

# Übungen zu Softwareentwicklung III, Funktionale Programmierung

Blatt 12, Woche 14

Kombinatorische Probleme

Leonie Dreschler-Fischer

WS 2013/2014

**Ausgabe:** Freitag, 24.1.2014,

**Abgabe der Lösungen:** bis Montag, 3.2.2014, 12:00 Uhr per email bei den Übungsgruppenleitern.

**Ziel:** Mit den Übungsaufgaben auf diesem Blatt vertiefen Sie Ihre Kenntnisse über Funktionen höherer Ordnung und üben die Lösung kombinatorischer Probleme mittels Backtracking oder nicht-deterministischer Suche.

**Bearbeitungsdauer:** Die Bearbeitung sollte insgesamt nicht länger als 4,5 Stunden dauern.

**Homepage:**

[http://kogs-www.informatik.uni-hamburg.de/~dreschle/teaching/Uebungen\\_Se\\_III/Uebungen\\_Se\\_III.html](http://kogs-www.informatik.uni-hamburg.de/~dreschle/teaching/Uebungen_Se_III/Uebungen_Se_III.html)

Bitte denken Sie daran, auf den von Ihnen eingereichten Lösungsvorschlägen *Ihren Namen und die Matrikelnummer, den Namen der Übungsgruppenleiterin / des Übungsgruppenleiters und Wochentag und Uhrzeit der Übungsgruppe* anzugeben, damit wir ihre Ausarbeitungen eindeutig zuordnen können.

# 1 Der Wert einer Dezimalzahl

(Bearbeitungszeit 1 Std.), 10 Pnkt.

Gegeben seien die Ziffern einer Zahl als Liste von ganzen Zahlen, sortiert nach der Wertigkeit der Ziffern. Die Einerstelle steht am Anfang der Liste, dann die die Zehnerziffern, die Hunderter usw.

Beispiel: Die Zahl 9865 wird repräsentiert durch die Liste '(5 6 8 9).

Der Wert einer Dezimalzahl  $z$  mit den Ziffern  $z_n \cdots z_1 z_0$  ist das Polynom

$$z = z_n \cdot 10^n + z_{n-1} \cdot 10^{n-1} + \cdots + z_1 \cdot 10^1 + z_0$$

1. Geben Sie eine *rekursive* Funktion an, die den Wert einer Dezimalzahl aus deren Ziffern berechnet.
2. Berechnen Sie den Wert einer Dezimalzahl aus einer Liste der Ziffern mithilfe einer geeigneten Funktion höherer Ordnung.

Ein Tip: Polynome lassen sich elegant mithilfe des Hornerschen Schemas berechnen, so daß die Zahl der Multiplikationen klein bleibt, also:

$$z = (((((((z_n \cdot 10) + \dots + z_3) \cdot 10) + z_2) \cdot 10) + z_1) \cdot 10) + z_0$$

## 2 Krypto-Arithmetik

Bei krypto-arithmetischen Rätseln werden die Ziffern von Zahlen durch Buchstaben chiffriert. Gleiche Buchstaben stehen für gleiche Ziffern. Ein Rätsel besteht aus einer arithmetischen Gleichung, bei der eine Summe oder Differenz zweier Zahlen dargestellt wird. Es gilt, die Substitutionstabelle für die Ziffern zu erraten, so daß die Gleichung aufgeht.

Beispiel:

SEND	D=? E=? M=? N=?
+ MORE	O=? R=? S=? Y=?
<hr/>	
= MONEY.	

## 2.1 Problemanalyse

(Bearbeitungszeit 1 Std.), 10 Pnkt.

- Geben sie eine Datenstruktur an, mit der Sie Krypto-Arithmetik-Rätsel repräsentieren können.
- Beschreiben Sie den Zustandsraum für das Krypto-Arithmetik-Problem, in dem nach einer Lösung zu suchen ist.  
Wie müssen die Zustände beschrieben werden, und welche Zustandsübergänge und Folgezustände gibt es jeweils?
- Machen Sie eine Vorschlag, mit welchem Algorithmus nach einer Lösung gesucht werden kann (eine kombinatorische Funktion, backtracking oder nicht-deterministische Suche mit amb) und entwerfen Sie die Funktionen für die wichtigen Teilaufgaben.

## 2.2 Implementation und Test

(Bearbeitungszeit 2 Std.), 20 Pnkt.

Implementieren Sie den von Ihnen entworfenen Lösungsalgorithmus für Krypto-Arithmetik-Rätsel und erproben Sie ihn an mehreren Beispielen.

**Ausnahme:** Suchalgorithmen können sehr rechenaufwendig sein. Wenn Sie daher für einen schnellen Zugriff auf die Chiffre eine Hash-Tabelle oder einen Vektor verwenden wollen, dürfen Sie für diesen Teil der Aufgabe ausnahmsweise modifizierende Sprachelemente verwenden (make-hash-table, hash-table-put!, hash-table-get, vector-set!).

**Erreichbare Punkte:** 40

**Erreichbare Zusatzpunkte:** 0