

# Übung 1: Domino für Übersetzerbauer

(Besprechung am Montag, den 15. Mai 2003)

## Aufgabe 1: Bootstrapping eines Java-Übersetzers

Auf einer Maschine mit Maschinsprache  $M$  soll ein Übersetzer von *Java* in *Java Byte Code* (*JBC*), den „Assembler“ der *Java Virtual Maschine* (*JVM*), installiert werden, der „eingeboren“ (*native*) ist, d.h. in der Maschinsprache  $M$  vorliegt.

Gegeben ist ein portabler Übersetzer von *Java* in *JBC*, der in einer Teilmenge  $Java^*$  geschrieben ist, in der es zum Beispiel keine Nebenläufigkeit und Verteiltheit gibt. Bei diesem Übersetzer wurde darauf geachtet, dass er zum einen *schnell* übersetzt, und zum anderen *guten Code* produziert (der effizient abläuft).

Als Hilfsmittel wurde außerdem ein Übersetzer von  $Java^*$  nach  $C$  in  $C$  geschrieben, der weder besonders effizient abläuft, noch guten Code erzeugt.

Außerdem gibt es für  $M$  einen eingeborenen Übersetzer von  $C$  nach  $M$ .

Wie kann der eingeborene *Java*-Übersetzer nach *JBC* erzeugt werden? Erläutert die Herstellung mit T-Diagrammen. Achtet darauf, dass der erzeugte Übersetzer effizient ist und guten Code produziert.

## Aufgabe 2: Bootstrapping der *Java Virtual Machine* (*JVM*)

Die *JVM* ist ein Interpreter für *JBC*. Was bräuchte man, um aus einer portablen *JVM*, die in *Java* selbst implementiert ist, eine eingeborene *JVM* zu erzeugen, die auf  $M$  direkt ausgeführt werden kann.

Was meint Ihr: Ist dieser Schritt einfacher oder aufwändiger als für den portablen *Java-Compiler*?