Institut für Informatik
und Praktische Mathematik
der Universität Kiel

Kiel, 18.11.1999

Übungen zu Informatik I
WS 1999/2000

Serie 3
Programmierübungen in KiR

Aufgabe P6 (12 Punkte)

Schreiben Sie KiR-Programme, die folgende Operationen auf Listen von Zahlen ausführen:

- Umkehren der Reihenfolge der Elemente:

$$\begin{aligned} \texttt{reverse [<>]} &= \texttt{<>} \\ \texttt{reverse [<}u\texttt{>]} &= \texttt{<}u\texttt{>} \\ \texttt{reverse [<}u_1\texttt{, } u_2\texttt{, } \ldots\texttt{, } u_{n-1}\texttt{, } u_n\texttt{>]} &= \texttt{<}u_n\texttt{, } u_{n-1}\texttt{, } \ldots\texttt{, } u_2\texttt{, } u_1\texttt{>} \quad . \end{aligned}$$

- Aussortieren mehrfacher Vorkommen von Elementen:

$$\begin{aligned} \texttt{singleton [<>]} &= \texttt{<>} \\ \texttt{singleton [<}u\texttt{>]} &= \texttt{<}u\texttt{>} \\ \texttt{singleton [<}\ldots\texttt{, } u_i\texttt{, } \ldots\texttt{, } u_j\texttt{, } \ldots\texttt{>]} &= \texttt{<}\ldots\texttt{, } u_i\texttt{, } \ldots\texttt{>} \quad \text{falls } u_i = u_j \quad . \end{aligned}$$

- Paarung von Elementen zweier Listen:

$$\begin{aligned} &\texttt{pair [<}u_1\texttt{, } u_2\texttt{, } \ldots\texttt{, } u_m\texttt{>, <}v_1\texttt{, } v_2\texttt{, } \ldots\texttt{, } v_n\texttt{>]} \;= \\ &\qquad \texttt{< <}u_1\texttt{, } v_1\texttt{>, <}u_2\texttt{, } v_2\texttt{>, } \ldots\texttt{, <}u_m\texttt{, } v_m\texttt{>, <}v_{m+1}\texttt{, } \ldots\texttt{, } v_n\texttt{> >} \quad \text{falls } n \geq m \\ &\qquad\qquad (\text{für } n < m \text{ entsprechend}) \quad . \end{aligned}$$

- Einsortieren der Elemente einer (ggf. zu sortierenden) Liste in eine andere (ggf. zu sortierende) Liste:

$$\texttt{merge [<}u_1\texttt{, } u_2\texttt{, } \ldots\texttt{, } u_m\texttt{>, <}v_1\texttt{, } v_2\texttt{, } \ldots\texttt{, } v_n\texttt{>]} \;=\; \texttt{<}z_1\texttt{, } z_2\texttt{, } \ldots\texttt{, } z_{m+n}\texttt{>}$$

wobei $\{u_1, \ldots, u_m, v_1, \ldots, v_n\} = \{z_1, \ldots, z_{m+n}\}$ gilt und die z_k aufsteigend sortiert sind.

Testen Sie Ihre Programme an Hand geeigneter Argumentlisten.

Aufgabe P7 (8 Punkte)

Gegeben seien sogenannte binäre Bäume, die wie folgt konstruiert sind:

$$\mathit{bin\_tree} \quad \Rightarrow \quad \mathit{atom} \quad | \quad \texttt{<}\ \mathit{bin\_tree}\ \texttt{,}\ \mathit{bin\_tree}\ \texttt{>}$$

Beispiel: `< <`a_1`, <`a_2`, `a_3`> >, < <`a_4`, `a_5`>, <`a_6`, `a_7`> > >`

- Schreiben Sie ein KiR-Programm, das diese Baumstrukturen in 'flache' Listendarstellungen umwandelt, indem eine Funktion

  `traverse [` atom `]` `=` atom
  `traverse [ <`tree_1`, `tree_2`> ]` `=` `'kon' traverse[`tree_1`] traverse[`tree_2`]`

  benutzt wird. `'kon'` ist dabei ein sogenannter Konstruktur, der als Zeichenkette realisiert wird. Man beachte, daß die obige Darstellung in eine Listenstruktur einzubetten ist, siehe das nachfolgende Beispiel.

  Das obige Beispiel transformiert sich durch `traverse` zu:

  `<'kon', 'kon', `a_1`, 'kon', `a_2`, `a_3`, 'kon', 'kon', `a_4`, `a_5`, 'kon', `a_6`, `a_7`>`

- Schreiben Sie ein weiteres KiR-Programm, das diese sogenannte pre-order-linearisierte Darstellung der Baumstruktur wieder in die ursprüngliche Form zurücktransformiert.

Abgabe: Do, 2.12.1999 vor der Vorlesung.