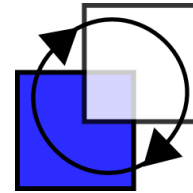


REACTIVE SYSTEMS GROUP

Universität des Saarlandes

Prof. Bernd Finkbeiner, Ph.D.

Markus Rabe, M.Sc.



Programmierung 1 (SS 2010) - 2. Übungsblatt

<http://react.cs.uni-saarland.de/prog1/>

Lesen Sie im Buch bis zum Ende von Kapitel 1.

Aufgabe 2.1

- Geben Sie ein Tupel mit 3 Positionen und nur einer Komponente an.
- Geben Sie einen Tupelausdruck an, der den Typ $int * (bool * (int * unit))$ hat.
- Geben Sie ein Paar an, dessen erste Komponente ein Paar und dessen zweite Komponente ein Tripel ist. Greifen Sie auf das erste Position des Tripels zu
- Geben Sie ein Paar aus Prozeduren an.

Aufgabe 2.2

Schreiben Sie eine Prozedur $teilbar : int * int \rightarrow bool$, die für (x, y) testet, ob x durch y ohne Rest teilbar ist.

Aufgabe 2.3

Oft gibt man eine Zeitdauer im *HMS-Format* mit Stunden, Minuten und Sekunden an.

- Schreiben Sie eine Prozedur $sec : int * int * int \rightarrow int$, die vom HMS-Format in Sekunden umrechnet. Beispielsweise soll $sec(1, 1, 6)$ die Zahl 3666 liefern.
- Schreiben Sie eine Prozedur $hms : int \rightarrow int * int * int$, die eine in Sekunden angegebene Zeit in das HMS-Format umrechnet. Beispielsweise soll $hms 3666$ das Tupel $(1, 1, 6)$ liefern. Berechnen Sie die Komponenten des Tupels mithilfe lokaler Deklarationen.

Aufgabe 2.4

Schreiben Sie eine Prozedur $min : int * int \rightarrow int$, die zu zwei Zahlen die kleinere liefert. Deklarieren Sie min analog zu $swap$ auf 3 verschiedene Arten: mit einem kartesischen Argumentmuster, mit einer lokalen Deklaration, und mit Projektionen.

Aufgabe 2.5

Schreiben Sie eine Prozedur $max : int * int * int \rightarrow int$, die zu drei Zahlen die größte liefert, auf zwei Arten:

- Benutzen Sie drei Konditionale und keine Hilfsprozedur.
- Benutzen Sie eine Hilfsprozedur und insgesamt nur ein Konditional.

Aufgabe 2.6

Sei die folgende rekursive Prozedurdeklaration gegeben:

```
fun f(n:int, a:int) : int = if n=0 then a else f(n-1, a*n)
```

Geben Sie die Rekursionsfolge und ein detailliertes Ausführungsprotokoll (vgl. Seite 14) für den Ausdruck $f(3, 1)$ an.

Aufgabe 2.7 (Binomialkoeffizienten)

Schreiben Sie eine rekursive Prozedur $binom : int * int \rightarrow int$, die für $n, k \geq 0$ den Binomialkoeffizienten $\binom{n}{k}$ berechnet. Verwenden Sie den durch die folgenden Gleichungen gegebenen Algorithmus:

$$\begin{aligned} \binom{n}{0} &= 1 \\ \binom{0}{k} &= 0 && \text{für } k > 0 \\ \binom{n}{k} &= \frac{n \cdot \binom{n-1}{k-1}}{k} && \text{für } n, k > 0 \end{aligned}$$

Aufgabe 2.8 (Sortieren von Tripeln)

- Schreiben Sie eine rekursive Prozedur $sort : int * int * int \rightarrow int * int * int$, die ganzzahlige Tripel sortiert. Beispielsweise soll $sort(7, 2, 5) = (2, 5, 7)$ gelten. Verwenden Sie nur 2 Konditionale.
- Schreiben Sie mithilfe von $sort$ eine Prozedur $max : int * int * int \rightarrow int$, die zu drei Zahlen die größte liefert. Verwenden Sie dabei keine kartesischen Muster.

Aufgabe 2.9

Schreiben Sie eine Prozedur $mul : int * int \rightarrow int$, die das Produkt einer natürlichen und einer ganzen Zahl durch wiederholte Addition berechnet.

- Schreiben sie eine rekursive Prozedur mul .

b) Schreiben sie die Prozedur *mul* mit einer endrekursiven Hilfsprozedur.

Beschreiben Sie den zugrunde liegenden Algorithmus zunächst mit Rekursionsgleichungen.

Aufgabe 2.10 (Fakultäten)

Für $n \geq 0$ können wir die so genannte n -te Fakultät $n!$ wie folgt definieren:

$$\begin{aligned} 0! &= 1 \\ n! &= 1 \cdot \dots \cdot n \quad \text{falls } n \geq 1 \end{aligned}$$

Beispielsweise gilt $4! = 1 \cdot 2 \cdot 3 \cdot 4 = 24$.

- a) Geben Sie zwei Gleichungen an, mit denen $n!$ berechnet werden kann.
- b) Realisieren Sie die Gleichungen mit einer Prozedur $fak : int \rightarrow int$. Schreiben Sie die Prozedur so, dass ihre Ausführung für negative Argumente wegen Speicherüberschreitung abgebrochen wird.
- c) Die Fakultäten werden schnell groß. Beispielsweise gilt $10! = 3628800$. Ermitteln Sie mit einem Interpreter das erste n , für das die Ausführung Ihrer Prozedur zu einem Überlauf führt.