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

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

Due to Monday, 3rd May 2010, *before* the exercise course begins.

Exercise 2.1: **(2 points)**

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' is 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

$$c \textbf{ resultis } a$$

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 operational semantics.

Exercise 2.2: **(1+2 points)**

(a) Write a WHILE program, which transforms a given decimal number x into its binary representation. Store the resulting binary number as an arithmetic or string expression (using the syntax and semantics from Ex. 1.1).

(b) Construct the derivation tree by means of operational semantics for this program starting in a state $\sigma \in \Sigma$ with $\sigma(x) = 3$.

Exercise 2.3: **(2 points)**

As already mentioned in the lecture well-foundedness is a required property for relations to support the application of the induction principle.

Definition 1 (Well-founded relations) *A* well-founded relation *is a binary relation $\prec$ on a set A such that there are no infinite descending chains $\cdots \prec a_i \prec \cdots \prec a_1 \prec a_0$.*

Show that the following proposition holds:

Proposition 1 *Let $\prec$ be a binary relation on a set A. The relation $\prec$ is well-founded iff any nonempty subset Q of A has a minimal element, i.e. an element m such that*

$$m \in Q \land \forall b \prec m : b \notin Q.$$

Exercise 2.4: **(1+1+2 points)**

(a) Extend the WHILE language by a loop construct of the form

$$\textbf{repeat } c \textbf{ until } b$$

and define its execution relation $\to$ without (explicitly) using the **while** statement.

(b) Establish the following semantic equivalence:

$$\textsf{\textbf{repeat}}\ c\ \textsf{\textbf{until}}\ b \quad \sim \quad c;\ \textsf{\textbf{if}}\ b\ \textsf{\textbf{then skip else}}\ (\textsf{\textbf{repeat}}\ c\ \textsf{\textbf{until}}\ b).$$

(c) Establish the following semantic equivalence:

$$\textsf{\textbf{repeat}}\ c\ \textsf{\textbf{until}}\ b \quad \sim \quad c;\ \textsf{\textbf{while}}\ \neg b\ \textsf{\textbf{do}}\ c.$$