Masterarbeit

Formaler Beweis der semantischen Kohärenz eines LLVM-Frontends

Problemstellung

Zur Sicherstellung des korrekten Verhaltens von Software ist der Einsatz formaler Methoden unerlässlich. Gerade im Kontext sicherheitskritischer Systeme, wie sie z.B. in industriellen Anlagen verbaut werden, können Programmanalysen genutzt werden, um Fehler in der implementierten Software zu finden, bevor sie im Livebetrieb auftreten.

Um formale Methoden für möglichst viele Programmiersprachen zur Verfügung zu stellen, arbeiten einige Analysewerkzeuge auf LLVM-IR, der intermediären Darstellung des LLVM Projektes, wodurch alle Sprachen analysiert werden können, für die ein LLVM-Frontend existiert, was bei vielen gängigen Sprachen, insbesondere C/C++, der Fall ist.

Im industriellen Alltag kommen jedoch oft speicherprogrammierbare Steuerungen (SPSen) zum Einsatz, für deren Sprachen kein LLVM-Frontend existiert. Um auf die Analysewerkzeuge zugreifen zu können, wurde daher in einer vergangenen Arbeit am i11 bereits ein Compiler von der SPS-Sprache Structured Text (ST) nach LLVM-IR entwickelt.



Aufgabenstellung

Aufgabe dieser Arbeit soll es sein, die Semantiken von ST und LLVM-IR anhand einer Literaturrecherche zu bestimmen, in einen geeigneten Formalismus zu überführen, und auf diesem Formalismus einen Beweisansatz zu führen, dass der Compiler ST-Code in semantisch äquivalenten LLVM-IR-Code überführt.

Hier ist abzugrenzen, dass a) keine Instrumentierung der konkreten Compiler-Implementierung stattfinden soll, d.h. es soll die Idee des Compilers bewiesen werden, aber nicht der Compiler selbst verifiziert werden und b) nicht alle Sprachfeatures von ST explizit „durchbewiesen“ werden müssen, viel mehr geht es darum, die Formalismen und Beweisschemata zu entwickeln, mit denen dann der vollständige Beweis geführt werden kann.

Das Beweisschema soll im Beweisassistenten Coq formalisiert werden, um den Beweisprozess sicherer und einfacher zu gestalten

Vorkenntnisse

Vorkenntnisse zur Semantik von Programmiersprachen und zu Coq sind sehr hilfreich, können aber auch während der Einarbeitungsphase erworben werden. Ein frei zugängliches Tutorial zu Beweisen über Programmiersprachen in Coq ist verfügbar.

Ansprechpartner

Dr. rer. nat. Marcus Völker
voelker@embedded.rwth-aachen.de