Musterlösung XII. Übungsblatt “Informatik I” WS 95/96

Christian Schulte

Aufgabe XII.5 (Umordnen von Strömen)

1. Siehe Programmdatei.
2. Beide evaluieren zu Strömen, die dieselben Elemente wie `s` in derselben Reihenfolge enthalten.
3. Die ersten sieben Zahlen sind: 1, 1, 1, 2, 5, 21, 233 (Siehe auch Programmdatei).

Aufgabe XII.6 (Akkumulieren über Listen)

1.
```
(define (foldl p i xs)
  (if (null? xs) i
      (foldl p (p i (car xs)) (cdr xs))))
```
2. Die durch die Prozedur implementierte Funktion f muß sich “umklammern” lassen, das heißt es muß gelten:
$$f(x, f(y, z)) = f(f(x, y), z)$$
Das bedeutet gerade, daß die Funktion f assoziativ sein muß.
3. `foldl` ist effizienter, die Summe kann iterativ berechnet werden. Dies ist für `foldr` nicht der Fall.
4. Hier ist `foldr` effizienter. Der Aufwand der Prozedur `append` hängt von der Länge der Liste ab, die als erstes Argument übergeben wird. Bei `foldr` ist dieses erste Argument immer ein Element der Ausgangsliste `xs`. Bei `foldl` ist es dagegen immer das Zwischenergebnis für den bereits berechneten Rest der Ausgangsliste `xs`. Wenn die Summe über die Länge der Listen in `xs` gleich n ist, so hat die Version mit `foldl` einen Zeitaufwand von $O(n^2)$, die Version mit `foldr` dagegen einen Zeitaufwand von $O(n)$

Alle anderen Lösungen befinden sich in der Programmdatei.