Theoretische Informatik und Logik
Übungsblatt 3 (SS 2017)
Lösungen

Aufgabe 3.1 Drücken Sie folgende Prädikate jeweils als Boolesche Ausdrücke aus oder argumentieren Sie, warum das nicht möglich ist. Verwenden Sie dabei die **offizielle Syntax** für $\mathcal{BA}(\mathcal{D})$, d.h. **keine** der zusätzlichen Notationsvereinbarungen, die auf den Vorlesungsfolien erwähnt werden.

a) Über $\mathcal{D} = \mathsf{FamX}$: x ist eine leibliche Tante von y.

b) Über $\mathcal{D} = \mathsf{FamX}$: x hat eine leibliche Tante.

c) Über $\mathcal{D} = \mathbb{Z}$: Wenn x kleiner gleich y ist, dann ist z positiv.

d) Über $\mathcal{D} = \mathbb{Z}$: x ist positiv und gerade.

e) Über $\mathcal{D} = \mathbb{N}$: x ist entweder 2 oder größer als des Vierfache von y.

f) Über $\mathcal{D} = \mathbb{S}$: Wenn x nicht leer ist, dann enthält y genau 2 Stackelemente (Binärzeichen).

Lösung

a) $(\mathit{weiblich}(\mathtt{x}) \wedge (\mathit{Geschwister}(\mathtt{x}, \mathit{Mutter}(\mathtt{y})) \vee \mathit{Geschwister}(\mathtt{x}, \mathit{Vater}(\mathtt{y}))))$

b) Man kann zwar einen Booleschen Ausdruck angeben, der ausdrückt, dass y die leibliche Tante von x ist; aber um das *einstellige* Prädikat 'hat eine leibliche Tante' auszudrücken benötigt man einen Existenzquantor.

c) $(\neg(\leq(\mathtt{x},\mathtt{y}) \vee \mathtt{x} = \mathtt{y}) \vee \leq(0,\mathtt{z}))$

d) "x ist gerade" lässt sich in üblicher mathematischer Notation als $\exists y\colon x = 2y$ schreiben. Der Existenzquantor ist nicht eliminierbar; daher gibt es keinen entsprechenden Booleschen Ausdruck.

e) $((\mathtt{x} = +(1,1)) \wedge \neg\leq(+(\mathtt{y},+(\mathtt{y},+(\mathtt{y},\mathtt{y}))),\mathtt{x})) \vee (\neg\mathtt{x} = +(1,1) \wedge \leq(+(\mathtt{y},+(\mathtt{y},+(\mathtt{y},\mathtt{y}))),\mathtt{x})))$
Anmerkung: Man könnte 'das Vierfache von y' auch mit dem Term $*(+(1,+(1,+(1,1))),\mathtt{y})$ ausdrücken; aber jedenfalls ist 4 keine Konstante des Datentyps $\mathbb{N}$.
Wenn man "entweder ... oder" (etwas problematisch) nicht als exklusives Oder interpretiert, dann ist bereits (z.B.) $(\mathtt{x} = +(1,1) \vee \leq(+(\mathtt{y},+(\mathtt{y},+(\mathtt{y},\mathtt{y}))),\mathtt{x}))$ eine Lösung.

f) $(\mathrm{istleer?}(\mathtt{x}) \vee (\mathrm{istleer?}(\mathrm{pop}(\mathrm{pop}(\mathtt{y}))) \wedge \neg\mathrm{istleer?}(\mathrm{pop}(\mathtt{y}))))$

Anmerkung: Es gibt natürlich jeweils auch weitere alternative Lösungen.

Aufgabe 3.2 Es sei $\mathbb{B}$ ein abstrakter Datentyp für binäre Bäume mit folgenden Komponenten:
– Eine Konstante ϵ für den leeren Baum (kein Knoten), sowie eine weitere Konstante $\square$ für den Baum, der nur aus einem Endknoten (= Wurzel) besteht;
– eine zweistellige Funktion f, die angewendet auf zwei Bäume b_1 und b_2 den Baum liefert, der (mit einem neuen Wurzelknoten) b_1 als linken und b_2 als rechten Teilbaum hat.
Es gelte $f(\epsilon, \epsilon) = \square$, sowie $f(b_1, b_2) = \epsilon$, falls entweder $b_1 = \epsilon$ oder $b_2 = \epsilon$, aber nicht beide.

a) Definieren Sie eine Signatur $\Sigma_{\mathbb{B}}$ zu $\mathbb{B}$ (analog zu $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{S}$) und geben Sie zu jedem binären Baum, der höchstens 4 Endknoten enthält, jeweils einen entsprechenden Term aus $\mathcal{T}(\mathbb{B})$ an.

b) Geben Sie einen Booleschen Ausdruck in $\mathcal{BA}(\mathbb{B})$ an, der ausdrückt, dass x und y die beiden Teilbäume von z sind, wobei es aber nicht darauf ankommt, welcher Teilbaum links und welcher rechts steht.

c) Geben Sie eine PL-Formel über $\Sigma_{\mathbb{B}}$ an, die ausdrückt, dass x der linke oder der rechte Teilbaum des binären Baums y ist.

d) Zeigen Sie durch Induktion, dass jeder Term, der in jeder Umgebung einen Baum mit mindestens $n \geq 1$ Endknoten als Wert hat, aus mindestens $3n - 2$ Zeichen besteht.

Lösung

a) Gemäß der Unterstreichungskonvention aus der Vorlesung erhält man $\Sigma_{\mathbb{B}} = \langle\{\}, \{\underline{\epsilon}, \underline{\square}\}, \{\underline{\mathtt{f}}\}\rangle$.
Es gibt also keine Prädikatensymbole, aber die Konstantensymbole $\underline{\epsilon}$ und $\underline{\square}$, sowie das zweistellige Funktionssymbol $\underline{\mathtt{f}}$.
Es gibt 10 verschiedene binäre Bäume mit höchstens 4 Endknoten.
Entsprechende Terme sind:
0 Endknoten: $\underline{\epsilon}$
1 Endknoten: $\underline{\square}$
2 Endknoten: $\underline{\mathtt{f}}(\underline{\square},\underline{\square})$
3 Endknoten: $\underline{\mathtt{f}}(\underline{\square},\underline{\mathtt{f}}(\underline{\square},\underline{\square}))$, $\underline{\mathtt{f}}(\underline{\mathtt{f}}(\underline{\square},\underline{\square}),\underline{\square})$
4 Endknoten: $\underline{\mathtt{f}}(\underline{\square},\underline{\mathtt{f}}(\underline{\square},\underline{\mathtt{f}}(\underline{\square},\underline{\square})))$, $\underline{\mathtt{f}}(\underline{\mathtt{f}}(\underline{\square},\underline{\square}),\underline{\mathtt{f}}(\underline{\square},\underline{\square}))$, $\underline{\mathtt{f}}(\underline{\square},\underline{\mathtt{f}}(\underline{\mathtt{f}}(\underline{\square},\underline{\square}),\underline{\square}))$
$\underline{\mathtt{f}}(\underline{\mathtt{f}}(\underline{\square},\underline{\mathtt{f}}(\underline{\square},\underline{\square})),\underline{\square})$, $\underline{\mathtt{f}}(\underline{\mathtt{f}}(\underline{\mathtt{f}}(\underline{\square},\underline{\square}),\underline{\square}),\underline{\square})$

b) $(\underline{\mathrm{z}}=\underline{\mathrm{f}}(\underline{\mathrm{x}},\underline{\mathrm{y}})\vee\underline{\mathrm{z}}=\underline{\mathrm{f}}(\underline{\mathrm{y}},\underline{\mathrm{x}}))$

c) $\exists\underline{\mathrm{z}}(\underline{\mathrm{y}}=\underline{\mathrm{f}}(\underline{\mathrm{x}},\underline{\mathrm{z}})\vee\underline{\mathrm{y}}=\underline{\mathrm{f}}(\underline{\mathrm{z}},\underline{\mathrm{x}}))$

d) Wir schreiben $|t|$ für die Anzahl der Zeichen im Term $t \in \mathcal{T}(\mathbb{B})$ und zeigen durch strukturelle Induktion, dass $|t| \geq 3n - 2$ gilt, wenn t in jeder Umgebung einen binären Baum mit mindestens n Endknoten darstellt.

Induktionsanfang: Der kleinste Term t, der in jeder Umgebung den einzigen Baum mit mindestens einem Endknoten als Wert hat, besteht aus der Konstante $\underline{\square}$.
Offensichtlich gilt $|\underline{\square}| = 1 \geq 3 \cdot 1 - 2$.
Natürlich gibt es auch größere Terme, die in jeder Umgebung einen binären Baum mit mindestens einem Endknoten darstellen. Aber das ändert nichts an der Korrektheit der Behauptung für $n = 1$.

Induktionsannahme: Für alle $1 \leq k \leq n$ gilt: Wenn t in jeder Umgebung einen Baum mit mindestens k Endknoten darstellt, so gilt $|t| \geq 3k - 2$.

Induktionsschritt: Ein Term t, der in jeder Umgebung einen Baum mit mindestens $n + 1$ Endknoten als Wert hat, muss die Form $\underline{\mathtt{f}}(t_1, t_2)$ haben.
Für $i \in \{1, 2\}$ sei n_i die Anzahl der Endknoten in t_i.
Es gilt $n_1 + n_2 \geq n + 1$. Für $n > 1$ gilt außerdem, dass keiner der beiden von t_1 bzw. t_2 repräsentierten Teilbäume leer sein kann. Daher ist die Induktionsannahme auf t_1 und t_2 anwendbar und es folgt für den Induktionsschritt:

$$|t| = 4 + |t_1| + |t_2| \geq 4 + 3n_1 - 2 + 3n_2 - 2 = 3(n_1 + n_2) \geq 3(n + 1) \geq 3(n + 1) - 2.$$

Zusammenfassend: Aus dem Induktionsanfang, welcher die zu zeigende Behauptung für $n = 1$ nachweist und dem Induktionsschritt, der die Induktionssannahme von n auf $n+1$ hochzieht, folgt, dass $|t| > 3n - 2$ für jeden Term $t \in \mathcal{T}(\mathbb{B})$ gilt, der einen mit Baum mindestens $n \geq 1$ Endknoten darstellt.

Aufgabe 3.3 Untersuchen Sie eine Variante $AL'(\mathbb{N})$ der Programmiersprache $AL(\mathbb{N})$, in der es keine $\underline{\text{while}}$-Schleife, dafür aber $\underline{\text{loop}}$-Schleifen gibt. *Genauer:*

a) In der Definition von $AL'(\mathbb{N})$ wird (AL4) durch folgende Klausel ersetzt:

(AL′4) Ist α aus $AL'(\mathbb{N})$ und $v \in IVS$ eine Variable, die in α nicht vorkommt, dann ist $\underline{\text{loop}}\ v\ \underline{\text{times}}\ \alpha \in AL'(\mathbb{N})$.

Formulieren Sie eine entsprechende Bedingung (MAL′4) zur Festlegung der Semantik. (Informell: In der Umgebung I wird α genau $I(v)$ mal ausgeführt.)
Hinweis: Formulieren Sie (MAL′4) direkt, nicht durch Rückführung auf die $\underline{\text{while}}$-Schleife!

b) Überprüfen Sie Ihre Definition durch schrittweise Auswertung des $AL'(\mathbb{N})$-Programms

$$\underline{\text{loop}}\ \underline{\text{x}}\ \underline{\text{times}}\ \underline{\text{loop}}\ \underline{\text{y}}\ \underline{\text{times}}\ \underline{\text{z}} \leftarrow \underline{\text{z}} + \underline{1}$$

in einer Umgebung I mit $I(\underline{\text{x}}) = 1$, $I(\underline{\text{y}}) = 2$, $I(\underline{\text{z}}) = 3$.

c) Wir haben in der Vorlesung festgestellt, dass $AL(\mathbb{N})$ universell ist. Ist auch $AL'(\mathbb{N})$ universell? Mit anderen Worten: Lassen sich alle partiell-berechenbaren Funktionen mit einem $AL'(\mathbb{N})$-Programm berechnen?

Lösung

a) (MAL'4) $\mathcal{M}_{AL}(I, \underline{\text{loop}}\ v\ \underline{\text{times}}\ \alpha) = \begin{cases} I & \text{falls } I(v) = 0 \\ \mathcal{M}_{AL}(I', \underline{\text{loop}}\ v\ \underline{\text{times}}\ \alpha) & \text{sonst} \end{cases}$

wobei $I'(w) = \mathcal{M}_{AL}(I, \alpha)(w)$ für alle $w \neq v$ und $I'(v) = I(v) - 1$.

b) I ist ein Environment über $\mathbb{N}$ mit $I(\underline{\text{x}}) = 1$, $I(\underline{\text{y}}) = 2$ und $I(\underline{\text{z}}) = 3$.

$$\begin{aligned}
&\mathcal{M}_{AL}(I, \underline{\text{loop}}\ \underline{\text{x}}\ \underline{\text{times}}\ \underline{\text{loop}}\ \underline{\text{y}}\ \underline{\text{times}}\ \underline{\text{z}} \leftarrow \underline{\text{z}} + \underline{1}) \\
&\qquad [\text{wegen } I(\underline{\text{x}}) = 1] \\
&\overset{MAL4'}{=} \mathcal{M}_{AL}(I', \underline{\text{loop}}\ \underline{\text{x}}\ \underline{\text{times}}\ \underline{\text{loop}}\ \underline{\text{y}}\ \underline{\text{times}}\ \underline{\text{z}} \leftarrow \underline{\text{z}} + \underline{1}) \\
&\qquad \text{wobei } I'(\underline{\text{x}}) = 0, \text{ und } I'(v) = \mathcal{M}_{AL}(I, \underline{\text{loop}}\ \underline{\text{y}}\ \underline{\text{times}}\ \underline{\text{z}} \leftarrow \underline{\text{z}} + \underline{1})(v) \text{ für alle } v \neq \underline{\text{x}} \\
&\overset{MAL4'}{=} I' \quad [\text{wegen } I'(\underline{\text{x}}) = 0]
\end{aligned}$$

Um $I'(v)$ für $v \neq \underline{\text{x}}$ zu bestimmen muss man die Auswertung wie folgt fortsetzen:

$$\begin{aligned}
&\mathcal{M}_{AL}(I, \underline{\text{loop}}\ \underline{\text{y}}\ \underline{\text{times}}\ \underline{\text{z}} \leftarrow \underline{\text{z}} + \underline{1}) \\
&\qquad [\text{wegen } I(\underline{\text{y}}) = 2] \\
&\overset{MAL4'}{=} \mathcal{M}_{AL}(I'', \underline{\text{loop}}\ \underline{\text{y}}\ \underline{\text{times}}\ \underline{\text{z}} \leftarrow \underline{\text{z}} + \underline{1}) \\
&\qquad \text{wobei } I''(\underline{\text{y}}) = 1, \text{ und } I''(v) = \mathcal{M}_{AL}(I, \underline{\text{z}} \leftarrow \underline{\text{z}} + \underline{1})(v) \text{ für } v \neq \underline{\text{y}}, \\
&\qquad \text{daher } I''(\underline{\text{z}}) = \mathcal{M}_{\mathcal{T}}(I, \underline{\text{z}} + \underline{1}) = I(\underline{\text{z}}) + 1 = 3 + 1 = 4 \\
&\overset{MAL4'}{=} \mathcal{M}_{AL}(I''', \underline{\text{loop}}\ \underline{\text{y}}\ \underline{\text{times}}\ \underline{\text{z}} \leftarrow \underline{\text{z}} + \underline{1}) \\
&\qquad \text{wobei } I'''(\underline{\text{y}}) = 0, \text{ und } I'''(v) = \mathcal{M}_{AL}(I'', \underline{\text{z}} \leftarrow \underline{\text{z}} + \underline{1})(v) \text{ für } v \neq \underline{\text{y}} \\
&\qquad \text{daher } I'''(\underline{\text{z}}) = \mathcal{M}_{\mathcal{T}}(I'', \underline{\text{z}} + \underline{1}) = I''(\underline{\text{z}}) + 1 = 4 + 1 = 5 \\
&\overset{MAL4'}{=} I''' \quad [\text{wegen } I'''(\underline{\text{y}}) = 0]
\end{aligned}$$

Für die insgesamt resultierende Umgebung J gilt also: $J(\underline{\text{x}}) = I'(0) = 0$, $J(\underline{\text{y}}) = I'''(\underline{\text{y}}) = 0$, $J(\underline{\text{z}}) = I'''(\underline{\text{z}}) = 5$. Für allen anderen Variablen v gilt $J(v) = I(v)$.

c) Es ist leicht zu sehen, dass jedes $AL'(\mathbb{N})$-Programm terminiert. Daher kann $AL'(\mathbb{N})$ nicht universell sein. Es gilt übrigens auch, dass es nicht nur partielle, sondern auch totale Funktionen gibt, die zwar auf einer Turingmaschine oder mit einem $AL(\mathbb{N})$-Programm berechenbar sind, aber nicht mit einem $AL'(\mathbb{N})$-Programm.

Aufgabe 3.4 Formalisieren Sie folgende Sätze als PL-Formeln. Wählen Sie dabei jeweils zunächst eine geeignete Signatur und geben Sie die Kategorie (inklusive Stelligkeit) und die intendierte Bedeutung aller Elemente der Signatur vollständig an.
Hinweis: Gehen Sie davon aus, dass die jeweilige Domäne ausschließlich aus Personen besteht.

a) Adam hat mindestens zwei Brüder, aber keine Schwester.

b) Jeder Schüler kennt einen Lehrer, der ihn nicht kennt.

c) Adina ist eine Professorin, die alle Studierenden benotet, die sie unterrichtet.

d) Die Chefin von Adam ist dreifache Mutter.

Lösung

a) Signatur $\langle\{B, S\}, \{a\}, \{\}\rangle$ mit folgender intendierter Bedeutung:
Prädikatensymbole:
$B(x, y)$ ... x ist Bruder von y (zweistellig)
$S(x, y)$ ... x ist Schwester von y (zweistellig)
Konstantensymbol:
a ... Adam

PL-Formel: $\exists x \exists y[x \neq y \wedge B(x, a) \wedge B(y, a) \wedge \neg \exists z S(z, a)]$

b) Signatur $\langle\{S, L, K\}, \{\}, \{\}\rangle$ mit folgender intendierter Bedeutung:
Prädikatensymbole:
$S(x)$ ... x ist ein Schüler (einstellig)
$L(x)$ ... x ist ein Lehrer (einstellig)
$K(x, y)$ ... x kennt y (zweistellig)

PL-Formel: $\forall x[S(x) \supset \exists y(L(y) \wedge K(x, y) \wedge \neg K(y, x))]$

c) Signatur $\langle\{P, S, U, B\}, \{a\}, \{\}\rangle$ mit folgender intendierter Bedeutung:
Prädikatensymbole:
$P(x)$ ... x ist eine Professorin (einstellig)
$S(x)$ ... x ist ein(e) Studierende(r) (einstellig)
$U(x, y)$ ... x unterrichtet y (zweistellig)
$B(x, y)$ ... x benotet y (zweistellig)
Konstantensymbol:
a ... Adina

PL-Formel: $P(a) \wedge \forall x[(S(x) \wedge U(a, x)) \supset B(a, x)]$

d) Signatur $\langle\{\}, \{a\}, \{m, c\}\rangle$ mit folgender intendierter Bedeutung:
Konstantensymbol:
a ... Adam
Funktionssymbole:
$m(x)$ ... die Mutter von x (einstellig)
$c(x)$ ... die Chefin von x (einstellig)

PL-Formel: $\exists x \exists y \exists z[x \neq y \wedge y \neq z \wedge x \neq z \wedge c(a) = m(x) \wedge c(a) = m(y) \wedge c(a) = m(z)$
$\wedge \forall u(c(a) = m(u) \supset (u = x \vee u = y \vee u = z))]$

Erläuterungen: Die Relation "ist Mutter von" ist funktional. Daher ist es, wie in der Vorlesung besprochen, besser ein entsprechendes Funktionssymbol für "die Mutter von" als ein zweistelliges Prädikatensymbol für "ist Mutter von" in die Signatur zu stellen.
Ähnliches gilt für "die Chefin von": Der bestimmte Artikel signalisiert, dass die Relation funktional ist und daher ein entsprechendes Funktionssymbol in der Signatur sein sollte.
Die Eigenschaft "ist dreifache Mutter" könnte möglicherweise auch als "hat *mindestens* drei Kinder" verstanden werden. Unter dieser Voraussetzung ist bereits die einfachere Formel

$$\exists x \exists y \exists z[x \neq y \wedge y \neq z \wedge x \neq z \wedge c(a) = m(x) \wedge c(a) = m(y) \wedge c(a) = m(z)]$$

eine korrekte Formalisierung.

Aufgabe 3.5 Beweisen oder widerlegen Sie folgende Behauptungen für die Theorie der Totalordnungen (Folie 420 - Axiome: Transitivität, Reflexivität, Antisymmetrie, Totalität):

a) Das Axiom der Reflexivität ist unabhängig von der Menge der anderen drei Axiome.

b) Das Axiom der Totalität ist unabhängig von der Menge der anderen drei Axiome.

Lösung

a) Wir zeigen, dass R nicht unabhängig vom Rest der Axiome ist:
Es gilt $To \models R$, da To nur dann wahr ist, wenn $\Phi(\preceq)(c,d) = \mathbf{t}$ oder $\Phi(\preceq)(d,c) = \mathbf{t}$ für alle Elemente c, d der Domäne D gilt; also insbesondere auch für $c = d$. Das heißt, dass R nicht unabhängig von $\{Tr, As, To\}$ ist, da R aus diesen Axiomen logisch folgt ($Tr, As, To \models R$).

b) Wir zeigen, dass Totalität unabhängig vom Rest der Axiome ist:
Es ist klar, dass es Totalordnungen gibt, also Modellstrukturen in denen Transitivität (Tr), Reflexivität (R), Totalität (To) und Antisymmetrie (As) gilt. Daher gilt $Tr, R, As \not\models \neg To$. Man nehme z.B. die übliche "kleiner-gleich"-Relation über den ganzen Zahlen. Also $\mathcal{I} = \langle Z, \Phi, \xi\rangle$ über der Signatur $\Sigma_O = \langle\{\preceq\}, \{\}, \{\}\rangle$, wobei $\Phi(\preceq)$="$\leq$" (ξ beliebig).

   Um zu zeigen, dass auch $Tr, R, As \not\models To$ gilt, müssen wir nun noch eine Interpretation über Σ_O angeben, die eine transitive, reflexive und antisymmetrische Relation bestimmt, die nicht total ist. Das geht über einem Gegenstandsbereich mit nur zwei Elementen wie folgt: $\mathcal{I}' = \langle\{a, b\}, \Phi, \xi\rangle$, wobei $\Phi(\preceq)(a,a) = \Phi(\preceq)(b,b) = \mathbf{t}$ und $\Phi(\preceq)(a,b) = \Phi(\preceq)(b,a) = \mathbf{f}$. Offensichtlich gilt $\mathsf{val}_{\mathcal{I}'}(Tr) = \mathbf{t}$, $\mathsf{val}_{\mathcal{I}'}(As) = \mathbf{t}$, $\mathsf{val}_{\mathcal{I}'}(R) = \mathbf{t}$, aber $\mathsf{val}_{\mathcal{I}'}(To) = \mathbf{f}$, und daher $Tr, R, As \not\models To$.
   (Alternativ könnte man z.B. auch auf $\mathcal{I}' = \langle\omega, \Phi, \xi\rangle$, mit $\Phi(\preceq)$="=" verweisen.)