





Effiziente Datenrepräsentation in KIV

(Abschlussarbeiten, Forschungs- und Projektmodule, Studentische Mitarbeiter)

Das KIV-System unterstützt eine Library mit abstrakten Datentypen (u.a. Listen, Maps, Sets, Arrays, Heap-Strukturen). KIV Programme verwenden Datentypen aus dieser Library und können mit KIV verifiziert werden. Mit Hilfe eines Codegenerators kann aus KIV Programmen und den Datentypen Code in einer der drei Programmiersprachen C, Rust oder Scala generiert werden.

Datentypen können als Interfaces angesehen werden, die durch verschiedene Klassen implementiert werden können. Listen können z.Bsp. durch doubly-linked lists, singly-linked lists oder ArrayLists repräsentiert werden. Graphen können als Adjazenzmatrix oder als Array von Knoten mit einer Liste von Nachfolgern repräsentiert werden.

Derzeit ist im KIV-Codegenerator eine Repräsentation festgelegt. Eine fixe Zahl von primitiven Operationen (bei Listen sind das u.a. leere Liste erstellen, Elemente vorne und hinten anhängen, zwei Listen zusammenhängen) sind von Hand in C, Rust und Scala implementiert.

Der Nachteil dieses Ansatzes ist, dass der von Hand geschriebene Code nicht verifiziert werden kann, und der Codegenerator für nicht primitive Operationen je nach Alternative manchmal eine ineffiziente Implementierung generiert.

In der Arbeit soll ein flexibles Konzept, das es ermöglicht zu einem Interface (hier: List) mehrere implementierende Klassen (doubly-linked list etc.) zu haben, in Scala programmiert und in KIV integriert werden. Je nach Umfang der Arbeit kann das Konzept dann evtl. auch schon in den Codegenerator integriert werden. Die Verifikation von Implementierungen ist nicht Gegenstand der Arbeit.

Hinweise:

- Grundkenntnisse der objektorientierten Programmierung sind erforderlich.
- Interesse an Scala, einer modernen Programmiersprache,
  die gegenüber Java viele neue Konzepte hat.

Beginn:	ab sofort möglich
Ansprechpartner:	Dr. Gerhard Schellhorn, Felix Pribyl
Raum:	3014, 3016
Email:	schellhorn@isse.de f.pribyl@uni-a.de