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

26. Januar 2024

Logik in der Informatik

Wintersemester 2023/2024

Übungsblatt 13

Abgabe: *bis 5. Februar 2024, 13.⁰⁰ Uhr*

Aufgabe 1: **(Moodle-Quiz)**

Absolvieren Sie das Quiz 13 auf der Moodle-Plattform.

Aufgabe 2: **(Präsenzaufgabe)**

(a) Sei R ein 2-stelliges Relationssymbol, f ein 1-stelliges Funktionssymbol und seien c und d Konstantensymbole.

Im Folgenden ist für jedes $i \in \{1,2\}$ eine Signatur σ_i und ein $\mathsf{FO}[\sigma_i]$-Satz φ_i gegeben.

(1) Sei $\sigma_1 := \{R, f, c\}$ und sei φ_1 der folgende $\mathsf{FO}[\sigma_1]$-Satz:

$$\forall x \forall y \Big(\big(R(x,y) \ \rightarrow \ y{=}f(x) \big) \ \wedge \ \big(y{=}f(x) \ \rightarrow \ R(x,y) \big) \Big)$$

(2) Sei $\sigma_2 := \{R, c, d\}$ und sei φ_2 der folgende $\mathsf{FO}[\sigma_2]$-Satz:

$$\exists x \exists y \big(R(x,d) \ \wedge \ R(c,y) \big) \ \wedge \ \forall x \forall y \big(R(x,y) \ \rightarrow \ \neg\, x{=}y \big)$$

Geben Sie für jedes $i \in \{1,2\}$ eine σ_i-Herbrandstruktur $\mathcal{A}_i$ und eine σ_i-Herbrandstruktur $\mathcal{B}_i$ an, sodass gilt:

$$\mathcal{A}_i \models \varphi_i \qquad \text{und} \qquad \mathcal{B}_i \not\models \varphi_i.$$

Begründen Sie jeweils, warum $\mathcal{A}_i \models \varphi_i$ bzw. $\mathcal{B}_i \not\models \varphi_i$ gilt.

(b) Sei $\sigma := \{R, f\}$, wobei R ein 2-stelliges Relationssymbol und f ein 1-stelliges Funktionssymbol ist. Transformieren Sie die $\mathsf{FO}[\sigma]$-Formel

$$\forall x \, \neg \big(\neg f(x){=}y \ \ \vee \ \ \forall y \, R(x,y) \big)$$

in einen zu φ erfüllbarkeitsäquivalenten gleichheitsfreien $\mathsf{FO}[\hat{\sigma}]$-Satz $\hat{\varphi}$ in Skolemform. Gehen Sie dabei vor wie im Beweis von Satz 4.52 im Vorlesungsskript. Geben Sie insbesondere auch die Signatur $\hat{\sigma}$ sowie die Formeln an, die nach jedem der Schritte 1, 2 und 3 des Beweises entstehen.

— auf der nächsten Seite geht's weiter —

(c) Betrachten Sie das folgende Logik-Programm Π.[1]

```
verwandt(luke, lea).
gute_seite(han).
mag(lea, han).
mag(luke, X) :- verwandt(luke, X).
mag(luke, X) :- gute_seite(X).
verfolgt(han, lea).
verfolgt(darth_vader, X) :- mag(luke, X).
```

- **(i)** Geben Sie einen Beweisbaum für den Term `verfolgt(darth_vader, lea)` aus Π an.
- **(ii)** Geben Sie die Bedeutung $\mathcal{B}(\Pi)$ des Logikprogramms Π an.

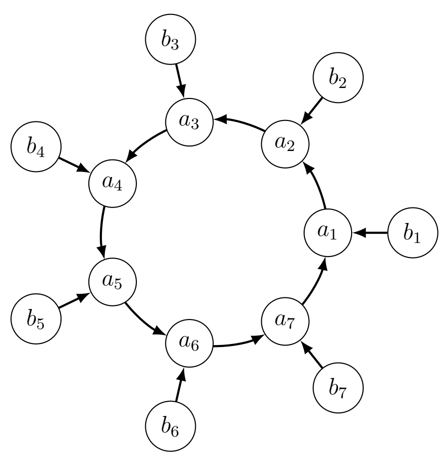
Aufgabe 3: (40 Punkte)

(a) Sei $\sigma := \{E\}$ die Signatur, die aus dem 2-stelligen Relationssymbol E besteht.

Definition: Für eine σ-Struktur $\mathcal{A} = (A, E^{\mathcal{A}})$ und eine natürliche Zahl $n \geq 2$ sagen wir, dass $\mathcal{A}$ *eine Krone der Länge n besitzt*, wenn es Elemente $a_1, \dots, a_n, b_1, \dots, b_n \in A$ mit $|\{a_1, \dots, a_n, b_1, \dots, b_n\}| = 2n$ gibt, sodass die Relation $E^{\mathcal{A}}$ die folgenden Kanten enthält:

- (a_i, a_{i+1}) für alle $i \in \{1, \dots, n-1\}$ und (a_n, a_1) und
- (b_i, a_i) für alle $i \in \{1, \dots, n\}$.

Eine Krone der Länge 7 sieht zum Beispiel wie folgt aus:



- **(i)** Geben Sie für jede natürliche Zahl $n \geq 2$ einen $\mathsf{FO}[\sigma]$-Satz φ_n an, sodass für jede σ-Struktur $\mathcal{A}$ gilt: $\mathcal{A} \models \varphi_n \iff \mathcal{A}$ enthält eine Krone der Länge n.
- **(ii)** Geben Sie eine Menge Ψ von $\mathsf{FO}[\sigma]$-Sätzen an, die die Klasse aller σ-Strukturen $\mathcal{A}$ axiomatisiert, die für alle $n \geq 2$ keine Krone der Länge n besitzen.
- **(iii)** Verwenden Sie den Endlichkeitssatz der Logik erster Stufe, um Folgendes zu beweisen: Die Klasse aller σ-Strukturen $\mathcal{A}$, die eine Krone der Länge ≥ 2 besitzen, ist *nicht* erststufig axiomatisierbar. Präzise: Zeigen Sie, dass es keine Menge Φ von $\mathsf{FO}[\sigma]$-Sätzen gibt, sodass für jede σ-Struktur $\mathcal{A}$ gilt:

$$\mathcal{A} \models \Phi \iff \text{Es gibt ein } n \in \mathbb{N} \text{ mit } n \geq 2, \text{ sodass } \mathcal{A} \text{ eine Krone der Länge } n \text{ besitzt.}$$

Hinweis: Orientieren Sie sich am Beweis von Satz 4.35 im Vorlesungsskript.

— auf der nächsten Seite geht's weiter —

[1] Sie haben es bereits als Prologprogramm auf Blatt 1 kennengelernt.

(b) Sei $\sigma := \{R, f_0, f_1, c\}$, wobei c ein Konstantensymbol, R ein 2-stelliges Relationssymbol und f_0, f_1 zwei 1-stellige Funktionssymbole sind.

Beweisen Sie folgende Aussagen aus Korollar 4.41 aus dem Vorlesungsskript:

(i) Das Unerfüllbarkeitsproblem für $\mathsf{FO}[\sigma]$ ist nicht entscheidbar.

(ii) Das Erfüllbarkeitsproblem für $\mathsf{FO}[\sigma]$ ist nicht semi-entscheidbar.

(c) Sie $\sigma' = \{f, c\}$ die Signatur mit dem 1-stelligen Funktionssymbol f und dem Konstantensymbol c. Beantworten Sie die folgenden zwei Fragen:

(i) Wie sieht das Universum einer konkreten σ'-Herbrandstruktur aus?

(ii) Wie viele σ'-Herbrandstrukturen gibt es?

Begründen Sie Ihre Antwort.

Aufgabe 4: (20 Punkte)

Lesen Sie Kapitel 8 aus dem Buch „Learn Prolog Now!“.

Achtung: *Die Bearbeitung der Aufgabe ist unter Beachtung der bekannten Abgabehinweise über Moodle abzugeben!*

(a) Implementieren Sie ein Prädikat `sat/1`, so dass eine Anfrage

```
?- sat(F).
```

für eine aussagenlogische Formel `F` genau dann erfolgreich ist, wenn `F` erfüllbar ist.

Hinweise: Ihr Prädikat soll zu der Formel `F` zuerst eine *erfüllbarkeitsäquivalente* 3-*KNF* konstruieren, und anschließend deren Erfüllbarkeit mit dem DPLL-Algorithmus testen. Es macht nichts, wenn Ihr Prädikat für eine erfüllbare Formel mehrfach

```
true.
```

ausgibt.

(b) Für Vergleiche von SAT-Solvern werden 3-KNF oft im sogenannten DIMACS-Format angegeben.[2] Implementieren Sie ein Prädikat `sat_dimacs/1`, welches als Argument den Namen einer Datei erhält, so dass beispielsweise die Anfrage

```
?- sat_dimacs('knf.cnf').
```

genau dann erfolgreich ist, wenn die in der Datei `knf.cnf` repräsentierte 3-KNF erfüllbar ist. Ist diese nicht erfüllbar, soll das Ergebnis `false.` sein.

Sie können in Ihrer Implementation davon ausgehen, dass die aufgerufene Datei im aktuellen Verzeichnis existiert und dem DIMACS-Standard entspricht.

Sie können zur Lösung dieser Aufgabe alle Prolog-Module verwenden, die Sie unter

`https://hu.berlin/prolog`

vorfinden. Dies gilt insbesondere für die Module `tseitin.pl` und `dpll.pl`.[3] Dort finden Sie auch Beispieldateien im DIMACS-Format.

---

[2] So waren auch auf der SAT Competition 2023 [Link: `http://www.satcompetition.org/`] die 2011 festgelegten Regeln gültig, welche auch für uns das DIMACS-Format definieren sollen (vgl. `http://www.satcompetition.org/2011/format-benchmarks2011.html`).

[3] Verfügbar ab 29.01.24 ca. 20:00 Uhr.