Symbolic Analysis Presentation

Authors: Jiří Bobek, Olivér Lelkes

Topic: Chapter 9.8 of the "Purple Dragon Book" [1]

Abstract

This presentation was made for the course Compiler Construction (VYPe) at Faculty of Information Technology, Brno University of Technology. Its purpose is to give an overview of the use of symbolic analysis in optimizing compilers.

In symbolic analysis, the compiler tracks the values of variables in programs symbolically as expressions of reference variables. This draws out the relationships between variables and can be used for optimization.

One important class of reference variables is loop indexes. A loop index is a variable introduced to count the number of iterations of a loop. Representing the values of other variables as affine expressions of loop indexes enables several code transformations, such as finding loop invariants, replacing multiplications with additions, and deciding whether or not a loop can be executed in parallel.

References

[1] Alfred V. Aho, Monica S. Lam, Ravi Sethi, and Jeffrey D. Ullman. *Compilers: Principles, Techniques, and Tools (2nd Edition)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2006.