RHEINISCH-
WESTFÄLISCHE
TECHNISCHE
HOCHSCHULE
AACHEN

LEHR- UND FORSCHUNGSGEBIET INFORMATIK II
RWTH Aachen · D-52056 Aachen · GERMANY
http://www-i2.informatik.rwth-aachen.de/lufgi2.html

Prof. Dr. Jürgen Giesl
Darius Dlugosz

Automatisierte Programmverifikation

Übungsblatt 9

Alle Aufgaben werden korrigiert. Die Abgabe ist bis Dienstag, 30. Januar 2001, in der Übung, Vorlesung oder im Raum 4211 möglich. Die Aufgaben werden in der Übung am 30. Januar 2001 besprochen.

Aufgabe 1: (3 Punkte)

Das Programm P bestehe aus folgenden Datenstrukturen und Algorithmen:

$$
\begin{array}{lcl}
\textbf{structure}\ \ \mathsf{bool} & & \\
\quad \mathsf{true} & : & \mathsf{bool} \\
\quad \mathsf{false} & : & \mathsf{bool} \\
& & \\
\textbf{structure}\ \ \mathsf{number} & & \\
\quad \mathcal{O} & : & \mathsf{number} \\
\quad \mathsf{succ} & : & \mathsf{number} \to \mathsf{number} \\
& & \\
\textbf{structure}\ \ \mathsf{list} & & \\
\quad \mathsf{empty} & : & \mathsf{list} \\
\quad \mathsf{add} & : & \mathsf{number} \times \mathsf{list} \to \mathsf{list} \\
& & \\
\multicolumn{3}{l}{\textbf{function}\ \ \mathsf{if} : \mathsf{bool} \times \mathsf{number} \times \mathsf{number} \to \mathsf{number}} \\
\quad \mathsf{if}(\mathsf{true}, x, y) & \equiv & x \\
\quad \mathsf{if}(\mathsf{false}, x, y) & \equiv & y \\
& & \\
\multicolumn{3}{l}{\textbf{function}\ \ \mathsf{plus} : \mathsf{number} \times \mathsf{number} \to \mathsf{number}} \\
\quad \mathsf{plus}(\mathcal{O}, y) & \equiv & y \\
\quad \mathsf{plus}(\mathsf{succ}(x), y) & \equiv & \mathsf{succ}(\mathsf{plus}(x, y)) \\
& & \\
\multicolumn{3}{l}{\textbf{function}\ \ \mathsf{times} : \mathsf{number} \times \mathsf{number} \to \mathsf{number}} \\
\quad \mathsf{times}(\mathcal{O}, y) & \equiv & \mathcal{O} \\
\quad \mathsf{times}(\mathsf{succ}(x), y) & \equiv & \mathsf{plus}(y, \mathsf{times}(x, y))
\end{array}
$$

function $\mathsf{minus} : \mathsf{number} \times \mathsf{number} \to \mathsf{number}$

$$\begin{array}{lcl} \mathsf{minus}(x, \mathcal{O}) & \equiv & x \\ \mathsf{minus}(\mathcal{O}, \mathsf{succ}(y)) & \equiv & \mathcal{O} \\ \mathsf{minus}(\mathsf{succ}(x), \mathsf{succ}(y)) & \equiv & \mathsf{minus}(x, y) \end{array}$$

function $\mathsf{fac} : \mathsf{number} \to \mathsf{number}$

$$\begin{array}{lcl} \mathsf{fac}(\mathcal{O}) & \equiv & \mathsf{succ}(\mathcal{O}) \\ \mathsf{fac}(\mathsf{succ}(x)) & \equiv & \mathsf{times}(\mathsf{succ}(x), \mathsf{fac}(x)) \end{array}$$

function $\mathsf{ge} : \mathsf{number} \times \mathsf{number} \to \mathsf{bool}$

$\mathsf{ge}(x, \mathcal{O}) \equiv \mathsf{true}$
$\mathsf{ge}(\mathcal{O}, \mathsf{succ}(y)) \equiv \mathsf{false}$
$\mathsf{ge}(\mathsf{succ}(x), \mathsf{succ}(y)) \equiv \mathsf{ge}(x, y)$

function $\mathsf{gcd} : \mathsf{number} \times \mathsf{number} \to \mathsf{number}$

$$\begin{array}{lcl} \mathsf{gcd}(x, \mathcal{O}) & \equiv & x \\ \mathsf{gcd}(\mathcal{O}, \mathsf{succ}(y)) & \equiv & \mathsf{succ}(y) \\ \mathsf{gcd}(\mathsf{succ}(x), \mathsf{succ}(y)) & \equiv & \mathsf{if}(\mathsf{ge}(x, y), \\ & & \quad \mathsf{gcd}(\mathsf{minus}(x, y), \mathsf{succ}(y)), \\ & & \quad \mathsf{gcd}(\mathsf{succ}(x), \mathsf{minus}(y, x))) \end{array}$$

function $\mathsf{mult} : \mathsf{number} \times \mathsf{number} \times \mathsf{number} \to \mathsf{number}$

$$\begin{array}{lcl} \mathsf{mult}(\mathcal{O}, y, z) & \equiv & z \\ \mathsf{mult}(\mathsf{succ}(x), y, z) & \equiv & \mathsf{mult}(x, y, \mathsf{plus}(z, y)) \end{array}$$

Beweisen Sie unter Verwendung von Lemmata die folgenden Aussagen:

(a) $\forall n, x_2, x_3 : \mathsf{number} \;\; \mathsf{plus}(\mathsf{fac}(n), \mathsf{plus}(x_2, x_3)) \equiv \mathsf{plus}(\mathsf{plus}(\mathsf{fac}(n), x_2), x_3) \quad \in Th_P$

(b) $\forall x_1, x_2, x_3 : \mathsf{number} \;\; \mathsf{times}(x_1, \mathsf{plus}(x_2, x_3)) \equiv \mathsf{plus}(\mathsf{times}(x_1, x_2), \mathsf{times}(x_1, x_3)) \quad \in Th_P$

(c) $\forall x_1, x_2 : \mathsf{number} \;\; \mathsf{gcd}(x_1, x_2) \equiv \mathsf{gcd}(x_2, x_1) \quad \in Th_P$

Die verwendeten Lemmata müssen nicht bewiesen werden.

Hinweis: In Teil (c) können folgende Lemmata erfolgreich verwendet werden:

$\forall x, y, z_1, z_2 : \mathsf{number} \;\; (x \equiv y \to z_1 \equiv z_2) \to (\mathsf{if}(\mathsf{ge}(x, y), z_1, z_2) \equiv \mathsf{if}(\mathsf{ge}(y, x), z_2, z_1))$

$\forall x, y, z_1, z_2 : \mathsf{number} \;\; (x \equiv y \to z_1 \equiv z_2) \;\to\; (x \equiv y \to \mathsf{gcd}(z_1, \mathsf{minus}(x, y)) \equiv \mathsf{gcd}(\mathsf{minus}(y, x), z_2))$

Aufgabe 2:

Sei P das Programm aus Aufgabe 1. Beweisen Sie unter Verwendung von Lemmata die folgende Aussage:

$$\forall x, y : \mathsf{number}\ \ \mathsf{mult}(x, y, \mathcal{O}) \equiv \mathsf{times}(x, y)$$

Zeigen Sie auch die Gültigkeit der verwendeten Lemmata.

Aufgabe 3: (1 Punkt)

Sei P das Programm aus Aufgabe 1. Geben Sie geeignete Induktionsformeln und die verwendete Induktionsrelation an zum Beweis der folgenden Aussage:

$$\forall x, y, z : \mathsf{number}\ \ \mathsf{ge}(x, y) \equiv \mathsf{true}\ \wedge\ \mathsf{ge}(y, z) \equiv \mathsf{true}\ \rightarrow\ \mathsf{ge}(x, z) \equiv \mathsf{true}$$

Skizzieren Sie den Beweis.

Bemerkung: Die Induktionsformeln müssen nicht mit denen in der Vorlesung vorgestellten Methoden erzeugt werden.

Aufgabe 4: (2 Punkte)

Beweisen Sie den Satz 6.1.1 (Terminierung von Programmen und Algorithmen) aus der Vorlesung:

Ein Programm terminiert genau dann, wenn alle seine Algorithmen terminieren.

Bemerkung: Ein Algorithmus f des Programms P terminiert gdw. kein Term der Form $f(t^*)$ eine unendliche Auswertung hat, wobei die Argumente t^* Konstruktorgrundterme sind.

Aufgabe 5: (1 Punkt)

Sei P ein Programm und seien $s, t \in \mathcal{T}(\Sigma, \mathcal{V})$ und $s^*, t^* \in \mathcal{T}(\Sigma, \mathcal{V})^*$. Wir betrachten die Relation $\succ$, für die jeweils gilt

(a) $f(s^*) \succ g(t^*)$ gdw. g im Algorithmus von f auftritt

(b) $s \succ t$ gdw. $\mathcal{V}(t) \subsetneqq \mathcal{V}(s)$

Untersuchen Sie die Relation $\succ$ auf Fundiertheit, Transitivität, Monotonie und Stabilität.