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

15. Januar 2021

Logik in der Informatik

Wintersemester 2020/2021

Übungsblatt 9

Abgabe: *bis 25. Januar 2021, 13.⁰⁰ Uhr via Moodle*

*Um unseren Tutor*innen die Arbeit etwas zu erleichtern, notieren Sie bitte auf dem ersten Blatt jeder schriftlich erstellten Abgabedatei oben rechts eingerahmt folgende Daten:*

<Vorname Nachname>, <Matrikelnr.>, <Moodle-Loginname>

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 $(\mathcal{B}, \overline{b})$.

Hinweis: Per Induktion nach m ist der Beweis einfach und kurz.

(b) Nutzen Sie zur Lösung dieser Aufgabe die Methode der logischen Reduktion (ähnlich wie im Beweis von Satz 3.59).
Sei 2-COL die Klasse aller gerichteten zweifärbbaren Graphen, d.h. aller $\{E/2\}$-Strukturen $\mathcal{A} = (A, E^{\mathcal{A}})$ für die gilt:

> Es gibt eine Funktion $f : A \to \{rot, blau\}$, so dass für jede Kante (a, b) in $E^{\mathcal{A}}$ gilt: $f(a) \neq f(b)$.

Zeigen Sie: Die Klasse 2-COL ist *nicht* FO*-definierbar*.

Aufgabe 3: (40 Punkte)

(a) Welche der folgenden Aussagen ist für jede Signatur σ und alle $\mathsf{FO}[\sigma]$-Formel φ und ψ korrekt, welche nicht? Beweisen Sie, dass ihre Antworten korrekt sind.

(i) $(\forall x\, \varphi \;\vee\; \forall x\, \psi) \;\equiv\; \forall x\, (\varphi \vee \psi)$

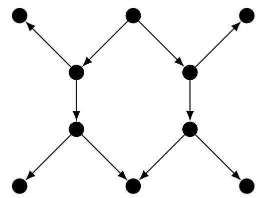
(ii) $\exists x\, \forall y\, \varphi \;\models\; \forall y\, \exists x\, \varphi$

(iii) $(\exists x\, \varphi \;\vee\; \exists x\, \psi) \;\equiv\; \exists x\, (\varphi \vee \psi)$

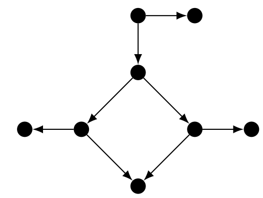
(iv) $\forall y\, \exists x\, \varphi \;\models\; \exists x\, \forall y\, \varphi$

(b) Sei $\sigma := \{E/2\}$. Betrachten Sie die folgenden gerichteten Graphen $\mathcal{A}$ und $\mathcal{B}$:

$\mathcal{A}$:



$\mathcal{B}$:



(i) Welches ist das kleinste m, so dass 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 Ehrenfeucht-Fraïssé Spiel und eine Gewinnstrategie für Duplicator im $(m-1)$-Runden Ehrenfeucht-Fraïssé Spiel beschreiben.

(ii) Finden Sie für Ihre Antwort m aus Teil **(i)** einen $\mathsf{FO}[\sigma]$-Satz ψ der Quantorentiefe m, so dass $\mathcal{A} \models \psi$ und $\mathcal{B} \models \neg\psi$.

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 Regie geführt 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.