schleifen in die Heuristik eingebracht werden.

Geplante Vorgehensweise

Zunächst soll untersucht werden, auf welchen Programmen die Standardheuristik keine guten Ergebnisse erzielt. Dann wird überprüft, welche *Policies* zu besseren Resultaten führen können, sodass schließlich aufbauend darauf eine Heuristik entwickelt werden kann.

Betreuer: Marcus Völker
06.11.2019

Informatik 11
Embedded Software
RWTH AACHEN UNIVERSITY