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

8. Februar 2023

Logik in der Informatik

Wintersemester 2022/2023

Übungsblatt 14

Abgabe: *Präsenzblatt für die letzte Woche.*

Aufgabe 1: (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 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 2: **(Präsenzaufgabe)**

In dieser Aufgabe bezeichnet AL' die Menge aller aussagenlogischen Formeln, die *keine Aussagensymbole* enthalten. Wir repräsentieren Formeln $\varphi \in \mathsf{AL}'$ wie folgt durch Terme $\mathtt{t}_\varphi \in \mathsf{T}_{\mathsf{LP}}$ der Logik-Programmierung:

- *Atomare Formeln:*
$$\mathtt{t}_{\mathbf{0}} := \mathtt{0} \quad \text{und} \quad \mathtt{t}_{\mathbf{1}} := \mathtt{1}$$
- *Rekursive Regeln:* Für Formeln $\varphi, \psi \in \mathsf{AL}'$ ist
$$\begin{aligned} \mathtt{t}_{\neg\varphi} &:= \mathtt{n(t}_\varphi\mathtt{)}, \\ \mathtt{t}_{(\varphi\vee\psi)} &:= \mathtt{o(t}_\varphi\mathtt{,\ t}_\psi\mathtt{)} \\ \mathtt{t}_{(\varphi\wedge\psi)} &:= \mathtt{u(t}_\varphi\mathtt{,\ t}_\psi\mathtt{)}. \end{aligned}$$

Beispielsweise wird die Formel $\big((\mathbf{1} \wedge \mathbf{0}) \vee \neg\mathbf{0}\big)$ durch den folgenden Term repräsentiert:

$$\mathtt{o(u(1,\ 0),\ n(0))}$$

Betrachten Sie das folgende Logik-Programm Π:

```
true(1).
false(0).
true(n(F))      :- false(F).
false(n(F))     :- true(F).
true(o(F, G))    :- true(F).
true(o(F, G))    :- true(G).
false(o(F, G))   :- false(F), false(G).
true(u(F, G))   :- true(F), true(G).
false(u(F, G)) :- false(F).
false(u(F, G)) :- false(G).
```

(a) Geben Sie einen Beweisbaum für den Term

$$\mathtt{true(o(u(1,\ 0),\ n(0)))}$$

aus Π an.

(b) Ist der folgende Term aus Π ableitbar? Begründen Sie Ihre Antwort.

$$\mathtt{false(n(u(1,0)))}$$

(c) Geben Sie die Bedeutung $\mathcal{B}(\Pi)$ von Π an.

(d) Schreiben Sie ein Logik-Programm Π', so dass gilt:

$$\mathcal{B}(\Pi') \quad = \quad \Big\{ \mathtt{dual(t}_\varphi\mathtt{,\ t}_{\tilde{\varphi}}\mathtt{)} :\ \varphi \in \mathsf{AL}' \Big\}.$$

Erinnerung: Für eine Formel $\varphi \in \mathsf{AL}'$ ist $\tilde{\varphi}$ die zu φ *duale Formel*, die aus φ entsteht, indem man überall $\mathbf{0}$ durch $\mathbf{1}$, $\mathbf{1}$ durch $\mathbf{0}$, $\wedge$ durch $\vee$ und $\vee$ durch $\wedge$ ersetzt.

Aufgabe 3: **(Präsenzaufgabe)**

Vor einigen Tagen führte ein schnuckliger Zombie-Hamster Alice auf die Fährte des zombifizierten Wissenschaftlers John, der beim Ausbruch des Zombie-Virus als erster mit dem Virus infiziert wurde und deshalb auch als *Patient Zero* bezeichnet wird. Unter Aufbietung einiger Raffinesse (und eines rostigen Rasenmähers) ist es Alice inzwischen gelungen, John einige DNA-Proben zu "entnehmen". Um das Genom des Zombie-Virus anhand dieser Proben zu bestimmen, stehen Alice leider nur die Mittel der weihnachtlichen Bastelecke des verwüsteten Einkaufszentrums von Hamster City zur Verfügung. Alice notiert sich ihre lückenhaften Ergebnisse als Terme, in denen sie die drei im Zombie-Virus vorkommenden Aminosäuren *Methionin*, *Tryptophan* und *Valin* durch Funktoren `m/3`, `t/2` und `v/1`, sowie das erstaunlicherweise ebenso vorkommende seltene Element *Promethium* durch `p` abkürzt. Um Lücken in den von ihr gefundenen Genom-Fragmenten zu markieren, die von ihrer Untersuchungsmethodik (oder auch dem Rasenmäher) hinterlassen wurden, setzt Alice Variablen ein.

$$\theta_1 := \texttt{m(v(X), v(Y), t(p, X))} \qquad \theta_2 := \texttt{m(v(Y), X, t(X, p))}$$
$$\theta_3 := \texttt{m(Z, p, t(Z, Z))} \qquad \theta_4 := \texttt{m(Z, v(Z), t(X, p))}$$

(a) Alice will ausschließen, dass sich in ihre Genom-Fragmente Messfehler eingeschlichen haben. Dazu will sie jeden der gefundenen Terme nur noch dann weiter untersuchen, wenn er mit mindestens einem anderen der Terme unifizierbar ist. Geben Sie deshalb für jedes $i \in \{2, 3, 4\}$ an, ob θ_1 und θ_i unifizierbar sind. Wenn ja, so geben Sie zusätzlich einen Unifikator für θ_1 und θ_i an.

(b) Alice überlegt, ob sie sich die Suche nach Unifikatoren für ihre Genom-Fragmente auch hätte einfacher machen können. Insbesondere fragt sie sich (und Sie!): Ist die Relation $U := \{(\theta, \eta) \colon \theta, \eta \in \mathsf{T}_{\mathsf{LP}}, \theta$ und η sind unifizierbar$\}$ eine Äquivalenzrelation, d.h. ist U reflexiv, transitiv und symmetrisch? Begründen Sie Ihre Antwort.

(c) Betrachten Sie die folgenden Terme und Substitutionen:

$$\theta_1 := \texttt{m(X,t(p,X),p)} \qquad \eta_1 := \texttt{m(p,t(p,Y),Y)} \qquad S_1 := \{\, \texttt{X} \mapsto \texttt{v(p)}, \texttt{Y} \mapsto \texttt{p} \,\}$$
$$\theta_2 := \texttt{t(t(p,X),Y)} \qquad \eta_2 := \texttt{t(Y,t(p,X))} \qquad S_2 := \{\, \texttt{X} \mapsto \texttt{p}, \texttt{Y} \mapsto \texttt{t(p, p)} \,\}$$
$$\theta_3 := \texttt{m(m(p,X,Y),p,X)} \qquad \eta_3 := \texttt{m(m(p,Y,X),Y,p)} \qquad S_3 := \{\, \texttt{X} \mapsto \texttt{p}, \texttt{Y} \mapsto \texttt{p} \,\}$$

Geben Sie für jedes $i \in [3]$ an, ob S_i ein allgemeinster Unifikator von θ_i und η_i ist. Ist S_i kein allgemeinster Unifikator von θ_i und η_i, so geben Sie einen allgemeinsten Unifikator für θ_i und η_i an.