JOHANNES KEPLER
UNIVERSITY LINZ

DI Josef Eisl
Institute for System Software

T +43 732 2468 4346
F +43 732 2468 4345
josef.eisl@jku.at

Secretary:
Birgit Kranzl
Ext 4341
birgit.kranzl@jku.at

Bachelor's Thesis / Project in Software Engineering

Implementation of a Register Allocation Verifier for the Graal Compiler

Student:
SKZ/Matr.Nr.: –- / –-----
Email:

Advisor: DI Josef Eisl

Start date: xx.xx.xxxx

Graal [1] is an effort to create a new just-in-time compiler for Java that is itself written in Java. It is based on a port of the HotSpot client compiler from C++ to Java.

Register allocation is the task of finding a mapping from an unlimited number of variables to a limited number of physical machine registers. Splitting and spilling of live-ranges, rematerialization and register coalescing are optimizations used to further improve the quality of the generated code.

Due to the modular design of Graal it is easy to add new register allocators. Currently, Graal incorporates two different implementations and more are in development. Creating a new allocator, or improving an existing one, is a tedious and error prone task. One reason therefor is the size of the compilation unit which can consist of thousands of instructions. Manual inspection of the result is not feasible. In addition, even if the generated code contains an error the program might not crash at all or only for specific inputs. Finding such problems is difficult and time consuming.

The goal of this thesis is to implement a register allocation verification framework based on static analysis. The verifier should be agnostic to the register allocator so that it can be used with all implementations.

The scope of this thesis is as follows:

- Implement SARAVerify, a static analysis register allocation verifier as proposed by Huang [2].
- Test the implementation on a variety of non-trivial Java programs.
- Verify the implementation by injecting faults in existing register allocators.

The progress of the project should be regularly discussed with the advisor. A time schedule and a milestone plan must be set up within the first 3 weeks. It should be continuously refined and monitored to make sure that the thesis will be completed in time. The final version of the thesis must be submitted not later than xx.xx.xxxx.

[1] http://openjdk.java.net/projects/graal
[2] Yuqiang Huang et al. Detecting Bugs in Register Allocation. 2010.