Formale Grundlagen der Informatik II
5. Übungsblatt

TECHNISCHE UNIVERSITÄT DARMSTADT

Fachbereich Mathematik
Prof. Dr. Martin Ziegler
Davorin Lešnik, Ph.D.
Stéphane Le Roux, Ph.D.

Sommersemester 2013
01. 07. 2013

Gruppenübung

Aufgabe G13 (Modellierung)
Ein Meteorologe versucht die zeitliche Entwicklung des Wetters an einem bestimmten Ort mit folgender Signatur in FO zu beschreiben:

$$S = \{0, N, <, S, R\}.$$

0	Konstante für Starttag
N	1-stelliges Funktionssymbol für „nächster Tag“
$<$	2-stelliges Relationssymbol für die zeitliche Ordnung der Tage
S, R	1-stellige Relationssymbole für Sonne und Regen

Formalisieren Sie die folgenden Aussagen in FO(S):

1. Auf Regen folgt (irgendwann) Sonnenschein.
2. Jeden zweiten Tag scheint die Sonne.
3. Wenn an einem Tag die Sonne scheint, gibt es innerhalb drei Tagen wieder Regen.

Lösung: Wir geben eine mögliche Lösung an (offenbar sind 1.-3. nicht unbedingt eindeutig in ihrer Semantik!).

1. $\forall x \,.\, \big(Rx \to \exists y \,.\, \big(x < y \wedge Sy\big)\big)$
2. $\forall x \,.\, (Sx \vee SNx)$
3. $\forall x \,.\, (Sx \to (RNx \vee RNNx \vee RNNNx))$

Aufgabe G14 (Sortieren)
Betrachten Sie die Struktur $\mathcal{N} = (\mathbb{N}, <)$, wobei $<$ die übliche Ordnung ist. Geben Sie einen Algorithmus an, der für in subquadratisch vielen Schritten entscheidet, ob eine Folge von Elementen $a_1, a_2, \ldots, a_n \in \mathbb{N}$ paarweise verschieden ist.

Lösung: Für $\mathcal{N}$ kann man z.B. den folgenden Algorithmus verwenden:

$b_1, \ldots, b_n \leftarrow \text{HEAPSORT}(a_1, \ldots, a_n)$
for $i = 1 \to n - 1$ **do**
  if $b_i =^{\mathcal{N}} b_{i+1}$ **then**
    return false
  end if
end for
return true

Dieser Algorithmus benötigt n Vergleiche plus die Vergleiche die Heapsort verwendet, also insgesamt $O(n \log(n) + n) = O(n \log(n))$.

Aufgabe G15 (Wörter und Sprachen)
Wir wollen Sprachen über dem Alphabet $\Sigma := \{a, b\}$ mit Hilfe der Prädikatenlogik definieren. Wie im Skript, S. 3, definieren wir zu einem nichtleeren Wort $w = w_1 \dots w_n \in \Sigma^+$ eine *Wortstruktur*

$$\mathcal{W}(w) = \big(\{1, \dots, n\}, <, P_a, P_b\big)$$

wobei

$$P_a := \big\{i \leq n \;\big|\; w_i = a\big\} \quad \text{und} \quad P_b := \big\{i \leq n \;\big|\; w_i = b\big\}.$$

(Wir schließen das leere Wort aus, da es keine leeren Strukturen gibt.) Ein Satz $\varphi \in \mathrm{FO}(<, P_a, P_b)$ definiert dann die Sprache $L(\varphi) := \big\{w \in \Sigma^+ \;\big|\; \mathcal{W}(w) \models \varphi\big\}$.

(a) Welche Sprachen definieren die folgenden Formeln?
   i. $\forall x \,.\, \forall y \,.\, \big(x < y \to ((P_a x \to P_a y) \wedge (P_b y \to P_b x))\big)$
   ii. $\forall x \,.\, \forall y \,.\, \big((x < y \wedge P_a x \wedge P_a y) \to \exists z \,.\, \big(x < z \wedge z < y \wedge P_b z\big)\big)$

(b) Geben Sie zu den folgenden Sprachen Formeln an, welche sie definieren.
   i. $\mathtt{L}((\mathtt{a} + \mathtt{b})^* \mathtt{bb} (\mathtt{a} + \mathtt{b})^*)$
   ii. $\mathtt{L}\big((\mathtt{ab})^+\big)$

Lösung:

(a) Der erste Teil der ersten Formel besagt, dass rechts von einem a nur a stehen dürfen. Analog sagt der zweite Teil, dass links von einem b nur b stehen dürfen, also wird die Sprache $\mathtt{L}(\mathtt{b}^*(\mathtt{a} + \mathtt{b})\mathtt{a}^*)$ definiert. Die zweite Formel besagt, dass zwischen zwei a jeweils ein b auftauchen muss, also ist die definierte Sprache $\mathtt{L}((\mathtt{b} + \mathtt{ab})^*(\mathtt{a} + \mathtt{b})\mathtt{b}^*)$.

(b)
$$\exists x \,.\, \exists y \,.\, \big(x < y \wedge \neg \exists z \,.\, \big(x < z \wedge z < y\big) \wedge P_b x \wedge P_b y\big)$$

und
$$\forall x \,.\, \forall y \,.\, \big((x < y \wedge \neg \exists z \,.\, \big(x < z \wedge z < y\big)\big) \to (P_a x \leftrightarrow P_b y)\big) \wedge$$
$$\forall x \,.\, \big(\neg \exists y \,.\, \big(y < x\big) \to P_a x\big) \wedge \forall x \,.\, \big(\neg \exists y \,.\, \big(x < y\big) \to P_b x\big)$$

Aufgabe G16 (Modellierung)
(a) Geben Sie eine FO-Formel an, die besagt, dass die Trägermenge mindestens n Elemente enthält.

(b) Drinking principle von Raymond Smullyan: Betrachte eine nicht leere Kneipe. Stellen Sie mithilfe einer Formel in FO den folgenden Satz: In der Kneipe gibt es jemanden, sodass wenn er oder sie trinkt, dann trinken alle. Begründen Sie, warum Ihre Formel allgemeingültig ist.

Lösung:

(a) $\forall x_1, \dots, x_{n-1} \,.\, \exists y \,.\, \bigwedge_{1 \leq i \leq n-1} y \neq x_i$

(b) Der Satz kann so dargestellt werden, wobei Tx bedeutet, dass x trinkt:

$$\varphi = \exists x \,.\, \big(Tx \to \forall y \,.\, T(y)\big).$$

Entweder gilt $T(a)$ für alle a, oder nicht. Wenn es für alle a gilt, dann ist $\forall y \,.\, T(y)$ wahr und damit auch φ. Wenn es eine Ausnahme gibt, also eine a, so dass $T(a)$ nicht gilt, können wir für x dieses a wählen. Dann ist die Annahme in $T(a) \to \forall y \,.\, T(y)$ falsch und damit ist die ganze Aussage richtig.

Hausübung

Aufgabe H11 (Modellierung (Vergleiche **Aufgabe G13**)) (2 Punkte)
Formalisieren Sie die folgenden Aussagen in FO(S):

1. Regen dauert nie länger als drei Tage.
2. Innerhalb jeder Periode von vier Tagen regnet es an mindestens zwei Tagen.

Lösung: Wir geben eine mögliche Lösung an (offenbar sind 1.-5. nicht unbedingt eindeutig in ihrer Semantik!).

1. [1 P.] $\neg\exists x\,.\,(Rx \wedge RNx \wedge RNNx \wedge RNNNx)$
2. [1 P.] $\forall x\,.\,\bigvee_{i,j<4,i\neq j}(RN^i x \wedge RN^j x)$

Aufgabe H12 (Sortieren (Vergleiche **Aufgabe G14**)) (3 Punkte)
Betrachten Sie die Signatur $(<)$ und eine Struktur $\mathscr{A} = (A, <^{\mathscr{A}})$ in dieser Signatur.

(a) Beschreiben Sie einen Algorithmus, der bei der Eingabe einer Folge von Elementen $a_1, a_2, \ldots, a_n$ aus A entscheidet, ob die a_i paarweise verschieden sind. Wieviele Schritte benötigt Ihr Algorithmus?

(b) Welche Eigenschaften von $\mathscr{N}$ haben Sie in **G14** benutzt und können Sie einen FO($<$) Satz φ angeben, so dass Ihr Algorithmus für alle Strukturen $(A, <^{\mathscr{A}}) \models \varphi$ funktioniert?

Lösung:

(a) [1,5 P.] Wir vergleichen jedes a_i mit jedem a_j mit $j \neq i$. Dies kann z.B. mit folgendem Algorithmus geschehen.

```
for i = 1 → n do
  for j = i + 1 → n do
    if a_i =^A a_j then
      return false
    end if
  end for
end for
return true
```

Wobei $=^{\mathscr{A}}$ die Gleichheit in $\mathscr{A}$ bezeichnet.

Insgesamt werden dafür $(n-1)+(n-2)+\cdots+1+0 = \frac{1}{2}n\cdot(n-1) = O(n^2)$ viele Vergleiche benötigt.

(b) [1,5 P.] Der Algorithmus funktioniert auf jeder Struktur, auf der Heapsort funktioniert. Das sind alle Strukturen, auf denen $<$ einen totale Ordnung beschreibt. Damit kann für φ z.B. der folgende Satz gewählt werden

$$\forall x\,.\,\forall y\,.\,\big((x<y \vee x=y \vee x>y) \wedge \neg(x<y \wedge y<x) \wedge \neg(x<x)\big) \wedge$$
$$\forall x\,.\,\forall y\,.\,\forall z\,.\,\big(x<y \wedge y<z \rightarrow x<z\big)\,.$$

Aufgabe H13 (Wörter und Sprachen (Vergleiche **Aufgabe G15**)) (2 Punkte)
Wir definieren die Menge der *$*$-freien regulären Ausdrücke* induktiv durch

- $\emptyset$ und jedes Element von Σ sind $*$-freie reguläre Ausdrücke;
- sind α und β $*$-freie reguläre Ausdrücke, so auch $\alpha\beta$, $\alpha+\beta$ und $\sim\alpha$.

Die Semantik eines solchen Ausdrucks ist wie für reguläre Ausdrücke definiert, wobei die Operation $\sim$ für die Komplementierung steht: $L(\sim\alpha) := \Sigma^* \setminus L(\alpha)$. Konstruieren Sie (induktiv) zu einem gegebenen $*$-freien regulären Ausdruck α eine Formel $\varphi_\alpha(x, y)$, so dass

$$\mathcal{W}(w_1 \dots w_n) \models \varphi_\alpha(i,k) \iff 1 \le i \le k \le n \text{ und } w_i w_{i+1} \cdots w_k \in L(\alpha).$$

Bemerkung: Man kann zeigen, dass die $*$-freien regulären Ausdrücke genau die Sprachen beschreiben, die man mit Prädikatenlogik definieren kann. Weiterhin gibt es reguläre Ausdrücke, die Sprachen beschreiben, die von keinem $*$-freien regulären Ausdruck beschrieben werden. Reguläre Ausdrücke können also mehr Sprachen beschreiben als Logik erster Stufe. Allerdings gibt es eine Erweiterung der Logik erster Stufe, die sogenannte *monadische Logik zweiter Stufe*, mit der man genau die Sprachen definieren kann, die auch von regulären Ausdrücken beschrieben werden.

Lösung: 2 P. Wir definieren zunächst die Formel ψ induktiv, die in der Definition von φ benutzt wird, weil ψ_α allgemeingültig ist, wenn $\emptyset \in L(\alpha)$, und sonst nicht erfüllbar:

- $\psi_\emptyset := \forall x\,.\,(x = x)$
- $\psi_a := \psi_b := \neg\psi_\emptyset$
- $\psi_{\alpha\beta} := \psi_\alpha \wedge \psi_\beta$
- $\psi_{\alpha+\beta} := \psi_\alpha \vee \psi_\beta$
- $\psi_{\sim\alpha} := \neg\psi_\alpha$

Für $\alpha = \emptyset$, $\varphi_\alpha(x,y) := \exists z\,.\,\neg(z = z)$. Für $\alpha = l \in \Sigma$, $\varphi_l(x,y) := (x = y) \wedge P_l x$. Für andere $*$-freien regulären Ausdrücke α wird $\varphi_\alpha(x,y)$ induktiv definiert durch:

$$\begin{aligned}
\varphi_{\alpha\beta}(x,y) &:= \big(\varphi_\alpha(x,y) \wedge \psi_\beta\big) \vee \big(\varphi_\beta(x,y) \wedge \psi_\alpha\big) \vee \\
&\quad \exists z\,.\,\exists t\,.\,\Big(\varphi_\alpha(x,z) \wedge \varphi_\beta(t,y) \wedge z < t \wedge \neg\exists u\,.\,(z < u \wedge u < t)\Big) \\
\varphi_{\alpha+\beta}(x,y) &:= \varphi_\alpha(x,y) \vee \varphi_\beta(x,y) \\
\varphi_{\sim\alpha}(x,y) &:= x \le y \wedge \neg\varphi_\alpha(x,y)
\end{aligned}$$

($x \le y$ ist eine Abkürzung für $x < y \vee x = y$.)

Aufgabe H14 (Unendliche Erfüllbarkeit) (3 Punkte)
Betrachte FO-Formel in der Signatur $\{f\}$ an, wobei f ein Symbol für 1-stellige Funktionen ist.

(a) Geben Sie eine FO-Formel in der Signatur $\{f\}$ an, die von einem Modell erfüllt wird genau dann, wenn die Interpretation von f injektiv ist.

(b) Geben Sie eine FO-Formel in der Signatur $\{f\}$ an, die von einem Modell erfüllt wird genau dann, wenn die Interpretation von f surjektiv ist.

(c) Geben Sie eine FO-Formel in der Signatur $\{f\}$ an, die erfüllbar ist, aber nur unendliche Modelle hat.

Lösung:

(a) 1 P. $\varphi_{\text{inj}} := \forall x\,.\,\forall y\,.\,\big(f(x) = f(y) \to x = y\big)$

(b) 1 P. $\varphi_{\text{surj}} := \forall y\,.\,\exists x\,.\,\big(f(x) = y\big)$

(c) 1 P. $\varphi_{\text{inj}} \wedge \neg\varphi_{\text{surj}}$ oder $(\neg\varphi_{\text{inj}}) \wedge \varphi_{\text{surj}}$ oder $\varphi_{\text{inj}} \oplus \varphi_{\text{surj}}$. Wenn A eine endliche menge ist, ist nämlich eine Funktion $f : A \to A$ injektiv genau dann, wenn sie surjektiv ist, genau dann, wenn sie bijektiv ist. Die Funktion $f : \mathbb{N} \to \mathbb{N}$ sodass $f(n) := 2n$ für alle $n \in \mathbb{N}$ ist aber injektiv und nicht surjektiv.

Minitest

Aufgabe M10 (Pränex Normalform)
Sei P ein beliebiges einstelliges Prädikat. Betrachte die folgenden Formeln in Pränex Normalform.

1. $\exists x . \exists y . \big(Pz \wedge \neg Px \wedge Py\big)$
2. $\exists y . \forall x . \big((Pz \wedge Py) \vee (\neg Px \wedge \neg Pz)\big)$
3. $\exists x . (Px \wedge \neg Pz)$
4. $\forall x . \exists y . \big(Px \wedge \neg Py\big)$

Zu welchen Formeln unten sind die obigen Formeln äquivalent?

( ) $\exists x . \big(\neg(Px \to Pz) \wedge \neg \exists y . \big(Py \wedge Pz\big)\big)$
( ) $\neg \forall y . \big((Pz \wedge Py) \to \forall x . Px\big)$
( ) $\big(\forall x . \neg(Px \vee Pz)\big) \vee \exists y . \big(Pz \wedge Py\big)$

Lösung:

(3.) $\exists x . \big(\neg(Px \to Pz) \wedge \neg \exists y . \big(Py \wedge Pz\big)\big)$
(1.) $\neg \forall y . \big((Pz \wedge Py) \to \forall x . Px\big)$
(2.) $\big(\forall x . \neg(Px \vee Pz)\big) \vee \exists y . \big(Pz \wedge Py\big)$

Aufgabe M11 (Allgemeingültigkeit)
Sei $P(x)$ ein beliebiges einstelliges Prädikat. Welche der folgenden Sätze in der Signatur (P) sind allgemeingültig?

☐ $\forall x . \exists y . x = y$
☐ $\exists x . \forall y . x = y$
☐ $\forall x . \big(P(x) \vee \exists y . \neg P(y)\big)$

Lösung:

☒ $\forall x . \exists y . x = y$
☐ $\exists x . \forall y . x = y$
☒ $\forall x . \big(P(x) \vee \exists y . \neg P(y)\big)$

Begründung: Der erste Satz ist allgemeingültig, weil y nach x gewählt wird und man damit $y := x$ setzen kann. Der zweite Satz ist nicht allgemeingültig, weil er z.B. der folgenden Struktur $(\{a, b\})$ nicht erfüllt wird (warum?). Der dritte ist allgemeingültig, weil er äquivalent zu $\forall x . P(x) \vee \neg \forall y . P(y)$. Er ist damit eine Instanz des tertium non datur, also von $\varphi \vee \neg\varphi$.