

11. Übungsblatt zu Programmiersprachenkonzepten

Manuel Schwarz, Michael Stypa

19. Januar 2011

Aufgabe 11.1

Unifikation Gegeben zwei prädikatenlogische Aussagen $A_1 = (X, Y, f(b))$ und $A_2 = (a, b, Z)$ mit Atomen a und b und Variablen X, Y und Z . *Unifikation* formt Aussagen mit Hilfe der *Substitution* so um, dass diese gleich aussehen. Ersetzt man in A_1 nun X durch a und Y durch b und in A_2 Z durch $f(b)$ so erhält man zwei gleiche Aussagen.

Backtracking Die Suche nach einer Lösung kann in Einzelschritte unterteilt werden. In jedem Schritt gibt es die Möglichkeit mit unterschiedlichen Folgeschritten fortzufahren. Stellt man zu einem Zeitpunkt fest, dass der Aktuelle Weg nicht zur Lösung führt so geht man die gewählten Schritte rückwärts und wählt einen anderen Weg. Dieses Suchverfahren lässt sich anschaulich in einer Baum-Struktur darstellen und man bezeichnet es als *Backtracking*.

Aufgabe 11.2

...

```
58 mutter(ellen, torben).
59
60 grossvater(X, Y) :- vater(X, Z), vater(Z, Y);
61                   vater(X, Z), mutter(Z, Y).
62 grossmutter(X, Y) :- mutter(X, Z), vater(Z, Y);
63                   mutter(X, Z), mutter(Z, Y).
64 schwester(X, Y)   :- w(X), mutter(Z, X), mutter(Z, Y), not(X = Y).
65 bruder(X, Y)      :- m(X), mutter(Z, X), mutter(Z, Y), not(X = Y).
66 tante(X, Y)       :- w(X), mutter(Z, Y), schwester(X, Z);
67                   w(X), vater(Z, Y), schwester(X, Z).
68 onkel(X, Y)        :- m(X), mutter(Z, Y), bruder(X, Z);
69                   m(X), vater(Z, Y), bruder(X, Z).
70 cousine(X, Y)      :- w(X), tante(Z, Y), mutter(Z, X);
71                   w(X), onkel(Z, Y), vater(Z, X).
72 cousin(X, Y)       :- m(X), tante(Z, Y), mutter(Z, X);
73                   m(X), onkel(Z, Y), vater(Z, X).
```

Aufgabe 11.3

```
1  laenge([], 0).
2  laenge([_|Rest], Lang) :- laenge(Rest, RLang), Lang is RLang + 1.
```

Aufgabe 11.4

```
12  direktverbindung(X, Y, Z) :- verbindung(X, Y, A, _, C), passt(Z, A).
13  zwischenreise(X, Y, Z)    :- verbindung(X, A, B, C, D), passt(Z, B),
14                               reise(A, Y, C).
15  reise(X, Y, Z)            :- direktverbindung(X, Y, Z);
16                               zwischenreise(X, Y, Z).
17  passt(X, Y)               :- X =< Y.
```