

Vorlesung Grundlagen der Programmiersprachen Sommer 2004 - Übungsblatt 7

Ausgabe: 20.07.2004 -- Abgabe: 28.07.2004, 13:00 Uhr, im Abgabekasten auf dem D3-Flur.

Aufgabe 19 (SML-Funktionen ausführen und verstehen)

Was leisten die folgenden SML-Funktionen? Beschreiben Sie verbal ihre Wirkungen und geben Sie ihre Signaturen an.

```
fun eins(x:int,y:int)    = x < y;
fun zwei z              = if z=1 then 1
                          else z + (zwei(z-1));
fun drei (e,(x,y))      = if e then (x,y) else (y,x);
fun vier a              = drei (eins a, a);
fun fuenf (x,y)         = if x=0 then y
                          else fuenf (x-1, y+1);
```

Überprüfen Sie ihre Ergebnisse, indem Sie die Funktionen mit einem SML-System (z.B. Moscow ML, `~gdp/mosml/bin/mosml`) ausprobieren. Stimmen die Signaturen, die das SML-System aus den Funktionsdefinitionen ableitet, mit Ihren Ergebnissen überein?

Mit SML praktisch arbeiten

Hinweis: Wir empfehlen dringend, Funktionen aus der Vorlesung und den Übungen sowie eigene Entwicklungen praktisch zu erproben. Dafür haben wir das System *Moscow ML* bereitgestellt.

Die lokal unter Solaris und Linux installierte Version von *Moscow ML* kann mit `~gdp/mosml/bin/mosml` aufgerufen werden.

Die SML-Funktionen können interaktiv oder als separate Textdatei eingegeben werden.
Beispiel: `mosml gdp.sml`

Moscow ML ist für alle gebräuchlichen Betriebssysteme (Windows, Linux, MacOS) verfügbar. Nähere Informationen und Download-Möglichkeiten gibt es im WWW unter <http://www.dina.dk/~sestoft/mosml.html>.

Aufgabe 20 (Funktionen auf Listen)

*** *Diese Aufgabe wird bewertet!* ***

Wir betrachten die Funktionen `MkList`, `Sum`, `Member` und `Append` von Folie 707.

- Schreiben Sie unter Verwendung von `MkList` und `Sum` die Funktion `SumBis n`, die zu einer positiven ganzen Zahl `n` die Summe der Zahlen von 1 bis `n` berechnet.
- Schreiben Sie eine Funktion `NonMember(l, m)`, die berechnet, ob `m` nicht Element von `l` ist.
- Schreiben Sie eine Funktion `UniqueElems(l)`, die mit Hilfe der Funktion `Member` berechnet, ob jedes Element von `l` nur einmal in `l` enthalten ist.
- Schreiben Sie eine Funktion `Filter(l, x)`, die aus einer Liste `l` eine neue Liste berechnet, die aus allen Elementen von `l` besteht außer denen, die den Wert `x` haben.

e) Welche Signatur ermittelt das Moscow ML-System für die Funktion

```
fun guess ( a , nil ) = [a]
  | guess ( a , h::t ) = if ( a > h ) then a::h::t
                        else h::guess(a,t);
```

Welchen Wert liefert der folgende Aufruf?

```
guess(3, MkList 7);
```

f) Wie oft wird die `::`-Operation bei der Abarbeitung des Aufrufes

```
Append(MkList 7, MkList 10);
```

ausgeführt?

Aufgabe 21 (Akkumulierender Parameter)

Betrachten Sie die Funktionen `Sum` und `ASum` auf Folie 711. Verdeutlichen Sie die Reihenfolge der ausgeführten Additionen bei der Berechnung von

```
Sum ([1,2,3,4,5])
```

und

```
ASum ([1,2,3,4,5], 0)
```

durch jeweils einen Baum: Die Blätter des Baumes sind die Zahlenwerte, die inneren Knoten das '+'-Symbol (Kantorowitsch-Bäume). Die inneren Knoten sollen in der Reihenfolge der Auswertung durchnummeriert sein.

Hier ist ein Beispiel für einen solchen Baum:

