Algorithmen und Programmieren 3, WS 1999/2000 — 4. Übungsblatt

Abgabe bis Dienstag, 23. November 1999

---

19. (0 Punkte) Definieren Sie die arithmetischen Operationen für die ganzen Zahlen, die um die Werte $\pm\infty$ und *undefiniert* erweitert wurden. Es sollten die erwarteten Rechenregeln gelten, zum Beispiel $a + (-\infty) = -\infty$, $a/(+\infty) = 0$, für $a \in \mathbb{Z}$, $(+\infty) + (-\infty) = undef$, usw. Verwenden Sie dazu einen algebraischen Typ, der etwa so aussehen könnte:

```
data Int' = Normal Int
     | PlusUnendlich | MinusUnendlich | Undefiniert deriving ....
```

20. (7 Punkte) Definieren Sie in Haskell einen algebraischen Datentyp `Menge a` für Mengen von Elementen des Typs `a`, die wahlweise durch Aufzählen der Elemente oder durch Angabe einer charakteristischen Eigenschaft definiert werden können. (Sie könne sich dabei an der vorigen Aufgabe orientieren.) Schreiben Sie folgende Funktionen:

```
mengeausListe      :: [a] -> Menge a            -- {x1, x2, ..., xn}
mengefürdiegilt    :: (a -> Bool) -> Menge a    -- { x | f(x) }
mengeDurchschnitt :: Menge a -> Menge a -> Menge a
mengeVereinigung   :: Menge a -> Menge a -> Menge a
mengeElementvon    :: a -> Menge a -> Bool      -- x ∈ M
```

21. (0 Punkte) Zwei Bäume heißen *isomorph*, wenn sie dieselbe Struktur (unabhängig von den Knotenbeschriftungen) haben. Definieren Sie ein Funktion
`isomorph :: Tree a -> Tree b -> Bool`
zum Testen, ob zwei Bäume isomorph sind.

22. (0 Punkte) Ein *vollständiger* binärer Baum der Höhe h ist ein Baum, bei dem jeder Knoten mit Tiefe $\leq h-2$ genau zwei Kinder hat.
   (a) Zeichnen Sie vollständige binäre Bäume mit 12 und mit 15 Knoten.
   (b) Wieviele Knoten kann ein vollständiger binärer Baum der Höhe h (mindestens und höchstens) haben?
   (c) Welche Höhe kann ein vollständiger binärer Baum mit n Knoten haben?

23. (15 Punkte) Das *Ungleichgewicht* eines Knotens x in einem binären Baum ist definiert als

$$|H(l(x)) - H(r(x))|,$$

wobei $l(x)$ und $r(x)$ der linke und der rechte Teilbaum von x ist und $H(b)$ die Höhe eines Baumes b ist. Das Ungleichgewicht eines *binären Baumes* ist das maximale Ungleichgewicht aller Knoten des Baumes.

   (a) (3 Punkte) Schreiben Sie ein Programm in Haskell, das das Ungleichgewicht eines Baumes berechnet.
   (b) (7 Punkte) Erweitern Sie die Java-Klasse `Baum` aus der Vorlesung, sodass man das Ungleichgewicht des Baumes (mit einer neuen Methode `getUngleichgewicht`) jederzeit ablesen kann, ohne es neu zu berechnen. (Hinweis: In der neuen Klasse sollte jeder Knoten ein Feld mit seinem Ungleichgewicht enthalten, und möglicherweise noch andere Felder. Gegebenenfalls müssen Sie die Methoden `einfügen` und `entfernen` abändern.)
   (c) (1 Punkt) Wie viele Knoten kann ein binärer Baum der Höhe h haben, dessen Ungleichgewicht 0 ist?
   (d) (4 Punkte) Ein *AVL-Baum* ist ein binärer Baum, dessen Ungleichgewicht höchstens 1 ist (nach Adel'son-Velski und Landis). Wie viele Knoten kann ein AVL-Baum der Höhe h (mindestens und höchstens) haben, für $h \leq 7$?
   (e) (freiwillige Zusatzfrage, 3 Punkte) Welche Höhe kann ein AVL-Baum mit n Knoten (mindestens und höchstens) haben?