



Aufgabenblatt 6

letzte Aktualisierung: 28. November, 12:04

Ausgabe: 30.11.2001

Abgabe: 9.12. / 10.12.2001 Prozent: 100

Thema: Funktionen höherer Ordnung; algebraische Datentypen

1. Aufgabe (30 Prozent): Funktionale auf Listen

1.1. Die Listenfunktionale reduce und zip (Tut) Diskutiert die Listenfunktionale `reduce` und `zip`. Implementiert die Funktionen `reduceSum` und `skalarProd` (für natürliche Zahlen) aus Aufgabenblatt 3 mit Hilfe von `reduce` und `zip`.

1.2. Linksreduktion (20 Prozent) Das Funktional `reduce` verknüpft die Listenelemente von rechts nach links. Implementiert das Funktional `fold`, das die Elemente von links nach rechts verknüpft, also

$$\text{fold}(\oplus, e)((x_1, x_2, \dots, x_n)) = (\dots((e \oplus x_1) \oplus x_2) \oplus \dots) \oplus x_n$$

Implementiert mit Hilfe dieses Funktionals die Funktion `foldDiff` aus Aufgabenblatt 3.

1.3. Verknüpfung mehrerer Sequenzen (10 Prozent) Implementiert das Funktional `zip3`, das analog zu `zip` Elemente aus drei Sequenzen durch eine gegebene Funktion verknüpft.

2. Aufgabe (20 Prozent): Algebraische Datentypen

2.1. Aufbau von Sequenzen (Tut) Diskutiert den Aufbau der Typs `seq`. Welche Signatur induziert der Typ? Welche Konstruktoren, Selektoren und Diskriminatoren gibt es?

2.2. Der Typ date (Tut) Welche Signatur induziert die folgende Datentypdeklaration?

```
TYPE date == date( day   : nat,
                  month : nat,
                  year   : nat)
```

Handelt es sich hierbei um einen Produkt- oder Summentyp?

2.3. Der Typ person (10 Prozent) Gegeben ist der folgende Datentyp:

```
TYPE person == person(name      : denotation,
                      surname   : denotation,
                      birthdate : date)
```

Welche Signatur induziert er?

2.4. Arbeiten mit algebraischen Typen (10 Prozent) Implementiert die folgenden Funktionen:

- `earlier?`, die überprüft, ob ein Wert vom Typ `date` ein früheres Datum repräsentiert als ein anderer,
- `older?`, die überprüft, ob eine Person älter als eine andere ist,
- `birthday`, die den Geburtstag (also Tag und Monat) einer Person liefert,
- `sameName?`, die überprüft, ob zwei Personen den gleichen Nachnamen haben.

3. Aufgabe (50 Prozent): Benutzerdefinierte Funktionale

Gegeben ist der folgende Datentyp:

```
TYPE safeNat == ok (value : nat)
              error (message : denotation)
```

3.1. Induzierte Signatur (10 Prozent) Welche Signatur induziert der Typ? Handelt es sich hierbei um einen Produkt- oder Summentyp?

3.2. Verwendung von safeNat (Tut) Wozu kann dieser Typ verwendet werden? Implementiert die Funktion `safePred`, die den Vorgänger eines Werts vom Typ `safeNat` oder eine Fehlermeldung liefert.

3.3. Totalisierung einstelliger Funktionen (Tut) Implementiert die Funktion `safeLift`, die eine einstellige Funktion auf natürlichen Zahlen in eine Funktion auf `safeNat` umwandelt, wobei der Definitionsbereich der Funktion durch ein Prädikat angegeben werden kann. Implementiert die Funktion `safePred` aus der vorherigen Unteraufgabe sowie die Funktion `safeSucc` mit Hilfe des neuen Funktionals.

3.4. Totalisierung zweistelliger Funktionen (20 Prozent) Implementiert die Funktion `safeLift2`, die eine zweistellige Funktion auf natürlichen Zahlen in eine Funktion auf `safeNat` umwandelt, wobei der Definitionsbereich der Funktion durch ein zweistelliges Prädikat angegeben werden kann.

3.5. Sichere Arithmetik (10 Prozent) Implementiert die Funktionen `safeAdd`, `safeSub`, `safeMult`, `safeDiv`, die die Addition, Subtraktion, Multiplikation und ganzzahlige Division für Werte vom Typ `safeNat` als totale Funktionen implementieren.

Hinweis: Verwendet wenn möglich die bereits definierten Funktionale.

3.6. safeNat in Sequenzen (10 Prozent) Implementiert das Funktional `safeMap`, das analog zu `map` eine Funktion auf alle natürlichen Zahlen in einer Sequenz anwendet und eine Sequenz von Werten vom Typ `safeNat` liefert. Der Definitionsbereich der Funktion kann durch ein Prädikat angegeben werden.

Hinweis: Diese Aufgabe kann bei Verwendung bereits definierter Funktionen durch einen Einzeler gelöst werden.