Semantik und Programmverifikation

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

Hausaufgabe mit Lösungsvorschlag 6

Hausaufgabe 6.1 ($COND(t,\pi)$) **(4 Punkte)**

Zeigen Sie, dass für alle $t \in \mathcal{T}(\Sigma(P))$ gilt (vgl. Übung 3.1.4):

(a) $COND(t,\pi\rho) \approx COND(t,\pi) \wedge COND(t|_\pi,\rho)$ für alle $\pi \in Occ(t)$ und alle $\rho \in Occ(t|_\pi)$, vgl. Definition 1.2.2.

Lösungsvorschlag:

Beweis mit Induktion über π:

Basisfall $\pi = \epsilon$**:** $COND(t,\pi\rho) = COND(t,\rho) \approx TRUE \wedge COND(t|_\epsilon,\rho)$
$= COND(t,\epsilon) \wedge COND(t|_\epsilon,\rho) = COND(t,\pi) \wedge COND(t|_\pi,\rho)$

Schrittfall $\pi = i\pi'$**:** $\Rightarrow t = f(t_1,\ldots,t_n)$

Fall $f \neq if$**:** $COND(t,i\pi'\rho) = COND(t_i,\pi'\rho) \overbrace{\approx}^{IH} COND(t_i,\pi') \wedge COND(t_i|_{\pi'},\rho)$
$= COND(t,i\pi') \wedge COND(t|_{i\pi'},\rho)$

Fall $f = if$ **und** $i = 1$**:** analog zum vorherigen Fall

Fall $f = if$ **und** $i = 2$**:** $COND(t,i\pi'\rho) = t_1 \equiv true \wedge COND(t_2,\pi'\rho)$
$\underbrace{\approx}_{IH} t_1 \equiv true \wedge COND(t_2,\pi') \wedge COND(t_2|_{\pi'},\rho) = COND(t,i\pi') \wedge COND(t|_{i\pi'},\rho)$

Fall $f = if$ **und** $i = 3$**:** analog zum vorherigen Fall

(b) $\mathsf{cbv\text{-}eval}_P(t) = \propto$ gdw. $COND(t,\pi) \Rightarrow_P^* TRUE$ und $\mathsf{cbv\text{-}eval}_P(t|_\pi) = \propto$ für ein $\pi \in Occ$.

Lösungsvorschlag:

„$\Longrightarrow$“**:** es gelte $\mathsf{cbv\text{-}eval}_P(t) = \propto \Rightarrow$ für ein π, nämlich für $\pi = \epsilon$ gilt:
$COND(t,\pi) \Rightarrow_P^* true$ und $\mathsf{cbv\text{-}eval}_P(t|_\pi) = \propto$

„$\Longleftarrow$“**:** Beweis durch strukturelle Induktion über t:

Basisfall $t \in \Sigma^c$**:** $\Rightarrow \forall \pi \in Occ(t).\mathsf{cbv\text{-}eval}_P(t|_\pi) \neq \propto$, die Behauptung gilt also trivialerweise

Schrittfall $t = f(t_1,\ldots,t_n)$**:** Sei $\pi \in Occ(t)$ mit $COND(t,\pi) = TRUE$ und $\mathsf{cbv\text{-}eval}_P(t|_\pi) = \propto$ und $\pi \neq \epsilon$

Fall $f \neq if$**:** $\Rightarrow COND(t,\pi) = COND(t,i\pi') = COND(t_i,\pi')$ mit Voraussetzung und Def. 3.1.4 folgt $COND(t_i,\pi') \Rightarrow_P^* TRUE$ Wegen $\mathsf{cbv\text{-}eval}_P(t|_\pi) = \mathsf{cbv\text{-}eval}_P(t_i|_{\pi'})$
$\Rightarrow \mathsf{cbv\text{-}eval}_P(t_i|_{\pi'}) = \propto \mathsf{cbv\text{-}eval}_P(t_i) \underbrace{=}_{IH} \propto \Rightarrow$ da $f \neq if$ $\mathsf{cbv\text{-}eval}_P(t) = \propto$

$f = if$ **und** $i = 1$**:** analog zum vorherigen Fall

$f = if$ **und** $i = 2$**:** $COND(t,\pi) = t_1 \equiv true \wedge COND(t_2,\pi') \Rightarrow \mathsf{cbv\text{-}eval}_P(t_1) = true$ und $COND(t_2,\pi') \Rightarrow_P^* TRUE(1)$.
Weiter gilt: $\mathsf{cbv\text{-}eval}_P(t|_\pi) = \propto \Rightarrow \mathsf{cbv\text{-}eval}_P(t_2|_{\pi'}) = \propto$ (2).
Mit (1), (2) und IH folgt $\mathsf{cbv\text{-}eval}_P(t_2) = \propto$. Und da $\mathsf{cbv\text{-}eval}_P(t_1) = true$ (s. o.) $\mathsf{cbv\text{-}eval}_P(t) = \propto$.

$f = if$ **und** $i = 3$**:** analog zum vorherigen Fall

Hausaufgabe 6.2 (Terminierung) **(3 Punkte)**

Seien F_{case_s}, F_f und F'_f Funktionsprozeduren mit $F_{case\_s} =$

function $case\_s(b : bool,\ x, y : s) : s \Leftarrow$ **if** b **then** x **else** y **fi**

für jede Sorte s. F'_f entsteht dabei aus F_f, indem jedes Funktionssymbol if_s in F_f durch $case\_s$ ersetzt wird. Formulieren Sie notwendige und hinreichende Bedingungen an F_f für

(a) F'_f terminiert *call-by-name*,

(b) F'_f terminiert *call-by-value*.

wenn alle anderen Funktionsprozeduren des Programms *call-by-value* terminieren (vgl. Übung 3.1.7).

Lösungsvorschlag:

(a) Behauptung: F'_f terminiert *call-by-name* $\Leftrightarrow$ F_f terminiert *call-by-name*

Beweis: wegen $\delta_{P,case_s} = \delta_{P,if_s}$ gilt: Es gibt eine fundierte Relation $\gg$ und einen Fixpunkt von $\mathfrak{I}_{F_f}$ mit $D(\phi)(s^*) \gg D(\phi)(\sigma(t^*))$ für alle $\sigma = \{x^*/t^*\}$ mit $s^* \in \mathcal{T}(\Sigma(P))_w$ für alle $\pi \in Occ(R_f)$ mit $R_f|_\pi = f(t^*)$ und $\sigma(R_f) \downarrow_P \pi$. $\Leftrightarrow$ Es gibt eine fundierte Relation $\gg$ und einen Fixpunkt von $\mathfrak{I}_{F'_f}$ mit $D(\phi)(s^*) \gg D(\phi)(\sigma(t^*))$ für alle $\sigma = \{x^*/t^*\}$ mit $s^* \in \mathcal{T}(\Sigma(P'))_w$ für alle $\pi \in Occ(R'_f)$ mit $R'_f|_\pi = f(t^*)$ und $\sigma(R'_f) \downarrow_{P'} \pi$.

(b) Behauptung: F'_f terminiert *call-by-value* $\Leftrightarrow$ F_f ist nicht rekursiv

Beweis:

„$\Longleftarrow$“: offensichtlich, da F'_f keine rekursiven Aufrufe enthält

„$\Longrightarrow$“: Beweis durch Widerspruch

Annahme:		$\exists \pi \in Occ(R_f).R_f|\pi = f(t^*)$
	$\Rightarrow$	$R'_f|_\pi = f(t^*)$
	$\Rightarrow$	$COND(\sigma(R'_f), \pi) = TRUE$, da R'_f keine if-Terme enthält
	$\Rightarrow$	es gibt eine fundierte Relation $\gg$ und einen Fixpunkt ϕ von $\mathfrak{I}_{F'_f}$ mit
(1)		$q^* \gg D(\phi)(\theta(t^*))$ für alle $\theta = \{x^*/q^*\}$ mit $q^* \in \mathcal{T}(\Sigma^c)_w$, da F'_f cbv-terminiert.
		F'_f terminiert cbn gemäß Satz 3.1.9.(i) und da F'_f cbv-terminiert
	$\Rightarrow$	F'_f terminiert gemäß Satz 3.1.4
	$\Rightarrow$	ϕ kann ω-total gewählt werden
	$\Rightarrow$	$D(\phi)(\theta(t^*)) \in \mathcal{T}(\Sigma^c)_s$
	$\Rightarrow$	$q^*_j \gg q^*_{j+1} \forall j \in \mathbb{N}$ mit $q^*_0 := q^*, q_{i+1} := D(\phi)(\theta_i(t^*)), \theta_i := \{x^*/q^*_i\}$
	$\Rightarrow$	mit (1) Widerspruch zur Fundiertheit von $\gg$

Hausaufgabe 6.3 (Primitiv-rekursive Funktionen) **(2 Punkte)**

Wir nennen eine Funktionsprozedur F *primitiv-rekursiv* gdw. F nicht rekursiv definiert oder aber von der Form

function $f(x : nat, y_1 : s_1, \ldots, y_k : s_k) : s \Leftarrow$
  if $x = 0$
    then $g(y_1, \ldots, y_k)$
    else $h(f(x-1, y_1, \ldots, y_k), x-1, y_1, \ldots, y_k)$
  fi

ist, wobei $g \in \Sigma(BM)$ oder **function** $g(y_1 : s_1, \ldots, y_k : s_k) : s \Leftarrow \ldots$ primitiv-rekursiv ist und ebenso $h \in \Sigma(BM)$ oder **function** $h(z : s, x : nat, y_1 : s_1, \ldots, y_k : s_k) : s \Leftarrow \ldots$ primitiv-rekursiv ist. Ein funktionales Programm P heißt dementsprechend *primitiv-rekursiv* gdw. jede Funktionsprozedur in P primitiv-rekursiv ist. Beweisen Sie, dass jedes primitiv-rekursive Programm stark terminiert.

Lösungsvorschlag:

Programm P ist primitiv rekursiv
$\Leftrightarrow$ jede Prozedur F in P ist primitiv rekursiv
$\Rightarrow$ jede Prozedur F ist normal-rekursiv (vgl. Def. 3.2.5)
$\Rightarrow$ jede Prozedur F in P terminiert stark gdw F cbv-terminiert (Satz 3.2.2).

Wir zeigen, dass jede primitiv-rekursive Prozedur cbv-terminiert:

Seien ϕ ein Fixpunkt, $q^* \in \mathcal{T}(\Sigma)_{nat} \times \mathcal{T}(\Sigma)_w$, $\theta = \{xy^*/q^*\}$, $\pi = 31$. Dann ist $R_f|_\pi = f(x-1, y^*)$.

$D(\phi) \models COND(\theta(R_f), \pi) \Rightarrow D(\phi) \models eq_{nat}(q_1, 0) \equiv false \Rightarrow q_1 = succ(r)$.

Sei $(a_1, \ldots, a_{k+1}) \gg (b_1, \ldots, b_{k+1}) \Leftrightarrow a_1 >_{\mathcal{T}} b_1$
$\Rightarrow$ $(q_1, q_2, \ldots, q_{k+1}) \gg D(\phi)(\theta(x-1, y^*)) = D(\phi)(r, y^*) = (r, y^*)$
$\Rightarrow$ Jede Prozedur F in P terminiert cbv
$\Rightarrow$ P terminiert cbv