

# Übungen 6: Listen

Programmiertechniken in der Computerlinguistik I · Wintersemester 2000/2001

Um keine Zeit mit Tippen und Tippfehlerkorrektur zu verschwenden, findest du die Programmtexte dieser Übung unter <http://www.ifi.unizh.ch/cl/sicemat/lehre/ws0001/pcl1/uebung6.txt>

## 1. Syntaxdiagramm der Listenschreibweise

a) Zeichne ein Syntaxdiagramm, das den Aufbau von Listentermen in der Listenschreibweise von Prolog beschreibt. D.H. es soll Ausdrücke wie

```
[ ]  
[a,b,4]  
[A,b|[4]]  
[a|Rest]
```

als korrekte Listenschreibweisen erlauben, Ausdrücke wie

```
[a,|Rest]  
[a|b]  
[ ]
```

jedoch nicht.

b) Wodurch wird im Syntaxdiagramm die Rekursivität der Listen ausgedrückt?

## 2. Listen unifizieren

Gib – ohne den Prolog-Interpreter zu benutzen – an, was das Ergebnis der folgenden Anfragen sein wird. Notiere dir *yes/no* und allfällige Variablenbindungen auf. Überprüfe anschliessend am Computer, ob Prolog tatsächlich die erwartete Antwort gibt.

- a) ?- X = [fido,gulasch].
- b) ?- [fido,frisst] = [frisst,Gulasch].
- c) ?- [fido,frisst,Was] = [Wer,frisst,gulasch].
- d) ?- [fido,gulasch] = [fido|X].
- e) ?- [fido|Rest] = [fido,frisst,gulasch].
- f) ?- [fido,Rest] = [fido,frisst,gulasch].
- g) ?- [fido,frisst|[X]] = [fido,frisst,gulasch|[ ]].
- h) ?- '.'(fido,'.'(frisst,[Gulasch])) = [fido,frisst,gulasch].
- i) ?- '.'(fido,'.'(frisst,[ ])) = [Fido|X].
- j) ?- '.'(frisst,gulasch) = [frisst,gulasch].
- k) ?- [[frisst],[fido,gulasch,'?']] = [[X|Y]|Z].

## 3. Termstruktur von Listen

Zeichne die Liste [[1,2],[ ],3] als Baum in der Dot-Notation.

## 4. Das dritte Element

Definiere ein Prädikat, das als erstes Argument eine Liste nimmt und als zweites Argument das dritte Element der Liste zurückliefert.

```
?- element_3([kiwi,apfel,birne,zwetschge], Drittes).  
Drittes = birne
```

Falls die Liste weniger als drei Elemente enthält, soll das Prädikat scheitern:

```
?- element_3([gurke,apfel], Drittes).  
no
```

Das Prädikat soll auch mit umgekehrter Instantiierung der Anfrage funktionieren:

```
?- element_3([kiwi,pfirsich,Drittes|_], birne).  
Drittes = birne
```

## 5. Prolog-Slang

a) Definiere ein Prädikat `slang/2`, das bei den Bemühungen hilft, den Wortschatz bestimmter Sprachgruppen nachzubilden. Dabei sollen etwas verblasste Adjektive wie "nett", "lieb" oder "dumm" zu kräftigeren "meganett", "supermegadumm" etc. werden. Es sollten mindestens 3 Verstärkungsstufen eingebaut werden. Die Adjektive sollen als Listen von Buchstaben dargestellt werden.

```
?- slang([n,e,t,t], X).  
X = [s,u,p,e,r,n,e,t,t] ;  
X = [s,u,p,e,r,s,u,p,e,r,n,e,t,t] ;  
X = [s,u,p,e,r,s,u,p,e,r,s,u,p,e,r,n,e,t,t] ;  
no
```

Es sollen nur bekannte Adjektive verstärkt werden:

```
?- slang([a,b,c,d], X).  
no
```

b) Schreibe einen Failure-Driven-Loop, der alle Slang-Adjektive herausschreibt (in der Listennotation, nicht in der natürlichen Schreibweise).