Ludwig-Maximilians-Universität München
Department "Institut für Informatik"
Prof. Dr. Christian Böhm
Elke Achtert, Arthur Zimek

München, 27.10.2006

Informatik 1
WS 2006/07

Übungsblatt 2: Funktionsdefinitionen, vollständige Induktion, Rekursion

Besprechung: 06.11.–10.11.2006

Abgabe aller mit **Hausaufgabe** markierten Aufgaben bis Freitag, 03.11.2006, 18:00 Uhr

Aufgabe 2-1 *Prädikat-Funktionen in SML für Relationen* (Hausaufgabe)

Wir geben im Folgenden einige Relationen an. Zu jeder Relation $R \subseteq M \times N$ sollen Sie eine Prädikat-Funktion `istR : M * N -> bool` in SML definieren, die für ein beliebiges Element aus $M \times N$ angibt, ob es auch Element der Relation R ist.

(a) Die Relation `Quadrat` $\subseteq \mathbb{N} \times \mathbb{N}$ enthält alle Paare von natürlichen Zahlen $(x, y) \in \mathbb{N} \times \mathbb{N}$, für die gilt: $y = x^2$. Geben Sie in SML eine Prädikat-Funktion `istQuadrat : int * int -> bool` an, die für ein gegebenes Paar aus $\mathbb{N} \times \mathbb{N}$ angibt, ob es Element der Relation `quadrat` ist.

(b) Die Relation `Summe` $\subseteq \mathbb{N} \times \mathbb{N} \times \mathbb{N}$ enthält alle Tripel von natürlichen Zahlen $(x, y, z) \in \mathbb{N} \times \mathbb{N} \times \mathbb{N}$, für die gilt: $x + y = z$. Geben Sie in SML eine Prädikat-Funktion `istSumme : int * int * int -> bool` an, die für ein gegebenes Tripel aus $\mathbb{N} \times \mathbb{N} \times \mathbb{N}$ angibt, ob es Element der Relation `summe` ist.

(c) Die Relation `Negation` $\subseteq$ `bool` $\times$ `bool` enthält alle Tupel von booleschen Werten $(x, y) \in$ `bool` $\times$ `bool`, für die gilt: x ist die Negation von y. Geben Sie in SML eine Prädikat-Funktion `istNegation : bool * bool -> bool` an, die für ein gegebenes Tupel aus `bool` $\times$ `bool` angibt, ob es Element der Relation `Negation` ist.

(d) Die Relation `DeltaGleich` $\subseteq \mathbb{R}_0^+ \times \mathbb{R} \times \mathbb{R}$ enthält alle Tupel von reellen Zahlen $(\delta, x, y) \in \mathbb{R}_0^+ \times \mathbb{R} \times \mathbb{R}$, für die gilt: x und y unterscheiden sich höchstens um δ. Geben Sie in SML eine Prädikat-Funktion `istDeltaGleich : real * real * real -> bool` an, die für ein gegebenes Tripel aus $\mathbb{R}_0^+ \times \mathbb{R} \times \mathbb{R}$ angibt, ob es Element der Relation `DeltaGleich` ist.

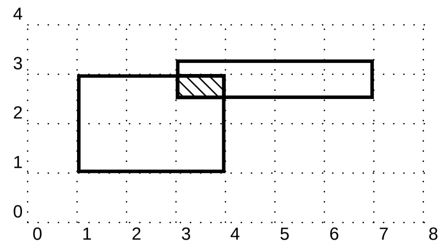
(e) Die Relation `GleicheOrdnung` $\subseteq (\mathbb{R} \times \mathbb{R}) \times (\mathbb{R} \times \mathbb{R})$ enthält alle Tupel von Paaren von reellen Zahlen $((m, n), (p, q)) \in (\mathbb{R} \times \mathbb{R}) \times (\mathbb{R} \times \mathbb{R})$, für die gilt: (m, n) stehen in der gleichen Ordnung wie (p, q). So gilt z.B. wenn $m < n$, dann $p < q$. Geben Sie in SML eine Prädikat-Funktion `istGleicheOrdnung : (real * real) * (real * real) -> bool` an, die für ein gegebenes Tupel aus $(\mathbb{R} \times \mathbb{R}) \times (\mathbb{R} \times \mathbb{R})$ angibt, ob es Element der Relation `GleicheOrdung` ist.

Ihre SML-Funktionen geben Sie bitte in einer Datei `2-1.sml` ab.

Aufgabe 2-2 *Überschneidung von Rechtecken* (Hausaufgabe)

Ein achsenparalleles Rechteck im zweidimensionalen euklidischen Raum definieren wir durch die Koordinate der linken unteren und der rechten oberen Ecke. Das untere Rechteck in der Skizze ist z.B. definiert durch die Koordinaten $(1.0, 1.0)$ und $(4.0, 3.0)$, das obere durch $(3.0, 2.5)$ und $(7.0, 3.3)$.

Ein Rechteck ist nur wohldefiniert, wenn die erste Koordinate links und unterhalb der zweiten Koordinate liegt. Wird ein Rechteck angegeben, dessen erste Koordinate rechts oder oberhalb der zweiten Koordinate liegt, soll der Flächeninhalt 0 sein.



Wenn sich zwei Rechtecke schneiden, ergibt sich ein neues Rechteck (wie das schraffierte in der Skizze). Geben Sie eine Funktion

```
flaecheSchnittRechteck :
((real * real) * (real * real)) * ((real * real) * (real * real)) -> real
```

in SML an, die zwei Rechtecke auf den Flächeninhalt des durch den Schnitt entstehenden Rechteckes abbildet.

Wenn Sie die Funktion definiert haben, sollten Sie sie auf das obige Beispiel anwenden können, z.B. mit einer Sitzung wie dieser:

```
- val r1 = ((1.0,1.0),(4.0,3.0));
val r1 = ((1.0,1.0),(4.0,3.0)) : (real * real) * (real * real)
- val r2 = ((3.0,2.5),(7.0,3.3));
val r2 = ((3.0,2.5),(7.0,3.3)) : (real * real) * (real * real)
- flaecheSchnittRechteck(r1, r2);
val it = 0.5 : real
```

Hinweis: Lösen Sie nicht alles auf einmal. Es wird leichter, wenn Sie Hilfsfunktionen schreiben, die nur Teilaufgaben lösen (z.B. eine Funktion, die den Flächeninhalt eines Rechteckes bestimmt und eine Funktion, die für zwei Rechtecke das Schnittrechteck bestimmt). Die Funktion `flaecheSchnittRechteck` ist dann einfach eine geeignete Kombination Ihrer Hilfsmethoden.

Geben Sie Ihre Lösung (mit allen eventuellen Hilfsfunktionen) in einer Datei `2-2.sml` ab.

Aufgabe 2-3 *Induktionsbeweis* **(Hausaufgabe)**

Beweisen Sie durch vollständige Induktion, dass für alle $n \in \mathbb{N}$ gilt:

(a) $\sum_{k=0}^{n} 2k = n(n+1)$

(b) $\sum_{k=0}^{n} k^2 = \frac{1}{6}n(n+1)(2n+1)$

Geben Sie die Beweise als Textdatei `2-3.txt` ab.

Aufgabe 2-4 *Zinsberechnung* **(Hausaufgabe)**

Sie wollen trotz Studiengebühren ein Vermögen ansparen und interessieren sich für Ihr Guthaben nach einer gewissen Anzahl von Jahren bzw. Monaten. Ihre Bank verzinst Ihr Guthaben mit einem festen Zinssatz z (in Prozent).

(a) Sie bezahlen zur Zeit 0 einen festen Betrag b auf Ihr Sparkonto ein. Dieser wird jährlich mit dem Zinssatz z verzinst. Wie hoch ist Ihr Guthaben nach n Jahren? Diese Frage wird von einer Funktion beantwortet, die ein Tupel $(z, b, n) \in \mathbb{R} \times \mathbb{R} \times \mathbb{N}$ auf eine reelle Zahl abbildet. Schreiben Sie eine entsprechende SML-Funktion `balance : real * real * int -> real`, um diese Funktion zu modellieren.

(b) Sie schließen mit Ihrer Bank einen Sparvertrag ab und bezahlen monatlich (beginnend mit Monat 0) einen Betrag b auf Ihr Konto ein. Die Zinsabrechnung für Ihr Konto erfolgt monatlich. Wie hoch ist der angesparte Betrag (mit Zinsen) nach n Monaten? Geben Sie eine SML-Funktion `savings` (mit der gleichen Signatur wie die Funktion `balance`) an, die zu gegebenem Jahreszinssatz z und monatlicher Einzahlung b den Stand Ihres Kontos nach n Monaten berechnet.

Geben Sie Ihre Lösungen zu dieser Aufgabe in einer Datei `2-4.sml` ab.

Hinweis: Mit Hilfe von Rekursion lassen sich diese Berechnungen einfach und elegant lösen.