Serge Autexier
Christoph Lüth

4. Übungsblatt

Ausgabe: 30.05.13
Abgabe: 13.06.13

4.1 *Bäume* *10 Punkte*

Bäume sind neben Listen einer der populärsten Datentypen der Informatik. In Isabelle/HOL können Binäre Bäume als algebraischer Datentyp definiert werden (genau wie in Haskell):

```
datatype 'a tree = Node 'a "'a tree" "'a tree"
                 | Leaf
```

(i) Definieren Sie für diesen Datentypen `tree` eine Funktion `map` (mit der offensichtlichen Semantik), und zeigen Sie das sogenannte *map fusion lemma* (wobei `o` die Komposition zweier Funktionen darstellt)

`map (f o g) = map f o map g.`

(ii) Definieren Sie eine Funktion

```
fun inorder :: "'a tree => 'a list"
```

welche die Liste der in dem Baum enthaltenen Knoten (in Inorder-Traversion) zurückgibt.

Jetzt wollen wir zeigen, dass diese Liste die gleichen Knoten wie der Baum enthält. Dazu definieren wir zwei Funktionen

```
tcount :: "'a tree => 'a=> nat"
lcount :: "'a list => 'a=> nat"
```

welche zählen, wie oft ein Element in einem Baum bzw. einer Liste auftritt (mit anderen Worten, sie berechnen die Multimenge der Elemente).

Formulieren und beweisen Sie mit Hilfe dieser Funktionen, dass die durch `inorder` erzeugte Liste dieselben Knoten (Multimenge der Elemente wie oben) enthält wie der ursprüngliche Baum.

Hinweis: Sie werden ein Hilfslemma über das Verhältnis von `lcount` und der Listenkonkatenation (Infix-Operator `@`) benötigen.

(iii) Beschreibt diese Spezifikation vollständig die `inorder`-Funktion? Wie könnte eine vollständige Spezifikation aussehen? Welche anderen Funktionen erfüllen diese Spezifikation?

4.2 *Punkte, Vektoren* *5 Punkte*

Punkte in einem Cartesischen Koordinatensystem sind Paare (x, y) von Zahlen, wobei x die X-Koordinate und y die Y-Koordinate angebit. Vektoren sind ebenfalls zwei Zahlen $\begin{pmatrix} u \\ v \end{pmatrix}$, wobei u die Länge auf der X-Koordinate und v die Länge auf der Y-Koordinate ist. Zwei Punkte $A = (x, y)$ und $B = (x', y')$ definieren einen Vektor von A nach B als $\overrightarrow{AB} = \begin{pmatrix} x' - x \\ y' - y \end{pmatrix}$. Das Kreuzprodukt zweier Vektoren $\begin{pmatrix} u \\ v \end{pmatrix}$ und $\begin{pmatrix} a \\ b \end{pmatrix}$ ist definiert als $\begin{pmatrix} u \\ v \end{pmatrix} \times \begin{pmatrix} a \\ b \end{pmatrix} = ub - va$. Das Kreuzprodukt erlaubt eine Charakterisierung des Winkels zwischen den beiden Vektoren: falls $\begin{pmatrix} u \\ v \end{pmatrix} \times \begin{pmatrix} a \\ b \end{pmatrix} = 0$, dann ist der Winkel zwischen beiden Vektoren 0 oder π; ist es kleiner als 0, dann ist der Winkel echt größer 0 und echt kleiner als π; und ist es größer als 0, dann ist der Winkel echt größer π und kleiner als 2π.

Ausgehend von der Theorie der rationalen Zahlen `Rat` in Isabelle/HOL, definieren Sie

(i) Datentypen für *Punkte* und *Vektoren*,

(ii) eine Funktionen zur Berechnung eines Vektors zu gegebenen zwei Punkten,

(iii) das Kreuzprodukt zweier Vektoren, und

(iv) ein Prädikat, das genau dann wahr ist für zwei Vektoren, wenn diese einen Winkel zwischen 0 und π bilden.

$\boxed{4.3}$ *Polygone* — *5 Punkte*

Ein Polygon ist eine mindestens dreielementige Liste von unterschiedlichen Punkten $P_1, \ldots P_n$. Die Seitenvektoren eines Polygons sind die Vektoren $v_i := \overrightarrow{P_i P_{i+1}}$ für $1 \leq i < n$ sowie $v_n = \overrightarrow{P_n P_1}$. Ein Polygon ist *konvex*, wenn für alle $0 \leq i < n$ gilt, dass der Winkel zwischen dem Seitenvektor v_i zum nächsten Seitenvektor v_{i+1} echt kleiner π ist.

Ausgehend von Ihrer Theorie der Punkte und Vektoren definieren Sie einen Typ für Polygone und ein Prädikate zur Kategorisierung der konvexen Polygone. Hierfür kann gegebenenfalls die Definition anderer Datentypen und Prädikate als Hilfsdatentypen und Prädikate hilfreich sein.

Revision 2196 vom 2013-06-13