Semesterarbeit

Dynamische Typprüfung für das Universe Type System

Aliasing tritt in der objektorientierten Programmierung auf, wenn ein Objekt von mehreren anderen Objekten referenziert wird. So wird zum Beispiel eine Zelle einer doppelt verketteten Liste im Allgemeinen von ihrem Vorgänger, ihrem Nachfolger und möglicherweise vom Listenkopf sowie von Iteratoren referenziert. Während Aliasing grundsätzlich zur Flexibilität und Effizienz objektorientierter Programme beiträgt, ist es in anderen Fällen unerwünscht und Ursache von Fehlern: Die Existenz von Aliasen kann dazu führen, dass Objektstrukturen in unvorhergesehener Weise manipuliert werden, was zu Inkonsistenzen und zur Verletzung von Invarianten führen kann. Zum Beispiel geht man bei der Implementierung doppelt verketteter Listen üblicherweise davon aus, dass die Zell-Struktur nur durch den Listenkopf verändert wird. Unerwünschte Referenzen auf Zellen könnten jedoch benutzt werden, um die Struktur zu zerstören.

In der Software Component Technology Group wurde das Universe Type System zur Alias-Kontrolle entwickelt und in einem Kompiler implementiert. Mit diesem Typsystem kann der Heapspeicher in sogenannten Universes strukturiert werden, wodurch man starke, statisch geprüfte Kontrolle über Referenzen zwischen Universes bekommt. Neben Vorteilen für die formale Beweisführung von Programmen ermöglicht das Typsystem die Schnittstelle von Komponenten genau zu definieren.

Aufgabenstellung dieser Semesterarbeit ist es die dynamische Typprüfung für das Universe Type System zu entwerfen und zu implementieren. Die notwendigen Prüfungen entsprechen den Java Laufzeitprüfungen, zB Casts, `instanceof` und schreibender Arrayzugriff. Das Problem kann in folgende Teile zerlegt werden:

1. Design einer Methode, mit der die Ownership-Beziehung zur Laufzeit gespeichert werden kann, zB durch entsprechende Referenzen in `java.lang.Object`.

2. Anpassung des Universe-Compilers, so dass der erzeugte Bytecode die dynamischen Prüfungen vornimmt. Dies soll auch durch entsprechende Source-Transformationen beschrieben werden.