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

24. Januar 2025

Logik in der Informatik

Wintersemester 2024/2025

Übungsblatt 13

Abgabe: *bis 3. Februar 2025, 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]

```
gesund(tomatensaft).
enthaelt(bloody_mary, wodka).
enthaelt(bloody_mary, tomatensaft).
trinkt(isabell, X) :- enthaelt(X, wodka).
trinkt(isabell, X) :- gesund(X).
leer(prosecco).
leer(X) :- trinkt(_, X).
```

(i) Geben Sie einen Beweisbaum für den Term `leer(bloody_mary)` aus Π an.

(ii) Geben Sie die Bedeutung $\mathcal{B}(\Pi)$ des Logikprogramms Π an.

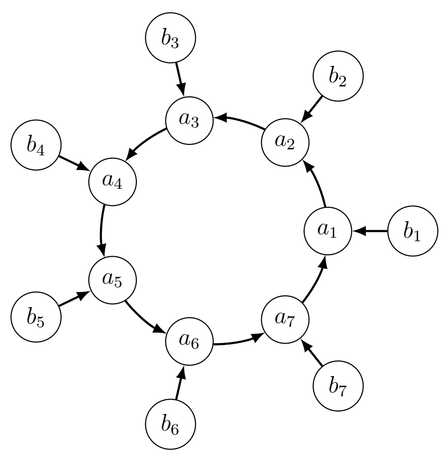
Aufgabe 3: (40 Punkte)

(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 $a_1, \ldots, a_n, b_1, \ldots, b_n \in A$ mit $|\{a_1, \ldots, a_n, b_1, \ldots, b_n\}| = 2n$ gibt, sodass die Relation $E^{\mathcal{A}}$ die folgenden Kanten enthält:

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

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:

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

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 Aussage aus Korollar 4.41 aus dem Vorlesungsskript:

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

(c) Sei $\sigma_2 = \{R, c, d\}$ die Signatur aus Aufgabe **2(a)** mit dem 2-stelligen Relationssymbol R und den Konstantensymbolen c und d. Beantworten Sie die folgenden zwei Fragen:

- **(i)** Wie sieht das Universum einer konkreten σ_2-Herbrandstruktur aus?
- **(ii)** Wie viele σ_2-Herbrandstrukturen gibt es?

Begründen Sie Ihre Antworten.

Aufgabe 4: (20 Punkte)

Lesen Sie Kapitel 7 aus dem Buch „Learn Prolog Now!“.
Achtung: *Die Bearbeitung der Aufgabe ist unter Beachtung der bekannten Abgabehinweise über Moodle abzugeben! Analog zu früheren Blättern finden Sie die benötigten Dateien auf der Seite zur Prolog-Übung. Machen Sie sich auch mit den neuen Prolog-Modulen* ***`unit_propagation`*** *und* ***`pure_literal`*** *vertraut.* [2]

(a) Importieren Sie im Modul `dpll` Ihrer Abgabe `blatt13.pl` die Prädikate aus den Prolog-Modulen, die Sie für die Lösung der folgenden Teilaufgabe benötigen.

(b) Wir kodieren Klauselmengen, wie gewohnt, als Listen von Listen von Literalen. Implementieren Sie das Prädikat `dpll/1`, sodass eine Anfrage

```
?- dpll(KM).
```

für eine Klauselmenge `KM` genau dann erfolgreich ist, wenn die Klauselmenge erfüllbar ist.

Beispielsweise sollte die Anfrage für die Klauselmenge

```
KM = [[x1,~x5,~x6,x7], [~x1,x2,~x5], [~x1,~x2,~x3,~x5,~x6],
      [x1,x2,~x4,x7], [~x4,~x6,~x7], [x3,~x5,x7], [x3,~x4,~x5],
      [x5,~x6], [x5,x4,~x8], [x1,x3,x5,x6,x7], [~x7,x8],
      [~x6,~x7,~x8]]
```

erfolgreich sein. Es macht hierbei nichts, wenn die Antwort `true.` durch das Backtracking mehrfach ausgegeben werden kann. Für die Klauselmenge

```
KM = [[~r,t,w], [~r,~s,~w], [~r,~t], [~q,s,t], [~q,r,~s],
      [r,s,w], [r,~t,~w], [q,u], [s,~u,~w], [q,w], [q,~s,~u]]
```

sollte dieselbe Anfrage jedoch scheitern.
Hinweise: Implementieren Sie dazu den *DPLL-Algorithmus*, wie er auf den Seiten 93/94 des Skripts beschrieben ist. Definieren Sie geeignete Hilfsprädikate. Nutzen Sie insbesondere die bereits auf Blatt 11 und 12 implementierten Vereinfachungsheuristiken *Unit Propagation* und *Pure Literal Rule*, die Sie aus den Modulen des entsprechenden Namens importieren können. Die Streichung von Klauseln, die Obermengen von anderen Klauseln sind, müssen Sie nicht implementieren.

— auf der nächsten Seite geht's weiter —

[2] Verfügbar spätestens ab 27.01.25 20:00 Uhr.

(c) Gegeben sei die kontextfreie Grammatik $G = (\Sigma, V, S, P)$ mit den Terminalsymbolen $\Sigma := \{\texttt{if}, \texttt{then}, \texttt{else}, \texttt{e1}, \texttt{e2}, \texttt{s1}, \texttt{s2}\}$, den Nichtterminalsymbolen $V := \{\texttt{stmt}, \texttt{expr}\}$, dem Startsymbol $S :=$ `stmt`, und den Produktionen $P :=$

$$\{\ \texttt{stmt} \longrightarrow \texttt{if expr then stmt}, \quad \texttt{stmt} \longrightarrow \texttt{if expr then stmt else stmt}, \\ \texttt{stmt} \longrightarrow \texttt{s1}, \quad \texttt{stmt} \longrightarrow \texttt{s2}, \quad \texttt{expr} \longrightarrow \texttt{e1}, \quad \texttt{expr} \longrightarrow \texttt{e2}\ \}$$

Bilden Sie für die kontextfreie Grammatik G eine *Definite Clause Grammar (DCG)*, sodass die Anfrage

```
?- stmt(X, []).
```

genau dann erfüllt wird, wenn `X` eine Liste von Terminalsymbolen aus Σ ist, die einem Wort der durch G beschriebenen Sprache entspricht. Dies gilt beispielsweise für die Liste

```
X = [ if, e1, then, if, e2, then, s1, else, s2] .
```

Fügen Sie Ihre Definite Clause Grammar der Datei `blatt13.pl` hinzu.