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

28. Januar 2022

Logik in der Informatik

Wintersemester 2021/2022

Übungsblatt 13

Abgabe: *bis 7. Februar 2022, 13.⁰⁰ Uhr*

Aufgabe 1: (Moodle-Quiz)

Absolvieren Sie das Quiz 13 auf der Moodle-Plattform.

Aufgabe 2: (Präsenzaufgabe)

Vier Wochen nach Ausbruch des *Zombie-Virus*: Alice sitzt in der Kleinstadt *Hamster City* fest, die inzwischen nicht mehr als besonders ruhig gelten kann. Um endlich ein Serum gegen das Virus entwickeln zu können beschließt Alice, den sogenannten *Patient Zero* zu finden, d.h. denjenigen, der als erstes mit dem Zombie-Virus infiziert wurde. Natürlich kann es sich nur um jemanden handeln, der sich zur Zeit des Viren-Ausbruchs im Forschungslabor befand.

Bei der Suche nach Patient Zero „hilft“ Alice der Zombie-Hamster Charly: Charlys Blutprobe ermöglicht es Alice, die Kette der Infektionen vom Forschungslabor, über verschiedene Wirte hinweg, bis zu Charly nachzuvollziehen. Dabei geht Alice davon aus, dass Menschen, die andere beißen, Zombies sind. Außerdem gilt als gesichert, dass Hamster zwar das Virus tragen können, es aber nicht weitergeben.

Alice hat ihre Erkenntnisse in einem Logikprogramm $\Pi \in \mathsf{LP}$ formuliert:

```
% Zombies
hamster(bianca).
hamster(charly).
hamster(herby).
mensch(edward).
mensch(john).
mensch(seymour).
% Wer hat wen gebissen?
beisst(bianca, charly).
beisst(edward, herby).
beisst(john, bianca).
beisst(john, seymour).
beisst(herby, charly).
beisst(seymour, charly).
% Wer war im Labor?
im_labor(edward).
im_labor(john).
% Wer hat wen infiziert?
infiziert(X, Y) :-
    mensch(X),
    beisst(X, Y).
infiziert(X, Y) :-
    mensch(X),
    beisst(X, Z),
    infiziert(Z, Y).
% Patient Zero war im Labor
% und hat Charly infiziert
patient0(X) :-
    im_labor(X),
    infiziert(X, charly).
```

Hierbei repräsentiert `beisst(X, Y)` die Aussage, dass `Y` von `X` gebissen wurde, und `infiziert(X, Y)` bedeutet, dass `Y` von `X` (eventuell über andere Wirte) infiziert wurde.

(a) Geben Sie einen Beweisbaum für den Term `patient0(john)` aus Π an.

(b) Geben Sie eine Ableitung des Terms `patient0(john)` aus Π an.

Aufgabe 3: **(40 Punkte)**

(a) Sei σ eine Signatur, die ein einstelliges Relationssymbol P enthält. Seien x und y zwei verschiedene Variablen. Leiten Sie die Sequenz

$$\{\ P(x),\ \forall x \forall y\, x{=}y\ \}\ \vdash\ \forall y\, P(y)$$

im Sequenzenkalkül $\mathfrak{K}_S$ ab.

(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 die folgende Aussage aus Korollar 4.41 aus dem Vorlesungsskript:

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

(c) Sei R ein 2-stelliges Relationssymbol, f ein 1-stelliges Funktionssymbol und sei c ein Konstantensymbol.

Sei $\sigma := \{R, f, c\}$ und sei φ der folgende $\mathsf{FO}[\sigma]$-Satz:

$$\forall x \forall y \forall z \left(R(x, f(x))\ \wedge\ \Big(\big(R(x,y) \wedge R(y,z) \big)\ \rightarrow\ R(x,z) \Big) \right)$$

Geben Sie eine σ-Herbrandstruktur $\mathcal{A}$ und eine σ-Herbrandstruktur $\mathcal{B}$ an, so dass gilt:

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

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

(d) Sei $\sigma := \{f, c\}$, wobei f ein 1-stelliges Funktionssymbol ist und c ein Konstantensymbol.

Zeigen Sie, dass Satz 4.46 aus dem Vorlesungsskript im Allgemeinen *nicht* für $\mathsf{FO}[\sigma]$-Sätze in Skolemform gilt, die *nicht* gleichheitsfrei sind.

Geben Sie dazu einen $\mathsf{FO}[\sigma]$-Satz φ in Skolemform an, so dass gilt:

$$\varphi \text{ ist erfüllbar,} \qquad \text{aber} \qquad \varphi \text{ besitzt kein Herbrand-Modell.}$$

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

$$\varphi(z)\ :=\ \exists x\ \big(\forall y\ R(x,y)\ \vee\ f(x) = z \big)$$

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 —

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.[1] 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 akutellen Verzeichnis existiert und dem DIMACS-Standard entspricht.

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

`https://t1p.de/proLog2122`

vorfinden. Dies gilt insbesondere für die Module `tseitin.pl` und `dpll.pl`.[2] Dort finden Sie auch Beispieldateien im DIMACS-Format.

[1] So waren auch auf der SAT Competition 2021 die 2011 festgelegten Regeln gültig, welche auch für uns das DIMACS-Format definieren sollen (vgl. `http://www.satcompetition.org/2011/format-benchmarks2011.html`).

[2] Verfügbar ab 31.01.22 ca.: 20:00 Uhr.