

9. Exercise sheet *Compiler Construction 2008*

Due to Wed., 9 July 2008, *before* the exercise course begins.

Exercise 9.1:

Which of the following procedure stacks can be the result of an execution of a program in the example programming language:

- a) 15 : 4 : 9 : -3 : 5 : 6 : 3 : 2 : 1 : 5 : 4 : 12 : 3 : 4 : 4 : 3 : 25 : 0 : 0 : 0 : 0 : 12
- b) 15 : 4 : 9 : -3 : 5 : 10 : 4 : 3 : 2 : 1 : 5 : 4 : 12 : 3 : 4 : 4 : 3 : 25 : 0 : 0 : 0 : 0 : 12
- c) 15 : 4 : 9 : -3 : 5 : 14 : 4 : 3 : 2 : 1 : 5 : 4 : 12 : 3 : 4 : 4 : 3 : 25 : 0 : 0 : 0 : 0 : 12

Explain!

Exercise 9.2:

In addition to `while`-loops we want to have Java-style `for`-loops with implicit declaration of the counter variable in our example programming language:

`for (var I := A ; B ; C1) C2`

- a) Extend the translation function *ct* accordingly.
- b) Generate intermediate code for

`for (var x := 0; x < 10; x := x + 1) P();`

without parameters for the `CALL` instruction generated for `P()`.

Exercise 9.3:

Consider the EPL-instruction

`if ((I=1) and (J=0)) or ((I=3) and (J=1)) then J := J+1 else I:=I+1`

and the symbol table $[I/(var, 1, 1), J/(var, 1, 2)]$. Compute $ct(C, st, 1, 1)$ and $sct(C, st, 1, 1)$ and determine the successor state of $(1, \varepsilon, 5 : 4 : 0 : 3 : 1 : 0 : 0 : 0)$ for both intermediate codes.