Prof. Dr. Peter Thiemann
Matthias Keil

Winter Semester 2016/2017

Compiler Construction

`http://proglang.informatik.uni-freiburg.de/teaching/compilerbau/2016ws/`

Exercise Sheet 6

1 From Kanga to MIPS (10 Points)

This last part of the compiler project focuses on the translation from Kanga to MIPS assembly. Since some of you may not have an old SGI workstation at home or may not want to void the warranty on their MIPS-based DSL router, we will use a virtual MIPS R2000 CPU simulated by *Spim*[1].
Syntax and semantics of MIPS can be found in plenty of books and ressources (e.g. longish "Einführung in die Assemblerprogrammierung mit dem MIPS-Simulator SPIM" by Reinhard Nitzsche[2], or shorter Appendix A of Hennessy&Patterson[3]). The main challenge of this exercise is to implement the stack layout for the method frames. You will have to store and re-construct the frame and stack pointers, and the return address for each procedure call as well as allocate enough stack space for each frame. If you solved the previous sheet, this will be refreshingly simple.
The grammar `spim.sable` covers a subset of the MIPS instruction set. It should prove sufficient to translate Kanga programs.

Project - Part 6

- Implement an AST transformation from Kanga to MIPS.
- Write a description of the frame layout you implemented. (A small picture might be helpful.)
- On the homepage, you will find a project template to get you started.

Runtime library

Kanga's PRINT and ERROR statements and the HALLOC expression can be simulated using system calls in Spim. The MIPS pretty-printer `mips.pretty.PrettyVisitor` will append wrapper procedures to your code during pretty-printing. They are:

- `_print`: prints the int in `a0`. Modifies `a0,v0`.

[1] `http://spimsimulator.sf.net/`

[2] `http://ti.itec.uka.de/TI-2/Spim/Spim_Tutorial.pdf`

[3] `http://spimsimulator.sourceforge.net/HP_AppA.pdf`

- `_halloc`: allocates the number of bytes in `a0` and leaves a pointer to the new memory in `v0`. Modifies `a0`.
- `_error`: prints “ERROR” and aborts.

Technicalities

- The library package contains a statically-linked Spim 9.0.5 binary for Linux, which the KangaToMips tests use to compare the output of Kanga programs in the interpreter with the output of your MIPS programs in Spim. Unfortunately, there is no headless Spim for Windows, so this test suite works on Linux only.
- If you want to run the test suite from Eclipse, you need to add the following VM arguments to your Run Configuration to tell `KangaToMipsTest` where it can find the Spim binary:

  ```
  -Dspim.executable="/wo/auch/immer/compiler-construction-jars/spim/spim"
  -Dspim.exceptionsFile="/wo/auch/immer/compiler-construction-jars/spim/exceptions.s"
  ```

- You can either work on the sixth project template in isolation, or copy the following parts of the sixth project template into your Spiglet-to-Kanga project: packages `mips.pretty`, `kanga.test`, `kanga.tomips`, grammar `grammar/spim.sable`, test data `testdata/kanga-to-mips`, build file `build.xml`.

Submission

- Deadline: **16.02.2017, 12:00 (noon)**. Late submissions will not be accepted.
- Submit your solution to the subversion repository. Your submission will consist of one folder (exercise6) which includes your solution.
- Rewrite method `kanga.tomips.KangaToMipsTranslator.mips.node.PProgram translateProgram` so that it calls your Kanga transformation for the given Kanga AST.
- Your solution will consist of: 1. a zip file as generated by `ant submission` with the implementation, and 2. a pdf `mips-<your name>.pdf` with a description of the implementation.
- You are strongly encouraged to test your solution with the provided test data. Add test cases as you might think necessary. You need not submit your own test cases.
- The description must be limited to two pages. Submitting more than one page will lead to reduction in points.
- The description may be either German or English. Clear and understandable style is required.