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

10. Januar 2025

Logik in der Informatik

Wintersemester 2024/2025

Übungsblatt 11

Abgabe: *bis 20. Januar 2025, 13.00 Uhr*

Aufgabe 1: (Moodle-Quiz)

Absolvieren Sie das Quiz 11 auf der Moodle-Plattform.

Aufgabe 2: (Präsenzaufgabe)

Sei $\varphi \in \mathsf{FO}[\sigma]$, wobei σ eine Signatur ist, die ein 1-stelliges Relationssymbol P enthält. Seien x und y zwei verschiedene Variablen. Leiten Sie ähnlich wie in Beispiel 4.19 aus dem Skript die folgenden beiden Sequenzen im Sequenzenkalkül $\mathfrak{K}_S$ ab.

(a) $\neg\neg\varphi \vdash \varphi$

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

Aufgabe 3: (40 Punkte)

(a) Sei σ eine Signatur, sei $\Gamma \subseteq_e \mathsf{FO}[\sigma]$, seien $\varphi, \psi, \chi \in \mathsf{FO}[\sigma]$ und seien $x, y \in \mathsf{VAR}$. Beweisen Sie die Korrektheit der folgenden Sequenzenregeln:

(i) $\vee$-Einführung im Antezedens ($\vee$A):

$$\frac{\Gamma,\ \varphi \vdash \chi \qquad \Gamma,\ \psi \vdash \chi}{\Gamma,\ (\varphi \vee \psi) \vdash \chi}$$

(ii) $\wedge$-Einführung im Sukzedens ($\wedge$S):

$$\frac{\Gamma \vdash \varphi \qquad \Gamma \vdash \psi}{\Gamma \vdash (\varphi \wedge \psi)}$$

(iii) $\exists$-Einführung im Antezedens ($\exists$A):

$$\frac{\Gamma, \varphi^{\,y}_{x} \vdash \psi}{\Gamma, \exists x \varphi \vdash \psi} \qquad \text{falls } y \notin \text{frei}(\Gamma,\, \exists x\varphi,\, \psi)$$

— auf der nächsten Seite geht's weiter —

(b) Wir betrachten in dieser Teilaufgabe Kalküle über der Menge $M := \mathsf{AL}(\{\neg, \rightarrow\})$.

Ein Kalkül $\mathfrak{K}$ über M heißt *korrekt*, falls für jede Menge $\Phi \subseteq M$ und jede Formel $\psi \in M$ gilt: Wenn $\psi \in \mathrm{abl}_{\mathfrak{K}}(\Phi)$, dann gilt $\Phi \models \psi$. Ein Kalkül $\mathfrak{K}$ über M heißt *vollständig*, falls für jede Menge $\Phi \subseteq M$ und jede Formel $\psi \in M$ gilt: Wenn $\Phi \models \psi$, dann ist $\psi \in \mathrm{abl}_{\mathfrak{K}}(\Phi)$.

Seien $\mathfrak{K}_{Syl}$ und $\mathfrak{K}_{Abd}$ die beiden folgenden Kalküle über der Menge M: Beide Kalküle enthalten für jede *allgemeingültige* aussagenlogische Formel $\varphi \in M$ das Axiom $\dfrac{}{\varphi}$.

Außerdem enthält

- $\mathfrak{K}_{Abd}$ für alle $\varphi, \psi \in M$ die Ableitungsregel

$$\frac{\psi \qquad (\varphi \rightarrow \psi)}{\varphi},$$

  die *Abduktion* genannt wird,
- $\mathfrak{K}_{Syl}$ für alle $\varphi, \psi, \chi \in M$ die Ableitungsregel

$$\frac{(\varphi \rightarrow \psi) \qquad (\psi \rightarrow \chi)}{(\varphi \rightarrow \chi)},$$

  die *Syllogismus* genannt wird.

(i) Geben Sie für $\varphi := \neg(A_0 \rightarrow A_0)$ und $\Phi := \emptyset$ eine möglichst kurze Ableitung von φ aus Φ in $\mathfrak{K}_{Abd}$ an.

(ii) Beweisen Sie, dass $\mathfrak{K}_{Abd}$ vollständig, aber nicht korrekt ist.

(iii) Beweisen Sie, dass $\mathfrak{K}_{Syl}$ korrekt, aber nicht vollständig ist.

(iv) Betrachten Sie den Kalkül $\mathfrak{K}$ über M, der alle Ableitungsregeln aus $\mathfrak{K}_{Syl}$ und alle Ableitungsregeln aus $\mathfrak{K}_{Abd}$ enthält. Ist $\mathfrak{K}$ korrekt? Ist $\mathfrak{K}$ vollständig? Begründen Sie jeweils Ihre Antwort.

Aufgabe 4: (20 Punkte)

Lesen Sie Kapitel 11 aus dem Buch „Learn Prolog Now!“.

Achtung: *Die Bearbeitung der Aufgabe ist unter Beachtung der bekannten Abgabehinweise über Moodle abzugeben! Importieren Sie in Ihrer Abgabedatei* `blatt11.pl`*, analog zu Blatt 7, die Dateien* `al.pl` *und* `kinodb.pl`*.*

(a) Schreiben Sie ein Prädikat `clooney/1`, sodass die Anfrage

```
?- clooney(R).
```

in der Liste `R` die Menge aller Tupel `(K, Z)` zurückgibt, sodass gilt: Zum Zeitpunkt `Z` läuft im Kino `K` ein Film, in dem George Clooney mitgespielt hat.

(b) Wir implementieren eine Reservierungsverwaltung für das Kino Babylon (in dem zurzeit aus technischen Gründen nur ein Saal in Betrieb ist). Der Umstand, dass eine Person `P` für den Film, der um `Z` Uhr beginnt, Sitzplatz `S` reserviert hat, soll durch einen Fakt `belegt(P, Z, S)` in der Wissensbasis ausgedrückt werden. Stellen Sie zu diesem Zweck Ihrer Datei `blatt11.pl` die Zeile

```
:- dynamic belegt/3.
```

voran.

— auf der nächsten Seite geht's weiter —

Schreiben Sie ein Prädikat `reservieren/3`, sodass die Anfrage

```
?- reservieren(P, Z, S).
```

den Sitzplatz `S` für Person `P` und `Z` Uhr reserviert, d.h., der Wissensbasis einen Fakt `belegt(P, Z, S)` hinzufügt. Dies soll allerdings nur möglich sein, wenn um `Z` Uhr im Babylon tatsächlich ein Film läuft, und wenn der Sitzplatz für diese Uhrzeit noch nicht belegt ist. Anderenfalls soll die Anfrage scheitern.

(c) Schreiben Sie ein Prädikat `stornieren/2`, sodass die Anfrage

```
?- stornieren(Z, S).
```

die Reservierung für den Sitzplatz `S` zur Zeit `Z` aufhebt, d.h., einen entsprechenden Fakt in der Wissensbasis löscht. Gibt es eine solche Reservierung nicht, so soll die Anfrage scheitern.

(d) Wir repräsentieren im folgenden Klauseln als Listen von Literalen und Klauselmengen als Listen von Klauseln. So repräsentiert `[[~x1, ~x2, ~x3, x4], [x1, ~x2], [x2]]` die Klauselmenge $\{\{\neg X_1, \neg X_2, \neg X_3, X_4\}, \{X_1, \neg X_2\}, \{X_2\}\}$. Schreiben Sie ein Prädikat `unit_propagation/2`, das die Vereinfachungsheuristik *Unit Propagation* des DPLL-Algorithmus implementiert. D.h., ist `K` eine Klauselmenge, dann sollte die Anfrage

```
?- unit_propagation(K, K2).
```

in `K2` die Klauselmenge zurückgeben, die aus `K` entsteht, indem die Unit Propagation so lange auf `K` angewendet wird, bis keine „Einerklausel“ mehr vorhanden ist. Beispielsweise sollte die Anfrage

```
?- unit_propagation([[~x1, ~x2, ~x3, x4], [x1, ~x2], [x2]], K2).
```

zu der folgenden (oder einer äquivalenten) Antwort führen:

```
K2 = [[~x3, x4]].
```

Hinweis: Führen Sie geeignete Hilfsprädikate ein.