

Theorie der Programmiersprachen

7. Übung

1. Aufgabe:

Man wende den Unifikationsalgorithmus auf die Literalmenge

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

an.

2. Aufgabe:

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

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

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

3. 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.