# Big Data Übungsblatt 05

Anton Bulat, Josephine Geiger, Julia Siekiera

November 17, 2017

## Aufgabe 1: Monoide

### a)

Handelt es sich im Folgenden um Monoide?
Ein Monoid ist eine Halbgruppe mit Einselement, also ein Tripel $(G, \circ, e)$ aus:

- Menge $G$
- assoziativer Verknüpfung $\circ : G \times G \to G$
- Einselement $e \in G$

### i)

Das Kreuzprodukt von Vektoren aus $\mathbb{R}^3$ ist kein Monoid, da die Verknüpfung des Kreuzproduktes nicht assoziativ ist.
Gegenbeispiel: Sei $e_i$ der $i$-te Einheitsvektor.

$$(e_1 \times e_1) \times e_2 = 0, \qquad e_1 \times (e_1 \times e_2) = -e_2.$$

### ii)

Die Addition von Matrizen aus $\mathbb{R}^{m \times n}$ mit $m \neq n$ ist ein Monoid mit dem Tripel $(\mathbb{R}^{m \times n}, +, 0_{mn})$, wobei $+$ die elementweise Addition ist und damit assoziativ und $0_{mn}$ die Nullmatrix ist. Addiert man zwei Elemente aus $\mathbb{R}^{m \times n}$, so ergibt sich ein Element, das ebenfalls in $\mathbb{R}^{m \times n}$ liegt.

### iii)

Die Bestimmung des größten gemeinsamen Teilers ganzer Zahlen ist ein Monoid mit dem Tripel $(\mathbb{Z}, ggT, 0)$, wobei die Verknüpfung ggT die Bestimmung des größten gemeinsamen Teilers sein soll mit dem geltenden Assoziativgesetz:

$$ggT(a, b, c) = ggT(ggT(a, b), c) = ggT(a, ggT(b, c)).$$

Die 0 ist hier das Einselement, da der $ggT(a, 0) = a \quad \forall a \in \mathbb{Z}$.

### iv)

Die binäre XOR-Verknüpfung Boolscher Werte ist ein Monoid mit dem Tripel $(\{False, True\}, XOR, False)$, wobei die XOR-Verknüpfung assoziativ ist wie in der Tabelle gezeigt (mit False $\hat{=}$ 0, True $\hat{=}$ 1):

| A | B | C | (A XOR B) XOR C | A XOR (B XOR C) |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 |

Eine äquivalente, mathematischere Formulierung für die XOR-Verknüpfung ist das Tripel $(\mathbb{Z}/2\mathbb{Z}, +, 0)$, wobei $\mathbb{Z}/2\mathbb{Z}$ isomorph ist zu der Menge $\{$True, False$\}$. Die Verknüpfung $+$ entspricht der Addition auf $\mathbb{Z}/2\mathbb{Z}$ und liefert dadurch auch die Assoziativität, und das Einselement ist hier die $0 \in \mathbb{Z}/2\mathbb{Z}$.

### v)

Die Vereinigung von Mengen ist ein Monoid mit dem Tripel $(\mathcal{P}(X), \cup, \emptyset)$, wobei $\mathcal{P}(X)$ die Potenzmenge einer Menge $X$ ist und $\cup$ die Vereinigungsverknüpfung darstellt, die nach Definition assoziativ ist. Das Einselement ist hier die leere Menge, da $A \cup \emptyset = A \quad \forall A \subseteq X$. Vereinigt man zwei Elemente aus $\mathcal{P}(X)$, so ergibt sich ein Element, das ebenfalls in $\mathcal{P}(X)$ liegt.

## b)

### i)

Welche Form haben die Eingaben der Map- und Reduceschritte?
Die Eingabe des Mappers ist $(k, v)$-Tupel mit $k$ als String und $v$ als Integer. Sie bildet ein Monoid bezüglich der Konkatenation.
Die Eingabe des Reducers ist von der Form $(k, l = [v_1, \ldots, v_n])$, also vom Typ $Map < String, List < Integer >>$ und ist ebenfalls ein Monoid bezüglich der Konkatenation mit dem Tripel $(Map < String, List < Integer >>, \otimes, \epsilon)$.
Welche Form hat die Ausgabe?
Die Ausgabe ist vom Typ $Map < String, Integer >$ und hat die Monoidstruktur mit dem Tripel $(Map < String, Integer >, \otimes, \epsilon)$.

**ii)**

Reduce ist ein Monoid-Homomorphismus und kann also als Combiner verwendet werden.
**Denn:** Die Eingabe für den Reducer sind $(k, l)$-Paare. Reduce summiert alle $l$-Werte für alle Schlüssel auf.
Bei einer Aufteilung der Reduce-Eingabe in zwei Stücke ($(k, l_{first})$- und $(k, l_{second})$-Paare, $l = l_{first} + l_{second}$) summiert Reduce $l_{first}$ und $l_{second}$ einzeln und erhält $l$, da die Summe der beiden Teilsummen die ganze Summe ergeben.