P. Thiemann, S. Wehr Wintersemester 2006/2007

Compilerbaupraktikum

`http://proglang.informatik.uni-freiburg.de/teaching/compilerpraktikum/2006ws/`

Aufgabenblatt 4

Instruktionsauswahl und Registerallokation

27.1.2006

Ziel dieses Aufgabenblatts ist die Transformation von `JIR` nach MIPS–Assembler. Zum Ausführen von MIPS–Assembler werden wir den SPIM Simulator verwenden (`http://www.cs.wisc.edu/~larus/spim.html`). Neben dem Simulator findet sich auf der Homepage auch eine gut lesbare Dokumentation zu MIPS–Assembler (`http://www.cs.wisc.edu/~larus/HP_AppA.pdf`).

Die Transformation nach Assembler besteht aus zwei Schritten:

1. *Instruktionsauswahl.* Da der MIPS–Assembler recht einfach ist, kann die Instruktionsauswahl mehr oder weniger ad-hoc erfolgen. Benutzen sie zum Erzeugen der Instruktionen das auf der Praktikumsseite verfügbare Modul `Instructions.ml`. (Die in der Vorlesung vorgestellte Transformation von A–Normal Form nach MIPS–Assember benützt ebenfalls dieses Modul.) Das Modul erzeugt dabei eine Datentyprepräsentation der Instruktionen, welche auch Informationen über die von den Instruktionen gelesenen und geschriebenen Register enthält.

2. *Registerallokation.* Im vorhergehenden Schritt werden immer noch virtuelle Register verwendet. Die Registerallokation bildet diese virtuellen Register auf Maschinenregister ab. Dazu ist zunächst eine Liveness Analyse zu implementieren. Dann sollen mittels des in der Vorlesung vorgestellten Graph–Coloring Algorithmus' virtuelle Register auf Maschinenregister abgebildet werden. Dabei müssen möglicherweise auch sogenannte Spill–Instruktionen zum Aus- und Einlagern von Registerinhalten in den Speicher eingefügt werden.

Fügen sie dem Treibermodule `Driver` eine neue Phase hinzu, welche die Übersetzung nach MIPS–Assembler aufruft. Vergessen sie nicht, Tests zu schreiben. Nach Abschluss dieses Aufgabenblatts sollten sie in der Lage sein, jedes Mini-Java Programm im SPIM Simulator ausführen zu können.

Deadline: 10.4.2007