Formal korrekte Übersetzung von Java Bytecode nach Maschinensprache

Michael Leuschel

Universität Düsseldorf

Ensuring the correctness of the compilation process is an important consideration in the construction of reliable software. If the compiler generates code that is not faithful to the original program code of a system, then all efforts spent on proving the correctness of the system could be futile. Proving that target code is correct w.r.t. the program source is especially important for high assurance systems

In earlier work, AWE together with Susan Stepney and the company Logica have developed the DeCCo compiler which translates a Pascal-like high-level language (called PASP) into machine code for the ASP processor. This undertaking was a Herculean task, and is arguably a big step towards one of the Grand Challanges of computer science, the Verifying Compiler.

We have investigate using a DeCCo style approach for Java Bytecode rather than PASP, to provide a reusable, demonstrably correct compiler backend. We have also moved from using Z to B. This allows us to replace the hand proofs by mechanical proofs, and also allows formal code generation, as well as powerful tool support in form of animation and model checking. A small development of a compiler from simplified Bytecode to a simplified RISC architecture, has proven the value of these tools (each finding different bugs), and has also shown the promise of the approach. While a number of research advances will certainly be required to bring such an ambitious project to completion, the fact that we start from intermediate code leads us to believe that the overall goal can be achieved within the lifetime of a research project.