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

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

Due to Wed., 18 April 2007, *before* the exercise course begins.

Exercise 1.1:

In this exercise we will discuss *alternative evaluation strategies* for Boolean expressions.

(a) *Sequential evaluation:* Define operational rules for Boolean expressions of the form $b_1 \wedge b_2$ and $b_1 \vee b_2$ which do *not* evaluate b_2 provided that the value of b_1 is **false** (**true**, respectively). (In these cases, the value of b_2 does not contribute to the overall result.)

(b) *Parallel evaluation:* Define operational rules which evaluate a Boolean expression of the form $b_1 \vee b_2$ to **true** if b_1 or b_2 evaluates to **true**, and which do not evaluate b_2 (b_1, respectively) in this case.

Exercise 1.2:

In the lecture we have defined a so–called *bigstep semantics* for expressions, i.e., a relation $\rightarrow \subseteq (AExp \cup BExp) \times \Sigma \times (\mathbb{Z} \cup \mathbb{B})$ which yields the value of an expression within one step: $\langle (3+3) * (9-2), \sigma \rangle \rightarrow 42$. (Thus the intermediate results of the computation are "hidden" in the derivation tree.)
Alternatively it is possible to explicitly represent the intermediate steps by defining a *single–step semantics*: $\langle (3+3) * (9-2), \sigma \rangle \rightarrow \langle 6 * (9-2), \sigma \rangle \rightarrow \langle 6 * 7, \sigma \rangle \rightarrow \langle 42, \sigma \rangle \rightarrow 42$. Give a complete specification of the single–step relation

(a) $\rightarrow_1^a \subseteq (AExp \times \Sigma) \times (AExp \times \Sigma \cup \mathbb{Z})$ for arithmetic expressions and

(b) $\rightarrow_1^b \subseteq (BExp \times \Sigma) \times (BExp \times \Sigma \cup \mathbb{B})$ for Boolean expressions.