Christian-Albrechts-Universität zu Kiel
Institut für Informatik
Lehrstuhl für Programmiersprachen und Übersetzerkonstruktion
Prof. Dr. M. Hanus, F. Reck

Institut für Informatik

7. Übung zur Vorlesung „Informatik I“

Abgabe am Freitag, 12. Dezember, 14 Uhr

Hinweis: Am Mittwoch den 17. Dezember, 16.15 Uhr findet in CAP.3-R.2 eine Übungsklausur statt. Die Teilnahme an der Übungsklausur ist verpflichtend. **In Ausnahmefällen beim Übungsleiter bis Mittwoch den 10. Dezember mit einer Begründung abmelden!** Die Übungsklausur wird anstelle der Übungsabgaben in dieser Woche bewertet.

Präsenzaufgabe 1

Betrachten Sie die Funktion:

```
(define (foldr f x l)
  (if  (empty? l) x
       (f (first l) (foldr f x (rest l)))))
```

(a) Was berechnet `(define (sum l) (foldr + 0 l))`?
Was ist im Allgemeinen der Wert von `(foldr f x (list x1 ... xn))`?

(b) Verwenden Sie `foldr` um die Funktionen `(map-list f l)` und `(filter-list ? l)` zu definieren.

Präsenzaufgabe 2

Ein Array ist ein abstrakter Datentyp, der Indizes auf Werte abbildet. Die Signatur dieses Datentyps besteht aus einer Konstanten `emptyarray`, die das leere Array repräsentiert, einer Funktion `putindex`, die einem Index im Array einen Wert zuordnet und einer Funktion `getindex`, die den Wert zu einem Index aus einem Array ausliest.
In der Vorlesung haben Sie eine Implementierung von Listen durch Funktionen höherer Ordnung kennengelernt. Überlegen Sie sich eine solche Implementierung für den abstrakten Datentyp Array.

Aufgabe 3 (Programmieraufgabe) 2+2+2 Punkte

Definieren Sie eine Funktion `(reverse-list l)`, die eine Liste mit den Elementen von `l` in umgekehrter Reihenfolge zurückgibt. Welche Laufzeit und welchen Speicherbedarf hat Ihre Lösung? Geben Sie (auch) eine Implementierung mit linearer Laufzeit an.

Aufgabe 4 (Programmieraufgabe) 6 Punkte

Definieren Sie eine Funktion `(sublist? xs ys)`, die testet, ob `xs` eine Teilliste von `ys` ist. Dabei ist `xs` genau dann eine Teilliste von `ys`, wenn es Listen `prefix` und `suffix` gibt, so dass

```
ys = (concat (list prefix xs suffix))
```

gilt.

Aufgabe 5 (Programmieraufgabe) 8 Punkte

Verdeutlichen Sie sich zunächst den Unterschied zwischen mehrstelligen Funktionen und geschachtelten Lambda-Ausdrücken der Form:

```
(lambda (x) (lambda (y) (lambda (z) ...)))
```

Definieren Sie dann eine Funktion `(magic n)`, die eine natürliche Zahl $n > 0$ als Parameter hat. Das Ergebnis von `(magic n)` soll nacheinander n Parameter verarbeiten und die Summe dieser Parameter zurückliefern. Zum Beispiel soll der Ausdruck

```
((((magic 3) 2) 7) 3)
```

zu 12 ausgewertet werden.