Prof. Dr. A. Goerdt
J. Pape-Lange

TU Chemnitz
Wintersemester 2021/2022
05.01.2022

Theorie der Programmiersprachen

11. Übung

1. 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.

2. 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.

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

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

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

4. Aufgabe: Bei endlichen aussagenlogischen Klauselmengen F ist $Res^*(F)$ immer eine endliche Menge. Geben Sie eine endliche prädikatenlogische Klauselmenge F an, so dass für alle n gilt:

$$Res^n(F) \neq Res^*(F).$$