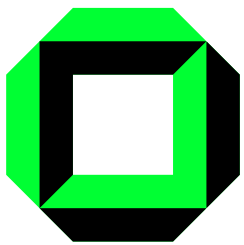
Info I – Übungsblatt 6

Joachim Breitner

mit Aufgaben von Christian Maier

http://www.joachim-breitner.de/wiki/Infotut

12. Dezember 2005



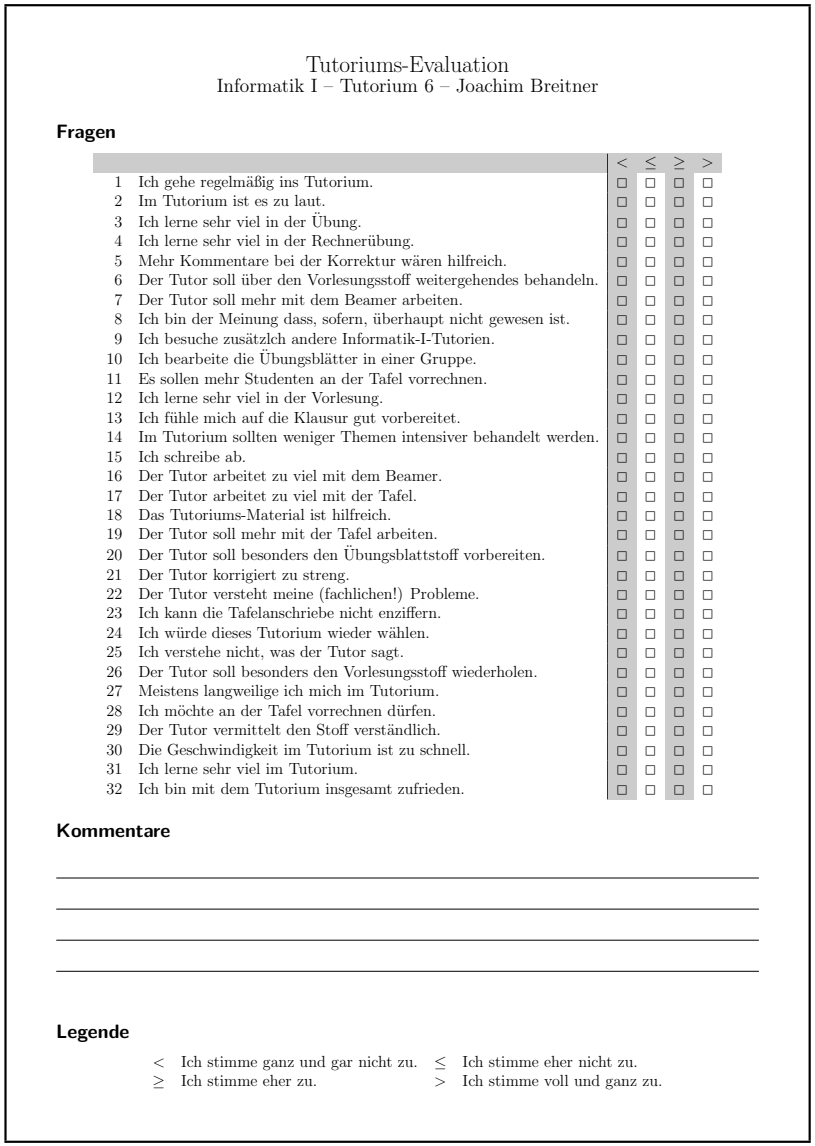
Unser Programm heute

1. Evaluation
2. Übungsblatt 4 und 5
3. Relationen und Graphen
4. Halbgruppen und Monoide
5. Java
6. Übungsblatt 6
7. Fragen

Tutoriums-Evaluation!

Bitte füllt die Fragebogen aus und legt sie mir nach dem Tutorium vorne hin.

- Alles anonym
- Kommentare sind sehr erwünscht
- Ich verspreche nichts!



Tutoriums-Evaluation
Informatik I – Tutorium 6 – Joachim Breitner

Fragen

		<	≤	≥	>
1	Ich gehe regelmäßig ins Tutorium.	□	□	□	□
2	Im Tutorium ist es zu laut.	□	□	□	□
3	Ich lerne sehr viel in der Übung.	□	□	□	□
4	Ich lerne sehr viel in der Rechnerübung.	□	□	□	□
5	Mehr Kommentare bei der Korrektur wären hilfreich.	□	□	□	□
6	Der Tutor soll über den Vorlesungsstoff weitergehendes behandeln.	□	□	□	□
7	Der Tutor soll mehr mit dem Beamer arbeiten.	□	□	□	□
8	Ich bin der Meinung dass, sofern, überhaupt nicht gewesen ist.	□	□	□	□
9	Ich besuche zusätzlch andere Informatik-I-Tutorien.	□	□	□	□
10	Ich bearbeite die Übungsblätter in einer Gruppe.	□	□	□	□
11	Es sollen mehr Studenten an der Tafel vorrechnen.	□	□	□	□
12	Ich lerne sehr viel in der Vorlesung.	□	□	□	□
13	Ich fühle mich auf die Klausur gut vorbereitet.	□	□	□	□
14	Im Tutorium sollten weniger Themen intensiver behandelt werden.	□	□	□	□
15	Ich schreibe ab.	□	□	□	□
16	Der Tutor arbeitet zu viel mit dem Beamer.	□	□	□	□
17	Der Tutor arbeitet zu viel mit der Tafel.	□	□	□	□
18	Das Tutoriums-Material ist hilfreich.	□	□	□	□
19	Der Tutor soll mehr mit der Tafel arbeiten.	□	□	□	□
20	Der Tutor soll besonders den Übungsblattstoff vorbereiten.	□	□	□	□
21	Der Tutor korrigiert zu streng.	□	□	□	□
22	Der Tutor versteht meine (fachlichen!) Probleme.	□	□	□	□
23	Ich kann die Tafelanschriebe nicht enziffern.	□	□	□	□
24	Ich würde dieses Tutorium wieder wählen.	□	□	□	□
25	Ich verstehe nicht, was der Tutor sagt.	□	□	□	□
26	Der Tutor soll besonders den Vorlesungsstoff wiederholen.	□	□	□	□
27	Meistens langweilige ich mich im Tutorium.	□	□	□	□
28	Ich möchte an der Tafel vorrechnen dürfen.	□	□	□	□
29	Der Tutor vermittelt den Stoff verständlich.	□	□	□	□
30	Die Geschwindigkeit im Tutorium ist zu schnell.	□	□	□	□
31	Ich lerne sehr viel im Tutorium.	□	□	□	□
32	Ich bin mit dem Tutorium insgesamt zufrieden.	□	□	□	□

Kommentare

Legende

< Ich stimme ganz und gar nicht zu. ≤ Ich stimme eher nicht zu.
≥ Ich stimme eher zu. > Ich stimme voll und ganz zu.

Übungsblatt-Rückblick

Statistik

- Blatt 4:
  Schnitt: 17 von 23 Punkten
- Blatt 5:
  Schnitt: 22 von 29 Punkten

Errata

Shuffle-Sort (auch „Bogo-Sort“ genannt) braucht $O(n \cdot n!)$, nicht $O(n \cdot n^n)$ Schritte. ($O(n!) \subset O(n^n)$ und $O(c^n) \subset O(n!)\ \forall c \in \mathbb{R}$)

Übungsblatt-Rückblick

Statistik

- Blatt 4:
  Schnitt: 17 von 23 Punkten
- Blatt 5:
  Schnitt: 22 von 29 Punkten

Errata

Shuffle-Sort (auch „Bogo-Sort“ genannt) braucht $O(n \cdot n!)$, nicht $O(n \cdot n^n)$ Schritte. ($O(n!) \subset O(n^n)$ und $O(c^n) \subset O(n!)\ \forall c \in \mathbb{R}$)

Relationen – Überblick

Durch Relationen werden Elemente einer oder mehrerer Mengen in Beziehung gesetzt

- Praktisch jede Aussage enhält Relationen.
- Beispiel: „Das Haus hat vier Außenwände“

In der Informatik werden Relationen zur Modellierung von Systemen benötigt.

- Relationen sind ein westentlcher Bestandteil der verschiedenen Diagramme der Unified Modeling Language

Aus der graphischen Darstellung von Relationen resultieren die Graphen.

Relationen mathematisch

Eine Relation ρ bezieht sich auf zwei Grundmengen U, V, und es gilt $\rho \subseteq U \times V$.
Eine Relation heißt homogen, wenn $U = V$ gilt. I.d.R. nennt man die Grundmenge dann E, also $\rho \subseteq E \times E$.
Für uns wichtig: Homogene Relationen mit einer endlichen Grundmenge E.

- Wir benennen die Elemente aus E mit e_i, $i = 0, \ldots, n-1$
- Diese kann man anschaulich als gerichtete Graphen darstellen.

Relationen mathematisch

Eine Relation ρ bezieht sich auf zwei Grundmengen U, V, und es gilt $\rho \subseteq U \times V$.
Eine Relation heißt homogen, wenn $U = V$ gilt. I.d.R. nennt man die Grundmenge dann E, also $\rho \subseteq E \times E$.
Für uns wichtig: Homogene Relationen mit einer endlichen Grundmenge E.

- Wir benennen die Elemente aus E mit e_i, $i = 0, \ldots, n-1$
- Diese kann man anschaulich als gerichtete Graphen darstellen.

Gerichteter Graph

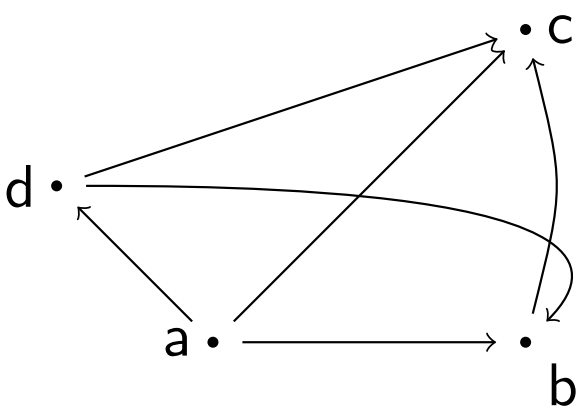
Ein gerichteter Graph G ist ein Tupel $G = (E, K)$ mit

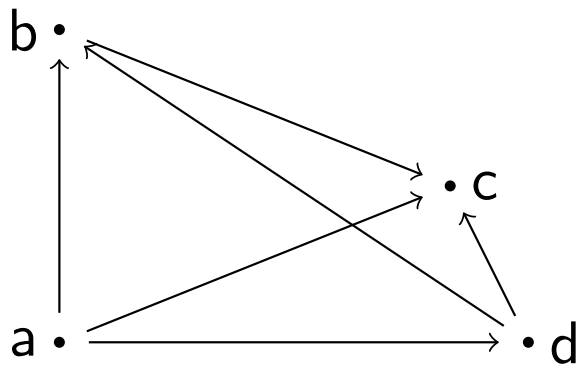
- der Grundmenge $E = \{e_i\}$ (die Menge der Ecken)
- der Relation $K \subseteq E \times E$ (die Menge der Kanten)
  Notationen für Kanten:
  - $(e, e') \in K$
  - $e \rightarrow_G e'$
  - $e \rightarrow e'$

Ein Graph heißt endlich, wenn E endlich ist ($|E| < \infty$)

Zeichnerische Darstellung von gerichteten Graphen

Man malt die Ecken als Punkte und die Kanten als Pfeile. Dabei ist die Anordnung beliebig:





$$G = (\{a, b, c, d, e\}, \{(a, b), (a, c), (a, d), (b, c), (d, b), (d, c)\})$$

Ungerichteter Graph

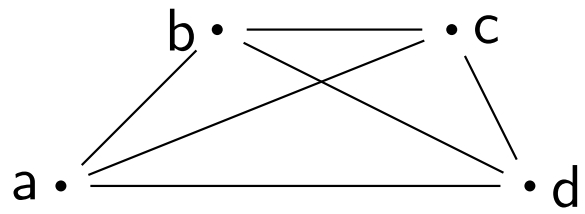
Ein ungerichteter Graph ist ein gerichteter Graph $G = (E, K)$ mit

$$\forall (e, e') \in K \Rightarrow (e', e) \in K$$

oder, anders formuliert:

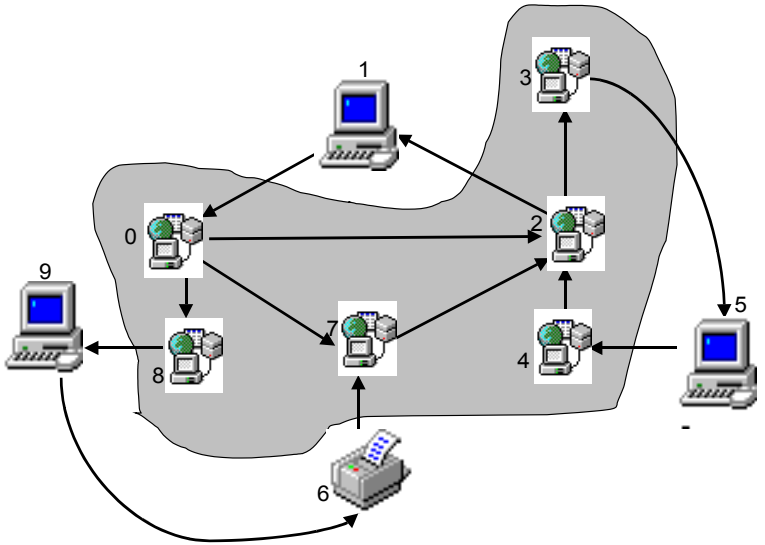
$$\forall e, e' \in E : e \rightarrow e' \Leftrightarrow e' \rightarrow e$$

Graphisch werden die Kanten als Verbindungslinen *ohne* Pfeilspitzen (oder manchmal auch mit Pfeilspitzen auf beiden Seiten) dargestellt.



Beispiel-Graph

Gegeben ist der folgende Netzausschnitt. Es werden vier Endsysteme mit sechs Routern teilweise vernetzt. Die Pfeile legen die Verbindungen und die Kommunikationsrichtung zwischen den Komponenten fest.

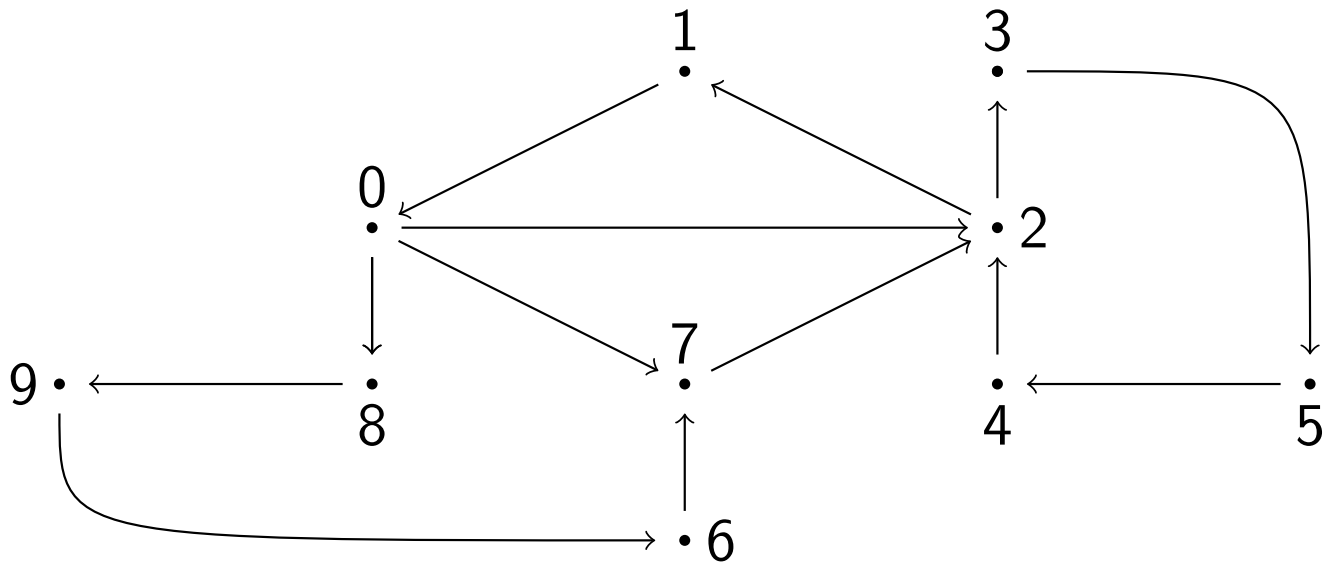


Aufgabe

1. Stelle das Kommunikationsnetz als Graph dar.
2. Was würde sich an der Aussage des Graphen ändern, wenn er ungerichtet wäre?
3. Gib für den gerichteten Graphen den Ein- und Ausgangsgrad jeder Ecke und den Grad des gesamten Graphen an.

Lösung

1. Abstrakte Darstellung des Netzausschnitts als Graph:



2. Wenn der Graph ungerichtet wäre, bedeutete dies, dass in beide Richtungen Daten ausgetauscht werden können. Die Verbindungen müssten also bidirektional verwendbar sein.

Lösung (Fortsetzung)

Ecke	Eingangsgrad $\lvert {}^{\cdot}e \rvert$	Ausgangsgrad $\lvert e^{\cdot} \rvert$
0	1	3
1	1	1
2	3	2
3	1	1
4	1	1
5	1	1
6	1	1
7	2	1
8	1	1
9	1	1

Gesamtgrad des Graphen: $\text{grad}(G) = 13$

Basteln mit Halbgruppen

Wir brauchen:

1. Ein Menge M mit einer Abbildung $* : M \times M \to M$
   Infix-Schreibweise bevorzugt ($a * b$ statt $*(a, b)$)
2. Diese Verknüpfung muss abgeschlossen sein
   - Die Schreibweise $* : M \times M \to M$ sagt das bereits.
   - Abgeschlossenheit bedeutet: Ich bekomme in jedem Fall ein Element aus M heraus.
3. Das Assoziativgesetzt muss gelten:
   $(a * b) * c = a * (b * c) =: a * b * c$

Und schon ist eine Halbgruppe fertig.

Beispiel für eine Halbgruppe

Zeichenketten

Die Menge Σ^+ über einem Alphabet Σ ist mit der Konkatenation eine Halbgruppe und enthält alle Zeichenketten.

In der Informatik ist Σ meist endlich. Dann ist Σ^+ abzählbar.

Andere Grundmengen

Die Grundmenge kann auch noch aus ganz anderen Dingen bestehen. Zum Beispiel könenn wir statt Σ folgendes U zu nehmen:
$U = \{$ `Apfel`, `Birne`, `Pflaume`, `Kirsche`, `Traube` $\}$
Die Elemente von U^* heißen dann Listen, Sequenzen oder Folgen und sehen so aus:
[`Apfel`, `Birne`] oder [`Pflaume`, `Traube`, `Pflaume`]
Statt U^* schreibt man auch $[U]$.

Beispiel für eine Halbgruppe

Zeichenketten

Die Menge Σ^+ über einem Alphabet Σ ist mit der Konkatenation eine Halbgruppe und enthält alle Zeichenketten.

In der Informatik ist Σ meist endlich. Dann ist Σ^+ abzählbar.

Andere Grundmengen

Die Grundmenge kann auch noch aus ganz anderen Dingen bestehen. Zum Beispiel könenn wir statt Σ folgendes U zu nehmen:
$U = \{$ `Apfel`, `Birne`, `Pflaume`, `Kirsche`, `Traube` $\}$
Die Elemente von U^* heißen dann Listen, Sequenzen oder Folgen und sehen so aus:
[`Apfel`, `Birne`] oder [`Pflaume`, `Traube`, `Pflaume`]
Statt U^* schreibt man auch $[U]$.

Beispiel für eine Halbgruppe

Zeichenketten

Die Menge Σ^+ über einem Alphabet Σ ist mit der Konkatenation eine Halbgruppe und enthält alle Zeichenketten.

In der Informatik ist Σ meist endlich. Dann ist Σ^+ abzählbar.

Andere Grundmengen

Die Grundmenge kann auch noch aus ganz anderen Dingen bestehen. Zum Beispiel könenn wir statt Σ folgendes U zu nehmen:
$U = \{$ `Apfel`, `Birne`, `Pflaume`, `Kirsche`, `Traube`$\}$
Die Elemente von U^* heißen dann Listen, Sequenzen oder Folgen und sehen so aus:
[`Apfel`, `Birne`] oder [`Pflaume`, `Traube`, `Pflaume`]
Statt U^* schreibt man auch $[U]$.

Basteln mit Monoiden

Dazu nehmen wir:

1. eine Halbgruppe (die haben wir ja schon gebastelt)
2. ein Einselement ε, für das folgende Eigenschaften gelten:

   $\varepsilon * a = a$

   $a * \varepsilon = a$

Gibt es ein Einselement, so gibt es genau eines. Man sagt, das Einselement ist eindeutig. (Beweis an der Tafel)

Beispiel für Monoid

Strings mit Konkatention

- $\Sigma^* := \Sigma^+ \cup \{\varepsilon\}$
- ε ist das leere Wort, d.h. Länge $|\varepsilon| = 0$
- falls w^n die n-fache Wiederholung des Wortes w bezeichnet, so gilt $w^0 = \varepsilon$
- Haben wir ein allgemeines U, so ist das Einselement von $[U]$ die leere Liste []

Nachhweis: Monoid

Ist eine gegebene Struktur ist ein Monoid? Zu überprüfen ist: Die Verknüpfung $*$ auf der Menge M ist abgeschlossen, erfüllt das Assoziativgesetz und besitzt ein Einselement ε.

Checkliste

Abgeschlossenheit:	$\forall x, y \in M : x * y \in M$
Assoziativität:	$\forall x, y, z \in M : (x * y) * z = x * (y * z)$
Einselement:	$\forall x \in M : \varepsilon * x = x * \varepsilon = x$

Nachweis: Halbgruppe

Ist die gegebene Struktur ist eine Halbgruppe, aber kein Monoid?
Zu zeigen ist: Die Verknüpfung $*$ auf der Menge M ist abgeschlossen, erfüllt das Assoziativgesetz, sie besitzt aber kein Einselement.

Beweis

Abgeschlossenheit:	$\forall x, y \in M : x * y \in M$
Assoziativität:	$\forall x, y, z \in M : (x * y) * z = x * (y * z)$
Einselement:	$\nexists \varepsilon \in M : \forall x \in M : x * \varepsilon = x$ und $\varepsilon * x = x$

Nachweis: Halbgruppe - Monoid

Überprüft, ob es sich bei folgenden Strukturen um Halbgruppen oder Monoide oder keines von beidem handelt. Beweist eure Aussage. Ihr dürft bei den Beweisen auf schon bekannte Eigenschaften aus der Vorlesung zurückgreifen. Zum Widerlegen genügen begründete Gegenbeispiele.

Aufgaben

Aufgabe a)

Die Addition $+$ auf der Menge $\mathbb{N}_0$ der natürlichen Zahlen mit der Zahl Null.

Aufgaben

Aufgabe a)

Die Addition $+$ auf der Menge $\mathbb{N}_0$ der natürlichen Zahlen mit der Zahl Null.

Die Struktur ist ein Monoid. Die Addition auf $\mathbb{N}_0$ ist abgeschlossen, erfüllt das Assoziativgesetz und besitzt das Einselement Null.

Beweis

Abgeschlossenheit:	$\forall x, y \in \mathbb{N}_0 : x + y \in \mathbb{N}_0$
Assoziativität:	$\forall x, y, z \in \mathbb{N}_0 : (x + y) + z = x + (y + z)$
Einselement:	$\forall x \in \mathbb{N}_0 : 0 + x = x + 0 = x$

Aufgaben

Aufgabe b)

Die Multiplikation $\cdot$ auf der Menge $M = \{r \in \mathbb{R} | r > 1\}$, wobei $\mathbb{R}$ die Menge der reelen Zahlen ist.

Aufgaben

Aufgabe b)

Die Multiplikation $\cdot$ auf der Menge $M = \{r \in \mathbb{R} | r > 1\}$, wobei $\mathbb{R}$ die Menge der reelen Zahlen ist.

Die Struktur ist eine Halbgruppe, aber kein Monoid. Die Multiplikation auf $\mathbb{R}$ ist abgeschlossen, erfüllt das Assoziativgesetz, sie besitzt aber kein Einselement.

Beweis

Abgeschlossenheit:	$\forall x, y \in \mathbb{R} : x \cdot y \in \mathbb{R}$
Assoziativität:	$\forall x, y, z \in \mathbb{R} : (x \cdot y) \cdot z = x \cdot (y \cdot z)$
Einselement:	$\forall x, y \in M : y > 1 \Rightarrow x \cdot y > x \Rightarrow x \cdot y \neq x$
	$\Rightarrow y$ ist kein Einselement.

Aufgaben

Aufgabe c)

Die Vektoraddition mit der Verknüpfung
$(a, b) + (c, d) := (a + c, b + d)$ auf der Menge $\mathbb{N}_0^2 = \mathbb{N}_0 \times \mathbb{N}_0$

Aufgaben

Aufgabe c)

Die Vektoraddition mit der Verknüpfung
$(a, b) + (c, d) := (a + c, b + d)$ auf der Menge $\mathbb{N}_0^2 = \mathbb{N}_0 \times \mathbb{N}_0$

Die Struktur ist ein Monoid.

Beweis

In beiden Stellen des Vektors findet unabhängig voneinander eine normale Addition wie bei der Struktur in Aufgabenteil a) statt. Von dieser ist bekannt, dass sie abgeschlossen und assoziativ ist und mit der Null ein Einselement besitzt.

Aufgaben

Aufgabe d)

Die Linksidentität mit der Verknüpfung $\#$ mit $a\#b := a$ auf einer beliebigen Menge $M \neq \emptyset$

Aufgaben

Aufgabe d)

Die Linksidentität mit der Verknüpfung $\#$ mit $a\#b := a$ auf einer beliebigen Menge $M \neq \emptyset$

Die Struktur ist eine Halbgruppe, aber kein Monoid. Die Linksidentität mit der Verknüpfung $\#$ auf M ist abgeschlossen, erfüllt das Assoziativgesetz, sie besitzt aber kein Einselement.

Beweis

Abgeschlossenheit:	$\forall x, y \in M : x\#y = x \in M$
Assoziativität:	$\forall x, y, z \in M : (x\#y)\#z = x\#y$
	$= x = x\#z = x\#(y\#z)$
Einselement:	$\forall x, y \in M, x \neq y : x\#y = x \neq y$
	$\Rightarrow y$ ist nicht Linkseins.

Aufgaben

Aufgabe e)

Die Konkatention von Zeichenketten (Strings) in Java.

Aufgaben

Aufgabe e)

Die Konkatention von Zeichenketten (Strings) in Java.

Die Struktur ist ein Monoid. Die Konkatention von Zeichenketten ist abgeschlossen, erfüllt das Assoziativgesetz und die leere Zeichenkette ist ihr Einselement.

Beweis

Abgeschlossenheit:	$\forall x, y \in \text{String}: xy \in \text{String}$
Assoziativität:	$\forall x, y, z \in \text{String}: (xy)z = x(yz)$
Einselement:	$\forall x \in \text{String}: ""+x = x+""= x$

Transpositions-Chiffren

Die Transposition von Buchstaben kann zur Chiffrierung eines Textes eingesetzt werden. So genannte Transpositions-Chiffren ordnen die Buchstaben eines Klartextes dabei nach einem Schema oder einer geometrischen Figur um, etwa nach einer zweidimensionalen Matrix. Bei der Spaltentransposition wird der zu chiffrierende Text zeilenweise z.B. in eine Matrix eingelesen und danach spaltenweise entsprechend einer vorgegebenen Reihenfolge ausgelesen.
Beispielausgabe für eine 2×3 - Matrix mit Ausgabe-Reihenfolge 3-2-1:

```
Zu chiffrierender Text : Informatik
Chiffrierter Text : fmnrIoitak
```

Die Matritzen sehen in diesem Fall folgendermaßen aus:

	1	2	3
1	I	n	f
2	o	r	m

	1	2	3
1	a	t	i
2	k		

Beachten Sie: Nicht besetzte Stellen der Matrix werden beim Auslesen der Spalten übergangen!

Transpositions-Chiffren (2)

Schreiben Sie ein Java-Programm, das eine Textzeile einliest und diese unter Verwendung einer 4×3-Matrix durch Spaltentransposition chiffriert und ausgibt. Die Spalten der Matrix sollen danach in einer im Programm fest definierten Reihenfolge 3-1-2 ausgegeben werden. Verwendne Sie dabei eine eigene Methode cipher(). Diese soll aus einem übergebenen Text und einer ebenfalls übergebenden Ausgabenreihenfolge der Spalten den chiffrierten Text erzeugen.

```
public class Transposition {

static String cipher(String text, int first, int ↵
    second, int third) {
    // Initialize result variable and matrix
    String encipheredText = "";
    char[][] cipherMatrix = new char[4][3];
    for (int m=0; m<(((text.length()-1)/12)+1); m++) {
      // Fill matrix and calculate number of matrices necessary for text
      // For each row and columns...
      for (int row=0; row<4; row++) {
        for (int column=0; column<3; column++) {
          if ((m*12+row*3+column)<text.length()) {
            // If there is a char left to insert into matrix, insert it
            cipherMatrix[row][column] = ↵
                text.charAt(m*12+row*3+column);
          }
        }
      }
```

```
// Get chars from „first“ column
for (int row=0; row<4; row++) {
  if ((m*12+row*3+first-1) < text.length()) {
    // only if in range
    encipheredText += ↵
        cipherMatrix[row][first-1];
  }
}
// Get chars from „second“ column
for (int row=0; row<4; row++) {
  if ((m*12+row*3+second-1) < text.length()) {
    encipheredText += ↵
        cipherMatrix[row][second-1];
  }
}
// Get chars from „third“ column
for (int row=0; row<4; row++) {
  if ((m*12+row*3+third-1) < text.length()) {
    encipheredText += cipherMatrix[row][third-1];
  }
}
```

```
        }
      return encipheredText; // Return result
    }
    public static void main(String[] args) {
      // Read text to cipher from keyboard
      Out.print("Zu chiffrierender Text: ");
      String plainText = In.readLine();
      // Cipher text and print result on screen
      Out.print("Chiffrierter Text      : '");
      Out.print(cipher(plainText,3,1,2));
      Out.println("'");
    }
}
```

geklaut von:

`http://www-fs.informatik.uni-tuebingen.de/~reinhard/krypto/German/3.1.d.html`

Tipps fürs nächste Übungsblatt

- Nur die jeweils angegebenen Funktionen benutzen bzw. *nicht* benutzen
- Aufgabe 1.3: Macht euch mit `StringBuffer` vertraut (siehe API, „Java ist auch eine Insel“)
  Wichtig: Ihr sollt eine Textersetzung durchführen, also nicht nur einen neuen String zusammenschustern.

Fragen

Fragen?