

Programmierung und Modellierung, SS 2008
 Übungsblatt 6

Abgabe: bis Di 3. Juni 2008 (nachts)

Besprechung am Do-Di 5-10. Juni

Aufgabe 6-1 Endrekursion

SML

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

Definieren Sie in einer Datei `6-1.sml` diese Funktion neu, so dass sie selbst nicht mehr rekursiv ist, aber eine lokale Funktion benutzt, die endrekursiv ist (also einen iterativen Berechnungsprozess auslöst, aber ohne Referenzen). Hinweis: vergleichen Sie mit der Fakultätsfunktion im Skriptum.

Aufgabe 6-2 Endrekursion

SML

`maxexp: int * int -> int`

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

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

`maxexp(2, 10368)` hat Wert 7, denn 10368 ist ein Vielfaches von 2^7 , aber nicht von 2^8 .
`maxexp(3, 10368)` hat Wert 4, denn 10368 ist ein Vielfaches von 3^4 , aber nicht von 3^5 .
`maxexp(4, 10368)` hat Wert 3, denn 10368 ist ein Vielfaches von 4^3 , aber nicht von 4^4 .
`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.

Aufgabe 6-3 Endrekursion

SML

Die Fibonacci-Funktion ist für natürliche Zahlen wie folgt definiert:

$$\text{fib}(n) = \begin{cases} 0 & \text{falls } n = 0 \text{ ist} \\ 1 & \text{falls } n = 1 \text{ ist} \\ \text{fib}(n-1) + \text{fib}(n-2) & \text{falls } n \geq 2 \text{ ist} \end{cases}$$

Geben Sie eine Definition der Funktion an, die für $n \in \mathbb{N}$ durch einen iterativen Berechnungsprozess die n -te Fibonacci-Zahl liefert.

Eine Nullstelle einer lokal differenzierbaren Funktion f kann mit dem Newton-Verfahren gefunden werden. Man beginnt mit einem Näherungswert x_0 , der sich nahe der Nullstelle befindet, und berechnet eine Folge von Näherungen durch

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

Ist $|x_{n+1} - x_n| < \epsilon$ für ein n hat man eine Nullstelle bis auf Genauigkeit ϵ gefunden. (Details siehe <http://de.wikipedia.org/wiki/Newton-Verfahren>).

Implementieren Sie eine endrekursive Funktion

```
fun newton(f: real -> real, f': real -> real, x: real): real,
```

die für f mit Ableitung f' ausgehend vom Startwert x eine Nullstelle von f berechnet, mit $\epsilon = 10^{-10}$.