Institut für Informatik
der Universität München
Prof. Dr. M. Hofmann
Dr. J. Johannsen, Dr. M. Lange

WS 03/04

18.12.03

Übungen zur Vorlesung
Informatik I

Blatt 9

Abgabe der Hausaufgaben spätestens am 12.01.04, 11:00 Uhr. Programmieraufgaben über
`http://miles.tcs.informatik.uni-muenchen.de/info1/abgabe.php`,
schriftliche Aufgaben auf Papier im Briefkasten in der Theresienstraße 39, 1. Stock. Notieren Sie Namen, Matrikelnummern und Ihre Übungsgruppe auf den Blättern. Bearbeitung in Gruppen zu max. 3 Personen ist zulässig. Besprechung der Aufgaben in den Übungen ab 19.01.04.

Schriftliche Aufgabe S-35: **4 Punkte**
Gegeben sei folgende Definition einer Ocaml-Funktion.

```
let rec append l k = match l with []   -> k
                                | h::t -> h::(append t k) ;;
```

Zeigen Sie, dass für alle Listen l_1 und l_2 gilt:

$$\texttt{append}\, l_1\; l_2 \;=\; l_1@l_2$$

Programmieraufgabe P-36 (`figur.ml`): **4 Punkte**
Eine Strecke ist definiert durch zwei Punkte, ein Polygon durch beliebig viele Punkte. Erweitern Sie die Variante `figur` auf Folie 212 um zwei Konstruktoren `Strecke` und `Polygon`.

Schreiben Sie eine Funktion `umfang: figur` $\rightarrow$ `float`, die zu einer geometrischen Figur ihren Umfang berechnet. Dabei ist

- der Umfang eines Kreises mit Radius r gleich $\pi \cdot r \cdot 2$.
- der Umfang einer Strecke zwischen (x_1, y_1) und (x_2, y_2) gleich ihrer Länge. Nach dem Satz des Pythagoras ist dies $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$. In Ocaml gibt es eine Funktion `sqrt: float` $\rightarrow$ `float`.
- der Umfang eines Dreiecks, eines Rechtecks und eines Polygons gleich der Summe der Längen der Strecken, die jeweils zwei benachbarte Punkte verbinden. Ein Polygon, das aus einem oder keinem Punkt besteht, hat den Umfang 0.

Programmieraufgabe P-37 (`teilliste.ml`): **4 Punkte**
Implementieren Sie eine Funktion `teilliste:` $'a$ `list` $\rightarrow$ `int` $\rightarrow$ `int` $\rightarrow$ $'a$ `list`, die als Argumente eine Liste l und zwei Zahlen n und m nimmt und diejenige Teilliste von l liefert, die mit dem n-ten Element beginnt und m Elemente lang ist. Desweiteren gilt:

1. Die Numerierung der Elemente von l beginnt mit 0.
2. Hat l kein n-tes Element, so soll eine einfache Exception ausgelöst werden. Dasselbe gilt für den Fall, das beginnend mit dem n-ten Element keine m Elemente in l vorhanden sind.

3. Ist $m = 0$ so soll in jedem Fall die leere Liste zurückgeliefert werden, auch wenn n kein Element in l definiert.

Hinweis: Es empfiehlt sich, zuerst zwei Funktionen zu definieren, die von einer Liste den Rest ab Position n, bzw. die ersten m Elemente berechnen.

Programmieraufgabe P-38 (`wortliste.ml`): **8 Punkte**

Schreiben Sie eine Funktion `wortliste: string -> (string * int) list`, die zu einem Text, der als Zeichenkette eingegeben wird, eine alphabetisch sortierte Liste der darin vorkommenden Wörter mit ihren Häufigkeiten erstellt.

Dabei gilt als Wort jeder maximale Teilstring, der nur aus Buchstaben besteht. Alle anderen Zeichen sollen ignoriert werden. Wörter, die sich nur durch Groß-/Kleinschreibung unterscheiden, sollen nicht unterschieden werden.

Für die folgende Eingabe:

```
"<head>Ocaml $$ list* typ array 34Typ oCaMl55
7c3ocaml 7676 </head> "
```

sollte beispielsweise die Ausgabe wie folgt aussehen:

```
[("array", 1); ("c", 1); ("head", 2); ("list", 1); ("ocaml",
3); ("typ", 2)]
```

Hinweis: Sie werden vermutlich mehrere Hilfsfunktionen programmieren müssen. Sie können die Funktionen `anfang: string -> char` und `rest: string -> string` aus der Lösung zur Aufgabe P-19 verwenden. Die Funktion `String.lowercase: string -> string` ersetzt in einer Zeichenkette jeden Großbuchstaben durch den entsprechenden Kleinbuchstaben.