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

15. Dezember 2023

Logik in der Informatik

Wintersemester 2023/2024

Übungsblatt 9

Abgabe: *bis 8. Januar 2024, 13.⁰⁰ Uhr*

Aufgabe 1: (Moodle-Quiz)

Absolvieren Sie das Quiz 9 auf der Moodle-Plattform.

Aufgabe 2: (Präsenzaufgabe)

(a) Beweisen Sie folgende Aussage:
Für alle $m \in \mathbb{N}$, alle relationalen Signaturen σ, alle σ-Strukturen $\mathcal{A}$ und $\mathcal{B}$, alle $k \in \mathbb{N}$, alle $\overline{a} := a_1, \ldots, a_k \in A$ und alle $\overline{b} := b_1, \ldots, b_k \in B$ gilt:

> Genau einer der beiden Spieler hat eine Gewinnstrategie im m-Runden EF-Spiel auf $(\mathcal{A}, \overline{a})$ und $(\mathcal{B}, \overline{b})$.

(b) Sei $\Sigma = \{a, b\}$ und $\sigma_\Sigma = \{\leqslant, P_a, P_b\}$ die Signatur mit dem 2-stelligen Relationssymbol $\leqslant$ und den zwei 1-stelligen Relationssymbolen P_a und P_b. Beweisen Sie, dass es keinen $\mathsf{FO}[\sigma_\Sigma]$-Satz gibt, der die Sprache aller nicht-leeren Worte aus $\{a, b\}^*$ beschreibt, in denen die Anzahl der in ihnen vorkommenden as gerade ist.

Zur Erinnerung: Ein $\mathsf{FO}[\sigma_\Sigma]$-Satz φ *beschreibt* eine Sprache $L \subseteq \Sigma^*$, falls für jedes nicht-leere Wort $w \in \Sigma^*$ gilt: $w \in L \iff \mathcal{A}_w \models \varphi$.

Aufgabe 3: **(40 Punkte)**

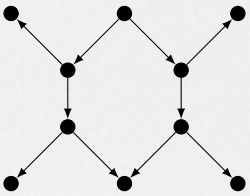
Sei $\sigma := \{E\}$ die Signatur mit dem 2-stelligen Relationssymbol E.

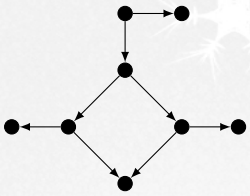
(a) Seien x, y, z drei paarweise verschiedene Variablen. Betrachten Sie die $\mathsf{FO}[\sigma]$-Formeln

$$\begin{aligned} \varphi &:= \Big(E(z,z) \ \vee \ \exists x\, \exists y\, x{=}y \Big) \\ \psi_1 &:= \Big(E(z,z) \ \vee \ \exists y\, \forall x\, x{=}y \Big) \\ \psi_2 &:= \Big(\forall z\, \exists y\, y{=}z \ \vee \ E(x,y) \Big) \end{aligned}$$

Geben Sie für jedes $i \in \{1,2\}$ an, ob $\varphi \equiv \psi_i$ gilt und beweisen Sie, dass Ihre Antwort korrekt ist.

(b) Betrachten Sie die folgenden gerichteten Graphen $\mathcal{A}$ und $\mathcal{B}$:

$\mathcal{A}$: 

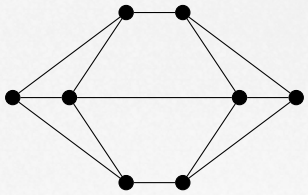
$\mathcal{B}$: 

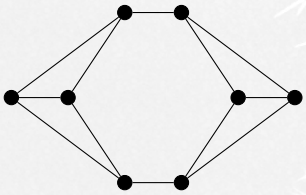
(i) Welches ist das kleinste m, sodass Spoiler eine Gewinnstrategie im m-Runden Ehrenfeucht-Fraïssé Spiel auf $\mathcal{A}$ und $\mathcal{B}$ hat? Begründen Sie Ihre Antwort, indem Sie eine Gewinnstrategie für Spoiler im m-Runden EF-Spiel und eine Gewinnstrategie für Duplicator im $(m{-}1)$-Runden EF-Spiel beschreiben.

(ii) Geben Sie für Ihre Antwort m aus Teil **(i)** einen $\mathsf{FO}[\sigma]$-Satz φ der Quantorentiefe m an, sodass $\mathcal{A} \models \varphi$ und $\mathcal{B} \models \neg\varphi$.

(c) In der folgenden Darstellung von Graphen repräsentiert jede ungerichtete Kante zwischen zwei Knoten u und v die beiden gerichteten Kanten (u,v) und (v,u).

Betrachten Sie die folgenden Graphen $\mathcal{A}$ und $\mathcal{B}$:

$\mathcal{A}$: 

$\mathcal{B}$: 

Sei $\varphi := \exists x\, \exists y\, \forall z\, (E(x,z) \vee E(y,z))$. Dann gilt: $\mathcal{A} \models \varphi$ und $\mathcal{B} \not\models \varphi$.

(i) Leiten Sie aus dem $\mathsf{FO}[\sigma]$-Satz φ eine Gewinnstrategie für Spoiler im EF-Spiel auf $\mathcal{A}$ und $\mathcal{B}$ her. Geben Sie an, wie viele Runden Spoiler benötigt, wenn er dieser Strategie folgt. Beschreiben Sie die Strategie ähnlich wie in der in der Vorlesung behandelten Beweisidee zu Satz 3.51 im Vorlesungsskript.

(ii) Existiert eine bessere Gewinnstrategie für Spoiler, d.h. eine Strategie, mit der er in weniger Runden das Spiel gewinnt?

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* `blatt9.pl`*, analog zu Blatt 7, die Dateien* `al.pl` *und* `kinodb.pl`*.*

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

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

in der Liste `R` die Menge aller Tupel `(K, Z)` zurückgibt, so dass 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 zur Zeit 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 `blatt9.pl` die Zeile

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

voran. Schreiben Sie ein Prädikat `reservieren/3`, so dass 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`, so dass 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.