Prof. Dr. A. Goerdt
L. Falke

TU Chemnitz
Wintersemester 2015/2016
6.10.2015

Theorie der Programmiersprachen

9. Übung

1. Aufgabe: Man zeige mittels *Grundresolution* und *prädikatenlogischer Resolution*, dass sowohl

$$F \;=\; \forall x\big(\neg P(x) \wedge P(f(a))\big)$$

als auch

$$G \;=\; \forall x\Big(\Big(\big(P(f(x)) \to P(x)\big) \wedge P(f(f(a)))\Big) \wedge \neg P(a)\Big)$$

unerfüllbar sind.

2. Aufgabe: Formalisieren Sie die Aussagen (a) und (b) als prädikatenlogische Formeln. Verwenden Sie die Notation $S(x, y)$ – x ist Student von y, $G(x)$ – x ist glücklich, $M(x)$ – x mag Logik.

(a) A = „Der Professor ist glücklich, wenn alle seine Studenten Logik mögen."

(b) B = „Der Professor ist glücklich, wenn er keine Studenten hat."

Zeigen Sie durch *Grundresolution* und *prädikatenlogische Resolution*, dass (b) eine Folgerung von (a) ist. Formulieren Sie dazu $A \wedge \neg B$ in Klauselform.

Gegeben seien zwei Strukturen $\mathcal{A}$ und $\mathcal{B}$ mit:

$U_{\mathcal{A}}$	=	$\{p, s_1, \ldots, s_n\}$	$U_{\mathcal{B}}$	=	$\mathbb{N} = \{1, 2, 3, \ldots\}$
$I_{\mathcal{A}}(G)$	=	$\{x \mid x$ ist glücklich$\}$	$I_{\mathcal{B}}(G)$	=	$\{x \mid x = 1\}$
$I_{\mathcal{A}}(M)$	=	$\{x \mid x$ mag Logik$\}$	$I_{\mathcal{B}}(M)$	=	$\{x \mid x > 1\}$
$I_{\mathcal{A}}(S)$	=	$\{(x, y) \mid x$ ist Student von $y\}$	$I_{\mathcal{B}}(S)$	=	$\{(x, y) \mid x > y\}$
$I_{\mathcal{A}}(p)$	=	$\{p\}$	$I_{\mathcal{B}}(p)$	=	$\{1\}$
$I_{\mathcal{A}}(q)$	=	$\{s_1\}$, q Skolemkonstante	$I_{\mathcal{B}}(q)$	=	$\{2\}$

Zeigen Sie durch „Hochgehen" in Ihren Beweisen, dass beide Strukturen kein Modell für die entsprechende Formel sind.

3. Aufgabe: Man drücke folgende Tatsachen als prädikatenlogische Formeln aus:

A = „Jeder Drache ist glücklich, wenn alle seine Kinder fliegen können."

B = „Grüne Drachen können fliegen."

C = „Ein Drache ist grün, wenn er Kind mindestens eines grünen Drachen ist."

Man zeige durch *Grundresolution* und *prädikatenlogische Resolution*, dass aus A, B und C folgt, dass *alle grünen Drachen glücklich sind.*

4. Aufgabe: Geben Sie *alle* prädikatenlogischen Resolventen von

$$\big\{P(f(x)), \neg Q(z), P(z)\big\} \quad \text{und} \quad \big\{\neg P(x), R(g(x), a)\big\}$$

an. (x, y, z sind Variablen, a ist eine Konstante)

5. Aufgabe: Man wende den Unifikationsalgorithmus auf die Literalmenge

$$L \;=\; \big\{P(x, y),\; P(f(a), g(x)),\; P(f(z), g(f(z)))\big\}$$

an.

6. Aufgabe: Aus Effizienzgründen wird in manchen Implementierungen des Unifikationsalgorithmus' auf den Test „kommt x in t vor“ (*occur check*) verzichtet. Man gebe ein Beispiel einer *nicht unifizierbaren*, zweielementigen Literalmenge L_1, L_2 an, so dass L_1 und L_2 keine gemeinsamen Variablen enthalten, und ein Unifikationsalgorithmus ohne *occur check* – je nach Implementierung – in eine unendliche Schleife gerät oder fälschlicherweise „unifizierbar“ konstatiert.

7. Aufgabe: Zeigen Sie, dass der Unifikationsalgorithmus (naiv implementiert) exponentielle Laufzeit haben kann.
Hinweis: Man betrachte das Beispiel:

$$L \;=\; \Big\{P\big(x_1, x_2, \ldots, x_n\big),\; P\big(f(x_0, x_0), f(x_1, x_1), \ldots, f(x_{n-1}, x_{n-1})\big)\Big\}$$

Man überlege sich eine geeignete Datenstruktur für Literale bzw. Literalmengen, so dass das Unifizieren effizienter durchgeführt werden kann.