

# Übung 13

## Musterlösung

### Aufgabe 13.1

a)

Der Fehler der Funktion *mymapcar* liegt darin, daß sie die Antwortliste in falscher Reihenfolge liefert. In jedem Durchlauf der Funktion *iter* wird an die bestehende, anfänglich leere Antwortliste, das jeweils erste und nun mit *f* bearbeitete Element vorne angehängt.:

```
(cons (funcall f (car liste)) ergebnisliste).
```

b)

```
(defun mymapcar2 (f liste)
  (defun iter (liste ergebnisliste)
    (cond ((null liste) (reverse(ergebnisliste)))
          (t (iter (cdr liste)
                    (cons (funcall f (car liste)) ergebnisliste)))))
  (iter liste ()))
```

c)

```
(mymapcar2 (lambda (x) (* x x)) '(1 2 3 4))
```

Ergebnis: (1 4 9 16)

### Aufgabe 13.2

a)

```
studentHoertVorlesung(balin, kryptographie).
studentHoertVorlesung(beutlin, compilerbau).
studentHoertVorlesung(boffin, softwaretechnik).
studentHoertVorlesung(brandybock, mustererkennung).
studentHoertVorlesung(pausbacken, mustererkennung).
studentHoertVorlesung(straffguertel, compilerbau).
studentHoertVorlesung(bolger, datenkommunikation).
studentHoertVorlesung(honblaeser, softwaretechnik).
studentHoertVorlesung(stolzfuss, datenkommunikation).
```

```
profLiestVorlesung(hromkovic, kryptographie).
profLiestVorlesung(indermark, compilerbau).
profLiestVorlesung(nagl, softwaretechnik).
profLiestVorlesung(spaniol, datenkommunikation).
profLiestVorlesung(neu, mustererkennung).
```

b)

```
balin_hoert_bei_hromkovic :- studentHoertVorlesung(balin, kryptographie),
                             profLiestVorlesung(hromkovic, kryptographie).
```

```
student_hoert_bei_nagl(Student) :- studentHoertVorlesung(Student,
                                                           softwaretechnik),
                                    profLiestVorlesung(nagl, softwaretechnik).
```

```
student_hoert_bei(Student, Prof) :- studentHoertVorlesung(Student, Vorlesung),
                                   profLiestVorlesung(Prof, Vorlesung).
```

```
beutlin_ist_kommilitoneVon_balin :- studentHoertVorlesung(beutlin, Vorlesung),
                                   studentHoertVorlesung(balin, Vorlesung).
```

```
kommilitoneVon(Stud1, Stud2) :- studentHoertVorlesung(Stud1, Vorlesung),
                                studentHoertVorlesung(Stud2, Vorlesung),
                                not(Stud1=Stud2).
```

c)

```
?- studentHoertVorlesung(balin, compilerbau).
NO
```

```
?- studentHoertVorlesung(balin, kryptographie).
YES
```

```
?- profLiestVorlesung(nagl, softwaretechnik).
YES
```

```
?- profLiestVorlesung(ney, compilerbau).
NO
```

```
?- studentHoertVorlesung(brandybock, V).
V = mustererkennung
NO
```

```
?- profLiestVorlesung(spaniol, V).
V = datenkommunikation
NO
```

```
?- profLiestVorlesung(P, mustererkennung).
P = ney
NO
```

```
?- studentHoertVorlesung(S, softwaretechnik).
S = boffin ;
S = hornblaeser ;
NO
```

```
?- student_hoert_bei(S, indermark).
S = beutlin ;
S = straffguertel ;
NO
```

```
?- student_hoert_bei(S, P).
S = balin
P = hromkovic ;
```

```
S = beutlin
P = indermark ;
```

```
S = boffin
P = nagl ;
```

```
S = bandybock
P = ney ;
```

```
S = pausbacken
P = ney ;
```

S = straffguertel  
P = indermark ;  
S = bolger  
P = spaniol ;

S = hornblaeser  
P = nagl ;

S = stlzfuss  
P = spaniol ;

NO

?- beutlin\_ist\_kommilitoneVon\_balin.  
NO

?- kommilitoneVon(beutlin, straffguertel).  
YES

?- kommilitoneVon(bolger, K).  
K = stolfuss ;  
NO

?- kommilitoneVon(K1, K2).  
K1 = beutlin  
K2 = straffguertel ;

K1 = boffin  
K2 = hornblaeser ;

K1 = brandybock  
K2 = pausbacken ;  
K1 = pausbacken  
K2 = brandybock ;

K1 = straffguertel  
K2 = beutlin ;

K1 = bolger  
K2 = stolfuss ;

K1 = hornblaeser  
K2 = boffin ;

K1 = stolfuss  
K2 = bolger ;

NO

## Aufgabe 13.3

a)

```
farbe(rot).  
farbe(gelb).  
farbe(blau).  
farbe(gruen).
```

```
verschieden(rot, gelb).  
verschieden(rot, gruen).  
verschieden(rot, blau).  
verschieden(gelb, rot).  
verschieden(gelb, gruen).  
verschieden(gelb, blau).  
verschieden(gruen, rot).  
verschieden(gruen, gelb).  
verschieden(gruen, blau).  
verschieden(blau, gelb).  
verschieden(blau, rot).  
verschieden(blau, gruen).
```

```
korrekteFaerbung(L1, L2, L3, L4, L5, L6, L7, L8) :-  
    verschieden(L1, L2), verschieden(L1, L3),  
    verschieden(L1, L4), verschieden(L1, L5),  
    verschieden(L2, L5), verschieden(L2, L6), verschieden(L2, L7),  
    verschieden(L3, L4), verschieden(L3, L5),  
    verschieden(L3, L6), verschieden(L4, L5),  
    verschieden(L5, L6), verschieden(L5, L7),  
    verschieden(L6, L7), verschieden(L7, L8).
```

b)

Eine Lösung des Problems erhalten wir, wenn wir nach einer Färbung der acht Länder fragen, bei der nur die vier erlaubten Farben benutzt werden und die auch korrekt ist:

```
?- farbe(L1), farbe(L2), farbe(L3), farbe(L4), farbe(L5), farbe(L6),  
    farbe(L8), korrekteFaerbung(L1, L2, L3, L4, L5, L6, L7, L8).
```

Eine korrekte Färbung der Landkarte lautet:

```
L1=rot, L2=gelb, L3=gelb, L4=blau, L5=gruen, L6=rot, L7=blau,  
L8=rot
```

c)

Insgesamt gibt es 360 Lösungen für eine korrekte Färbung der Landkarte.