

Funktionale Programmierung SS 2013 - Aufgabenblatt 6

Prof. Dr. U. Kastens

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

24.06.2013

Aufgabe 1 (Arbeiten mit Strömen)

In dieser Aufgabe sollen einfache Stromproduzenten, -konsumenten und -transformer geschrieben werden (siehe Folie 702). Verwenden Sie dazu die folgenden Definitionen:

```
datatype 'a seq = Nil | Cons of 'a * (unit-> 'a seq);
fun from k = Cons(k, fn() => from(k+1));
fun take(xq, 0) = []
  | take(Cons(x,xq),n) = x :: take(xq(),n-1)
  ;
```

- a) Definieren Sie drei Stromwerte:
- `nat`, die Folge $0, 1, 2, 3, \dots$
 - `even`, die Folge $0, 2, 4, \dots$
 - `odd`, die Folge $1, 3, 5, \dots$

Verwenden Sie jeweils `take` um sich die ersten Werte im Strom in Form einer Liste anzusehen.

Hinweis: Damit der SML-Interpreter die Ausgabe von Listenwerten nicht zu früh abbricht, können Sie folgende Direktive eingeben:

```
Control.Print.printLength := 200;
```

- b) Schreiben Sie eine Funktion `pair2(s,t)`, die aus den beiden Strömen `s` und `t` einen Strom von Paaren aus den Elementen von `s` und `t` generiert.

Beispiel: `pair2(even,odd)` ergibt $(0,1), (2,3), (4,5), (6,7), \dots$

- c) Schreiben Sie eine Funktion `pair1(s)`, die aus dem Strom `s` einen Strom von Paaren aus je zwei aufeinander folgenden Elementen aus `s` generiert.

Beispiel: `pair1(nat)` ergibt $(0,1), (2,3), (4,5), (6,7), \dots$

Hinweis: Verwenden Sie das `let`-Konstrukt, wie z.B. auf Folie 707.

- d) Schreiben Sie eine Funktion `interleave3`, die `nat`, `even` und `odd` verzahnt.
- e) Schreiben Sie eine Funktion `pfib`, die den Strom der Fibonacci-Zahlen liefert. Definieren Sie damit den Stromwert `fibs`, die Folge $1, 1, 2, 3, 5, 8, 13, 21, \dots$
- f) Schreiben Sie einen Stromkonsumenten, der aus dem Stromwert `fibs` eine Liste berechnet, die alle Elemente des Stromes enthält, die kleiner als 100 sind.

Aufgabe 2 (Verarbeitung von Character Streams)

- a) Geben ist ein Text in Form einer Zeichenkette, z. B.

```
val txt = "Standard ML is a formally defined programming language. The 'Definition " ^  
          "of Standard ML (Revised)' is the formal definition of the language.";
```

Schreiben Sie einen Stromproduzenten `string2CharStream`, der daraus einen Strom über dem Grundtyp `char` generiert. Nach dem Ende des Strings soll der Strom durch ein Nil-Element abgeschlossen werden.

Hinweis: Verwenden Sie `String.sub(s, i)` um auf den `i`-ten Buchstaben im String `s` zuzugreifen. `size(s)` liefert die Anzahl der Zeichen des Strings `s`.

- b) Schreiben Sie einen Stromkonsumenten `read(s)`, der einen solchen Strom liest und in eine Liste von Buchstaben transformiert.

Aufgabe 3 (Funktionale für Ströme)

Verwenden Sie für die folgenden Aufgaben die Funktionale auf Folie 705.

- a) Schreiben Sie mit Hilfe von `iterates` einen Strom-Erzeuger `ppow2`, der den Strom aller Zweierpotenzen `1, 2, 4, 8, ...` liefert.

Dies ist die Definition:

```
fun iterates f x = Cons(x, fn() => iterates f (f x));
```

- b) Schreiben Sie mit Hilfe von `filter` einen Strom-Filter `alpha`, der aus Zeichen-Strömen (siehe Aufg. 2) alle Nicht-Buchstaben entfernt.

Dies ist die Definition:

```
fun filter pred Nil = Nil  
  | filter pred (Cons(x,xf)) = if pred x then Cons(x, fn()=> filter pred (xf()))  
                               else filter pred (xf());
```

Hinweis: Verwenden Sie `Char.isAlpha(c)` um zu prüfen, ob `c` ein Buchstabe ist.

Aufgabe 4 (Anwendung: Zinsberechnung)

Erläutern Sie die folgenden Definitionen und geben Sie jeweils die Signatur an:

```
fun vermoegen zins kap = iterates (fn(kap) => kap + kap*zins/100.0) kap;  
val entwicklung1 = vermoegen 2.5;  
val konto1 = entwicklung1 1000.00;
```

Geben Sie eine Funktion `howlong(e, k)` an, die berechnet, nach wie vielen Jahren in der Vermögensentwicklung `e` ein Stand von `k` Euro erreicht wird.

Wie groß ist `howlong(konto1, 2000.00)`?