Semantik und Programmverifikation

Prof. Dr. Christoph Walther / Simon Siegler
Technische Universität Darmstadt — Wintersemester 2008/09

Übungsblatt 8

Aufgabe 8.1 (Normal-rekursive und tail-rekursive Funktionen)

Bestimmen Sie alle normal-rekursiven und alle tail-rekursiven Funktionsprozeduren des funktionalen Programms $P_{hsort} \bigoplus S_{sort}$, wobei P_{hsort} und S_{sort} wie in Abschnitt 3.3 definiert ist.

Aufgabe 8.2 (Terminierungsanalyse)

Sei die Funktionsprozedur F_{funny} definiert durch

function $funny(x : nat) : nat \Leftarrow$
  if $x = 0$
    then M
    else if $funny(x - 1) = N$
      then K
      else $funny(x + 1)$
    fi
  fi.

Beweisen oder widerlegen Sie für alle $M, N, K \in \{0, 1\}$ die Aussage: F_{funny} terminiert *stark*.

Hinweis: Verwenden Sie die Ergebnisse von Aufgabe 7.3, wenn möglich.

Aufgabe 8.3 (Semantik der Spezifikationssprache)

(a) Beweisen Sie, dass $[t \equiv M_P(t)] \in Th_P$ für alle $t \in \mathcal{T}(\Sigma(P))$ gilt.

(b) Beweisen Sie, dass eq_s und $\equiv$ für jedes terminierende Programm übereinstimmen, d. h., dass der folgende Satz gilt: „Sei P ein terminierendes funktionales Programm, $x^* \in \mathcal{V}_w$ und $t_1, t_2 \in \mathcal{T}(\Sigma(P), \mathcal{V}(x^*))_s$. Dann gilt:

$$[\forall x^* : w.\ t_1 \equiv t_2] \in Th_P \Leftrightarrow [\forall x^* : w.\ eq_s(t_1, t_2) \equiv true] \in Th_P\text{“}$$

(c) Beweisen Sie, dass eq_s und $\equiv$ für terminierende Programme nicht in $(AX_P)^{\models}$ übereinstimmen, d.h., dass der folgende Satz gilt: „Sei P ein terminierendes funktionales Programm. Dann existieren $x^* \in \mathcal{V}_w$ und $t_1, t_2 \in \mathcal{T}(\Sigma(P), \mathcal{V}(x^*))_s$ mit

$$[\forall x^* : w.\ t_1 \equiv t_2 \leftrightarrow \forall x^* : w.\ eq_s(t_1, t_2) \equiv true] \notin (AX_P)^{\models}\text{.“}$$

(vgl. Übung 3.4.1).

Aufgabe 8.4 (Korrektheitsbeweis durch Induktion)

Sei $P_{\text{double}} = \langle F_{\text{ge}}, F_{\text{half}}, F_{\text{double}} \rangle$ das funktionale Programm mit

$$
\begin{aligned}
F_{\text{ge}} = \ & \textbf{function } ge(x, y : nat) : bool \Leftarrow \\
& \quad \textbf{if } y = 0 \textbf{ then } true \textbf{ else } (\textbf{if } x = 0 \textbf{ then } false \textbf{ else } ge(x-1, y-1) \textbf{ fi}) \textbf{ fi}, \\
F_{\text{half}} = \ & \textbf{function } half(x : nat) : nat \Leftarrow \\
& \quad \textbf{if } x = 0 \textbf{ then } 0 \textbf{ else } (\textbf{if } x = 1 \textbf{ then } 0 \textbf{ else } 1 + half(x-2) \textbf{ fi}) \textbf{ fi}, \\
F_{\text{double}} = \ & \textbf{function } double(x : nat) : nat \Leftarrow \textbf{if } x = 0 \textbf{ then } 0 \textbf{ else } 2 + double(x-1) \textbf{ fi}.
\end{aligned}
$$

Definieren Sie eine unendliche entscheidbare Menge $\mathsf{IND}_{P_{\text{double}}}$ von Induktionsaxiomen für P_{double}. Zeigen Sie $\mathsf{IND}_{P_{\text{double}}} \subseteq Th_{P_{\text{double}}}$ und beweisen Sie mit Hilfe von $\mathsf{IND}_{P_{\text{double}}}$

$$[\forall x : nat.\ ge(x, double(half(x))) \equiv true] \in Th_{P_{double}}.$$

(vgl. Übung 3.7.2)

Aufgabe 8.5 (Nicht-Äquivalenzmodell)

Definieren Sie ein Nicht-Äquivalenzmodell N von AX_{BM} mit $N \not\models \forall x : nat.\ \neg x \equiv succ(x)$ (vgl. Übung 3.6.1.(i)).

Aufgabe 8.6 (Programme, Axiome, Modelle)

Sei das funktionale Programm $P = \langle F_{plus} \rangle$ gegeben durch

$$
\begin{aligned}
&\textbf{function } plus(x, y : nat) : nat \ \Leftarrow \\
&\ \textbf{if } x = 0 \textbf{ then } y \textbf{ else } 1 + plus(x-1, y) \textbf{ fi}.
\end{aligned}
$$

Sei weiter die Σ-Algebra $A = (\mathcal{A}, \alpha)$ gegeben durch

$$
\begin{aligned}
\mathcal{A}_{bool} &:= \{\top, \bot\} \\
\mathcal{A}_{nat} &:= \{2^i \mid i \in \mathbb{N}\} \\
\alpha_{true} &:= \top \\
\alpha_{false} &:= \bot \\
\alpha_0 &:= 1 \\
\alpha_{succ}(n) &:= n * 2 \\
\alpha_{pred}(n) &:= \begin{cases} n/2, \text{ falls } n \neq 1 \\ 1, \text{ falls } n = 1 \end{cases} \\
\alpha_{eq}(n, m) &:= \begin{cases} \top, \text{ falls } n = m \\ \bot, \text{ falls } n \neq m \end{cases} \\
\alpha_{if}(b, n, m) &:= \begin{cases} n, \text{ falls } b = \top \\ m, \text{ falls } b = \bot \end{cases} \\
\alpha_{plus}(n, m) &:= n * m
\end{aligned}
$$

Beweisen oder widerlegen Sie:

(a) A ist ein Äquivalenzmodell von $AX_{P_{plus}}$.

(b) A ist ein Standardmodell von $AX_{P_{plus}}$.