Formale Grundlagen der Informatik II
4. Übungsblatt

TECHNISCHE UNIVERSITÄT DARMSTADT

Fachbereich Mathematik **SS 2011**
Prof. Dr. Martin Ziegler **22.06.11**
Alexander Kreuzer
Carsten Rösnick

Minitest Lösung

a) Sei $P(x)$ ein beliebiges einstelliges Prädikat. Welche der folgende Formeln in der Signatur (P) ist eine syntaktisch korrekte FO-Formel?

- ☒ $\forall x \, \forall y \, P(x) \wedge P(y)$
- ☒ $\forall x \, \exists x P(x)$
- ☐ $P(x) \, \forall x$
- ☐ $\forall P \, \forall x \, \forall y \, (P(x) \leftrightarrow P(y))$

Begründung: Der dritte Satz ist syntaktisch nicht korrekt, weil der Quantor an der falschen Stelle steht. Der vierte Satz ist falsch, weil $\forall P$, also quantifizieren über Prädikate, in FO nicht möglich ist.

b) Welche der folgenden Sätze in der Signatur (P) ist allgemeingültig.

- ☒ $\forall x \, \exists y \; x = y$
- ☐ $\exists x \, \forall y \; x = y$
- ☒ $\forall x \, P(x) \vee \exists y \, \neg P(y)$.

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 $\varphi \vee \neg\varphi$.

Gruppenübung

Aufgabe G1 (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 $\mathrm{FO}(S)$:

1) Auf Regen folgt Sonnenschein.

2) Jeden zweiten Tag scheint die Sonne.

3) Wenn an einem Tag die Sonne scheint, gibt es innerhalb drei Tagen wieder Regen.

4) Regen dauert nie länger als drei Tage.

5) Innerhalb einer Periode von vier Tagen regnet es an mindestens zwei Tagen.

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

1) $\forall x\,(Rx \to \exists y\,(x < y \wedge Sy))$

2) $\forall x\,(Sx \vee SNx)$

3) $\forall x\,(Sx \to (RNx \vee RNNx \vee RNNNx))$

4) $\neg\exists x\,(Rx \wedge RNx \wedge RNNx \wedge RNNNx)$

5) $\forall x \bigvee_{i,j<4, i\neq j} (RN^i x \wedge RN^j x)$

Aufgabe G2

Betrachten Sie die Signatur $(<)$ und eine Struktur $\mathcal{A} = (A, <)$ in dieser Signatur.

(i) 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 Vergleiche mit $=$ oder $<$ benötigt Ihr Algorithmus.

(ii) Betrachten Sie nun die Struktur $\mathcal{N} = (\mathbb{N}, <)$. Geben Sie einen Algorithmus an, der für $\mathcal{N}$ schneller als für allgemeine Strukturen $\mathcal{A}$ entscheidet, ob eine Folge von Elementen $a_1, a_2, \ldots, a_n \in \mathbb{N}$ paarweise verschieden ist.

Welche Eigenschaften von $\mathcal{N}$ haben Sie benutzt und können Sie einen Satz φ angeben, so dass Ihr Algorithmus für alle Strukturen $\mathcal{A} \models \varphi$ funktioniert?

Hinweis: Erinnern Sie sich an Heapsort oder Mergesort und nutzen Sie die Eigenschaft, dass jeder dieser Algorithmen eine Liste in $\mathrm{O}(n\ \log(n))$ Zeit und damit insbesondere mit $\mathrm{O}(n\,log(n))$ Vergleichen sortiert.

Lösungsskizze:

(i) Wir vergleichen jedes a_i mit jedem a_j mit $j \neq i$. Dies kann z.B. mit folgendem Algorithmus geschehen.

for $i = 1 \to n$ **do**
  for $j = i + 1 \to n$ **do**
    if $a_i =^{\mathcal{A}} a_j$ **then**
      return false
    end if
  end for
end for
return true

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

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

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

$b_1, \ldots, b_n \leftarrow$ 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 $\mathrm{O}(n\ \log(n) + n) = \mathrm{O}(n\ \log(n))$.
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

$$\forall x \forall y (x < y \vee x = y \vee x > y) \wedge \forall x \forall y \forall z (x < y \wedge y < z \to x < z).$$

Aufgabe G3
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 $w = a_1 \dots a_n \in \Sigma^+$ eine *Wortstruktur*

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

wobei

$$P_a := \{i \leq n \mid a_i = a\} \quad \text{und} \quad P_b := \{i \leq n \mid a_i = b\}\,.$$

(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 $L(\varphi) := \{w \in \Sigma^+ \mid \mathcal{W}(w) \models \varphi\}$.

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

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

(c) **Zusatzaufgabe:** 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

$$\mathcal{W}(a_1 \dots a_n) \models \varphi_\alpha(i, k) \quad \Longleftrightarrow \quad 1 \leq i \leq k \leq n \text{ und } a_i a_{i+1} \cdots a_k \in L(\alpha)\,.$$

   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.

Lösungsskizze:

(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 $L(b^*(a+b)a^*)$ definiert. Die zweite Formel besagt, dass zwischen zwei a jeweils ein b auftauchen muss, also ist die definierte Sprache $L((b+ab)^*(a+b)b^*)$.

(b)

$$\exists x \exists y [x < y \wedge \neg \exists z (x < z \wedge z < y) \wedge P_b x \wedge P_b y]$$

und

$$\forall x \forall y [(x < y \wedge \neg \exists z (x < z \wedge z < y)) \rightarrow (P_a x \leftrightarrow P_b y)] \wedge$$
$$\forall x (\neg \exists y (y < x) \rightarrow P_a x) \wedge \forall x (\neg \exists y (x < y) \rightarrow P_b x)$$

(c) 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:

$$\begin{aligned} \varphi_{\alpha\beta}(x, y) &:= \exists z [x \leq z \wedge z \leq y \wedge \varphi_\alpha(x, z) \wedge \varphi_\beta(z, y)] \\ \varphi_{\alpha+\beta}(x, y) &:= \varphi_\alpha(x, y) \vee \varphi_\beta(x, y) \\ \varphi_{\sim\alpha}(x, y) &:= x \leq y \wedge \neg \varphi_\alpha(x, y) \end{aligned}$$

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

Hausübung

Aufgabe H1

(a) Geben Sie eine FO-Formel an, die besagt, dass die Trägermenge genau n Elemente enthält.

(b) Geben Sie eine FO-Formel an, die erfüllbar ist, aber nur unendliche Modelle hat.
Hinweis: Betrachten Sie die Signatur $(<)$.

(c) Begründen Sie, warum die folgende Formel wahr ist.

$$\varphi = \exists x \, (\, Px \rightarrow \forall y P(y) \,).$$

Lösungsskizze:

(a) Die Trägermenge enthält mindestens n Elemente:

$$\varphi = \exists x_1 \ldots \exists x_n \bigwedge_{i \neq j} (x_i \neq x_j)$$

Die Trägermenge enthält höchstens n Elemente:

$$\psi = \forall x_1 \ldots \forall x_{n+1} \bigvee_{i \neq j} (x_i = x_j).$$

Die Trägermenge enthält genau n Elemente:

$$\varphi \wedge \psi.$$

(b) Zum Beispiel:

$$\forall x \exists y \, (x < y) \wedge \forall x \, \neg (x < x) \wedge \forall x \forall y \forall z \, (x < y \wedge y < z \rightarrow x < z).$$

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

Aufgabe H2 (Monoide) (7 Punkte)

Betrachten Sie die Signatur $(*, e)$, wobei $*$ eine 2-stellige Funktion und e eine Konstante ist.

(a) Ein Struktur $\mathcal{A}$ für diese Signatur heißt Monoid, wenn $*$ assoziativ und e ein neutrales Element ist für $*$, siehe Skript FGdI1 1.1.19.

Geben Sie einen Satz φ an, so dass $\mathcal{A}$ ein Monoid ist genau dann wenn $\mathcal{A} \models \varphi$.

(b) Wenn es zu jedem Element von $\mathcal{A}$ ein Inverses gibt, dann kann $\mathcal{A}$ zu einer Gruppe erweitert werden. Geben Sie einen Satz φ an, so dass $\mathcal{A} \models \varphi$ genau dann wenn $\mathcal{A}$ zu einer Gruppe erweitert werden kann.

(c) Betrachten Sie die folgenden Monoide:

1) $\mathcal{A}_1 = (\mathbb{Z}, +, 0)$
2) $\mathcal{A}_2 = (\mathbb{N}, +, 0)$
3) $\mathcal{A}_3 = (\mathbb{N}, \max, 0)$, wobei $\max(x, y)$ das Maximum von x und y bezeichnet.
4) $\mathcal{A}_4 = (\Sigma, \cdot, \varepsilon)$, wobei $\Sigma = \{a, b\}$. Das ist der Wortmonoid, siehe Skript FGdI1 1.1.21.

Geben Sie Sätze $\varphi_1, \varphi_2, \varphi_3, \varphi_4$, so dass für jedes $i \in \{1, 2, 3, 4\}$

$$\mathcal{A}_i \vDash \varphi_i \qquad \text{und für } j \neq i \quad \mathcal{A}_j \nvDash \varphi_i.$$

D.h. mit den Sätzen φ_i können die Strukturen unterschieden werden.

(d) Geben Sie eine Struktur $\mathcal{A}$ für die Signatur an, die *kein* Monoid ist.

Lösungsskizze:

(a) Assoziativität wird durch $\varphi_A :\equiv \forall x \forall y \forall z (x * y) * z = x * (y * z)$ ausgedrückt. Das Eigenschaft, dass e das neutrale Element ist, wird z.B. durch $\varphi_e :\equiv \forall x (x * e = x \wedge e * x = x)$ beschreiben. Damit ist

$$\begin{aligned} \varphi &:\equiv \varphi_A \wedge \varphi_e \\ &\equiv \forall x \forall y \forall z (x * y) * z = x * (y * z) \wedge \forall x (x * e = x \wedge e * x = x) \end{aligned}$$

eine mögliche Lösung.

(b) $\varphi :\equiv \forall x \exists y (x * y = e \wedge y * x = e)$

(c) 1) Für φ_1 kann der Satz φ aus b) gewählt werden, da sich weder $\mathcal{A}_2$, $\mathcal{A}_3$, $\mathcal{A}_4$ zu einer Gruppe erweitern lassen.

2) Für φ_2 wählen wir $\neg\varphi_1 \wedge \neg\varphi_3 \wedge \neg\varphi_4$, wobei φ_3 und φ_4 später bestimmt werden.

3) $\varphi_3 :\equiv \forall x (x * x = x)$, weil nur $\max$ idempotent ist, d.h. nur für $\max$ gilt $\max(x, x) = x$ und nicht für $+$ oder $\cdot$.

4) $\varphi_4 :\equiv \neg\forall x \forall y (x * y = y * x)$. Dieser Satz drückt aus, dass der Monoid nicht kommutativ ist. Der Monoid $\mathcal{A}_4$ ist der einzige Monoid in der Liste, der nicht kommutativ ist. (Z.B. $a \cdot b \neq b \cdot a$)

(d) Z.B. $(\mathbb{Z}, \max, 0)$ ist kein Monoid, weil aus $\max(0, -1) \neq -1$ folgt, dass 0 nicht das neutrale Element ist.