Simple Code Generation

Daniel Klimaj (xklima22@stud.fit.vutbr.cz)
Filip Pokorný (xpokor62@stud.fit.vutbr.cz)

Abstract

Simple code generation is a fixed rewriting of the nodes in the AST to target code, where the AST is scheduled by following the data flow inside expressions and the flow of control elsewhere. Simple code generation is suitable for two machine models, the pure stack machine and the pure register machine.

There are operands supplied to and retrieved from the top of the stack in the pure stack machines and the operations work with them there. Each machine instruction corresponds exactly to one node in the expression AST. There is an important order of the instructions to be done because the operation may only be issued when operand(s) are on the top of the stack. The depth-first is a suitable order for evaluation, which coincides with data-flow arrows in the AST of an expression. According to the depth-first, code generation algorithm is established.

The pure register machine has a memory to store values in and a set of registers to perform operations on. The inputs and outputs are mentioned explicitly, as numbered registers. The interface conventions are that the output register of an instruction must be used immediately as an input register of the parent instruction in the AST. We use depth-first code generation again where we have to determine the target to store the result of the expression. There is an available set of auxiliary registers to help get it there.