Humboldt-Universität zu Berlin
Lehrstuhl Logik in der Informatik
Prof. Dr. Nicole Schweikardt

5. Dezember 2019

Logik in der Informatik

Wintersemester 2019/2020

Übungsblatt 8

Abgabe: *bis 17. Dezember 2019, 11.15 Uhr (vor der Vorlesung oder im Briefkasten zwischen den Räumen 3.401 und 3.402 im Johann von Neumann-Haus (Rudower Chaussee 25))*

Aufgabe 1: (25 Punkte)

Betrachten Sie die Kinodatenbank $\mathcal{D}$ aus der Vorlesung (Skript, Kapitel 3.1).

(a) Geben Sie für die folgenden Anfragen jeweils eine $\mathsf{FO}[\sigma_{\mathsf{KINO}}]$-Formel φ und ein Variablentupel $(x_1, \dots, x_n)$ mit $\text{frei}(\varphi) \subseteq \{x_1, \dots, x_n\}$ an, die die Anfrage beschreiben. Berechnen Sie jeweils auch die Relation $[\![\varphi(x_1, \dots, x_n)]\!]^{\mathcal{D}}$.

- **(i)** Geben Sie alle Schauspieler aus, die in mindestens einem Film mitspielen, in dem George Clooney mitspielt.
- **(ii)** Geben Sie alle Kinos und deren Adresse aus, in denen um '20:00' ein Film beginnt.
- **(iii)** Geben Sie die Titel aller Filme aus, die in genau einem Kino laufen.

(b) Geben Sie umgangssprachlich an, welche Anfragen durch die Formeln φ_1, φ_2 und φ_3 beschrieben werden.

(i) $\varphi_1(x) := \exists x_Z\, R_{Prog}(x, \text{'Alien'}, x_Z)$

(ii) $\varphi_2(x_1, x_2) := \exists x_A \exists x_T \Big(R_{Kino}(x_1, x_A, x_2, x_T) \ \wedge\ \exists x_R \exists x_S\ R_{Film}(x_1, x_R, x_S)\Big)$

(iii)
$$\varphi_3(x_1, x_2, x_3, x_4) := \Big(\ \exists x_T\, R_{Kino}(x_1, x_2, x_3, x_T) \wedge \exists x_{Z_1}\Big(R_{Prog}(x_1, x_4, x_{Z_1}) \wedge \forall x_K \forall x_{Z_2} (R_{Prog}(x_K, x_4, x_{Z_2}) \ \rightarrow\ x_K{=}x_1)\Big)\Big)$$

Aufgabe 2: (30 Punkte)

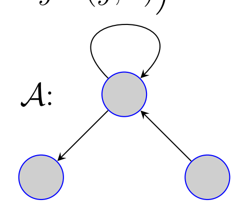
Seien $\sigma := \{E, g, c\}$ und $\sigma' := \{E\}$ Signaturen. Hierbei ist E ein 2-stelliges Relationssymbol, g ein 1-stelliges Funktionssymbol und c ein Konstantensymbol.

(a) **Achtung:** Dieser Aufgabenteil ist bis zum Abgabetermin durch die Beantwortung eines Quiz in Moodle **von jeder Teilnehmerin/jedem Teilnehmer einzeln** abzugeben.

Betrachten Sie jede der folgenden $\mathsf{FO}[\sigma]$-Formeln φ, bestimmen Sie jeweils die freien und die gebundenen Vorkommen von Variablen in φ, und entscheiden Sie, ob φ ein $\mathsf{FO}[\sigma]$-Satz ist.

- **(i)** $g(x) = g(c)$
- **(ii)** $\exists z \Big(E(x, z) \ \wedge\ E(z, y)\Big)$
- **(iii)** $\forall x \forall y \Big(\exists z\, g(z) = x \ \rightarrow\ \exists z\, g(z) = y\Big)$
- **(iv)** $\Big(\forall z \exists x\, E(x, c) \ \wedge\ \forall y\, E(y, z)\Big)$

(b) Betrachten Sie die σ'-Struktur $\mathcal{A} = (A, E^{\mathcal{A}})$, die durch den Graphen in der nebenstehenden Abbildung repräsentiert wird. Geben Sie einen $\mathsf{FO}[\sigma']$-Satz φ an, der die Struktur eindeutig beschreibt. D.h. es soll für alle σ'-Strukturen $\mathcal{B}$ gelten:



$$\mathcal{A} \cong \mathcal{B} \quad \Longleftrightarrow \quad \mathcal{B} \models \varphi$$

(c) Geben Sie für die $\mathsf{FO}[\sigma']$-Formel

$$\varphi(x) \quad := \quad \forall y \Big(E(x,y) \rightarrow \big(\exists z \, (E(x,z) \wedge E(y,z)) \big) \Big)$$

eine σ'-Struktur $\mathcal{A}$, deren Universum aus höchstens 4 Elementen besteht, und zwei σ'-Interpretationen $\mathcal{I}_1 = (\mathcal{A}, \beta_1)$ und $\mathcal{I}_2 = (\mathcal{A}, \beta_2)$ an, so dass $\mathcal{I}_1 \models \varphi$ und $\mathcal{I}_2 \models \neg\varphi$ gilt.

Aufgabe 3: (20 Punkte)

(a) Beweisen Sie das Koinzidenzlemma für Terme (Satz 3.27) per Induktion über den Aufbau von Termen.

(b) Beweisen Sie das Koinzidenzlemma für Formeln der Logik erster Stufe (Satz 3.28) per Induktion über den Aufbau von Formeln.

Aufgabe 4: (25 Punkte)

Lesen Sie Kapitel 6 aus dem Buch „Learn Prolog Now!“.

Achtung: *Die Bearbeitung der Teilaufgaben (b) und (c) ist unter Beachtung der bekannten Abgabehinweise über Moodle abzugeben!*

(a) Zeichnen Sie den Suchbaum für die folgende Anfrage:

```
?- append(X, Y, [1,2] ).
```

(b) *Binärbäume* seien definiert, wie in der dritten Prolog-Übungsstunde (vgl. Folie zum Spiegeln von Binärbäumen auf der Website zur Prolog-Übung). Beispielsweise wird der linke dort abgebildete Binärbaum $\mathcal{B}$ repräsentiert durch den folgenden Prolog-Term:

B := `tree(tree(leaf(1),tree(leaf(2),leaf(3))),leaf(4))`

Schreiben Sie ein Prädikat `label/2`, so dass die Anfrage `?- label(B, X).` für eine Repräsentation `B` eines Binärbaums und einen Prolog-Term `X` genau dann erfüllt ist, wenn `X` die Beschriftung eines Blattes von `B` ist.

Für unseren Baum $\mathcal{B}$ soll beispielsweise die Anfrage

```
?- label(B, X).
```

die Antworten

```
X = 1;      X = 2;       X = 3;       X = 4.
```

liefern.

(c) Schreiben Sie ein Prädikat `labels/2`, so dass die Anfrage `?- labels(B, Y).` für eine Repräsentation `B` eines Binärbaums und eine Liste `Y` von Prolog-Termen genau dann erfüllt ist, wenn `Y` eine Auflistung der Beschriftungen aller Blätter von `B` ist, und zwar in der Reihenfolge vom am weitesten links zum am weitesten rechts stehenden Blatt.

Für unseren Baum $\mathcal{B}$ soll beispielsweise die Anfrage

```
?- labels(B, Y).
```

die Antwort

```
Y = [1, 2, 3, 4].
```

liefern.

Hinweis: Benutzen Sie gegebenenfalls das Prädikat `append/3`.