

# Funktionale Programmierung SS 2013 - Lösung 6

Prof. Dr. U. Kastens

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

14.06.2013

## Lösung zu Aufgabe 1

a)

```
fun from k = Cons(k, fn() => from(k+1));
fun from2step k = Cons(k, fn() => from2step(k+2));

val nat = from 0;
val even = from2step 0;
val odd = from2step 1;
```

b)

```
fun pair2(Cons(x,xf),Cons(y,yf)) = Cons((x,y), fn()=>pair2(xf(),yf()))
| pair2(_,_) = Nil;
```

c)

```
fun pair1(Cons(x,xf)) = let val Cons(y,yf) = xf()
                        in Cons((x,y), fn()=>pair1(yf())) end
| pair1(Nil) = Nil;
```

d)

```
fun interleave3(Cons(x,xf), y, z) = Cons(x, fn() => interleave3(y, z, xf()));
```

e)

```
fun pfib(a,b) = Cons(a, fn () => pfib(b, a+b));
val fibs = pfib(1,1);
```

f)

```
fun fibs100(Cons(x,xq)) = if x < 100 then x :: fibs100(xq())
                          else []
| fibs100(Nil) = [];
```

## Lösung zu Aufgabe 2

a)

```
fun string2CharStream (s,n) = if n >= size(s) then Nil
                              else Cons(String.sub(s,n), fn()=>string2CharStream (s,n+1));
```

b)

```
fun read (Nil) = []
| read(Cons(x,xq)) = x :: read(xq())
;
```

## Lösung zu Aufgabe 3

a)

```
fun ppow2 x = iterates (fn x => 2*x) x;
```

b)

```
val onlyletters = filter Char.isAlpha charstream;
```

## Lösung zu Aufgabe 4

```
fun vermoegen zins kap = iterates (fn(kap) => kap + kap*zins/100.0) kap;  
Signatur: real -> real -> real seq
```

Erzeugt einen Strom von reellen Zahlen, beginnend mit dem Wert `kap`, gefolgt von Werten, die jeweils um `zins` Prozent gegenüber dem Vorgänger-Wert steigen. Interessant ist hier die Verwendung der freien Variablen `zins` im Strom-Funktional `iterates`.

```
val entwicklung1 = vermoegen 2.5;  
Signatur: real -> real seq
```

Bindung eines Zins-Wertes an die Funktion `vermoegen`.

```
val konto1 = entwicklung1 1000.00;  
Signatur: real seq
```

Bindung eines Kapitalwertes an die Funktion `entwicklung1`.

```
fun howlong (Cons(x:real,xf), kap) = if x >= kap then 0  
                                   else 1 + howlong(xf(),kap)  
|   howlong(Nil) = 99999;    (* steht für niemals *)
```

ergibt 29.