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

München, 08.12.2006

Informatik 1
WS 2006/07

Übungsblatt 8: Zeichen und Zeichenfolgen, Listen, Zusammengesetzte Typen

Besprechung: 18.12.–22.12.2006

Abgabe aller mit **Hausaufgabe** markierten Aufgaben bis Freitag, 15.12.2006, 18:00 Uhr

Aufgabe 8-1 *Zeichen und Zeichenfolgen* **(Hausaufgabe)**

Definieren Sie folgende Funktionen mit rein funktionalen Mitteln in einer Datei `8-1.sml`.

(a) `isNat : string -> bool`
Prädikat über Strings. Liefert `true`, wenn der String eine natürliche Zahl repräsentiert (d.h. nur aus Ziffern besteht, mit beliebig vielen führenden `"0"`), `false` sonst.

(b) `normalizedNat : string -> string`
Gibt für einen String, der eine natürliche Zahl repräsentiert, einen normalisierten String zurück, der keine führende `"0"` enthält. Nur der String, der die Zahl 0 repräsentiert, soll mit einer Ziffer `"0"` beginnen, aber sonst keine Zeichen enthalten.

(c) `valueOf : string -> int`
Gibt für einen String, der eine natürliche Zahl repräsentiert, den entsprechenden `int`-Wert zurück. Wenn der gegebene String keine natürliche Zahl repräsentiert, soll der Funktionswert -1 sein.

(d) `toString : int -> string`
Gibt für eine natürliche Zahl die entsprechende String-Repräsentation zurück.

Hinweis: Außer den in der Vorlesung vorgestellten Funktionen könnte auch die SML-Funktion
`substring : string * int * int -> string`
von Nutzen sein. Sie können sich unter `http://www.standardml.org/Basis/string.html` über ihr Verhalten informieren.

Aufgabe 8-2 *Listen* **(Hausaufgabe)**

Definieren Sie folgende Funktionen mit rein funktionalen Mitteln in einer Datei `8-2.sml`.

(a) `listensumme : int list -> int`
Berechnet die Summe aller Elemente in einer Liste. Die Definition soll einen rekursiven Berechnungsprozess auslösen. Die Listensumme der leeren Liste ist 0 (analog zur leeren Summe $\sum_{i=2}^{1} x_i = 0$).

(b) `listensumme' : int list -> int`
Gleiche Ergebnisse wie oben, aber so dass ein iterativer Berechnungsprozess ausgelöst wird.

(c) `mittelwert : int list -> real`
Berechnet den Mittelwert aller Elemente in einer Liste. Wird die Funktion mit der leeren Liste aufgerufen, soll sie keinen Fehler und keine Ausnahme verursachen, sondern den dafür sinnvollsten Wert des Ergebnistyps liefern (siehe Abschnitt 5.2 der Vorlesung).

Aufgabe 8-3 *Zusammengesetzte Typen* **(Hausaufgabe)**

Die Datei `8-3.sml` enthält eine Definition

```
type student = {name:string, vorname:string, fach:string, matnr:int}
```

sowie eine Liste von Beispiel-Werten dieses Typs. Geben Sie eine Kopie der Datei ab, in der zusätzlich eine Funktion `fach_studenten` vom Typ `string * student list -> student list` definiert ist.

`fach_studenten(f,studenten)` berechnet die Liste derjenigen Elemente von `studenten`, deren Komponente `fach` den gleichen Wert hat wie `f`. Die Datei enthält einen Kommentar mit einem Beispiel.

Aufgabe 8-4 *Zusammengesetzte Typen*

Gegeben seien die Deklarationen

```
val eins = 1;
val zwei = 2;
```

Welche Werte und Typen haben in dieser Umgebung die folgenden Ausdrücke:

(a) `[eins=2, zwei=1]`

(b) `[2=eins, 1=zwei]`

(c) `(eins=2, zwei=1)`

(d) `(2=eins, 1=zwei)`

(e) `{eins=2, zwei=1}`

(f) `{2=eins, 1=zwei}`