

Semantics of Programming Languages

Exercise Sheet 7

Exercise 7.1 A different instruction set architecture

We consider a different instruction set which evaluates boolean expressions on the stack, similar to arithmetic expressions:

- The boolean value *False* is represented by the number 0, the boolean value *True* is represented by any number not equal to 0.
- For every boolean operation exists a corresponding instruction which, similar to arithmetic instructions, operates on values on top of the stack.
- The new instruction set introduces a conditional jump which pops the top-most element from the stack and jumps over a given amount of instructions, if the popped value corresponds to *False*, and otherwise goes to the next instruction.

Modify the theory *Compiler* by defining a suitable set of instructions, by adapting the execution model and the compiler and by updating the correctness proof.

Homework 7 Compiler optimization

Submission until Wednesday, December 15, 2010, 12:00 (noon).

A common programming idiom is *IF b THEN c*, i.e., the else-branch consists of a single *SKIP* command.

- Look at how the program *IF Less (V 7) (N 5) THEN 4 ::= N 3 ELSE SKIP* is compiled by *ccomp* and identify a possible compiler optimization.
- Implement an optimized compiler (by modifying *ccomp*) which reduces the number of instructions for programs of the form *IF b THEN c*.
- Extend the proof of *comp_correct* to your modified compiler.