Ludwig-Maximilians-Universität München WS 2005/2006
Institut für Informatik Blatt 9
Prof. Dr. F. Kröger, Dr. M. Hölzl, F. Hacklinger

Übungen zu Informatik I

Aufgabe 9-1 Pfade in Bäumen (keine Abgabe)

Ein *Pfad* zu einem Knoten k beschreibt den „Weg“, den man in einem Binärbaum von der Wurzel aus zurücklegen muss, um zu k zu gelangen. In SML kann man Pfade durch folgende Datentypen realisieren:

```
datatype step = Left | Right;
type path = step list;
```

Es gibt genau einen Pfad zu jedem Knoten in einem Baum. Die *Tiefe* $T(k)$ eines Knotens k ist die Länge des Pfades zu k. Ein Knoten k_1 ist *links von* einem Knoten k_2, wenn für die Pfade p_1 zu k_1 und p_2 zu k_2 gilt: (1) Die Pfade stimmen auf einem (möglicherweise leeren) Anfangsstück überein. (2) An der ersten Stelle, an der p_1 nicht mit p_2 übereinstimmt, enthält p_1 den Wert *Left* und p_2 den Wert *Right*.

Geben Sie eine SML-Funktion *target_node* vom Typ $'a$ **bintree** $\rightarrow$ **path** $\rightarrow$ $'a$ **bintree** mit folgenden Eigenschaften an:

- Falls in einem Aufruf *target_node b p* der Pfad p zu einem Knoten k von b führt, so wird k zurückgegeben.
- Andernfalls wird eine Ausnahme *No_Node_For_Path* ausgelöst.

Aufgabe 9-2 Breitensuche für Bäume (keine Abgabe)

Geben Sie eine SML-Funktion *breadth_first_lin* vom Typ $'a$ **bintree** $\rightarrow$ $'a$ **list** mit folgender Eigenschaft an: Ein Aufruf von *breadth_first_lin b* liefert die Liste der Knoten von b in „Breitenordnung“. Die „Breitenordnung“ $\prec$ ist die Relation, in der für alle Knoten k_1, k_2 gilt:

- Falls $T(k_1) < T(k_2)$, dann $k_1 \prec k_2$.
- Falls $T(k1) = T(k_2)$ so gilt $k_1 \prec k_2$ genau dann, wenn k_1 im Baum links von k_2 vorkommt.

Aufgabe 9-3 Funktionen für Bäume (6 Punkte)

a) Geben Sie eine SML-Funktion *istblatt* vom Typ $'a$ **bintree** $\rightarrow$ **bool** an, die bestimmt, ob der Eingabebaum ein Blatt ist.

b) Geben Sie eine SML-Funktion *blattanz* vom Typ $'a$ **bintree** $\rightarrow$ **int** an, welche die Anzahl der Blätter des Eingabebaums bestimmt.

c) Geben Sie eine SML-Funktion *nichtblattanz0* vom Typ **int bintree** $\rightarrow$ **int** an, welche die Anzahl der Knoten k des Eingabebaums bestimmt, die keine Blätter sind und für die gilt $root(k) > 0$.

d) Geben Sie eine SML-Funktion *allbintree* vom Typ $('a \rightarrow$ **bool**$) \rightarrow$ $'a$ **bintree** $\rightarrow$ **bool** an, die zu einer Funktion p diejenige Funktion bestimmt, die entscheidet, ob $p(x)$ für alle Knoten x eines Binärbaums 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 *geloescht* vom Typ *dekor* **bintree** → **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** → **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** → *dekor* **bintree** an, die jede Kugel des Eingabebaums durch einen Stern ersetzt.

d) Geben Sie eine SML-Funktion *anzkug* vom Typ *dekor* **bintree** → **int** an, welche die Anzahl der Kugeln des Eingabebaums berechnet.

e) 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** → **int** an, die die Summe der Preise der Dekorationen eines Baumes bestimmt.

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

Abgabe: Montag, 16.01.2006, 8:00 Uhr