Berthold Hoffmann
Markus Roggenbach

Kurs "Programmiersprachen"
Aufgabenblatt Logisches Programmieren

Aufgabe 1:

Unifiziere die folgenden Klauselmengen:

1. $\{\neg P(f(z, g(a, y)), h(z)), \neg P(f(f(u, v), w), h(f(a, b)))\}$

   Dabei ist P ein zweistelliges Prädikat, f und g sind zweistellige Funktionssymbole, h ist ein einstelliges Funktionssymbol, u, v, w, x, y, z sind Variable und a, b sind Konstanten.

2. $\{A(a, x, x), A(a, y, f(y))\}$

   Dabei ist A ein dreistelliges Prädikat, f ist ein einstelliges Funktionssymbol, x, y sind Variable und a ist eine Konstante.

   Was passiert, wenn bei der Unifikation der Occur-Check nicht ausgeführt wird?

Aufgabe 2:

Sei der Grundriss eines Hauses gegeben als eine Menge von Fakten der Form `door(`a`,`b`)`, die besagen, dass eine Tür von Raum a in Raum b führt, und mindestens ein Faktum `hasTelephone(`c`)`, das besagt, dass in Raum c ein Telefon steht. (Hierbei sind a, b, c Atome, die Räume bezeichnen.)

Definiere ein Prädikat `go(`X, Y, T`)`, das einen Weg von Raum X in Raum Y findet und die Folge der aufgesuchten Räume in der Liste T aufsammelt.

Aufgabe 3:

Definiere ein Prädikat `Diff(`F, DF`)`, das eine Funktion F nach x ableitet. Funktionen (und ihre Ableitungen) sind Terme über den Atomen x, 0, 1, $+$, $*$, sin und cos. Was berechnet für die Zielklausel `Diff(`$x * sin(x)$`, H)`?