# Theoretische Informatik und Logik
# Übungsblatt 3 (2021W)
# 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.
*Hinweis:* Versuchen Sie möglichst kurze Ausdrücke zu finden.

a) Über $\mathcal{D} = \mathbb{N}$: $3xy = z - 2$.

b) Über $\mathcal{D} = \mathbb{N}$: $x + y$ ist gerade.

c) Über $\mathcal{D} = \mathbb{Z}$: Wenn $|x| > y$, dann ist $z < y$.

d) Über $\mathcal{D} = \text{FamX}$: $x$ ist eine (leibliche) Tante von $y$.

e) Über $\mathcal{D} = \text{FamX}$: $x$ hat eine (leibliche) Tante.

f) Über $\mathcal{D} = \mathbb{S}$: Die Tiefe von $x$ ist größer als 3.
*Hinweis:* Die Tiefe eines Stacks ist die Anzahl der darin vorkommenden Stackelemente ($\underline{\mathtt{0}}$ oder $\underline{\mathtt{1}}$).

g) Über $\mathcal{D} = \mathbb{S}$: $x$ ist genau dann leer, wenn $y$ nicht leer ist.

**Lösung 3.1**

a) $\underline{+}\underline{(}\underline{*}\underline{(}\underline{+}\underline{(}\underline{\mathtt{1}}\underline{,}\underline{+}\underline{(}\underline{\mathtt{1}}\underline{,}\underline{\mathtt{1}}\underline{)}\underline{)}\underline{,}\underline{*}\underline{(}\underline{\mathtt{x}}\underline{,}\underline{\mathtt{y}}\underline{)}\underline{)}\underline{,}\underline{+}\underline{(}\underline{\mathtt{1}}\underline{,}\underline{\mathtt{1}}\underline{)}\underline{)}\underline{=}\underline{\mathtt{z}}$
Beachten Sie, dass 2 und 3 in $\mathbb{N}$ nicht als Konstantensymbole zur Verfügung stehen.
Außerdem kann man '$-$' nicht durch '$\dot{-}$' ersetzen. Daher wurde zunächst 2 (als $1 + 1$) auf die andere Seite der Gleichung gebracht.
Es gibt aber natürlich alternative Lösungen.

b) Das Prädikat 'gerade' lässt sich nicht als Boolescher Ausdruck darstellen. Dazu würde man einen Existenzquantor benötigen. ('$x + y$ ist gerade' wird durch die Formel $\exists\underline{\mathtt{z}}\,\underline{*}\underline{(}\underline{+}\underline{(}\underline{\mathtt{1}}\underline{,}\underline{\mathtt{1}}\underline{)}\underline{,}\underline{\mathtt{z}}\underline{)}\underline{=}\underline{+}\underline{(}\underline{\mathtt{x}}\underline{,}\underline{\mathtt{y}}\underline{)}$ ausgedrückt. Diese Formel ist aber kein Boolescher Ausdruck.)

c) Beachten Sie, dass 'Wenn A, dann B' zu 'Nicht A oder B' umgeformt werden muss.
$|x| > y$ lässt sich ausdrücken als '$y$ ist negativ oder $x^2 > y^2$'.
$\underline{(}\underline{\neg}\underline{(}\underline{\leq}\underline{(}\underline{\mathtt{y}}\underline{,}\underline{\mathtt{0}}\underline{)}\underline{\vee}\underline{\leq}\underline{(}\underline{*}\underline{(}\underline{\mathtt{y}}\underline{,}\underline{\mathtt{y}}\underline{)}\underline{,}\underline{*}\underline{(}\underline{\mathtt{x}}\underline{,}\underline{\mathtt{x}}\underline{)}\underline{)}\underline{)}\underline{\vee}\underline{\leq}\underline{(}\underline{\mathtt{z}}\underline{,}\underline{\mathtt{y}}\underline{)}\underline{)}$
Natürlich gibt es auch hier wieder viele alternative Lösungen.

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

e) Es gibt keinen Booleschen Ausdruck, der über der Signatur $\Sigma_{\text{FamX}}$ das einstellige Prädikat 'hat eine (leibliche) Tante' ausdrückt. Dazu benötigt man (ähnlich wie in Aufgabe b) einen Existenzquantor.

f) $\underline{\neg}\,\underline{\text{istleer?}}\underline{(}\underline{\text{pop}}\underline{(}\underline{\text{pop}}\underline{(}\underline{\text{pop}}\underline{(}\underline{\mathtt{x}}\underline{)}\underline{)}\underline{)}\underline{)}$

g) $\underline{(}\underline{(}\underline{\text{istleer?}}\underline{(}\underline{\mathtt{x}}\underline{)}\underline{\wedge}\underline{\neg}\,\underline{\text{istleer?}}\underline{(}\underline{\mathtt{y}}\underline{)}\underline{)}\underline{\vee}\underline{(}\underline{\text{istleer?}}\underline{(}\underline{\mathtt{y}}\underline{)}\underline{\wedge}\underline{\neg}\underline{\text{istleer?}}\underline{(}\underline{\mathtt{x}}\underline{)}\underline{)}\underline{)}$

**Aufgabe 3.2**
Es sei $\mathbb{B}$ ein abstrakter Datentyp für sogenannte 2-3-Bäume (kurz "Bäume") mit folgenden Komponenten:
– Eine Konstante $\bullet$ für den Baum, der nur aus einem Knoten (= 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;
– eine dreistellige Funktion $g$, die angewendet auf drei Bäume $b_1$, $b_2$, $b_3$ den Baum liefert, der unter dem (neuen) Wurzelknoten diese drei Teilbäume, in dieser Reihenfolge, hat.

a) Definieren Sie eine Signatur $\Sigma_{\mathbb{B}}$ zu $\mathbb{B}$ (analog zu $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{S}$) und geben Sie zu jedem 2-3-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 $x$ höchstens die Tiefe 1 hat. (*Hinweis:* Der Baum, der nur aus einem Knoten besteht, hat die Tiefe 0.)

c) Geben Sie eine PL-Formel über $\Sigma_{\mathbb{B}}$ an, die ausdrückt, dass $x$ mindestens die Tiefe 2 hat.

d) Zeigen Sie durch Induktion, dass Folgendes für alle $n \geq 1$ gilt: Wenn ein Term $t \in \mathcal{T}(\mathbb{B})$ in jeder Umgebung $I$ einen Baum mit $n$ Endknoten als Wert hat, so besteht $t$ aus mindestens $3n - 2$ Zeichen.

**Lösung 3.2**

a) Gemäß der Unterstreichungskonvention aus der Vorlesung erhält man $\Sigma_{\mathbb{B}} = \langle\{\}, \{\underline{\bullet}\}, \{\underline{\mathtt{f}}, \underline{\mathtt{g}}\}\rangle$.
Es gibt also keine Prädikatensymbole, aber das Konstantensymbol $\underline{\bullet}$, sowie das zweistellige Funktionssymbol $\underline{\mathtt{f}}$ und das dreistellige Funktionssymbol $\underline{\mathtt{g}}$.
Es gibt 15 verschiedene 2-3-Bäume mit höchstens 4 Endknoten.
Entsprechende Terme sind:
1 Endknoten: $\bullet$
2 Endknoten: $\mathtt{f}(\bullet,\bullet)$
3 Endknoten: $\mathtt{g}(\bullet,\bullet,\bullet)$, $\mathtt{f}(\bullet,\mathtt{f}(\bullet,\bullet))$, $\mathtt{f}(\mathtt{f}(\bullet,\bullet),\bullet)$
4 Endknoten: $\mathtt{g}(\mathtt{f}(\bullet,\bullet),\bullet,\bullet)$, $\mathtt{g}(\bullet,\mathtt{f}(\bullet,\bullet),\bullet)$, $\mathtt{g}(\bullet,\bullet,\mathtt{f}(\bullet,\bullet))$, $\mathtt{f}(\bullet,\mathtt{f}(\bullet,\mathtt{f}(\bullet,\bullet)))$, $\mathtt{f}(\mathtt{f}(\bullet,\bullet),\mathtt{f}(\bullet,\bullet))$, $\mathtt{f}(\bullet,\mathtt{f}(\mathtt{f}(\bullet,\bullet),\bullet))$, $\mathtt{f}(\mathtt{f}(\bullet,\mathtt{f}(\bullet,\bullet)),\bullet)$, $\mathtt{f}(\mathtt{f}(\mathtt{f}(\bullet,\bullet),\bullet),\bullet)$, $\mathtt{f}(\mathtt{g}(\bullet,\bullet,\bullet),\bullet)$, $\mathtt{f}(\bullet,\mathtt{g}(\bullet,\bullet,\bullet))$

b) $(\mathtt{x}{=}\bullet\vee(\mathtt{x}{=}\mathtt{f}(\bullet,\bullet)\vee\mathtt{x}{=}\mathtt{g}(\bullet,\bullet,\bullet)))$

c) Die einfächste Lösung besteht wohl darin, einfach den Booleschen Ausdruck von Aufgabe b) zu negieren: $\neg(\mathtt{x}{=}\bullet\vee(\mathtt{x}{=}\mathtt{f}(\bullet,\bullet)\vee\mathtt{x}{=}\mathtt{g}(\bullet,\bullet,\bullet)))$. (Beachte: Boolesche Ausdrücke sind spezielle PL-Formeln.)
Eine explizitere Beschreibung von '$x$ hat mindestens die Tiefe 2' leistet, z.B., folgende PL-Formel:
$(\exists\mathtt{y}\exists\mathtt{z}(\mathtt{x}{=}\mathtt{f}(\mathtt{y},\mathtt{z})\wedge\neg(\mathtt{y}{=}\bullet\wedge\mathtt{z}{=}\bullet))\vee\exists\mathtt{y}\exists\mathtt{z}\exists\mathtt{u}(\mathtt{x}{=}\mathtt{g}(\mathtt{y},\mathtt{z},\mathtt{u})\wedge\neg(\mathtt{y}{=}\bullet\wedge(\mathtt{z}{=}\bullet\wedge\mathtt{u}{=}\bullet))))$
Natürlich gibt es auch alternative Lösungen.

d) *Beobachtung:* Ein Term kann nur dann in allen Umgebungen einen Baum mit einer bestimmten Anzahl von Enknoten als Wert haben, wenn er kein Variablensymbol enthält. Wir können also im Folgenden davon ausgehen, dass alle Terme $t$, von denen hier die Rede ist, variablenfrei sind.
Wir schreiben $|t|$ für die Anzahl der Zeichen (Symbole) in $t$. (Beispiel: $|t| = 6$ für $t = \underline{f}(\bullet,\bullet)$.)

**Induktionsanfang:** Da jeder Term aus mindestens einem Symbol besteht und $3 \cdot 1 - 2 = 1$ gilt, folgt die zu beweisende Behauptung für $n = 1$ unmittelbar.

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

**Induktionsschritt:** Ein variablenfreier Term $t$, der einen Baum mit $n + 1$ (und daher mindestens 2) Endknoten als Wert hat, muss die Form $\mathtt{f}(t_1,t_2)$ oder $\mathtt{g}(t_1,t_2,t_3)$ haben.
Wir unterscheiden daher die folgenden beiden Fälle:

$t = \mathtt{f}(t_1,t_2)$**:** Für $i \in \{1, 2\}$ sei $n_i$ die Anzahl der Endknoten im von $t_i$ dargestellten Baum. Es gilt $n_1 + n_2 = n + 1$. Außerdem enthält $t$ 4 Symbole, die noch nicht in $t_1$ oder $t_2$ vorkommen. Daher gilt:

$$\begin{aligned}|t| &= 4 + |t_1| + |t_2| \\ &\geq 4 + 3n_1 - 2 + 3n_2 - 2 = 3(n_1 + n_2) = 3(n + 1) \\ &\geq 3(n + 1) - 2.\end{aligned}$$

$t = \mathtt{g}(t_1,t_2,t_3)$**:** Für $i \in \{1, 2, 3\}$ sei $n_i$ die Anzahl der Endknoten im von $t_i$ dargestellten Baum. Es gilt $n_1 + n_2 + n_3 = n + 1$. Außerdem enthält $t$ 5 Symbole, die noch nicht in $t_1$, $t_2$ oder $t_3$ vorkommen. Daher gilt:

$$\begin{aligned}|t| &= 5 + |t_1| + |t_2| + |t_3| \\ &\geq 5 + 3n_1 - 2 + 3n_2 - 2 + 3n_3 - 2 \\ &= 3(n_1 + n_2 + n_3) - 1 = 3(n + 1) - 1 \\ &\geq 3(n + 1) - 2.\end{aligned}$$

**Aufgabe 3.3**

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

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

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

Formulieren Sie eine entsprechende Bedingung (MAL′4) zur formalen Festlegung der Semantik. Informell lautet die Semantik: $\alpha$ wird in der Umgebung $I$ genau $I(v)$ mal ausgeführt. Formulieren Sie (MAL′4) direkt (induktiv) und nicht durch Rückführung auf die $\underline{\text{while}}$-Schleife.

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

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

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? Anders formuliert: Lassen sich alle partiell-berechenbaren Funktionen mit einem $AL'(\mathbb{N})$-Programm berechnen? (Begründen Sie Ihre Antwort.)

**Lösung 3.3**

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 $I(\underline{\text{z}}) = 3$.

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

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}} \underline{\leftarrow} \underline{\text{z}}\underline{*}\underline{(}\underline{1}\underline{+}\underline{1}\underline{)})\\
&\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}} \underline{\leftarrow} \underline{\text{z}}\underline{*}\underline{(}\underline{1}\underline{+}\underline{1}\underline{)})\\
&\qquad \text{wobei } I''(\underline{\text{y}}) = 1, \text{ und } I''(v) = \mathcal{M}_{AL}(I, \underline{\text{z}} \underline{\leftarrow} \underline{\text{z}}\underline{*}\underline{(}\underline{1}\underline{+}\underline{1}\underline{)})(v) \text{ für } v \neq \underline{\text{y}},\\
&\qquad \text{daher } I''(\underline{\text{z}}) = \mathcal{M}_{\mathcal{T}}(I, \underline{\text{z}}\underline{*}\underline{(}\underline{1}\underline{+}\underline{1}\underline{)}) = I(\underline{\text{z}}) \cdot \mathcal{M}_{\mathcal{T}}(I, \underline{(}\underline{1}\underline{+}\underline{1}\underline{)}) = 3 \cdot 2 = 6\\
\overset{MAL4'}{=}\ &\mathcal{M}_{AL}(I''', \underline{\text{loop}}\ \underline{\text{y}}\ \underline{\text{times}}\ \underline{\text{z}} \underline{\leftarrow} \underline{\text{z}}\underline{*}\underline{(}\underline{1}\underline{+}\underline{1}\underline{)})\\
&\qquad \text{wobei } I'''(\underline{\text{y}}) = 0, \text{ und } I'''(v) = \mathcal{M}_{AL}(I'', \underline{\text{z}} \underline{\leftarrow} \underline{\text{z}}\underline{*}\underline{(}\underline{1}\underline{+}\underline{1}\underline{)})(v) \text{ für } v \neq \underline{\text{y}}\\
&\qquad \text{daher } I'''(\underline{\text{z}}) = \mathcal{M}_{\mathcal{T}}(I'', \underline{\text{z}}\underline{*}\underline{(}\underline{1}\underline{+}\underline{1}\underline{)}) = I''(\underline{\text{z}}) \cdot \mathcal{M}_{\mathcal{T}}(I'', \underline{(}\underline{1}\underline{+}\underline{1}\underline{)}) = 6 \cdot 2 = 12\\
\overset{MAL4'}{=}\ &I''' \quad [\text{wegen } I'''(\underline{\text{y}}) = 0]
\end{aligned}$$

Für die insgesamt resultierende Umgebung $J$ gilt also: $J(\underline{\text{x}}) = I'(0) = 0$, $J(\underline{\text{y}}) = I'''(\underline{\text{y}}) = 0$, $J(\underline{\text{z}}) = I'''(\underline{\text{z}}) = 12$. 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. (Die sogenannte Ackermannfunktion ist ein konkretes Beispiel hierfür.)

**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.

a) Manche Elefanten fürchten sich vor jeder Maus, die sie sehen.

b) Aishas Vater besucht beide Eltern von Berta, aber nicht die Mutter von Chen.

c) Zu jedem Haus, in dem ein Kind wohnt, gibt es ein anderes Haus, in dem genau ein Erwachsener wohnt.

d) Mohan hat zwei Schwestern, die beide Ärztinnen sind.

Wenn Ihnen ein Satz mehrdeutig erscheint, so diskutieren Sie alternative Interpretationen.

**Lösung 3.4**

a) Signatur $\langle\{E, M, F, S\}, \{\}, \{\}\rangle$ mit folgender intendierter Bedeutung:
*Prädikatensymbole:*
$E(x)$ ... $x$ ist ein Elefant (einstellig)
$M(x)$ ... $x$ ist eine Maus (einstellig)
$F(x, y)$ ... $x$ fürchtet sich vor $y$ (zweistellig)
$S(x, y)$ ... $x$ sieht $y$ (zweistellig)

PL-Formel: $\exists x[E(x) \wedge \forall y(M(y) \wedge S(x, y) \supset F(x, y))]$

b) Signatur $\langle\{B\}, \{a, b, c\}, \{m, v\}\rangle$ mit folgender intendierter Bedeutung:
*Prädikatensymbol:*
$B(x, y)$ ... $x$ besucht $y$ (zweistellig)
*Funktionssymbole:*
$m(x)$ ... die Mutter von $x$ (einstellig)
$v(x)$ ... der Vater von $x$ (einstellig)
*Konstantensymbole:*
$a$ ... Aisha
$b$ ... Berta
$c$ ... Chen

PL-Formel: $B(v(a), m(b)) \wedge B(v(a), v(b)) \wedge \neg B(v(a), m(c))$.
*Beachte:* Es wäre zwar prinzipiell möglich, aber ziemlich ungeschickt Mutter und Vater nicht als (einstellige) Funktionssymbole, sondern, indirekt, über zweistellige Prädikate *ist_Mutter_von*$(x, y)$ bzw. *ist_Vater_von*$(x, y)$ auszudrücken. Wie in der Vorlesung diskutiert, sollte man grundsätzlich nacheindeutige Relationen (also funktionale Beziehungen) kompakter und expliziter mit Funktionssymbolen ausdrücken, wenn immer das möglich ist.

In jedem Fall ist es adäquat zu berücksichtigen, dass mit den Eltern einer Person deren Mutter und Vater gemeint sind. Die Einführung eines eigenen Prädikats für Eltern (*ist_Elternteil_von*$(x, y)$ oder ähnliches) ist inadäquat, da so die semantische Beziehung zwischen den Wörtern *Mutter*, *Vater* und *Eltern* unberücksichtigt bleibt.

c) Signatur $\langle\{H, K, E, W\}, \{\}, \{\}\rangle$ mit folgender intendierter Bedeutung:
*Prädikatensymbole:*
$H(x)$ ... $x$ ist ein Haus (einstellig)
$K(x)$ ... $x$ ist ein Kind (einstellig)
$E(x)$ ... $x$ ist ein Erwachsener (einstellig)
$W(x, y)$ ... $x$ wohnt in $y$ (zweistellig)

PL-Formel:
$\forall x[H(x) \wedge \exists y(K(y) \wedge W(y, x)) \supset \exists z(H(z) \wedge z \neq x \wedge \exists u(E(u) \wedge W(u, z) \wedge \forall v(E(v) \wedge W(v, z) \supset v = u)]$

d) Signatur $\langle\{A, S\}, \{m\}, \{\}\rangle$ mit folgender intendierter Bedeutung:
*Prädikatensymbole:*
$A(x)$ ... $x$ ist eine Ärztin (einstellig)
$S(x, y)$ ... $x$ ist eine Schwester von $y$ (zweistellig)
*Konstantensymbol:*
$m$ ... Mohan

Der Ausdruck ‘hat zwei Schwestern’ kann entweder als ‘hat mindestens zwei Schwestern’ (Lesart 1) oder als ‘hat genau zwei Schwestern’ (Lesart 2) verstanden werden.
PL-Formel für Lesart 1: $\exists x \exists y[x \neq y \wedge S(x,m) \wedge S(y,m) \wedge A(x) \wedge A(y)]$
PL-Formel für Lesart 2: $\exists x \exists y[x \neq y \wedge S(x,m) \wedge S(y,m) \wedge A(x) \wedge A(y) \wedge \forall z(S(z,m) \supset z = x \vee z = y)]$

**Aufgabe 3.5**

Spezifizieren Sie zu folgenden Formeln jeweils ein Modell $\mathcal{I}$ und ein Gegenbeispiel $\mathcal{J}$ über dem angegebenen Gegenstandsbereich $D$. Argumentieren Sie jeweils, warum $\mathcal{I}$ ein Modell und $\mathcal{J}$ ein Gegenbeispiel ist. Geben Sie außerdem für jede Formel an, welche Variablen dort frei bzw. gebunden vorkommen. (Beachten Sie die in der Vorlesung eingeführten Notationsvereinbarungen und Klammereinsparungsregeln.)

a) Über $D = Z$ (ganze Zahlen): $\exists y[P(x, f(y), c) \supset \forall x \neg P(x, c, f(x)) \vee \exists z P(y, x, z)]$

b) Über $D = \omega$ (natürliche Zahlen): $\exists x \forall y[Q(f(x)) \supset P(g(a, x), y)] \wedge \exists z(f(z) = g(x, y) \vee f(z) = a)$

c) Über $D = \{\underline{0}, \underline{1}\}^*$ (Binärstrings): $\forall x \exists y R(f(x, y), g(x)) \supset \exists x \forall y R(g(y), f(y, x))$

**Lösung 3.5**

Jede Interpretation, also jedes Modell und jedes Gegenbeispiel einer Formel, besteht aus 3 Bestandteilen: Dem Gegenstandsbereich $D$, der Signaturinterpretation $\Phi$ und der Variablenbelegung $\xi$. Da die Variablenbelegung nur für die *frei* vorkommenden Variablen relevant ist, ist es sinnvoll $\xi$ nur für letztere festzulegen.

a) Die Variablen $y$ und $z$ kommen nur gebunden vor; $x$ kommt (in $\forall x \neg P(x, c, f(x))$ zwei mal gebunden und zwei mal ungebunden vor. ($c$ ist keine Variable, sondern gemäß den Notationsvereinbarungen ein Konstantensymbol.)

   Die gesamte Formel ist eine existenziell quantifizierte Implikation. Man erhält daher ein Modell $\mathcal{I} = \langle Z, \Phi_{\mathcal{I}}, \xi_{\mathcal{I}} \rangle$, wenn man $\Phi_{\mathcal{I}}(P)(k, \ell, m) = \mathbf{f}$ für alle $k, \ell, m \in Z$ setzt. (Wenn die linke Seite einer Implikation falsch ist, so ist die Implikation wahr.) Alle anderen Bestandteile von $\mathcal{I}$ sind beliebig.

   Für ein Gegenbeispiel $\mathcal{J} = \langle Z, \Phi_{\mathcal{J}}, \xi_{\mathcal{J}} \rangle$ muss für wenigstens ein $y$ die linke Teilformel der Implikation wahr und die rechte Teilformel falsch werden. Dies erreicht man beispielsweise mit folgender Festlegung:

$$\Phi_{\mathcal{J}} = (P)(k, \ell, m) \begin{cases} \mathbf{f} & \text{falls } \ell = 0 \\ \mathbf{t} & \text{sonst,} \end{cases}$$

   $\Phi_{\mathcal{J}}(f)(n) = 1$, $\Phi_{\mathcal{J}}(c) = 1$, $\xi_{\mathcal{J}}(x) = 0$. Unter der Interpretation $\mathcal{J}$ sind die Formeln $\forall x \neg P(x, c, f(x))$ und $\exists z P(y, x, z)$ (für beliebiges $y$) beide falsch. Daher ist auch die Disjunktion auf der rechten Seite der Implikation falsch. Hingegen ist die linke Seite der Implikation ($P(x, f(y), c)$) wahr. Also ist die gesamte Formel unter $\mathcal{J}$ falsch.

b) Die Variablen $x$ und $y$ kommen im linken Konjunkt gebunden und im rechten Konjunkt frei vor; $z$ kommt nur gebunden vor. ($a$ ist keine Variable, sondern gemäß den Notationsvereinbarungen ein Konstantensymbol.)

   Ein Modell $\mathcal{I} = \langle \omega, \Phi_{\mathcal{I}}, \xi_{\mathcal{I}} \rangle$ erhält man beispielsweise wie folgt: Für alle $n \in \omega$, $\Phi_{\mathcal{I}}(Q)(n) = \mathbf{f}$, $\Phi_{\mathcal{I}}(f)(n) = 0$, $\Phi_{\mathcal{I}}(a) = 0$. ($\Phi_{\mathcal{I}}(P)$, $\Phi_{\mathcal{I}}(g)$, sowie $\xi_{\mathcal{I}}$ sind beliebig.) Da $Q(f(x))$ unter $\mathcal{I}$ immer falsch ist, ist die Implikation im linken Konjunkt der Formel wahr. Da $f(z) = a$ unter $\mathcal{I}$ immer wahr ist, ist auch das rechte Konjunkt und somit die gesamte Formel wahr.

   Für ein Gegenbeispiel $\mathcal{J} = \langle \omega, \Phi_{\mathcal{J}}, \xi_{\mathcal{J}} \rangle$ genügt es, für alle $m, n \in \omega$, $\Phi_{\mathcal{J}}(Q)(n) = \mathbf{t}$ und $\Phi_{\mathcal{J}}(P)(m, n) = \mathbf{f}$ zu setzen. Damit wird die Implikation, also das linke Konjunkt und somit die gesamte Formel falsch.

c) Die Variablen $x$ und $y$ kommen nur gebunden vor.

   Für ein Modell $\mathcal{I} = \langle \{\underline{0}, \underline{1}\}^*, \Phi_{\mathcal{I}}, \xi_{\mathcal{I}} \rangle$ genügt es für alle Binärstrings $\Phi_{\mathcal{I}}(R)(s, t) = \mathbf{f}$ oder, alternativ, $\Phi_{\mathcal{I}}(R)(s, t) = \mathbf{t}$ zu setzten. (Alle anderen Bestandteile von $\mathcal{I}$ sind beliebig.) In beiden Fällen wird die gesamte Formel (Implikation) wahr.

Für ein Gegenbeispiel $\mathcal{J} = \langle \{\underline{\mathtt{0}}, \underline{\mathtt{1}}\}^*, \Phi_{\mathcal{J}}, \xi_{\mathcal{J}} \rangle$ bestimmen wir $\Phi_{\mathcal{J}}(R)$, $\Phi_{\mathcal{J}}(g)$ und $\Phi_{\mathcal{J}}(f)$ wie folgt:
$\Phi_{\mathcal{J}}(R)(s,t) \iff$ '$s$ enthält $t$ als echten Teilstring,'
$\Phi_{\mathcal{J}}(f)(s,t) = \underline{\mathtt{0}}st$,
$\Phi_{\mathcal{J}}(g)(s) = s$.
Unter der Interpretation $\mathcal{J}$ ist die Formel falsch. Die linke Seite der Implikation ist wahr: Es gibt zu jedem Binärstring $s$ einen Binärstring $t$, sodass $t$ in $\underline{\mathtt{0}}st$ echt enthalten ist. Aber die rechte Seite der Implikation ist falsch: Es gibt keinen Binärstring $s$, sodass für beliebige $t$ der Binärstring $\underline{\mathtt{0}}st$ echt in $t$ enthalten ist.