Webb, Brae J.; Utting, Mark; Hayes, Ian J.
A formal semantics of the GraalVM intermediate representation. (English) [Zbl 1497.68104]

Summary: The optimization phase of a compiler is responsible for transforming an intermediate representation (IR) of a program into a more efficient form. Modern optimizers, such as that used in the GraalVM compiler, use an IR consisting of a sophisticated graph data structure that combines data flow and control flow into the one structure. As part of a wider project on the verification of optimization passes of GraalVM, this paper describes a semantics for its IR within Isabelle/HOL. The semantics consists of a big-step operational semantics for data nodes (which are represented in a graph-based static single assignment (SSA) form) and a small-step operational semantics for handling control flow including heap-based reads and writes, exceptions, and method calls. We have proved a suite of canonicalization optimizations and conditional elimination optimizations with respect to the semantics.

For the entire collection see [Zbl 1489.68022].

MSC:
68N20 Theory of compilers and interpreters
68Q55 Semantics in the theory of computing
68V20 Formalization of mathematics in connection with theorem provers

Software:
GitHub; Isabelle/HOL; CompCert; Vellvm; CakeML; LLVM; GraalVM

Full Text: DOI arXiv

References:


This reference list is based on information provided by the publisher or from digital mathematics libraries. Its items are heuristically matched to zbMATH identifiers and may contain data conversion errors. It attempts to reflect the references listed in the original paper as accurately as possible without claiming the completeness or perfect precision of the matching.