5. Übungsaufgabe zu
Fortgeschrittene Funktionale Programmierung
Themen: Spezifikationsbasiertes Testen
ausgegeben: 19.06.2006, fällig: 28.06.2006

Für dieses Aufgabenblatt sollen Sie Haskell-Rechenvorschriften zur Lösung der im folgenden angegebenen Aufgabenstellungen entwickeln und für die Abgabe in einer Datei namens `AufgabeFFP5.hs` in Ihrem Homeverzeichnis ablegen, wie gewohnt auf oberstem Niveau. Kommentieren Sie Ihre Programme aussagekräftig und benutzen Sie, wo sinnvoll, Hilfsfunktionen und Konstanten.

Ein Editorpuffer kann als Zeichenreihe zusammen mit einer Cursorposition in folgender Weise in Haskell modelliert werden:

```
type Buffer = (Int,String)
empty   :: Buffer                   -- the empty buffer
insert  :: Char -> Buffer -> Buffer -- insert character before cursor
delete  :: Buffer -> Buffer         -- delete character before cursor
left    :: Buffer -> Buffer         -- move cursor left one character
right   :: Buffer -> Buffer         -- move cursor right one character
atLeft  :: Buffer -> Bool           -- is cursor at left end?
atRight :: Buffer -> Bool           -- is cursor at right end?
```

- Geben Sie Implementierungen für die obigen Funktionen eines Editorpuffers an.
- Eine Modellierung des Editorpuffers mithilfe des Haskelltyps

  ```
  type BufferI = (String,String)
  ```

  wobei die erste Zeichenreihe in umgekehrter Reihenfolge die Zeichen bis zur Cursorposition, die zweite Zeichenreihe die Zeichen ab der Cursorposition enthält (einschließlich des Zeichens an der Cursorposition), erlaubt eine effizientere Implementierung der Pufferoperationen.

  Geben Sie Implementierungen für die obigen Pufferfunktionen auf dem neuen Datentyp `BufferI` an, die mit `emptyI`, `insertI`, usw. bezeichnet werden sollen.

- Definieren Sie eine Haskell-Rechenvorschrift `retrieve` mit der Signatur `retrieve :: BufferI -> Buffer`, die Puffer in effizienter Darstellung in ihr bedeutungsgleiches Äquivalent in Standarddarstellung überführt.
- Überprüfen Sie mithilfe von `QuickCheck` die Korrektheit der Implementierung der Pufferfunktionen auf `BufferI` bezüglich der entsprechenden Funktionen auf `Buffer`, die wir zu diesem Zweck als Spezifikation der Pufferoperationen auffassen. Definieren Sie dazu analog zum Beispiel über Schlangen aus der Vorlesung in Ihrem Haskell-Programm für jede der obigen Funktionen ein (oder mehrere) entsprechende Eigenschaft(en) und übernehmen Sie für die Bezeichnung dieser Eigenschaften die Namenskonventionen aus dem Beispiel über Schlangen aus der Vorlesung.