

Logik in der Informatik

Wintersemester 2020/2021

Übungsblatt 8

Abgabe: bis 18. Januar 2021, 13.00 Uhr via Moodle

Um unseren Tutor*innen die Arbeit etwas zu erleichtern, notieren Sie bitte auf dem ersten Blatt jeder schriftlich erstellten Abgabedatei oben rechts eingerahmt folgende Daten:

<Vorname Nachname>, <Matrikelnr.>, <Moodle-Loginname>

Aufgabe 1:

(Moodle-Quiz)

Absolvieren Sie das Quiz 8 auf der Moodle-Plattform.

Aufgabe 2:

(Präsenzaufgabe)

Betrachten Sie die Kinodatenbank \mathcal{D} aus der Vorlesung.

- (a) Geben Sie für die folgenden Anfragen jeweils eine FO[σ_{KINO}]-Formel φ und ein Variablen-tupel (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 $\llbracket \varphi(x_1, \dots, x_n) \rrbracket^{\mathcal{D}}$.
- (i) Geben Sie alle Kombinationen aus Name eines Kinos und Uhrzeit aus, die einer Vorstellung des Films Gravity entsprechen.
 - (ii) Geben Sie alle Paare von Namen von Kinos aus, die sich im gleichen Stadtteil befinden.
 - (iii) Geben Sie die Namen aller Kinos aus, in denen mindestens ein Film gezeigt wird, in dem exakt zwei verschiedene Regisseure Regie geführt haben.
- (b) Geben Sie umgangssprachlich an, welche Anfragen durch die Formeln φ_1 , φ_2 und φ_3 beschrieben werden.

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

(ii) $\varphi_2 := \exists x_{S_1} \exists x_T (R_{Kino}(x_1, x_2, x_{S_1}, x_T) \wedge \exists x_R \exists x_{S_2} R_{Film}(x_1, x_R, x_{S_2}))$

(iii) $\varphi_3 := \left(\exists x_K R_{Prog}(x_K, x_2, x_3) \wedge \exists x_{S_1} (R_{Film}(x_2, x_1, x_{S_1}) \wedge \forall x_F \forall x_{S_2} (R_{Film}(x_F, x_1, x_{S_2}) \rightarrow x_F = x_2)) \right)$

Aufgabe 3:

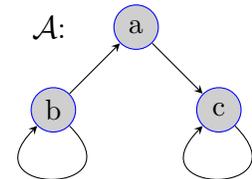
(40 Punkte)

- (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.

Sei $\sigma := \{E\}$ eine Signatur mit dem 2-stelligen Relationssymbol E .

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

$$\mathcal{B} \models \varphi \iff \mathcal{B} \cong \mathcal{A}$$



Erläutern Sie Ihren FO[σ]-Satz φ .

- (d) Geben Sie für die FO[σ]-Formel

$$\varphi(x) := \forall y \exists z \left((\neg y = z \wedge E(y, z)) \wedge (E(z, x) \rightarrow E(x, y)) \right)$$

eine σ -Struktur \mathcal{A} , deren Universum aus höchstens 4 Elementen besteht, und zwei Interpretationen $\mathcal{I}_1 = (\mathcal{A}, \beta_1)$ und $\mathcal{I}_2 = (\mathcal{A}, \beta_2)$ an, so dass gilt: $\mathcal{I}_1 \models \varphi$ und $\mathcal{I}_2 \not\models \varphi$.

Aufgabe 4:

(20 Punkte)

Lesen Sie Kapitel 10 aus dem Buch „Learn Prolog Now!“.

Achtung: Fertigen Sie Ihre Lösung für Aufgabenteil (a) auf einem Zettel an. Die Lösung der Aufgabenteile (b), (c) und (d) muss unter Beachtung der bekannten Abgabehinweise für Prolog-Code (in einer Datei für diese drei Aufgabenteile zusammen) in einem extra-Abgabefach bei Moodle eingereicht werden!

- (a) Gegeben sei das folgende Prolog-Programm:

```
1 a(X, Y) :- b(X, Y).           5 b(X, Y) :- c(X), c(Y).
2 a(1, 1).                     6 c(2).
3 b(X, X) :- c(X).             7 c(3).
4 b(X, Y) :- c(X), !, c(Y).
```

Zeichnen Sie einen Suchbaum für die folgende Anfrage: `?- a(X, Y).`

- (b) Schreiben Sie in `blatt8.pl` ein Prädikat `not_member/2`, so dass `not_member(X, L)` für einen Term X und eine Liste L genau dann erfüllt ist, wenn X mit *keinem* Element von L unifiziert werden kann. Verwenden Sie dabei abgesehen vom Cut und dem in SWI-Prolog vordefinierten Prädikat `fail/0` keine weiteren Prädikate, und insbesondere nicht `\=/2`.
- (c) Führen Sie in `blatt8.pl` einen neuen Operator `<=>` für die Biimplikation \leftrightarrow ein, der den gleichen Typ und die gleiche Präzedenz wie der in `a1.pl` definierte Operator `=>` hat.

(d) Implementieren Sie in `blatt8.pl`, analog zu Beispiel 2.54 im Vorlesungsskript, Schritt 1 des Tseitin-Verfahrens. D.h., schreiben Sie ein Prädikat `tseitin/2`, so dass die Anfrage `tseitin(F, L)` für eine aussagenlogische Formel `F` eine Liste `L` aussagenlogischer Formeln ausgibt, die die folgenden Eigenschaften hat:

- Die Konjunktion der Formeln in der Liste `L` ist erfüllbarkeitsäquivalent zu `F`.
- Die Liste `L` enthält für jede Teilformel von `F` (abgesehen von Literalen) genau eine Formel.
- In jeder Formel aus `L` kommen höchstens 3 verschiedene Aussagensymbole vor.

Beispielsweise sollte Prolog auf die Anfrage:

```
tseitin((p => ~q) \\/ (~ (p /\ q) /\ r), L).
```

wie folgt antworten:

```
L = [x1, x1<=>x2\/x3, x2<=> (p=> ~q), x3<=>x4/\r, x4<=> ~x5, x5<=>p/\q].
```

Hierbei sind die konkrete Wahl der neuen Aussagensymbole sowie die Reihenfolge der Formeln in der Repräsentation der Menge unwesentlich.

Hinweise:

- Benutzen Sie zur Erzeugung neuer Aussagensymbole das in SWI-Prolog eingebaute Prädikat `gensym/2`. Das Prädikat `gensym/2` instantiiert bei dem Aufruf `gensym(x, A)` die Variable `A` mit einem Atom der Form `xn`, wobei eine Zahl `n` so gewählt wird, dass das Atom `xn` in diesem Lauf von SWI-Prolog noch nicht verwendet wurde.
- Benutzen Sie den in Teilaufgabe (c) definierten Operator `<=>`.
- Nutzen Sie ggf. Cut oder Negation. Führen Sie bei Bedarf Hilfsprädikate ein.