

Aufgaben zu Haskell

Aufgabe 1

1. Definieren Sie einen Datentyp, der einen binären Baum darstellt. Die Daten sollen nur in den inneren Knoten, nicht in Blättern gespeichert werden. Halten Sie den gespeicherten Datentyp polymorph.

Geben Sie in den folgenden Teilaufgaben für alle geforderten Funktionen zusätzlich den allgemeinstmöglichen Typen inklusive eventuell vorhandener Typklasseneinschränkungen an.

2. Schreiben Sie eine Funktion `search`, die angibt, ob ein gegebenes Element in einem Baum vorkommt.

Im Folgenden befassen wir uns mit binären Suchbäumen. In binären Suchbäumen gilt für jeden inneren Knoten, dass die im linken Unterbaum gespeicherten Elemente kleiner oder gleich, die im rechten Unterbaum größer oder gleich dem im Knoten gespeicherten Element sind. Nehmen Sie für die folgenden Teilaufgaben an, dass die als Parameter übergebenen Bäume die Suchbaumeigenschaft erfüllen (Sie müssen die Eigenschaft nicht nachprüfen).

3. Geben Sie nun eine Funktion `search'` an, die angibt, ob ein gegebenes Element in einem Suchbaum vorkommt. Die Laufzeit der Funktion muss linear in der Höhe des Baumes sein.
4. Geben Sie eine Funktion `insert` an, die in einen Baum ein gegebenes Element einfügt. Elemente dürfen mehrfach im Baum vorkommen. Das Ergebnis der Funktion muss wieder ein gültiger Suchbaum sein.
5. Geben Sie eine Funktion `list2Tree` an, die eine (als endlich angenommene) Liste in einen binären Suchbaum umwandelt.
6. Geben Sie eine Funktion `tree2List` an, die einen Baum in eine Liste umwandelt. Die ausgegebene Liste muss aufsteigend sortiert sein.
7. Verwenden Sie die in den vorigen Teilaufgaben definierten Funktionen, um eine Funktion zu schreiben, die eine Liste sortiert.