

5. Übungsblatt - Revision 1 Diskrete Strukturen (WS 14/15)

Bauhaus-Universität Weimar, Professur für Mediensicherheit
Prof. Dr. Stefan Lucks, Christian Forler, Eik List
URL: <https://www.uni-weimar.de/de/medien/professuren/mediensicherheit/teaching/ws-2014/diskrete-strukturen-vorlesung/>

Abgabe und Besprechung: 16.12.2014 (13:30 Uhr) in der Übung

Aufgabe 1 (4 Punkte) Isomorphismus

Zeigen oder widerlegen Sie, dass die folgenden (Halb-)Gruppen isomorph sind. Die Definition der Isomorphie zwischen zwei Halbgruppen ist analog zu Definition 58 (Kapitel 4, Folie 154).

- a) $(\mathbb{Z}_6, +)$ und $(\mathbb{Z}_7^*, *)$
- b) $(\mathbb{Z}_6, +)$ und $(\mathbb{Z}_6, -)$

Aufgabe 2 (6 Punkte) Gruppeneigenschaften

Untersuchen Sie die folgenden Verknüpfungen auf ihre Gruppeneigenschaften (Kommutativität, Assoziativität, Existenz des neutralen Elements sowie Existenz des inversen Elements.)

- a) (\mathbb{Z}, \circ) mit $\circ(a, b) = a - b$
- b) (\mathbb{Z}, \circ) mit $\circ(a, b) = a^2 + b^2$
- c) (\mathbb{Q}, \circ) mit $\circ(a, b) = 2ab$

Aufgabe 3 (2 Punkte) Generatoren

Was sind die Generatoren (erzeugende Elemente) von $(\mathbb{Z}_7^*, *)$. Geben Sie den Lösungsweg mit an.

Aufgabe 4 (2 Punkte) Monoid

Sei $M = (S, \circ)$ ein Monoid mit neutralem Element 1. Wir nehmen an, dass für alle $x \in S$ gilt $x \circ x = 1$. Zeigen Sie, dass M abelsch (kommutativ) ist, d.h. es gilt

$$\forall x, y \in S : x \circ y = y \circ x$$

Aufgabe 5 (2 Punkte) Eindeutigkeit der Inversen (Satz 55)

Zeigen Sie, dass es in einem Monoid zu jedem Element höchstens ein Inverses gibt. Hat x ein Inverses x^{-1} , dann ist x selbst das Inverse von x^{-1} .

Aufgabe 6 (4 Punkte) Zyklische Untergruppen

Implementieren Sie ein Skript in Python welches per Kommandozeilenparameter eine Zahl $n \geq 1$ entgegennimmt. Das Programm soll alle Untergruppen von $(\mathbb{Z}_n^*, *)$ berechnen.

Bevor Sie Ihre Implementation einreichen sollten sie die folgenden Tests korrekt beantworten.

Testfall	Untergruppen
1	[[1]]
4	[[1], [1, 3]]
7	[[1], [1, 2, 4], [1, 2, 3, 4, 5, 6], [1, 6]]
18	[[1], [1, 5, 7, 11, 13, 17], [1, 7, 13], [1, 17]]

Beispielaufruf:

```
# python3 subgroups.py_123456.py 9
[[1], [1, 2, 4, 5, 7, 8], [1, 4, 7], [1, 8]]
```

(Abgabe: Schicken Sie Ihre Lösung als Anhang einer E-Mail `subgroups_<MatrNr>.py` an `christian.forler@uni-weimar.de` mit dem Betreff “[DS] Subgroups”)

Hinweise zur Programmieraufgabe

1. Mit Hilfe der Funktion `sorted` können Sie eine Liste sortieren.

```
>>> l = [2, 7, 1, 3]
>>> sorted(l)
[1, 2, 3, 7]
```

2. Mit der Funktion `append` können Sie eine Liste in einer Liste hinzufügen.

```
>>> a = []
>>> a.append([1, 2, 3])
>>> b = [5, 6, 7]
>>> a.append(b)
>>> print a
[[1, 2, 3], [5, 6, 7]]
```

3. Mit dem *Operator* `in` können Sie testen ob ein Element bereits in einer Liste ist.

```
>>> a = []
>>> a.append([1, 2, 3])
>>> b = [5, 6, 7]
>>> c = [1, 2, 3]
>>> a.append(b)
>>> c in a
True
```

4. Bei `[1, 2, 3]` handelt es sich um eine Liste von Zahlen, bei `[[1, 2, 3], [5, 6, 7]]` hingegen handelt es sich um eine Liste von Listen.

