

Übungen zu Einführung in die Informatik II

Aufgabe 1 Listen

a) Quadrierung

Definieren Sie eine Funktion, die zu der `int`-Liste $[x_1, x_2, \dots, x_n]$ die `int`-Liste $[x_1^2, x_2^2, \dots, x_n^2]$ berechnet.

b) Minimum

Definieren Sie eine Funktion, die das Minimum einer Liste berechnet.

c) Mergesort

Mergesort ist ein rekursiver Sortieralgorithmus, der gemäß dem Divide-and-Conquer-Prinzip arbeitet. Hier soll Mergesort zum Sortieren von Listen implementiert werden. Die Arbeitsweise des Algorithmus läßt sich wie folgt skizzieren.

- Null-/und einelementige Listen sind sortiert
- Listen mit mehr als einem Element werden in zwei möglichst gleich große Teillisten aufgeteilt. Die Teillisten werden durch einen rekursiven Aufruf von Mergesort sortiert und anschließend in einem Mischschritt zu einer sortierten Liste zusammengesetzt.

Definieren Sie eine Funktion `mergesort : 'a list -> 'a list`, die den skizzierten Algorithmus implementiert. Folgende Funktionen sollten Sie dazu zunächst definieren:

- Die Funktion `split : 'a list -> 'a list * 'a list` soll eine Liste in zwei möglichst gleich große Teillisten aufteilen.
- Die Funktion `merge : 'a list * 'a list -> 'a list` soll den Mischschritt implementieren. Als Argument erhält sie ein Paar (l_1, l_2) von bereits sortierten Listen. Als Ergebnis liefert sie die sortierte Liste aller Elemente aus l_1 und l_2 zurück.

Aufgabe 2 Arithmetische Ausdrücke

Schreiben Sie ein OCaml Programm, mit dem man arithmetische Ausdrücke auswerten kann.

- a) Definieren Sie zunächst einen Typ `expr`, der *konstante* Ausdrücke beschreibt, d.h. diese Ausdrücke können die Grundrechenarten („+“, „-“, „*“ und „/“) und `int` Zahlen enthalten. Außerdem benötigen Sie eine Funktion

```
val eval: expr -> int,
```

die Ihre Ausdrücke auswertet.

- b) Definieren Sie nun einen Typ `expr` mit einer entsprechenden Funktion `eval`, der zusätzlich auch Variablen enthalten kann. Sie können allerdings für die Aufgabe davon ausgehen, dass nur die Variablen `x`, `y` und `z` vorkommen können.

Bedenken Sie, dass Sie nun zum Auswerten eine *Variablen-Umgebung* `ρ` benötigen, in der die aktuellen Werte der drei Variablen nachgeschlagen werden können. (Tip: Sie können `ρ` als Funktion an `eval` übergeben!)