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

Priv.-Doz. T. Noll noll@cs.rwth-aachen.de
D. Willems willems@cs.rwth-aachen.de

3. Exercise sheet *Semantics and Verification of Software 2007*

Due to Wed., 2 May 2007, *before* the exercise course begins.

Exercise 3.1:

In our WHILE language, the evaluation of (arithmetic) expressions has no *side effects*—it does not change the state. If we were to model side effects it would be natural to consider an evaluation relation of the form

$$\langle a, \sigma \rangle \to \langle z, \sigma' \rangle$$

where σ' ist the state that results from the evaluation of a in the original state σ. To introduce side effects in WHILE, extend the arithmetic expressions by a construct

$$c \ \textsf{\textbf{resultis}}\ a$$

where $c \in \mathbf{Cmd}$ and $a \in \mathbf{AExp}$. To evaluate such an expression, c is first executed and then a is evaluated in the new state. Formalize this idea by giving it an operational semantics.

Exercise 3.2:

Show that the operational and the denotational semantics of arithmetic expressions coincide, i.e., prove the following result.
For every $a \in \mathbf{AExp}$, $\sigma \in \Sigma$, and $z \in \mathbb{Z}$:

$$\langle a, \sigma \rangle \to z \quad \text{iff} \quad \mathfrak{A}[\![a]\!](\sigma) = z.$$

Exercise 3.3:

Consider the following fragment of the factorial program (see Exercise 7):

$$\textsf{\textbf{while}}\ \neg(x = 1)\ \textsf{\textbf{do}}\ (y := y * x;\ x := x - 1).$$

(a) Determine the corresponding functional $\Phi : (\Sigma \dashrightarrow \Sigma) \to (\Sigma \dashrightarrow \Sigma)$.

(b) Give at least two fixpoints of Φ.

Exercise 3.4:

Develop a proof for Lemma 5.6 of the course, stating that the set of partial state transformations, $\Sigma \dashrightarrow \Sigma$, together with the relation $\sqsubseteq$ given by graph inclusion forms a partial order.