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

Jürgen Giesl
Darius Dlugosz

Automatisierte Programmverifikation

Übungsblatt 2

Alle Aufgaben werden korrigiert. Die Abgabe ist bis Dienstag, 28. November 2000, in der Übung, Vorlesung oder im Raum 4211 möglich. Die Aufgaben werden in der Übung am 28. November 2000 besprochen.

Aufgabe 1: (1 Punkt) Seien φ und ψ beliebige Formeln über Σ und $\mathcal{V}$. Beweisen Sie das sogenannte *Deduktionstheorem*:

$\{\varphi\} \models \psi$ gdw. $\varnothing \models \varphi \rightarrow \psi$

Aufgabe 2: (1+1+0 Punkte) Zeigen Sie, dass für jede Interpretation I gilt:

(a) $Th(I) \neq \varnothing$,

(b) für alle $\varphi \in Th(I)$ gilt: $\neg\varphi \notin Th(I)$, d.h. $Th(I)$ ist *konsistent*,

(c) für alle $\varphi \in \mathcal{F}_g(\Sigma, \mathcal{V})$ gilt: $\varphi \in Th(I) \vee \neg\varphi \in Th(I)$, d.h. $Th(I)$ ist *vollständig*.

Aufgabe 3: (2 Punkte) Beweisen Sie unter Verwendung der strukturellen Induktion das *Substitutionslemma*:

Sei $I = (\mathcal{A}, \alpha, \beta)$ eine Interpretation für eine Signatur Σ und $\sigma = \{x_1/t_1, \ldots, x_n/t_n\}$ eine Substitution. Dann gilt:

$I(\sigma(t)) = I[\![x_1/I(t_1), \ldots, x_n/I(t_n)]\!](t)$ *für alle* $t \in \mathcal{T}(\Sigma, \mathcal{V})$

Aufgabe 4: (2 Punkte) Sei P ein Programm mit zugehöriger Signatur Σ und seien $t, r_1, r_2 \in \mathcal{T}(\Sigma)$ und $\pi \in Occ(t)$, wobei $t|_\pi$ und r_1, r_2 die gleiche Sorte haben. Zeigen Sie: Aus $r_1 \Rightarrow_P r_2$ folgt $t[r_1]_\pi \Rightarrow_P t[r_2]_\pi$.

Aufgabe 5: (0.5 + 0.5 + 0 Punkte) Das Programm P bestehe aus den folgenden Algorithmen:

$$
\begin{array}{l}
\textbf{function}\ \ \mathsf{plus} : \mathsf{number} \times \mathsf{number} \to \mathsf{number} \\
\quad \mathsf{plus}(\mathcal{O}, y) \quad\quad \equiv \quad y \\
\quad \mathsf{plus}(\mathsf{succ}(x), y) \quad \equiv \quad \mathsf{succ}(\mathsf{plus}(x, y)) \\
\\
\textbf{function}\ \ \mathsf{times} : \mathsf{number} \times \mathsf{number} \to \mathsf{number} \\
\quad \mathsf{times}(\mathcal{O}, y) \quad\quad \equiv \quad \mathcal{O} \\
\quad \mathsf{times}(\mathsf{succ}(x), y) \quad \equiv \quad \mathsf{plus}(y, \mathsf{times}(x, y)) \\
\\
\textbf{function}\ \ \mathsf{null} : \mathsf{number} \times \mathsf{number} \to \mathsf{number} \\
\quad \mathsf{null}(\mathcal{O}, x) \quad\quad \equiv \quad \mathcal{O} \\
\quad \mathsf{null}(\mathsf{succ}(x), y) \quad \equiv \quad \mathsf{null}(x, \mathsf{null}(\mathsf{succ}(x), y)) \\
\\
\textbf{function}\ \ \mathsf{infty} : \mathsf{number} \to \mathsf{number} \\
\quad \mathsf{infty}(x) \quad\quad\quad\quad \equiv \quad \mathsf{infty}(\mathsf{succ}(x))
\end{array}
$$

Berechnen Sie durch Angabe der Auswertungsfolgen die Ergebnisse von

(a) $eval_P(\mathsf{times}(\mathcal{O}, \mathsf{null}(\mathsf{succ}(\mathcal{O}), \mathcal{O})))$,

(b) $eval_P(\mathsf{times}(\mathsf{succ}(\mathsf{plus}(\mathsf{null}(\mathcal{O}, \mathsf{succ}(\mathcal{O})), \mathcal{O})), \mathsf{succ}(\mathcal{O})))$,

(c) $eval_P(\mathsf{null}(\mathsf{times}(\mathsf{plus}(\mathsf{succ}(\mathcal{O}), \mathcal{O}), \mathcal{O}), \mathsf{succ}(\mathsf{infty}(\mathcal{O})))))$.