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

21. Januar 2015

Logik in der Informatik

Wintersemester 2014/2015

Übungsblatt 11

Abgabe: *bis 28. Januar 2015, 9.15 Uhr (vor der Vorlesung oder im Briefkasten zwischen den Räumen 3.401 und 3.402 im Johann von Neumann-Haus (Rudower Chaussee 25))*

Aufgabe 1: (24 Punkte)

(a) Welche der folgenden Aussagen sind für alle Signaturen σ und alle $\mathsf{FO}[\sigma]$-Formeln φ und ψ korrekt, welche nicht? (Sie brauchen Ihre Antwort nicht zu begründen.)

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

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

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

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

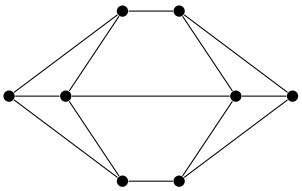
(b) Beweisen Sie, dass ihre Antworten zu **(i)** und **(iii)** korrekt sind.

Aufgabe 2: (25 Punkte)

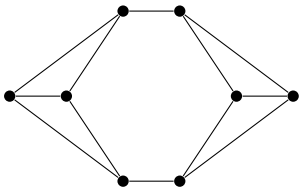
Sei $\sigma := \{E/2\}$. In der folgenden Darstellung von Graphen repräsentiert jede ungerichtete Kante zwischen zwei Knoten u und v die beiden gerichteten Kanten (u, v) und (v, u).

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 $\mathcal{B} \not\models \varphi$.

(a) 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 4.51.

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

Aufgabe 3: (26 Punkte)

Nutzen Sie für diese Aufgabe das in der Vorlesung vorgestellte Verfahren der logischen Reduktion.

(a) Sei $\Sigma = \{a, b\}$ und $\sigma_\Sigma = \{P_a, P_b, \leqslant\}$ die Signatur, die aus dem 2-stelligen Relationssymbol $\leqslant$, sowie zwei 1-stelligen Relationssymbolen P_a und P_b besteht. Beweisen Sie, dass es keinen $\mathsf{FO}[\sigma_\Sigma]$-Satz gibt, der die Sprache aller 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 nichtleere Wort $w \in \Sigma^*$ gilt: $w \in L \iff \mathcal{W}_w \models \varphi$.

(b) 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 $f : A \to \{\mathit{rot}, \mathit{blau}\}$, 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 4: (25 Punkte)

Lesen Sie Kapitel 7 aus dem Buch "Learn Prolog Now!".

Achtung: *Die Bearbeitung der folgenden Aufgabe ist in einer Datei* `hal.pl` *digital über das GOYA-System abzugeben!*

Nachdem *David Bowman* den zunehmend neurotischen Computer *HAL 9000* der *Discovery* abgeschaltet hat, benötigt dessen Sprachausgabe eine neue kontextfreie Grammatik. *HAL 9000* soll Sätze der Form

```
I can not X, Dave.
```

formulieren können, wobei *X* eine aus den Aktivitäten `open the pod bay doors`, `make coffee` und `do this` ist. Außerdem soll *HAL 9000* Sätze der Form

```
I can neither X, Dave.
```

formulieren können, wobei *X* eine durch "`nor`" getrennte Aufzählung der genannten Aktivitäten ist. Diese Aufzählung muss mindestens die Länge zwei haben, kann aber durchaus auch Aktivitäten mehrfach enthalten.

Implementieren Sie eine kontextfreie Grammatik für die beschriebene Sprache als *"Definite Clause Grammar" (DCG)*. Die Grammatik soll das Startsymbol `hal` und die Terminalsymbole

"`I`", "`,`", "`can`", "`not`", "`neither`", "`nor`", "`open the pod bay doors`", "`make coffee`", "`do this`", "`Dave`", "`.`"

besitzen. Beispielsweise sollen die folgenden Anfragen an Prolog erfüllt sein:

```
?- hal(['I', 'can', 'not', 'do this', ',', 'Dave', '.'], []).
?- hal(['I', 'can', 'neither', 'make coffee', 'nor', 'do this',
        'nor', 'open the pod bay doors', ',', 'Dave', '.'], []).
```

die folgende Anfrage jedoch nicht:

```
?- hal(['I', 'can', 'neither', 'do this', ',', 'Dave', '.'], []).
```

Hinweis: *Implementieren Sie die Grammatik mit Hilfe der* `-->` *Notation. Achten Sie auch darauf, dass Ihre Grammatik keine linksrekursiven Regeln enthält!*