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

8. Dezember 2016

Logik in der Informatik

Wintersemester 2016/17

Übungsblatt 8

Abgabe: *3. Januar 2017*

Aufgabe 1: **(25 Punkte)**

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

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

(i) Gib die Telefonnummern der Kinos aus, die um 20:15 Uhr oder um 20:30 Uhr eine Vorstellung haben.

(ii) Gib alle Regisseure aus, die in einem Film Regie geführt und auch selbst mitgespielt haben, der gerade nicht im Kino läuft.

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

(i) $\varphi_1(x_{\mathrm{T}}) \quad :=$

$$\Big(\exists x_{\mathrm{Z}}\, R_{Prog}(\text{‘Babylon’}, x_{\mathrm{T}}, x_{\mathrm{Z}}) \quad \wedge \quad \neg \exists x_{\mathrm{Z}}\, R_{Prog}(\text{‘Casablanca’}, x_{\mathrm{T}}, x_{\mathrm{Z}}) \Big)$$

(ii) $\varphi_2(x_{\mathrm{K}}, x_{\mathrm{Z}}) \quad :=$

$$\Big(\exists x_{\mathrm{T}}\, R_{Prog}(x_{\mathrm{K}}, x_T, x_Z) \quad \wedge \quad \forall y_{\mathrm{T}} \forall y_{\mathrm{Z}} \Big(R_{Prog}(x_{\mathrm{K}}, y_T, y_Z) \quad \rightarrow$$

$$\neg \exists x_{\mathrm{S}} \exists x_{\mathrm{R}} \Big(R_{Filme}(y_{\mathrm{T}}, x_{\mathrm{R}}, x_{\mathrm{S}}) \wedge (x_{\mathrm{S}} = \text{‘George Clooney’} \vee x_{\mathrm{R}} = \text{‘George Clooney’}) \Big) \Big) \Big)$$

Aufgabe 2: **(25 Punkte)**

(a) Sei die Signatur $\sigma := \{E, f\}$. Hierbei ist E ein zweistelliges Relationssymbol und f ein einstelliges Funktionssymbol. Welche der folgenden Aussagen sind korrekt, welche nicht? (Sie brauchen Ihre Antwort nicht zu begründen.)

(i) $\forall x \exists y\ E(x,y) \quad \equiv \quad \exists y \forall x\ E(x,y)$

(ii) $\forall x \forall y \Big(f(x) = y \ \rightarrow \ E(y,x)\Big) \quad \equiv \quad \forall y \forall x \Big(\neg E(y,x) \ \rightarrow \ \neg f(x) = y\Big)$

(iii) $\forall x \forall y \neg f(x) = y \quad \equiv \quad \neg \forall x \exists y \Big(\big(x = y \ \vee \ E(x,y)\big) \ \rightarrow \ \exists z \big(z = y \ \vee \ E(z,y)\big)\Big)$

(b) Welche der folgenden Aussagen sind für alle Signaturen σ und alle $\mathsf{FO}[\sigma]$-Formeln φ und ψ korrekt, welche nicht? (Sie brauchen Ihre Antwort nicht zu begründen.)

(i) $(\forall x\, \varphi \ \wedge \ \forall x\, \psi) \ \equiv \ \forall x\, (\varphi \wedge \psi)$

(ii) $(\forall x\, \varphi \ \vee \ \forall x\, \psi) \ \equiv \ \forall x\, (\varphi \vee \psi)$

(iii) $(\exists x\, \varphi \ \wedge \ \exists x\, \psi) \ \equiv \ \exists x\, (\varphi \wedge \psi)$

(iv) $(\exists x\, \varphi \ \vee \ \exists x\, \psi) \ \equiv \ \exists x\, (\varphi \vee \psi)$

(c) Beweisen Sie, dass Ihre Antworten zu **(i)** und **(iii)** in Aufgabenteil **(b)** korrekt sind.

Aufgabe 3: **(25 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.

— auf der nächsten Seite geht's weiter —

Aufgabe 4: (25 Punkte)

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

Erstellen Sie eine Datei `blatt08.pl`, die mit der Zeile `:- ensure_loaded([al]).` beginnt. Laden Sie in das selbe Verzeichnis die Datei `al.pl` von der URL `http://www2.informatik.hu-berlin.de/logik/lehre/WS16-17/Logik/downloads/al.pl`.

Achtung: *Geben Sie die von Ihnen erstellte Datei* `blatt08.pl` *mit Ihren Lösungen der Teilaufgaben* **(b)**, **(c)** *und* **(d)** *über Moodle ab!* ***Außerdem gilt:*** *Lösungsansätze, die von SWI-Prolog nicht geladen werden können, werden nicht bewertet!*

(a) Gegeben sei das folgende Prolog-Programm:

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

Zeichnen Sie einen Suchbaum für die folgende Anfrage:

```
?- a(X, Y).
```

(b) Schreiben Sie in `blatt08.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`.

Hinweis: Das Prädikat `fail/0` schlägt immer fehl. D.h., das Ziel `fail` in einer Anfrage löst immer das Backtracking aus.

(c) Führen Sie in `blatt08.pl` einen neuen Operator `<=>` für die Biimplikation $\leftrightarrow$ ein, der den gleichen Typ und die gleiche Präzedenz wie der in `al.pl` definierte Operator `=>` hat.

(d) Implementieren Sie in `blatt08.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 die Anfrage

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

zu der Ausgabe

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

führen. Hierbei ist 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 `x`n, wobei eine Zahl n so gewählt wird, dass das Atom `x`n 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.