Prof. Dr. Manfred Schmidt-Schauß
Künstliche Intelligenz/Softwaretechnologie

Fachbereich Informatik (20)
Johann Wolfgang Goethe-Universität Frankfurt am Main

Einführung in das funktionale Programmieren

Wintersemester 1995/96

Übungsblatt Nr. 7

The sun was shining on the sea,
Shining with all his might:
He did his very best to make
The billows smooth and bright-
And this was odd, because it was
The middle of the night.

aus Lewis Carrol: Through the Looking-Glass

Aufgabe 1

In dieser Aufgabe wollen wir *list comprehensions* „entzuckern":

(1) *dotdot expressions*. Diese werden im Gofer-Manual durch die Standard-Prelude Funktionen `enumFrom`, `enumFromTo`, `enumFromThen` und `enumFromThenTo` übersetzt:

```
[n ..]          enumFrom n
[n .. m]        enumFromTo
[n, m ..]       enumFromThen n m
[n, m .. k]     enumFromThenTo n m k
```

(2) *ZF-expressions* (nach Zermelo & Fraenkel) von der Form:

```
[exp | qualifier]       "list of all exp such that qualifiers",
```

Im Gofer-Handbuch sind folgende Übersetzungsregeln angegeben:

(i)
```
[e|pat<-exp] = loop exp
    where
      loop []        = []
      loop (pat:exp) = e : loop xs
      loop (_:exp)   = loop xs
```
(ii) `[e|cond] |cond = [e] |otherwise = []`

(iii) `[e | q1,q2] = concat [ [e | q2] | q1]`

a) Entzuckern Sie die *list comprehension*

`[x | y <- [1..] , x <- [1..10]].`

b) Entzuckern Sie die *list comprehension*

`[x | y <- [1/0] , x <- [1..10]].`

c) Schreiben und entzuckern Sie eine *list comprehension* `phyt n`, so daß

$$\texttt{phyt n} = [(a,b,c) \in \{1,\ldots,n\} \times \{1,\ldots,n\} \times \{1,\ldots,n\} \mid a < b < c \text{ und } a^2 + b^2 = c^2]$$

Aufgabe 2
Programmieren Sie die Funktion `map`, indem sie

a) eine *List-comprehension* benutzen.

a) die Funktion `foldr` benutzen.

Aufgabe 3
Wir wollen in dieser Aufgabe einfache I/O-Operationen üben.
Schreiben Sie einen kleinen Zeileneditor in Gofer. Er soll folgende Befehle verstehen:

- `:l datei` Es soll die Datei mit Namen `datei` in den Buffer geladen werden.
- `:d n` Es soll die Zeile der Nummer `n` der im Buffer befindlichen Datei angezeigt werden.
- `:a n string` Es soll hinter die Zeile der Nummer `n` der im Buffer befindlichen Datei die neue Zeile `string` eingefügt werden.
- `:c n string` Es soll die Zeile der Nummer `n` der im Buffer befindlichen Datei ersetzt werden durch die neue Zeile `string`
- `:?` Es soll die Anzahl der Zeilen die die Datei im Buffer enthält angezeigt werden.
- `:s` Es soll der Inhalt der im Buffer befindlichen Datei angezeigt werden.
- `:dd n` Es soll die Zeile der Nummer `n` der im Buffer befindlichen Datei gelöscht werden.
- `:s datei` Es soll der Inhalt der im Buffer befindlichen Datei als die Datei `datei` abgespeichert werden.
- `:q` Das Programm soll verlassen werden.