Goethe-Universität Frankfurt am Main
Institut für Informatik
Theorie komplexer Systeme
Prof. Dr. Nicole Schweikardt

24. Oktober 2013

Logik in der Informatik

Wintersemester 2013/14

Übungsblatt 2

Zu bearbeiten bis 31. Oktober 2013

Aufgabe 1: **(11 + 11 = 22 Punkte)**

Die Signatur σ bestehe aus einem 2-stelligen Relationssymbol E und einem 2-stelligen Funktionssymbol f. Betrachten Sie die FO[σ]-Formeln

(a) $\varphi_1 \ := \ \forall v_0 \, \forall v_1 \, \big(\exists v_2 (E(v_0, v_2) \land E(v_2, v_1)) \to E(v_0, v_1) \big)$

(b) $\varphi_2 \ := \ \forall v_0 \, \forall v_1 \, \big(E(v_0, v_1) \ \leftrightarrow \ E(f(v_0, v_1), f(v_1, v_0)) \big)$

Geben Sie für jedes $i \in \{1, 2\}$ eine σ-Interpretation $\mathcal{I}_i$ und eine σ-Interpretation $\mathcal{J}_i$ an mit $\mathcal{I}_i \models \varphi_i$ und $\mathcal{J}_i \not\models \varphi_i$.

Aufgabe 2: **(7 + 7 + 7 + 7 = 28 Punkte)**

Es sei R ein 2-stelliges Relationssymbol und f ein 2-stelliges Funktionssymbol. Berechnen Sie

(a) $\big(f(v_1, v_0) = v_2 \ \lor \ R(v_1, v_2) \big) \frac{f(v_0, v_1), \ f(v_4, v_1)}{v_0 \quad , \quad v_2}$

(b) $\exists v_1 \big(R(v_0, v_1) \ \to \ \forall v_2 \, f(v_1, v_2) = v_1 \big) \frac{f(v_2, v_0), \ v_3}{v_1 \quad , \ v_2}$

(c) $\forall v_0 \big(\neg R(v_2, v_0) \ \land \ f(v_1, v_2) = v_1 \big) \frac{v_2, \ v_3}{v_1, \ v_2}$

(d) $\forall v_1 \big(\neg R(v_0, v_1) \ \lor \ \exists v_0 \, R(v_1, f(v_4, v_0)) \big) \frac{f(v_1, v_2), \ v_0}{v_0 \quad , \ v_3}$

Aufgabe 3: **(25 Punkte)**

Sei σ eine relationale Signatur.
Zeigen Sie, dass es für jede FO[σ]-Formel φ eine zu φ äquivalente FO[σ]-Formel $\tilde{\varphi}$ mit $\operatorname{frei}(\tilde{\varphi}) = \operatorname{frei}(\varphi)$ gibt, in der höchstens $\operatorname{br}(\varphi)$ viele verschiedene Variablen vorkommen.

Aufgabe 4: **(25 Punkte)**

Entwickeln Sie einen Algorithmus, der bei Eingabe einer beliebigen FO[σ]-Formel φ eine zu φ äquivalente Formel φ' mit $\operatorname{frei}(\varphi') = \operatorname{frei}(\varphi)$ in pränexer Normalform erzeugt.
Analysieren Sie die Laufzeit Ihres Algorithmus (in Abhängigkeit von der Länge der Eingabe φ).