

Übungen zum Schemekurs

Aufgabe 9.

- a) Man definiere eine Prozedur `double` mit einem Argument, die eine Prozedur zurückgibt, welche die Originalprozedur zweimal anwendet. Ist zum Beispiel `inc` eine Prozedur, die ihr Argument um 1 vergrößert, dann soll `(double inc)` eine Prozedur sein, die 2 addiert. Was ist der Wert von `((double (double double)) inc) 5`?
- b) Seien f und g zwei einstellige Prozeduren. Die *Komposition* von f und g ist definiert als die Funktion $x \mapsto f(g(x))$. Definieren Sie eine Prozedur `compose`, die die Komposition implementiert. Was ist der Wert von `((compose square inc) 27)`?

Lösung.

```
(define (square x) (* x x))
(define (inc x) (+ x 1))

(define (double f)
  (lambda(x) (f (f x))))

(define (compose f g)
  (lambda(x) (f (g x))))

((compose square inc) 27)

(((double (double double)) inc) 5)
```

Aufgabe 10. Man bestimme die Projektionsfunktionen zur folgenden Paarbildungsoperation.

```
(define (make-pair a b)
  (lambda (h) (h a b)))
```

Lösung.

```

(define (make-pair a b)
  (lambda (h) (h a b)))

(define (new-car pair)
  (pair (lambda(a b) a)))

(define (new-cdr pair)
  (pair (lambda(a b) b)))

(new-car (make-pair 1 2))
(new-cdr (make-pair 1 2))

```

Aufgabe 11. Man schreibe eine Prozedur `natseq`, die jeder natürlichen Zahl n die aufsteigende Liste der Zahlen von 0 bis $n - 1$ zuordnet.

Lösung.

```

(define (natseq n)
  (if (= n 0)
      '()
      (append (natseq (- n 1)) (list (- n 1)))))

(natseq 5)

```

Aufgabe 12.* Man schreibe eine Prozedur, die das Pascalsche Dreieck Stufe für Stufe berechnet:

```

      1
    1 1
  1 2 1
1 3 3 1
1 4 6 4 1

```

Man benutze dazu Listen.

Lösung.

```

; von Stefan Wolf
(define (pascal n)
  (define (calc-new a)
    (map + (cons 0 a) (append a (list 0))))
  (if (= n 0)
      '(1)
      (calc-new (pascal (- n 1)))))

```

Abgabe. Donnerstag, den 7. Oktober 2004, vor der Vorlesung oder (besser) per Email an urban@mathematik.uni-muenchen.de