

Funktionale Programmierung SS 2013 - Lösung 1

Prof. Dr. U. Kastens

Institut für Informatik, Fakultät für Elektrotechnik, Informatik und Mathematik, Universität Paderborn

15.04.2013

Lösung zu Aufgabe 1

a)

```
(+ 13 -12)
```

b)

```
(- (* 5 (+ 1 2)) 16)
```

c)

```
(setq sinn (* 2 3 7))
```

d)

```
(cons 1 (cons 2 NIL))
```

e)

```
(setq mylist (quote (1 2 3 4)))
```

oder

```
(setq mylist '(1 2 3 4))
```

f)

```
(defun avg (a b) (/ (+ a b) 2))  
(avg 2 4)  
(avg 10 15)
```

g)

```
(defun sign (x) (cond ((< x 0) -1) ((> x 0) 1) (T 0)))
```

Lösung zu Aufgabe 2

a)

```
(defun sum (n) (cond ((= n 0) 0)(T (+ n (sum (- n 1))))))
```

b)

```
(defun pow (b e) (cond ((= e 0) 1)(T (* b (pow b (- e 1))))))
```

Lösung zu Aufgabe 3

a)

```
(defun ntes (n l) (cond ((null l) nil)  
                        ((= n 0) (car l))  
                        (T (ntes (- n 1) (cdr l)))))
```

b)

```
(defun myappend (left right) (cond ((null left) right)  
                                   (T (cons (car left) (myappend (cdr left) right)))))
```

Lösung zu Aufgabe 4

a)

```
(defun twice (f x) (funcall f (funcall f x)))
```

b)

```
(defun mymap (l f) (if (null l) nil
                      (cons (funcall f (car l))
                            (mymap (cdr l) f))))

(mymap '(1 0 -23 -1 41) (lambda (x)(+ x 1)))
(mymap '(1 0 -23 -1 41) (lambda (x)(if (< x 0) 0 x)))
(mymap '(1 0 -23 -1 41) (lambda (x)(cons x nil)))
```

Das Problem in der Formulierung ist, dass der Eindruck entstehen könnte, dass Listenelemente verändert werden. Beispiel für den Versuch einer präziseren Aufgabenformulierung:
Verwenden Sie `mymap`, um aus einer Liste von Zahlen eine gleich lange Liste von Zahlen zu erzeugen, deren Elemente die um 1 erhöhten Werte der Elemente der ersten Liste haben.