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

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

Due to Wed., 23 May 2007, *before* the exercise course begins.

Exercise 6.1:

Let $c \in \mathbf{Cmd}$ be given by

$$z := 0; \textsf{while } y \leq x \textsf{ do } (z := z + 1; x := x - y).$$

(a) Give a partial correctness property for c which formalizes the following observation: if the execution of c is started in a state $\sigma \in \Sigma$ with $\sigma(x) \geq 0$ und $\sigma(y) > 0$, and if it terminates in a state $\sigma' \in \Sigma$, then $\sigma'(z) = \sigma(x)$ **div** $\sigma(y)$ and $\sigma'(x) = \sigma(x)$ **mod** $\sigma(y)$.

(b) Establish the validity of this correctness property using the proof system from the lecture.

Exercise 6.2:

(a) Develop a proof rule for statements of the form **for** $x := a_1$ **to** a_2 **do** c where $x \in \mathbf{Var}$, $a_1, a_2 \in \mathbf{AExp}$, and $c \in \mathbf{Cmd}$ (without assuming the presence of a **while** statement in the programming language).

(b) Using this rule (and the known proof system), establish the validity of the following partial correctness property:

$$\{y \geq 0\} z := 0; \textsf{for } x := 1 \textsf{ to } y \textsf{ do } z := z + x \left\{ z = \frac{y(y+1)}{2} \right\}$$

Exercise 6.3:

(a) Show that the *greatest common divisor* of two positive integers $i, j \in \mathbb{Z}$, denoted by $\gcd(i, j)$, has the following properties:

(i) $i > j \Rightarrow \gcd(i, j) = \gcd(i - j, j)$,

(ii) $\gcd(i, j) = \gcd(j, i)$, and

(iii) $\gcd(i, i) = i$.

(b) Using the Hoare rules, prove that the statement $c \in \mathbf{Cmd}$ given by

$$\textsf{while } \neg(x = y) \textsf{ do if } x \leq y \textsf{ then } y := y - x \textsf{ else } x := x - y,$$

satisfies the following partial correctness property:

$$\{x = i \wedge y = j \wedge i \geq 1 \wedge j \geq 1\}\ c\ \{x = \gcd(x, y) = \gcd(i, j)\}.$$