Prof. Dr. A. Goerdt
Julian Pape-Lange

TU Chemnitz
Wintersemester 2019/2020
19.12.2019

Theorie der Programmiersprachen

10. Übung

1. Aufgabe: Zeigen Sie mittels *Grundresolution* und des *Algorithmus von Gilmore*, dass sowohl

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

als auch

$$G = \forall x\Bigg(\bigg(\big(P(f(x)) \to P(x)\big) \land P(f(f(a)))\bigg) \land \neg P(a)\Bigg)$$

unerfüllbar sind.

2. Aufgabe:

(a) 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.

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

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

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

3. Aufgabe: Drücken Sie 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: Wenden Sie den Unifikationsalgorithmus auf die Literalmenge

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

an.

5. Aufgabe: Aus Effizienzgründen wird in manchen Implementierungen des Unifikationsalgorithmus' auf den Test „kommt x in t vor“ (*occur check*) verzichtet. Geben Sie 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.

6. Aufgabe: Der Unifikationsalgorithmus (naiv implementiert) kann exponentielle Laufzeit haben. Das kann zum Beispiel bei

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

passieren.
Überlegen Sie sich eine geeignete Datenstruktur für Literale bzw. Literalmengen, so dass das Unifizieren effizienter durchgeführt werden kann.

7. Aufgabe: Transformieren Sie die Formel

$$F = \forall x \forall y \forall z \Bigg(\Big(\big(P(x) \leftrightarrow Q(f(x))\big) \vee \big(\neg Q(x) \to R(g(x), h(y))\big)\Big) \to \big(R(g(z), x) \wedge Q(y)\big)\Bigg)$$

in eine erfüllbarkeitsäquivalente Formel der Form

$$\widetilde{F} = \forall x \forall y \forall z F'.$$

Dabei soll F' in 3-KNF sein. Verallgemeinern Sie hierzu den Algorithmus zur Erstellung einer erfüllbarkeitsäquivalenten 3-KNF aus der Aussagenlogik.