

Programmierung und Modellierung

Aufgabe 8-1

Endrekursion

Gegeben sei die folgende Implementierung der Funktion $\text{summe}(n) = \sum_{i=1}^n i$:

```
fun summe(n) = if n = 0
               then 0
               else n + summe(n-1);
```

Definieren Sie in einer Datei `8-1.sml` diese Funktion neu, so dass sie selbst nicht mehr rekursiv ist, aber eine lokale Funktion benutzt, die endrekursiv ist.

Aufgabe 8-2

Endrekursion II

Auf dem letzten Übungsblatt in Aufgabe 7-1 b) und c) waren Funktionen angegeben worden, die von der Verwendung eines „if“ zu Pattern-Matching umgeschrieben werden sollten. Im folgenden sind Lösungsvorschläge für diese Aufgabe angegeben, die allerdings noch nicht endrekursiv sind. Bringen Sie die beiden Funktionen in endrekursive Form, und geben Sie diese in einer Datei `8-2.sml` ab.

a)

```
fun konstant nil = true
   | konstant (x :: nil) = true
   | konstant (x :: y :: l) = (x = y) andalso konstant(y :: l);
```

b) Unter Verwendung von `6-2.sml`:

```
fun linkerPfad Empty = []
   | linkerPfad (Build(x,l,_)) = x :: (linkerPfad l);
```

Aufgabe 8-3

Endrekursion III

Wir betrachten die Funktion $\text{maxexp}: \text{int} * \text{int} \rightarrow \text{int}$, die wie folgt definiert ist:

$$\text{maxexp}(m, n) = \begin{cases} \text{undefiniert} & \text{für } m < 2 \text{ oder } n \leq 0 \\ \text{das maximale } k \in \mathbb{N}, \text{ für das gilt:} & \text{für } m \geq 2 \text{ und } n > 0 \\ n \text{ ist ein ganzzahliges Vielfaches von} & \\ m^k & \end{cases}$$

Zum Beispiel ist $10368 = 2^7 \cdot 3^4$, also gilt:

$\text{maxexp}(2, 10368)$ hat Wert 7, denn 10368 ist ein Vielfaches von 2^7 , aber nicht von 2^8 .
 $\text{maxexp}(3, 10368)$ hat Wert 4, denn 10368 ist ein Vielfaches von 3^4 , aber nicht von 3^5 .
 $\text{maxexp}(4, 10368)$ hat Wert 3, denn 10368 ist ein Vielfaches von 4^3 , aber nicht von 4^4 .
 $\text{maxexp}(5, 10368)$ hat Wert 0, denn 10368 ist ein Vielfaches von 5^0 , aber nicht von 5^1 .

Definieren Sie die Funktion maxexp , so dass sie selbst nicht rekursiv ist, aber eine lokale Funktion benutzt, die endrekursiv ist. Geben Sie diese in einer Datei `8-3.sml` ab. Eventuell ist es dabei hilfreich, erst eine nicht-endrekursive Version zu implementieren, und diese dann zu übersetzen.

Aufgabe 8-4

Effizienter Algorithmus

Der folgende Algorithmus entscheidet, ob ein Binärbaum (mit der bekannten Definitionen aus `6-2.sml`) ausgeglichen ist:

```
fun balanced Empty = true
   | balanced (Build(e,l,r)) = (abs((tiefe l) - (tiefe r)) <= 1)
                               andalso (balanced l)
                               andalso (balanced r);
```

$\text{abs} : \text{int} \rightarrow \text{int}$ liefert den absoluten Betrag des Arguments, also gilt z.B. $\text{abs}(5) = 5$ als auch $\text{abs}(\sim 5) = 5$.

- a) Geben Sie die worst-case-Laufzeitkomplexität von `balanced` an.
- b) Geben Sie eine alternative Definition von `balanced` an, die lineare Zeitkomplexität besitzt. Laden Sie dazu von der Vorlesungsseite die Datei `8-4.sml` herunter, und ergänzen Sie diese entsprechend. Die Lösung von a) können Sie als Kommentar in diese Datei schreiben.

Abgabe: Montag, den 29.6.2009, 12 Uhr, per UniWorx.