Semantik und Programmverifikation

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

Hausaufgabe mit Lösungsvorschlag 7

Hausaufgabe 7.1 (Grenzen der formalen Verifikation) **(8 Punkte)**

Gegeben sei das funktionale Programm $P_{plus} = \langle F_{plus} \rangle$, wobei P_{plus} definiert ist durch

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

Zeigen Sie

(a) $\varphi_{kom}, \varphi_{kom,1}, \varphi_{kom,2} \in Th_{P_{plus}}$ und

(b) $\varphi_{kom}, \varphi_{kom,1}, \varphi_{kom,2} \notin (AX_{P_{plus}})^{\models}$.

Dabei sind $\varphi_{kom}, \varphi_{kom,1}$ und $\varphi_{kom,2}$ definiert durch

$$\begin{array}{rcl}\varphi_{kom} & = & \forall x, y : nat.\ plus(x, y) \equiv plus(y, x), \\ \varphi_{kom,1} & = & \forall y : nat.\ plus(0, y) \equiv plus(y, 0), \\ \varphi_{kom,2} & = & \forall x : nat.\ [\forall y : nat.plus(x, y) \equiv plus(y, x) \\ & & \rightarrow\ \forall y : nat.plus(succ(x), y) \equiv plus(y, succ(x))].\end{array}$$

(vgl. Übung 3.6.1.(ii))

Lösungsvorschlag:

(a) Wir gehen wie in Abschnitt 3.7 des Buches skizziert vor: Zunächst zeigen wir $\varphi_{kom,1} \in (AX_{P_{plus}})^{\models}$ und $\varphi_{kom,2} \in (AX_{P_{plus}})^{\models}$ jeweils durch Induktion.
Damit gilt das Induktionsaxiom $\varphi_{kom,1} \wedge \varphi_{kom,2} \rightarrow \varphi_{kom}$ in $Th_{P_{plus}}$ und es folgt $\varphi_{kom} \in Th_{P_{plus}}$.
Zum Beweis von $\varphi_{kom,1} \in Th_{P_{plus}}$ weisen wir nach, dass die beiden Induktionsformeln

$$\varphi_{kom,1,1} = 0 \equiv plus(0, 0)$$

und

$$\varphi_{kom,1,2} = \forall y : nat\ (y \equiv plus(y, 0) \rightarrow succ(y) \equiv plus(succ(y), 0))$$

in $(AX_{P_{plus}})^{\models}$sind.

$$AX_{P_{plus}} \models 0 \equiv plus(0, 0) \Leftrightarrow \alpha_0 = \alpha_{plus}(\alpha_0, \alpha_0) = \alpha_{if}(\alpha_{eq}(\alpha_0, \alpha_0), \alpha_0, \ldots) = \alpha_0$$

Also folgt $\varphi_{kom,1,1} \in (AX_{P_{plus}})^{\models}$.
Für $\varphi_{kom,1,2} \in (AX_{P_{plus}})^{\models}$ ist für jedes $a \in \mathcal{A}_{nat}$ zu zeigen:

$$a = \alpha_{plus}(a, \alpha_0) \rightarrow succ(a) = \alpha_{plus}(\alpha_{succ}(a), \alpha_0)$$

Sei also $a = \alpha_{plus}(a, \alpha_0)$. Es gilt

$$\begin{aligned}\alpha_{plus}(\alpha_{succ}(a), \alpha_0) &= \alpha_{if}(\alpha_{eq}(\alpha_{succ}(a), \alpha_0), \alpha_0, \alpha_{succ}(\alpha_{plus}(\alpha_{pred}(\alpha_{succ}(a)), \alpha_0))) \\ &= \alpha_{succ}(\alpha_{plus}(a, \alpha_0)) \\ &= \alpha_{succ}(a)\end{aligned}$$

Damit folgt $\varphi_{kom,1,2} \in (AX_{P_{plus}})^{\models}$.

Da also $\varphi_{kom,1,1} \in (AX_{P_{plus}})^{\models}$ und $\varphi_{kom,1,2} \in (AX_{P_{plus}})^{\models}$ folgt mit dem Induktionsaxiom für $\varphi_{kom,1}$ dann $\varphi_{kom,1} \in Th_{P_{plus}}$.

Für den Beweis von $\varphi_{kom,2} \in Th_{P_{plus}}$ erhalten wir nach Generalisierung die beiden Induktionsformeln

$$\varphi_{kom,2,1} = \forall x : nat\ succ(plus(0, x)) \equiv plus(0, succ(x)))$$

und

$$\begin{aligned}\varphi_{kom,2,2} = \forall y : nat\ (\forall x : nat\ succ(plus(y, x)) \equiv plus(y, succ(x)) \rightarrow \\ \forall x : nat\ succ(plus(succ(y), x)) \equiv plus(succ(y), succ(x))).\end{aligned}$$

Es gilt

$$\forall a \in \mathcal{A}_{nat}.\ (\alpha_{plus}(\alpha_0, a)) = a.$$

Damit folgt

$$\forall a \in \mathcal{A}_{nat}.\ \alpha_{succ}(\alpha_{plus}(\alpha_0, a)) = \alpha_{succ}(a) = \alpha_{plus}(\alpha_0, \alpha_{succ}(a))$$

und somit $\varphi_{kom,2,1} \in (AX_{P_{plus}})^{\models}$.

Für $\varphi_{kom,2,2} \in (AX_{P_{plus}})^{\models}$ ist zu zeigen:

$$\begin{aligned}\forall a : nat\ (\forall b : nat\ \alpha_{succ}(\alpha_{plus}(a, b)) = \alpha_{plus}(a, \alpha_{succ}(b)) \rightarrow \\ \forall b : nat\ \alpha_{succ}(\alpha_{plus}(\alpha_{succ}(a), b)) = \alpha_{plus}(\alpha_{succ}(a), \alpha_{succ}(b)))\end{aligned}$$

Sei also $a \in \mathcal{A}_{nat}$ beliebig und gelte

$$\forall b : nat\ \alpha_{succ}(\alpha_{plus}(a, b)) = \alpha_{plus}(a, \alpha_{succ}(b))$$

Dann folgt

$$\begin{aligned}\alpha_{plus}(\alpha_{succ}(a), \alpha_{succ}(b)) &= \alpha_{if}(\alpha_{eq}(\alpha_{succ}(a), \alpha_0), \alpha_{succ}(b), \alpha_{succ}(\alpha_{plus}(\alpha_{pred}(\alpha_{succ}(a)), \alpha_{succ}(b)))) \\ &= \alpha_{succ}(\alpha_{plus}(\alpha_{pred}(\alpha_{succ}(a)), \alpha_{succ}(b))) \\ &= \alpha_{succ}(\alpha_{plus}(a, \alpha_{succ}(b))) \\ &= \alpha_{succ}(\alpha_{succ}(\alpha_{plus}(a, b))) \\ &= \alpha_{succ}(\alpha_{if}(\alpha_{eq}(\alpha_{succ}(a), \alpha_0), b, \alpha_{succ}(\alpha_{plus}(\alpha_{pred}(\alpha_{succ}(a)), b)))) \\ &= \alpha_{succ}(\alpha_{plus}(\alpha_{succ}(a), b))\end{aligned}$$

Also gilt $\varphi_{kom,2,2} \in (AX_{P_{plus}})^{\models}$

Mit $\varphi_{kom,2,1} \in (AX_{P_{plus}})^{\models}$ und $\varphi_{kom,2,2} \in (AX_{P_{plus}})^{\models}$ folgt mit dem Induktionsaxiom $\varphi_{kom,2} \in Th_{P_{plus}}$.

Wie bereits erwähnt folgt mit $\varphi_{kom,2,} \in Th_{P_{plus}}$ und $\varphi_{kom,1} \in Th_{P_{plus}}$ dann $\varphi_{kom} \in Th_{P_{plus}}$ nach dem Induktionsaxiom.

(b) Wir betrachten die Alebra A aus Beispiel 3.6.1. Es gilt $A \models AX_{P_{plus}}$. Weiter gilt $\alpha_{plus}(1.5, 0.5) = 1.5 \neq -0.5 = \alpha_{plus}(0.5, 1.5)$, also $A \not\models \varphi_{kom}$

Wir veränderen die Definition von α_{plus} wie folgt:

$$\alpha_{plus}(n, m) = \begin{cases} (n - m) + 2 & \text{, falls } n \notin \mathbb{N} \\ n + m & \text{, sonst} \end{cases}$$

Der Beweis für $A \models AX_{P_{plus}}$ erfolgt analog zu dem im Beispiel.

Nun gilt $\alpha_{plus}(0, 0.5) = 0.5 \neq 2.5 = \alpha_{plus}(0.5, 0)$, also $A \not\models \varphi_{kom,1}$.

Schließlich gilt $\forall m : nat\ \alpha_{plus}(1, m) = 1 + m = \alpha_{plus}(m, 1)$, aber $\alpha_{plus}(2, 0.5) = 2.5 \neq 0.5 = \alpha_{plus}(0.5, 2)$, also $A \not\models \varphi_{kom,2}$.

Hausaufgabe 7.2 (Induktionsaxiome) (5 Punkte)

Seien P_{plus} und φ_{kom} wie zuvor definiert. Weiter sei $\mathsf{IND}_{P_{\text{plus}}}$ die Menge aller Formeln der Form

$$\begin{aligned}
&\psi[x/0, y/0]\\
\wedge\ \ &\forall x, y : nat\ (\ \psi[x/0, y] \to \psi[x/0, y/succ(y)]\ )\\
\wedge\ \ &\forall x, y : nat\ (\forall z : nat\ \psi[x, y/z] \to \psi[x/succ(x), y/0]\ )\\
\wedge\ \ &\forall x, y : nat\ (\forall z : nat\ \psi[x, y/z] \wedge \psi[x/succ(x), y] \to \psi[x/succ(x), y/succ(y)]\ )\\
&\to\ \forall x, y : nat\ \psi[x, y],
\end{aligned}$$

mit $\psi[x, y] \in \mathcal{F}(\Sigma, \mathcal{V})$, $x, y \in \mathcal{V}_{nat}$ und $\mathcal{V}_f(\psi[x, y]) = \{x, y\}$.

(a) Zeigen Sie $\mathsf{IND}_{P_{plus}} \subseteq Th_{P_{plus}}$.

Lösungsvorschlag:

Wir betrachten die lexikographische Ordnung $\gg \subseteq \mathcal{T}(\Sigma^c)^2_{nat} \times \mathcal{T}(\Sigma^c)^2_{nat}$ mit

$$(n, m) \gg (n', m') \Leftrightarrow n = succ(n') \text{ oder } (n = n' \text{ und } m = succ(m'))$$

Mit Satz 1.4.1 (iii) ist $\gg$ fundiert, also gilt das Noethersche Induktionslemma:

$$\begin{aligned}
&\Big[\forall n, m \in \mathcal{T}(\Sigma^c)_{nat}.\\
&\quad \big(\forall n', m' \in \mathcal{T}(\Sigma^c)_{nat}.\ (n', m') \gg (n, m) \Rightarrow \psi[x/n', y/m'] \in Th_{P_{plus}}\big)\\
&\quad\ \Rightarrow \psi[x/n, y/m] \in Th_{P_{plus}}\Big]\\
&\Rightarrow \forall n, m \in \mathcal{T}(\Sigma^c)_{nat} \psi[x/n, y/m] \in Th_{P_{plus}}
\end{aligned}$$

Die Bedingung $(n', m') \gg (n, m)$ kann man nun in die verschiedenen Fälle aufteilen:

$$\begin{aligned}
&\psi[x/0, y/0] \in Th_{P_{plus}} \wedge\\
&\forall m \in \mathcal{T}(\Sigma^c)_{nat}.\ (\psi[x/0, y/m] \in Th_{P_{plus}} \Rightarrow \psi[x/0, y/succ(m)] \in Th_{P_{plus}}) \wedge\\
&\forall n \in \mathcal{T}(\Sigma^c)_{nat}.\ (\forall l \in \mathcal{T}(\Sigma^c)_{nat}.\ \psi[x/n, y/l] \in Th_{P_{plus}} \Rightarrow \psi[x/succ(n), y/0] \in Th_{P_{plus}}) \wedge\\
&\forall n, m \in \mathcal{T}(\Sigma^c)_{nat}.\ (\forall l \in \mathcal{T}(\Sigma^c)_{nat}.\ \psi[x/n, y/l] \in Th_{P_{plus}} \Rightarrow \psi[x/succ(n), y/succ(m)] \in Th_{P_{plus}})\\
&\Rightarrow \forall n.m \in \mathcal{T}(\Sigma^c)_{nat} \psi[x/n, y/m] \in Th_{P_{plus}}
\end{aligned}$$

Dies lässt sich vereinfachen zu:

$$\begin{aligned}
&\Big[\psi[x/0, y/0]\\
&\ \wedge\ \forall m \in \mathcal{T}(\Sigma^c)_{nat}.\ (\psi[x/0, y/m] \to \psi[x/0, y/succ(m)])\\
&\ \wedge\ \forall n \in \mathcal{T}(\Sigma^c)_{nat}.\ (\forall l \in \mathcal{T}(\Sigma^c)_{nat}.\ \psi[x/n, y/l] \to \psi[x/succ(n), y/0])\\
&\ \wedge\ \forall n, m \in \mathcal{T}(\Sigma^c)_{nat}.\ (\forall l \in \mathcal{T}(\Sigma^c)_{nat}.\ \psi[x/n, y/l] \to \psi[x/succ(n), y/succ(m)])\\
&\ \to \forall n.m \in \mathcal{T}(\Sigma^c)_{nat} \psi[x/n, y/m]\Big] \in Th_{P_{plus}}
\end{aligned}$$

(b) Beweisen Sie $\varphi_{kom} \in Th_{\text{P}_{\text{plus}}}$ unter Verwendung von $\mathsf{IND}_{P_{\text{plus}}}$.

Lösungsvorschlag:

Sei A eine beliebige Σ-Algebra mit $A \models (AX_{P_{plus}})^{\models}$.

1) $A \models plus(0,0) \equiv plus(0,0)$ gilt offensichtlich.
2) Zu zeigen:

$$\forall x, y : nat\ (plus(0,y) \equiv plus(y,0) \rightarrow plus(0, succ(y)) \equiv plus(succ(y), 0)) \in (AX_{P_{plus}})^{\models}$$

Seien also $a, b \in \mathcal{A}_{nat}$ beliebig und gelte

$$\alpha_{plus}(\alpha_0, b) = \alpha_{plus}(b, \alpha_0).$$

Dann gilt

$$\alpha_{plus}(\alpha_0, \alpha_{succ}(b)) = \alpha_{succ}(b) = \alpha_{succ}(\alpha_{plus}(\alpha_0, b)) = \alpha_{succ}(\alpha_{plus}(b, \alpha_0)) = \alpha_{plus}(\alpha_{succ}(b), \alpha_0)$$

3) Zu zeigen:

$$\forall x, y : nat\ (\forall z : nat\ plus(x,z) \equiv plus(z,x) \rightarrow plus(succ(x), 0) \equiv plus(0, succ(x))) \in (AX_{P_{plus}})^{\models}$$

Seien also $a, b \in \mathcal{A}_{nat}$ beliebig und gelte

$$\forall c \in \mathcal{A}_{nat}.\ \alpha_{plus}(a, c) = \alpha_{plus}(c, a).$$

Dann gilt

$$\alpha_{plus}(\alpha_{succ}(a), \alpha_0) = \alpha_{succ}(\alpha_{plus}(a, \alpha_0)) = \alpha_{succ}(\alpha_{plus}(\alpha_0, a)) = \alpha_{succ}(a) = \alpha_{plus}(\alpha_0, \alpha_{succ}(a))$$

4) Zu zeigen:

$$\begin{aligned}\forall x, y : nat\ (\forall z : nat\ plus(x,z) \equiv plus(z,x) \wedge plus(succ(x), y) \equiv plus(y, succ(x))\\ \rightarrow plus(succ(x), succ(y)) \equiv plus(succ(y), succ(x))) \in (AX_{P_{plus}})^{\models}\end{aligned}$$

Seien also $a, b \in \mathcal{A}_{nat}$ beliebig und gelte

$$\forall c \in \mathcal{A}_{nat}.\ \alpha_{plus}(a, c) = \alpha_{plus}(c, a) \wedge \alpha_{plus}(\alpha_{succ}(a), b) = \alpha_{plus}(b, \alpha_{succ}(a))$$

Dann gilt.

$$\begin{aligned}\alpha_{plus}(\alpha_{succ}(a), \alpha_{succ}(b)) &= \alpha_{succ}(\alpha_{plus}(a, \alpha_{succ}(b)))\\ &= \alpha_{succ}(\alpha_{plus}(\alpha_{succ}(b), a))\\ &= \alpha_{succ}(\alpha_{succ}(\alpha_{plus}(b, a)))\\ &= \alpha_{succ}(\alpha_{succ}(\alpha_{plus}(a, b)))\\ &= \alpha_{succ}(\alpha_{plus}(\alpha_{succ}(a), b))\\ &= \alpha_{succ}(\alpha_{plus}(\alpha_{succ}(a), b))\\ &= \alpha_{succ}(\alpha_{plus}(b, \alpha_{succ}(a)))\\ &= \alpha_{plus}(\alpha_{succ}(b), \alpha_{succ}(a))\end{aligned}$$

(c) Der im Buch angegebene Beweis für $\varphi_{kom} \in Th_{P_{plus}}$ erfolgt unter Verwendung von Formel (3.7.12). Diskutieren Sie den Unterschied zwischen $\mathsf{IND}_{P_{\mathrm{plus}}}$ und (3.7.12) beim Beweis von $\varphi_{kom} \in Th_{P_{plus}}$.

Lösungsvorschlag:

Der Beweis im Buch verwendet eine geschachtelte Induktion, während der hier durchgeführte Beweis die einzelnen Induktionsformeln ohne Induktion beweist. Die hier verwendete Ordnung kodiert quasi bereits die Induktionsschritte, die im Buch in der geschachtelten Induktion gemacht werden.

Hausaufgabe 7.3 (Spezifikation und Verifikation) **(6 Punkte)**

Das funktionale Programm $P_{set} = \langle D_{set}, F_{insert} \rangle$ sei durch folgende Datenstrukturdefinition und folgende Funktionsprozedur definiert:

$$D_{set} = \textsf{structure } \Diamond, cons(element : nat, rest : set) : set$$

$$
\begin{array}{ll}
F_{insert} = & \textbf{function } insert(n : nat, s : set) : set \;\Leftarrow \\
& \quad \textbf{if } s = \Diamond \\
& \qquad \textbf{then } cons(n, \Diamond) \\
& \qquad \textbf{else if } n = element(s) \\
& \qquad\qquad\quad \textbf{then } s \\
& \qquad\qquad\quad \textbf{else } cons(element(s), insert(n, rest(s))) \\
& \qquad\qquad \textbf{fi} \\
& \quad \textbf{fi}
\end{array}
$$

(a) Formulieren Sie durch geeignete Erweiterung des funktionalen Programmes P_{set} durch einen Spezifikationsteil S_{set} die folgenden Aussagen:

(i) Fügt man zweimal hintereinander eine Zahl n in eine Menge s ein, so resultiert daraus die gleiche Menge wie beim einmaligen Einfügen dieser Zahl n in s.

(ii) Eine Zahl, die noch nicht in einer Menge s vorhanden ist, wird beim Einfügen in s an das Ende (der Listenstruktur) gestellt.

Lösungsvorschlag:

Wie definieren folgende Prozeduren für den Spezifikationsteil:

$$
\begin{array}{ll}
F_{contains} = & \textbf{function } contains(n : nat, s : set) : bool \;\Leftarrow \\
& \quad \textbf{if } s = \Diamond \\
& \qquad \textbf{then } false \\
& \qquad \textbf{else if } n = element(s) \\
& \qquad\qquad\quad \textbf{then } true \\
& \qquad\qquad\quad \textbf{else } contains(n, rest(s)) \\
& \qquad\qquad \textbf{fi} \\
& \quad \textbf{fi}
\end{array}
$$

$$
\begin{array}{ll}
F_{last} = & \textbf{function } last(s : set) : bool \;\Leftarrow \\
& \quad \textbf{if } s = \Diamond \\
& \qquad \textbf{then } 0 \\
& \qquad \textbf{else if } rest(s) = \Diamond \\
& \qquad\qquad\quad \textbf{then } element(s) \\
& \qquad\qquad\quad \textbf{else } last(rest(s)) \\
& \qquad\qquad \textbf{fi} \\
& \quad \textbf{fi}
\end{array}
$$

Damit formulieren wir die geforderten Aussagen:

(i) $\forall n : nat \; \forall s : set \; insert(n, insert(n, s)) \equiv insert(n, s)$

(ii) $\forall n : nat \; \forall s : set \; contains(n, s) \equiv false \rightarrow eq(last(insert(n, s)), n) \equiv true$

(b) Geben Sie die Axiomenmenge $AX_{P_{set}} \setminus AX_{BM}$ an.

Lösungsvorschlag:

- $eq(\Diamond, \Diamond) \equiv true$
- $\forall n : nat\, \forall s : set\ element(cons(n, s)) \equiv n$
- $\forall n : nat\, \forall s : set\ rest(cons(n, s)) \equiv s$
- $\forall n, m : nat,\ \forall s, t : set\ eq_{set}(cons(n, s), cons(m, t)) \equiv if(eq_{nat}(n, m), eq_{set}(s, t), false)$
- $element(\Diamond) = 0$
- $rest(\Diamond) = \Diamond$
- $\forall n : nat\, \forall s : set\ eq_{set}(cons(n, s), \Diamond) \equiv false$
- $\forall n : nat\, \forall s : set\ eq_{set}(\Diamond, cons(n, s)) \equiv false$
- $\forall s, t : set\ if_{set}(true, s, t) \equiv s$
- $\forall s, t : set\ if_{set}(false, s, t) \equiv t$
- $\forall n : nat\, \forall s : set\ insert(n, s) \equiv R_{insert}$

(c) Beweisen Sie für die unter (i) aufgestellte Formel ϕ die Aussage $\phi \in Th_P$. (Zeigen Sie hierzu $M_{P_{set}} \models \phi$ mit einem Induktionsbeweis.)

Lösungsvorschlag:

Zu zeigen ist $M_{P_{set}} \models \forall n : nat\ \forall s : set\ insert(n, insert(n, s)) \equiv insert(n, s)$ Dies ist äquivalent zu $\forall a \in \mathcal{T}(\Sigma^c)_{nat}.\ \forall b \in \mathcal{T}(\Sigma^c)_{set}.\delta_{insert}(a, \delta_{insert}(a, b)) = \delta_{insert}(a, b)$. Wir zeigen dies durch strukturelle Induktion über b:

- $b = \Diamond$
  Es gilt $\delta_{insert}(a, b) = cons(a, \Diamond)$ Damit folgt
$$\begin{aligned}\delta_{insert}(a, \delta_{insert}(a, b)) &= \delta_{insert}(a, cons(a, \Diamond))\\ &= cons(a, \Diamond)\\ &= \delta_{insert}(a, b).\end{aligned}$$
- $b = cons(a'b')$
  Als Induktionshypothese gilt $\forall c \in \mathcal{T}(\Sigma^c)_{nat}\ \delta_{insert}(c, \delta_{insert}(c, b')) = \delta_{insert}(c, b')$
  Es gilt $\delta_{insert}(a, \delta_{insert}(a, b)) = \delta_{insert}(a, \delta_{if_{set}}(\delta_{eq_{set}}(a, a'), s, cons(a', \delta_{insert}(a, b')))).$
  Eine Fallunterscheidung über $\delta_{eq_{set}}(a'a')$ liefert:
  - $\delta_{eq_{set}}(a'a') = true$
    $\delta_{insert}(a, \delta_{if_{set}}(\delta_{eq_{set}}(a, a'), s, cons(a', \delta_{insert}(a, b')))) = \delta_{insert}(a, s)$
  - $\delta_{eq_{set}}(a'a') = false$
$$\begin{aligned}\delta_{insert}(a, \delta_{if_{set}}(\delta_{eq_{set}}(a, a'), s, cons(a', \delta_{insert}(a, b')))) &= \delta_{insert}(a, cons(a', \delta_{insert}(a, b')))\\ &= cons(a', \delta_{insert}(a, \delta_{insert}(a, b')))\\ &= cons(a', \delta_{insert}(a, b'))\\ &= \delta_{insert}(a, b)\end{aligned}$$