

Übungen zu Informatik I

Aufgabe 9-1 Map und Faltung für Reihungen (keine Abgabe)

In der folgenden Aufgabe sollen für Listen definierte Funktionen auf Reihungen übertragen werden. Sie können davon ausgehen, dass alle Reihungen mindestens ein Element enthalten.

- a) Schreiben Sie eine SML-Funktion *mapvect* vom Typ $('a \rightarrow 'b) \rightarrow 'a \text{ vect} \rightarrow 'b \text{ vect}$, welche analog zur Funktion *map* bei Listen definiert ist. Ein Aufruf *mapvect f v* liefert also eine neue Reihung zurück, deren Elemente durch komponentenweise Anwendung von *f* auf die Elemente von *v* entstehen.
- b) Schreiben Sie eine SML-Funktion *finit* vom Typ $\text{int} * (\text{int} \rightarrow 'a) \rightarrow 'a \text{ vect}$ mit folgender Eigenschaft: Ein Aufruf *finit(n, f)* gibt eine Reihung der Länge *n* zurück, deren Elemente die Werte $f(1), \dots, f(n)$ haben. Definieren Sie mit Hilfe der Funktion *finit* eine Funktion *linit* vom Typ $'a \text{ list} \rightarrow 'a \text{ vect}$, für die ein Aufruf der Form *linit l* eine Reihung zurückgibt, die die gleichen Elemente wie *l* (in der gleichen Reihenfolge) enthält.
- c) Schreiben Sie eine SML-Funktion *foldlvect* vom Typ $('a * 'b \rightarrow 'b) \rightarrow 'b \rightarrow 'a \text{ vect} \rightarrow 'b$, welche, analog zur Faltung von Listen, die Linksfaltung einer Reihung berechnet.
- d) Das *Skalarprodukt* einer reellen Zahl mit einem Vektor reeller Zahlen ist definiert durch $x \cdot (y_1, \dots, y_n) = (x \cdot y_1, \dots, x \cdot y_n)$. Schreiben Sie eine SML-Funktion *sprod* vom Typ $\text{real} \rightarrow \text{real vect} \rightarrow \text{real vect}$, die das Skalarprodukt einer reellen Zahl mit einer Reihung berechnet.
- e) Die Norm $\|x\|$ eines Vektors (x_1, \dots, x_n) aus \mathbb{R}^n ist definiert durch $\|x\| = \sqrt{x_1^2 + \dots + x_n^2}$. Schreiben Sie eine SML-Funktion *norm*, die die Norm einer Reihung berechnet.

Aufgabe 9-2 Faltung, Quantoren und Map für Bäume (keine Abgabe)

Ähnlich wie bei Listen kann man die Faltung, Quantoren und Map auch für Binärbäume definieren.

- a) Geben Sie eine SML-Funktion *folds* vom Typ $('b * 'a * 'b \rightarrow 'b) \rightarrow 'b \rightarrow 'a \text{ bintree} \rightarrow 'b$ an, welche, analog zur Faltung von Listen, die Faltung eines Baumes berechnet. Wie ist die Faltung von Bäumen definiert?
- b) Geben Sie mit Hilfe der Faltung zwei SML-Funktionen *allbintree* und *existsbintree* vom Typ $('a \rightarrow \text{bool}) \rightarrow 'a \text{ bintree} \rightarrow \text{bool}$ an, die zu einer Funktion *p* diejenige Funktion bestimmen, die entscheidet, ob *p(x)* für alle Knoten *x* eines Binärbaums zutrifft, bzw. ob in einem Binärbaum ein Knoten *x* existiert, auf den *p(x)* zutrifft.
- c) Geben Sie eine polymorphe SML-Funktion *mapbintree* vom Typ $('a \rightarrow 'b) \rightarrow 'a \text{ bintree} \rightarrow 'b \text{ bintree}$ mit folgender Eigenschaft an: *mapbintree(f)(b)* ist derjenige Binärbaum, der entsteht, wenn jeder Knoten *x* von *b* durch den Knoten *f(x)* ersetzt wird.

Aufgabe 9-3 Reihungen (6 Punkte)

Auch in dieser Aufgabe enthalten alle vorkommenden Reihungen mindestens ein Element.

- a) Schreiben Sie eine SML-Funktion *reversevect* vom Typ $'a \text{ vect} \rightarrow 'a \text{ vect}$, für die gilt: Ein Aufruf *reversevect v* gibt die Reihung zurück, die die gleichen Elemente wie *v* in umgekehrter Reihenfolge enthält.

- b) Schreiben Sie eine SML-Funktion *zipvect* vom Typ $'a \text{ vect} \rightarrow 'b \text{ vect} \rightarrow ('a * 'b) \text{ vect}$, für die gilt: Ein Aufruf *zipvect* $v_1 v_2$ gibt eine Reihung zurück, deren i -te Komponente das Paar aus der i -ten Komponente von v_1 und der i -ten Komponente von v_2 ist, falls v_1 und v_2 die gleiche Länge haben. Andernfalls wird eine Ausnahme *differentlength* ausgelöst.

- c) Das *innere Produkt* zweier Vektoren aus \mathbb{R}^n ist definiert durch

$$(x_1, \dots, x_n) \cdot (y_1, \dots, y_n) = (x_1 \cdot y_1, \dots, x_n \cdot y_n).$$

Schreiben Sie eine SML-Funktion *intprod* vom Typ $real \text{ vect} \rightarrow real \text{ vect} \rightarrow real \text{ vect}$, die das innere Produkt zweier Reihungen berechnet. *Hinweis:* Die Funktion *zipvect* leistet Ihnen dabei gute Dienste.

- d) Das *Vektorprodukt* zweier Vektoren aus \mathbb{R}^3 ist definiert durch

$$(x_1, x_2, x_3) \times (y_1, y_2, y_3) = (x_2 \cdot y_3 - y_2 \cdot x_3, y_1 \cdot x_3 - x_1 \cdot y_3, x_1 \cdot y_2 - y_1 \cdot x_2).$$

Schreiben Sie eine SML-Funktion *intprod* vom Typ $real \text{ vect} \rightarrow real \text{ vect} \rightarrow real \text{ vect}$, die das Vektorprodukt zweier Reihungen der Länge 3 berechnet und eine Ausnahme *wronglength* auslöst, falls (mindestens) eine Reihung eine andere Länge hat.

- e) Schreiben Sie eine SML-Funktion *foldrvect* vom Typ $('a * 'b \rightarrow 'b) \rightarrow 'b \rightarrow 'a \text{ vect} \rightarrow 'b$, welche, analog zur Faltung von Listen, die Rechtsfaltung einer Reihung berechnet.
- f) Schreiben Sie SML-Funktionen *existsvect* und *allvect* vom Typ $('a \rightarrow \text{bool}) \rightarrow 'a \text{ vect} \rightarrow \text{bool}$, die zu einer Funktion p diejenige Funktion bestimmt, die entscheidet, ob $p(x)$ für alle Komponenten x einer Reihung zutrifft, bzw. ob in einer Reihung eine Komponente x existiert, auf die $p(x)$ zutrifft.

Aufgabe 9-4

Geschmückte Bäume

(6 Punkte)

Gegeben sei **datatype** *dekor* = *Stern* | *Kugel of int* | *Kerze of bool*. Hierbei steht *Kerze(true)* für eine brennende Kerze, *Kerze(false)* für eine Kerze, die nicht brennt und *Kugel(n)* für eine Kugel der Größe n .

- a) Geben sie zwei SML-Funktionen *brennend* und *gelöscht* vom Typ *dekor bintree* \rightarrow **bool** an, die bestimmen, ob alle Kerzen des Eingabebaums brennen bzw. ob alle Kerzen des Eingabebaums gelöscht sind.
- b) Geben Sie eine SML-Funktion *maxkug* vom Typ *dekor bintree* \rightarrow **int** an, welche die Größe der größten Kugel des Eingabebaums bestimmt. Hängen keine Kugeln am Eingabebaum, so soll *maxkug* den Wert 0 zurückliefern.
- c) Geben Sie eine SML-Funktion *kugelstern* vom Typ *dekor bintree* \rightarrow *dekor bintree* an, die jede Kugel durch einen Stern ersetzt.
- d) Geben Sie eine SML-Funktion *blattanz* vom Typ *dekor bintree* \rightarrow **int** an, welche die Anzahl der Blätter des Eingabebaums bestimmt.
- e) Geben Sie eine SML-Funktion *anzkug* vom Typ *dekor bintree* \rightarrow **int** an, welche die Anzahl der Kugeln des Eingabebaums berechnet.
- f) Der Preis einer Kerze beträgt 5 €, eine Kugel der Größe n kostet $2 \cdot n$ €, und der Erwerb eines Sterns schlägt mit 10 € zu Buche. Geben Sie eine SML-Funktion *kosten* vom Typ *dekor bintree* \rightarrow **int** an, die die Summe der Preise der Dekorationen eines Baumes bestimmt.

Hinweise zur Verwendung der Reihungen und Binärbäume aus der Vorlesung finden sich auf der InfoI-Homepage.

Abgabe: Dienstag, 17.12.2002, 12:00 Uhr