Software-Modellierung und Verifikation
Informatik 2
Prof. J.-P. Katoen
RWTH Aachen

Priv.-Doz. T. Noll noll@cs.rwth-aachen.de
C. Jansen christina.jansen@cs.rwth-aachen.de

12. Exercise sheet *Semantics and Verification of Software SoSe2010*

Due to Monday, 19th July 2010, *before* the exercise course begins.

Exercise 12.1: **(0.5 + 2 points)**

Perform an *available expression analysis* for the following nonterminating program:

$z := x + y;$
while true do skip

(a) Provide the control flow graph of the given program.

(b) Determine all solutions of the corresponding equation system resulting from the *available expression analysis*.

Exercise 12.2: **(2.5 points)**

Perform a *live variable analysis* for the following program:

$x := 2;$
$y := 4;$
$x := 1;$
if $y > x$ **then** $z := y$ **else** $z := y * y;$
$x := z;$

Exercise 12.3: **(3+1 points)**

(a) Develop a (non-trivial) *sign analysis* for program variables based on an abstraction that maps all negative numbers to the symbol $-$, zero to the symbol 0 and all positive numbers to $+$. E.g. the set $\{0, 1, 2, \ldots\}$ is abstracted to $\{0, +\}$.

(b) Perform your analysis on the following program:

$x := 0;$
$y := -1;$
$z := x * y;$
$x := x + 1;$
$y := y - x;$
$z := x * y;$

Exercise 12.4: **(3 points)**

In the lecture you learned about dataflow analysis for purposes like identifying available expression and live variables. By intuition the presented approaches seem correct. Nevertheless proving the semantic correctness of the analyses is necessary not only from the theoretic point of view.
In order to provide correctness proofs, a connection between formal semantics and the flow analysis must be drawn.

- Can you think of a suitable connection to proof correctness of live variable analysis, what is the idea behind your approach?
- Which of the learned semantics is most suitable for a correctness proof and why?