

Diskrete Strukturen

Manuel Kauers

Institut für Algebra
Johannes Kepler Universität

Version: 4. November 2016

Inhalt

1	Mengen und Formeln	4
2	Zeichenketten	7
3	Funktionen	9
4	Relationen	17
5	Graphen	23
6	Gruppen	35
7	Modulare Arithmetik	48
8	Kombinatorik	58
9	Summen und Rekurrenzen	58

1 Mengen und Formeln

Eine *Menge* (engl. *set*) ist ein abstraktes Objekt, das andere abstrakte Objekte beinhaltet, die sogenannten *Elemente* (engl. *element*) der Menge.

Beispiel.

1. \mathbb{N} ist die Menge aller natürlicher Zahlen $0, 1, 2, \dots$.
2. \mathbb{Z} ist die Menge aller ganzer Zahlen.
3. \mathbb{Q} ist die Menge aller rationaler Zahlen.
4. \mathbb{R} ist die Menge aller reeller Zahlen.
5. $\{\square, \circ, \triangle, \blacksquare\}$ ist die Menge, die die vier Objekte $\square, \circ, \triangle$ und \blacksquare beinhaltet.

Die Notation $1 \in \mathbb{N}$ bedeutet, dass das Objekt 1 ein Element der Menge \mathbb{N} ist. Die Notation $\frac{1}{3} \notin \mathbb{N}$ bedeutet, dass das Objekt $\frac{1}{3}$ kein Element der Menge \mathbb{N} ist. Die Notation $\mathbb{N} \subseteq \mathbb{Q}$ bedeutet, dass \mathbb{N} eine *Teilmenge* (engl. *subset*) von \mathbb{Q} ist, d.h. jedes Element von \mathbb{N} ist auch ein Element von \mathbb{Q} (aber nicht unbedingt umgekehrt).

Wenn M eine Menge ist, dann bezeichnet $\mathcal{P}(M)$ die *Potenzmenge* (engl. *power set*) von M . Das ist die Menge aller Teilmengen von M . Es gilt also

$$A \subseteq M \iff A \in \mathcal{P}(M)$$

für alle Mengen A und M .

Beispiel. Für $M = \{1, 2, 3\}$ gilt

$$\mathcal{P}(M) = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}\}.$$

Dabei bezeichnet das Symbol \emptyset die *leere Menge* ($\emptyset = \{\}$).

Man beachte, dass die Elemente einer Menge ihrerseits selbst Mengen sein können. Davon darf man sich nicht verwirren lassen. Es gilt zum Beispiel $1 \in \{1\}$ und $\{1\} \in \{\{1\}\}$, aber $1 \notin \{\{1\}\}$ und $\{1\} \notin \{1\}$. Insbesondere gilt $1 \neq \{1\}$: das Objekt 1 ist etwas anderes als die Menge, deren einziges Element dieses Objekt ist. Beachten Sie auch, dass eine Menge niemals ein Element von sich selbst sein kann.

Wenn M eine Menge ist, dann muss für **jedes** Objekt x entweder $x \in M$ oder $x \notin M$ gelten. Weitere Informationen codiert die Menge nicht. Insbesondere nicht, „wie oft“ ein Element in der Menge enthalten ist (z.B. gilt $\{1, 2, 3\} = \{1, 2, 2, 2, 2, 3, 3\}$) oder „in welcher Reihenfolge“ (z.B. gilt $\{1, 2, 3\} = \{3, 1, 2\}$). Zwei Mengen A, B sind gleich, wenn für jedes Objekt x gilt $x \in A \iff x \in B$.

Man kann sich eine Menge vorstellen als Codierung einer Eigenschaft, die ein Objekt haben kann. Element der Menge zu sein bedeutet die betreffende Eigenschaft zu besitzen; nicht Element der Menge zu sein bedeutet sie nicht zu besitzen. Zum Beispiel ist \mathbb{N} die Menge aller Objekte, die die Eigenschaft haben, eine natürliche Zahl zu sein. Umgekehrt: ist A irgendeine Menge, dann haben ihre Elemente die Eigenschaft, Elemente von A zu sein. Das ist zumindest insofern eine Eigenschaft, als es die Elemente von A von allen anderen Objekten unterscheidet.

Eigenschaften von Objekten einer gegebenen Menge A entsprechen Teilmengen dieser Menge. Solche Teilmengen lassen sich auch durch eine Funktion $p: A \rightarrow \{\text{True}, \text{False}\}$ beschreiben, die für jedes Element $x \in A$ angibt, ob es die Eigenschaft hat ($p(x) = \text{True}$) oder nicht ($p(x) = \text{False}$). Man verwendet die Notation $\{x \in A : p(x)\}$ oder $\{x \in A \mid p(x)\}$ für die Menge aller x aus A , für die $p(x)$ wahr ist. Das ist eine Teilmenge von A .

Beispiel.

1. $\{x \in \mathbb{Z} : \text{die letzte Ziffer von } x \text{ ist } 3\} = \{\dots, -23, -13, -3, 3, 13, 23, \dots\}$
2. $\{p \in \mathbb{N} : p \text{ ist eine Primzahl}\} = \{2, 3, 5, 7, 11, 13, 17, \dots\}$
3. $\{x \in \mathbb{R} : x > 0\}$ ist die Menge aller positiven reellen Zahlen.
4. $\{u \in \{\square, \blacksquare, \circ, \triangle\} : u \text{ ist schwarz}\} = \{\blacksquare\}$
5. $\{s \in K : s \text{ hat bestanden}\}$, wobei K die Menge der Teilnehmer der Vorlesungsklausur ist.

Es gibt noch eine weitere Notation, mit der man Mengen beschreiben kann. Dazu geht man von zwei bekannten Mengen A, B aus und betrachtet eine Funktion $f: A \rightarrow B$ (siehe Abschnitt 3 für eine genaue Erklärung des Begriffs Funktion). Dann schreibt man $\{f(x) : x \in A\}$ oder $\{f(x) \mid x \in A\}$ für die Menge aller $y \in B$, so dass es ein $x \in A$ gibt mit $y = f(x)$. Zum Beispiel ist $\{x^2 : x \in \mathbb{Z}\}$ die Menge aller Quadratzahlen.

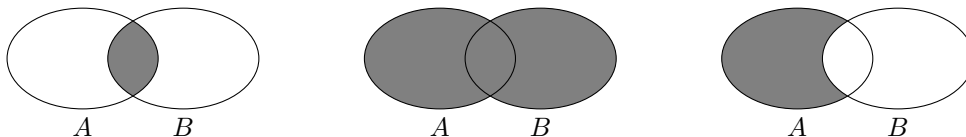
Man kann diese Notation auch mit der vorherigen kombinieren: $\{f(x) : x \in A, p(x)\}$ bezeichnet die Menge aller $y \in B$, so dass es ein $x \in A$ mit der Eigenschaft $p(x) = \text{True}$ gibt, für das $y = f(x)$ gilt. Zum Beispiel ist $\{x^2 : x \in \mathbb{N}, n \text{ prim}\} = \{4, 9, 25, 49, 121, 169, \dots\}$.

Sind A, B Mengen, so sind die Vereinigung $A \cup B$, der Schnitt $A \cap B$ und die Differenz $A \setminus B$ definiert durch

$$\forall x : (x \in A \cap B \iff x \in A \wedge x \in B)$$

$$\forall x : (x \in A \cup B \iff x \in A \vee x \in B)$$

$$\forall x : (x \in A \setminus B \iff x \in A \wedge x \notin B)$$



Die Notation $\forall x : \dots$ bedeutet „für alle Objekte x gilt/gelte ...“. Mit der Notation $\exists x : \dots$ meint man „es gibt ein Objekt x , für das gilt/gelte ...“. Die Symbole \forall und \exists bezeichnet man als *Quantoren* (engl. *quantifier*).

Beispiel. $\forall x : (x \in \mathbb{Q} \Rightarrow \exists y : (y \in \mathbb{Q} \wedge x \cdot y = 1))$ bedeutet:

„für alle Objekte x gilt, wenn sie zu \mathbb{Q} gehören, dass es ein (zu x passendes) Objekt y gibt, dass zu \mathbb{Q} gehört, und für das gilt $x \cdot y = 1$ “.

So etwas nennt man eine *Formel* (engl. *formula*). Eine Formel kann wahr oder falsch sein. Dagegen ist z. B. $x^2 - 7x + 2$ keine Formel sondern ein *Ausdruck* (engl. *expression*). Die Formel im obigen Beispiel ist falsch. Eine Formel ist genau dann falsch, wenn ihre *Negation* (ihr „Gegenteil“) wahr ist.

Beachten Sie, dass sich logische Zeichen beim Negieren von Formeln ändern können. Insbesondere gilt

- $\neg(\forall x : p(x)) \iff \exists x : \neg p(x)$
- $\neg(A \Rightarrow B) \iff (A \wedge \neg B)$
- $\neg(A \wedge B) \iff (\neg A) \vee (\neg B)$, usw.

Beispiel. Die Negation der Formel aus dem vorherigen Beispiel lautet

$$\exists x : (x \in \mathbb{Q} \wedge \forall y : (y \notin \mathbb{Q} \vee x \cdot y \neq 1)).$$

(„Es gibt ein Objekt x , das zu \mathbb{Q} gehört, so dass für alle Objekte y gilt: y gehört nicht zu \mathbb{Q} oder $x \cdot y = 1$ “)

Um eine Formel des Typs $\exists x : p(x)$ zu beweisen, genügt es, ein konkretes Objekt x anzugeben, und dann zu beweisen, dass $p(x)$ für diese Wahl von x wahr ist. Im obigen Beispiel können wir $x = 0$ wählen, denn für diese Wahl von x gilt sicher

$$0 \in \mathbb{Q} \wedge \forall y : (y \notin \mathbb{Q} \vee 0 \cdot y \neq 1),$$

weil ja $0 \cdot y = 0 \neq 1$ für alle $y \in \mathbb{Q}$ gilt. Damit ist gezeigt, dass die Negation der ursprünglichen Formel wahr ist. Also war die ursprüngliche Formel falsch.

Die modifizierte Formel

$$\forall x : (x \in \mathbb{Q} \setminus \{0\} \Rightarrow \exists y : (y \in \mathbb{Q} \wedge x \cdot y = 1))$$

ist dagegen wahr. Hier haben wir eine Formel des Typs $\forall x : p(x)$ zu beweisen. Dazu betrachtet man ein beliebig aber fest gewähltes Objekt x . Anders als vorher treffen wir keine konkrete Wahl sondern machen im Gegenteil keine weiteren Annahmen über x . Wir zeigen

$$x \in \mathbb{Q} \setminus \{0\} \Rightarrow \exists y : (y \in \mathbb{Q} \wedge x \cdot y = 1).$$

Es gibt zwei Fälle zu unterscheiden: $x \in \mathbb{Q} \setminus \{0\}$ und $x \notin \mathbb{Q} \setminus \{0\}$. Im Fall $x \notin \mathbb{Q} \setminus \{0\}$ ist nichts zu zeigen. (Warum nicht?) Betrachten wir also den Fall $x \in \mathbb{Q} \setminus \{0\}$. In diesem Fall ist zu zeigen

$$\exists y : (y \in \mathbb{Q} \wedge x \cdot y = 1),$$

wobei x das beliebig aber fest gewählte x aus $\mathbb{Q} \setminus \{0\}$ ist. Betrachte die konkrete Wahl $y = \frac{1}{x}$. Wegen $x \in \mathbb{Q} \setminus \{0\}$ ist y ein Element von \mathbb{Q} und es gilt $x \cdot y = x \cdot \frac{1}{x} = 1$, wie gefordert. Damit ist der Beweis fertig.

Also noch einmal zusammengefasst:

- Wenn man $\forall x : p(x)$ beweisen soll, schreibt man „Wähle ein beliebiges Objekt x “. Man wählt dieses Objekt aber nicht als Autor des Beweises, sondern überlässt die Wahl quasi dem Leser. Dann beweist man $p(x)$ für das „beliebige“ Objekt x , über das man keine weiteren Annahmen getroffen hat. Nur wenn das gelingt, hat man gezeigt, dass $p(x)$ wirklich für alle Objekte x gilt.
- Wenn man $\exists x : p(x)$ beweisen soll, darf man als Autor des Beweises ein geeignetes Objekt x auswählen und schreibt „Betrachte das Objekt $x = \dots$ “. Dann beweist man $p(x)$ für diese konkrete Wahl. Wenn das gelingt, hat man gezeigt, dass es ein Objekt gibt, für das $p(x)$ gilt, nämlich das, das man angegeben hat. Die Wahl des Objekts kann durchaus von anderen „beliebigen“ Objekten abhängig gemacht werden, so wie die Wahl von y in Abhängigkeit von x im obigen Beispiel.

Komplementär dazu gibt es die Situation, dass man bekannte Fakten in einem Beweis ausnutzen will.

- Wenn wir schon wissen, dass $\forall x : p(x)$ gilt, dann können wir für jedes beliebige Objekt y , von dem im Beweis die Rede ist, ohne weiteres annehmen, dass $p(y)$ gilt.
- Wenn wir schon wissen, dass $\exists x : p(x)$ gilt, dann können wir uns in einem Beweis ein solches Objekt verschaffen, indem wir sagen „Sei x ein Element mit der Eigenschaft $p(x)$ “.

Mehr über die Behandlung von Quantoren im besonderen und logischer Formeln im allgemeinen erfahren Sie in anderen Lehrveranstaltungen.

Da man es fast immer mit Objekten zu tun hat, die zu irgendeiner Menge gehören, ist es zweckmäßig, folgende Kurzschreibweisen einzuführen:

- $\forall x \in M : \dots$ für $\forall x : (x \in M \Rightarrow \dots)$ („für alle Elemente x von M gilt ...“)
- $\exists x \in M : \dots$ für $\exists x : (x \in M \wedge \dots)$ („es gibt ein Element x von M , so dass ...“)

Neben den allgemeinen logischen Schlussregeln, die wir oben kurz skizziert haben, gibt es auch noch einige Beweistechniken, die sich nur auf bestimmte Typen von Objekten anwenden lassen. Das wichtigste Beispiel ist das Prinzip der *vollständigen Induktion*. Dieses Prinzip bietet sich immer dann an, wenn man es mit einer Aussage über die natürlichen Zahlen zu tun hat:

$$\forall n \in \mathbb{N} : p(n).$$

Die Idee ist, dass man zunächst $p(0)$ beweist (was meistens sehr einfach ist), und dann die Formel

$$\forall n \in \mathbb{N} : p(n) \Rightarrow p(n+1).$$

Wenn diese nämlich wahr ist, dann folgt aus $p(0)$, dass auch $p(1)$ gilt, und daraus folgt, dass auch $p(2)$ gilt, und daraus folgt, dass auch $p(3)$ gilt, und so weiter. Es gilt dann also $p(n)$ für alle $n \in \mathbb{N}$, wie behauptet.

Beispiel. Wir beweisen die Summationsformel $\forall n \in \mathbb{N} : \sum_{k=0}^n k^2 = \frac{1}{6}n(n+1)(2n+1)$.

Induktionsanfang: Für $n = 0$ ist die Formel wahr, denn es gilt $\sum_{k=0}^0 k^2 = 0^2 = 0$ und $\frac{1}{6}0(0+1)(2 \cdot 0 + 1) = 0$.

Induktionsschluss $n \rightarrow n+1$: Sei $n \in \mathbb{N}$ beliebig, und nehmen wir an, dass die Formel für dieses n gilt. Wir zeigen, dass sie dann auch für $n+1$ anstelle von n gilt. Dazu rechnen wir:

$$\begin{aligned} \sum_{k=0}^{n+1} k^2 &= \sum_{k=0}^n k^2 + (n+1)^2 && \text{(Abspalten des letzten Terms)} \\ &= \frac{1}{6}n(n+1)(2n+1) + (n+1)^2 && \text{(Verwendung der Voraussetzung)} \\ &= \frac{1}{6}(n+1)((n+1)+1)(2(n+1)+1) && \text{(Nachrechnen).} \end{aligned}$$

Damit ist der Beweis fertig.

2 Zeichenketten

Sei Ω eine beliebige endliche Menge. Wir fassen die Elemente von Ω als Buchstaben auf und wollen aus diesen Buchstaben Wörter bilden. Ein Wort ist dabei einfach eine Aneinanderreihung von endlich vielen Buchstaben. Es kommt nicht darauf an, ob es einen Sinn hat.

Beispiel. $\Omega = \{a, b, c\}$. Dann sind zum Beispiel *aabcbacc* und *accbaccba* zwei Wörter.

Beachte: Die Reihenfolge der Buchstaben ist relevant: *ab* und *ba* sind zwei verschiedene Wörter.

Formal sind Wörter nichts anderes als Tupel von Buchstaben. Was ist ein Tupel? Zunächst: Sind A, B zwei Mengen, so bezeichnet $A \times B$ die Menge aller Paare (a, b) mit $a \in A$ und $b \in B$.

Beispiel. $\{1, 2\} \times \{3, 4\} = \{(1, 3), (1, 4), (2, 3), (2, 4)\}$.

Der Begriff des Tupels ist eine Verallgemeinerung: sind A_1, \dots, A_n Mengen, so ist $A_1 \times \dots \times A_n$ die Menge aller *Tupel* (a_1, \dots, a_n) mit $a_1 \in A_1, \dots, a_n \in A_n$. Man schreibt A^n statt $A \times A \times \dots \times A$. Im Fall $n = 1$ ist $A^1 = A$ und im Fall $n = 0$ ist $A^0 = \{()\}$ die Menge, die nur das leere Tupel enthält.

Wenn wir diese Begriffe als bekannt voraussetzen, können wir Wörter wie folgt formal definieren.

Definition 1. Sei Ω eine endliche Menge. Dann heißt

$$\Omega^* := \Omega^0 \cup \Omega^1 \cup \Omega^2 \cup \dots$$

die Menge aller *Wörter* oder *Zeichenketten* (engl. *string*) über dem *Alphabet* Ω . Es soll also gelten $\omega \in \Omega^* \iff \exists n \in \mathbb{N} : \omega \in \Omega^n$.

Statt (a_1, a_2, \dots, a_n) schreibt man in diesem Zusammenhang einfach $a_1 a_2 \dots a_n$.

Das Wort $\epsilon = () \in \Omega^0 \subseteq \Omega^*$ heißt das *leere Wort*.

Die *Länge* eines Wortes $\omega \in \Omega^*$ ist definiert als das (eindeutig bestimmte) $n \in \mathbb{N}$ mit $\omega \in \Omega^n$, und wird mit $|\omega| := n$ bezeichnet.

Beispiel. $\Omega = \{a, b, c\}$, $|abcba| = 5$, $|\epsilon| = 0$.

Definition 2. Seien $\omega = a_1 a_2 \dots a_n$, $\chi = b_1 b_2 \dots b_m$ zwei Wörter über einem Alphabet Ω . Dann heißt $\omega \circ \chi := a_1 a_2 \dots a_n b_1 b_2 \dots b_m \in \Omega^*$ die *Konkatenation* (engl. *concatenation*) von ω und χ .

Folgende Beobachtungen über die Konkatenation lassen sich leicht nachvollziehen:

1. Alle Wörter lassen sich durch Konkatenation aus Elementen des Alphabets bilden, z.B. gilt $abcca = a \circ b \circ c \circ c \circ a$.
2. $\forall \omega, \chi \in \Omega^* : |\omega \circ \chi| = |\omega| + |\chi|$
(Die Schreibweise $\forall x, y \in M$ bedeutet $\forall x \in M \forall y \in M$; die entsprechende Abkürzung für \exists ist auch erlaubt.)
3. Im allgemeinen gilt $\omega \circ \chi \neq \chi \circ \omega$
4. $\forall \omega \in \Omega^* : \omega \circ \epsilon = \epsilon \circ \omega = \omega$
5. $\forall \omega, \chi, \sigma \in \Omega^* : (\omega \circ \chi) \circ \sigma = \omega \circ (\chi \circ \sigma)$

Die letzten beiden Formeln haben verblüffende Ähnlichkeit mit bekannten Rechengesetzen für Zahlen:

$$\forall a \in \mathbb{Z} : a + 0 = 0 + a = a \quad \text{und} \quad \forall a, b, c \in \mathbb{Z} : (a + b) + c = a + (b + c).$$

Das wirft interessante Fragen auf:

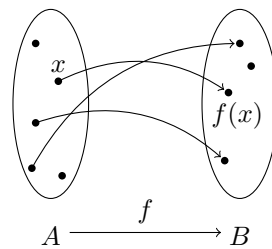
- Kann man mit Wörtern rechnen wie mit Zahlen?
- Wenn man Funktionen auf Wörtern definiert, wie verhalten sich diese zur Konkatenation?
- Wie kann man Gleichungen für Wörter lösen?
- Was, wenn man bestimmte Wörter als „im wesentlichen gleich“ auffassen will (z.B. gleich bis auf Groß-Kleinschreibung)?

Um solche Fragen zu klären, werden wir uns in den folgenden beiden Abschnitten Funktionen und Relationen genauer ansehen. Einige weitere interessante Fragen sind zum Beispiel:

- wie viele Wörter der Länge n gibt es, die ein bestimmtes vorgegebenes Wort nicht als Teilwort enthalten?
- wie findet man heraus, ob ein Wort in einem anderen enthalten ist?

Wir werden auf solche und ähnliche Fragen im Verlauf der Vorlesung zurückkommen. Wörter und Mengen von Wörtern sind dabei immer nur als Beispiel für einen (relativ einfachen) Typ diskreter Strukturen anzusehen. Ähnliche Fragen lassen sich auch für kompliziertere Strukturen stellen, denen wir später begegnen werden.

3 Funktionen



Definition 3. Seien A, B zwei Mengen. Eine Teilmenge $f \subseteq A \times B$ heißt *Funktion* (engl. *function*) von A nach B , falls gilt:

1. $\forall a \in A \exists b \in B : (a, b) \in f$
2. $\forall a \in A \forall b_1, b_2 \in B : (a, b_1) \in f \wedge (a, b_2) \in f \Rightarrow b_1 = b_2$.

Wenn f eine Funktion ist, schreibt man $f: A \rightarrow B$ statt $f \subseteq A \times B$ und $f(a) = b$ statt $(a, b) \in f$.

Man nennt A den Definitionsbereich und B den Bildbereich von f .

Beispiel.

1. Seien $A = \{1, 2, 3\}$ und $B = \{a, b, c\}$. Eine Teilmenge von $A \times B$ lässt sich in Tabellenform notieren, indem man die Zellen $(x, y) \in A \times B$ markiert, die zur Teilmenge gehören sollen. Welche der folgenden Teilmengen von $A \times B$ sind Funktionen von A nach B ?

c		●	
b			●
a	●		
	1	2	3

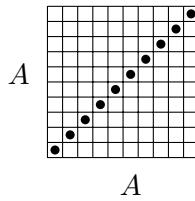
c			
b		●	●
a	●		
	1	2	3

c			
b			●
a	●		
	1	2	3

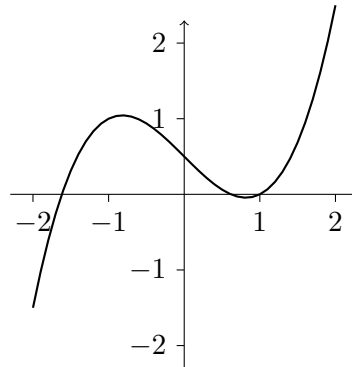
c		●	
b		●	
a	●		●
	1	2	3

Die ersten beiden sind Funktionen, die anderen nicht. Die dritte verletzt die erste Bedingung der Definition und die vierte die zweite Bedingung der Definition. Nur wenn beide Bedingungen erfüllt sind, handelt es sich um eine Funktion.

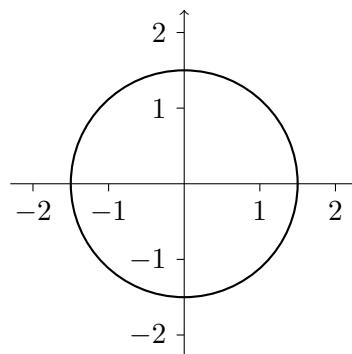
2. Sei A irgendeine Menge. Dann ist $\text{id}_A: A \rightarrow A, \text{id}_A(x) = x$ eine Funktion, die sogenannte *Identitätsfunktion*.



3. Bei $f: \mathbb{R} \rightarrow \mathbb{R}$, $f(x) = \frac{1}{2}x^3 - x + \frac{1}{2}$ handelt es sich um eine Funktion:



Dagegen entspricht die Kreislinie um den Ursprung mit Radius $3/2$ nicht einer Funktion:



Für $x < -1$ und $x > 1$ verletzt sie die erste Bedingung der Definition, und für $-1 < x < 1$ verletzt sie die zweite.

4. $f: \Omega^* \rightarrow \mathbb{N}$, $f(\omega) = |\omega|$ ist eine Funktion.
5. Manche Funktionen lassen sich am leichtesten durch eine Fallunterscheidung beschreiben, z.B. die Funktion

$$f: \mathbb{Z} \rightarrow \mathbb{Z}, \quad f(x) = \begin{cases} 1 & \text{falls } x \text{ eine Primzahl ist} \\ 0 & \text{falls nicht} \end{cases}$$

Um eine zulässige Funktionsdefinition zu bekommen, muss die Fallunterscheidung geeignet gewählt sein. Damit die erste Bedingung erfüllt ist, muss jedes x von mindestens einem Fall abgedeckt sein. Die zweite Bedingung der Definition ist dann erfüllt, wenn jedes x von höchstens einem Fall abgedeckt ist.

6. Bei $f: \mathbb{R} \rightarrow \mathbb{R}, f(x) = \frac{1}{x}$ handelt es sich **nicht** um eine Funktion, weil $f(0)$ nicht definiert ist. Auch

$$f: \Omega^* \rightarrow \Omega, f(\omega) := \text{erster Buchstabe von } \omega$$

ist keine gültige Definition, weil in diesem Fall $f(\epsilon)$ nicht definiert ist.

Man sollte in solchen Fällen erst gar nicht die Funktionsnotation $f: A \rightarrow B$ verwenden, sondern entweder allgemein von einer Teilmenge f von $A \times B$ sprechen oder die Definition so korrigieren, dass es sich wirklich um eine Funktion handelt. Zum Beispiel sind

$$f: \mathbb{R} \setminus \{0\} \rightarrow \mathbb{R}, \quad f(x) = \frac{1}{x}$$

$$f: \Omega^* \rightarrow \Omega \cup \{\epsilon\}, \quad f(\omega) = \begin{cases} \text{erster Buchstabe von } \omega & \text{falls } |\omega| \geq 1 \\ \epsilon & \text{falls } |\omega| = 0 \end{cases}$$

gültige Funktionsdefinitionen.

7. Seien $\Omega, \bar{\Omega}$ zwei Alphabete und sei $f: \Omega \rightarrow \bar{\Omega}$ eine Funktion.

Dann ist auch $F: \Omega^* \rightarrow \bar{\Omega}^*$ mit $F(a_1 a_2 \dots a_n) := f(a_1) f(a_2) \dots f(a_n)$ eine Funktion.

Diese Funktion ist mit der Komposition verträglich in dem Sinn, dass gilt

$$\forall \omega, \chi \in \Omega^* : F(\omega \circ \chi) = F(\omega) \circ F(\chi).$$

Eine Funktion mit dieser Eigenschaft heißt *Worthomomorphismus*.

8. Betrachte folgenden Algorithmus:

INPUT: eine Zahl $x \in \mathbb{Z}$

OUTPUT: eine Zahl $y \in \mathbb{Z}$

- 1 setze $z = 1$
- 2 falls $x < 0$ ist,
- 3 setze $z = -1$ und $x = -x$.
- 4 setze $y = 0$
- 5 solange $x \neq 0$ ist:
- 6 falls x ungerade ist:
- 7 setze $y = y + 1$ und $x = x - 1$
- 8 setze $x = x/2$.
- 9 gib $z \cdot y$ als Ergebnis zurück.

Dieser Algorithmus definiert eine Funktion $f: \mathbb{Z} \rightarrow \mathbb{Z}$. Um das einzusehen, muss man sich überlegen, dass der Algorithmus (a) für jede mögliche Eingabe eine Ausgabe liefert (und nicht etwa einen Fehler produziert oder in einer Endlosschleife landet), und (b) dass die Ausgabe nur von der Eingabe abhängt (und nicht etwa auch von einem Zufallsgenerator oder der Uhrzeit oder sonstigen externen Größen, die bei verschiedenen Aufrufen verschiedene Werte annehmen können).

Intuitiv kann man sich eine Funktion grundsätzlich vorstellen als etwas, das einen Input nimmt und dazu einen passenden Output produziert. Aber Vorsicht: nicht jede (mathematische) Funktion lässt sich durch einen Algorithmus berechnen. Es gibt sogenannte

„unberechenbare“ Funktionen, für die man beweisen kann, dass es keinen Algorithmus gibt, der diese Funktionen berechnet. Wie solche Funktionen aussehen, erfahren Sie in Vorlesungen über theoretische Informatik.

9. Für den (mathematischen) Begriff einer Funktion ist nicht wesentlich, dass man zu jedem x aus dem Definitionsbereich den zugehörigen Funktionswert $f(x)$ explizit bestimmen kann. Es genügt, dass die Funktionswerte durch die Definition eindeutig festgelegt sind. Zum Beispiel ist

$$f: \mathbb{R} \rightarrow \mathbb{R}, \quad f(x) = \begin{cases} 1 & \text{falls am 1.8. des Jahres 50 n.Chr. ein römischer Soldat} \\ & \text{innerhalb der heutigen Stadtgrenzen von Linz ein Glas} \\ & \text{Milch verschüttet hat.} \\ 0 & \text{falls nicht} \end{cases}$$

eine gültige Funktionsdefinition. Es handelt sich nämlich definitiv um eine der folgenden beiden Funktionen:

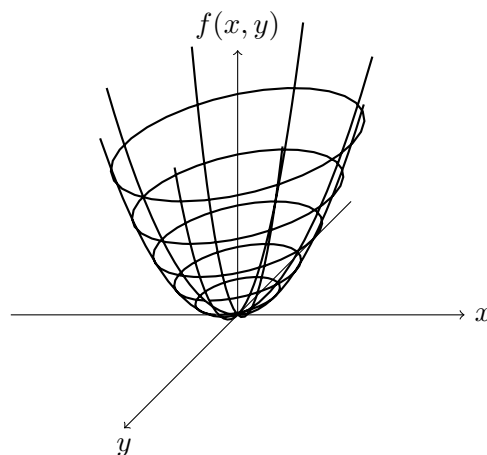
$$g: \mathbb{R} \rightarrow \mathbb{R}, g(x) = 1 \quad \begin{array}{c} \uparrow \\ \text{---} \\ \downarrow \\ \text{---} \end{array} \quad \text{oder} \quad h: \mathbb{R} \rightarrow \mathbb{R}, h(x) = 0 \quad \begin{array}{c} \uparrow \\ \text{---} \\ \downarrow \\ \text{---} \end{array}$$

Dass es sich nicht ohne weiteres herausfinden lässt, um welche der beiden, mag ein praktisches Problem sein, ein theoretisches ist es nicht.

10. Funktionen, die von mehreren Variablen abhängen, sind formal dasselbe wie Funktionen, deren Definitionsbereich Tupel sind. Ein Beispiel ist die Funktion

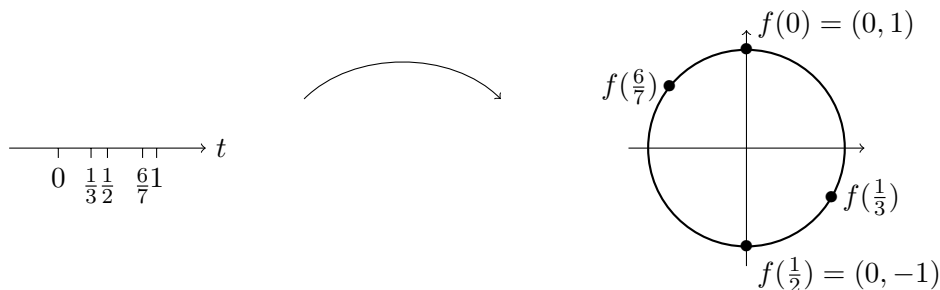
$$f: \mathbb{R}^2 \rightarrow \mathbb{R}, \quad f(x, y) = x^2 + y^2,$$

die jedem Paar (x, y) zweier reeller Zahlen eine reelle Zahl $f(x, y)$ zuordnet. Eine solche Funktion kann man sich als Gebirgslandschaft vorstellen:



Umgekehrt kann auch ein Funktionswert ein Tupel sein. Zum Beispiel lässt sich mit einer Funktion $f: [0, 1] \rightarrow \mathbb{R}^3$ die Flugbahn eines Teilchens im dreidimensionalen Raum beschreiben: zum Zeitpunkt $t = 0$ befindet sich das Teilchen am Punkt $f(0)$, zum Zeitpunkt $t = 1/2$ am Punkt $f(1/2)$, usw. Man kann mit solche Funktionen auch Kurven

in der Ebene beschreiben. Zum Beispiel die Kreislinie mit der Funktion $f: [0, 1] \rightarrow \mathbb{R}^2$, $f(t) = (\sin(2\pi t), \cos(2\pi t))$. Bei dieser Funktion wird der Kreis beginnend beim Punkt $(0, 1)$ im Uhrzeigersinn durchlaufen:



Definition 4. Sei $f: A \rightarrow B$ eine Funktion.

1. f heißt *injektiv* (engl. *injective*), falls gilt:

$$\forall a_1, a_2 \in A : f(a_1) = f(a_2) \Rightarrow a_1 = a_2.$$

2. f heißt *surjektiv* (engl. *surjective*), falls gilt:

$$\forall b \in B \exists a \in A : f(a) = b.$$

3. f heißt *bijektiv* (engl. *bijective*), falls f sowohl injektiv als auch surjektiv ist.

Injektiv bedeutet, dass jedes Element des Bildbereichs *höchstens* einmal getroffen wird. Surjektiv bedeutet, dass jedes Element des Bildbereichs *mindestens* einmal getroffen wird. Bijektiv bedeutet, dass jedes Element des Bildbereichs *genau* einmal getroffen wird. Man vergleiche die beiden Formeln in obiger Definition mit den beiden Formeln in Def. 3.

Beispiel.

1. $A = \{1, 2, 3\}$, $B = \{a, b, c, d\}$. Von den folgenden beiden Funktionen ist die linke injektiv, die rechte nicht (weil der Wert c mehr als einmal getroffen wird).

d			
c	●		
b			●
a	●		
	1	2	3

d			
c		●	●
b			
a	●		
	1	2	3

Eine surjektive Funktion von A nach B kann es in diesem Fall offenbar nicht geben, weil die Menge B mehr Elemente hat als A .

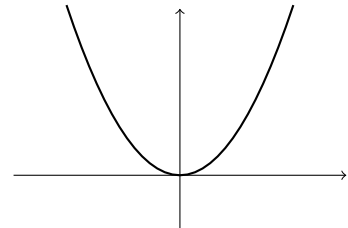
2. $A = \{1, 2, 3, 4\}$, $B = \{a, b, c\}$. Von den folgenden beiden Funktionen ist die linke surjektiv, die rechte nicht (weil der Wert b nicht getroffen wird).

c		●		
b	●			
a			●	●
	1	2	3	4

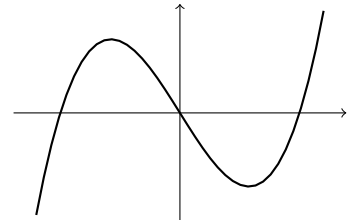
c	●			
b				
a	●		●	●
	1	2	3	4

Eine injektive Funktion von A nach B kann es in diesem Fall offenbar nicht geben, weil die Menge A mehr Elemente hat als B .

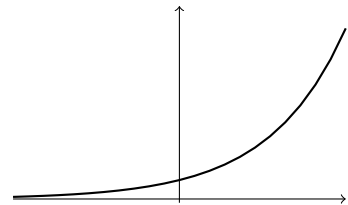
3. Die Funktion $f: \mathbb{R} \rightarrow \mathbb{R}$, $f(x) = x^2$ ist weder injektiv noch surjektiv. Sie ist nicht injektiv, weil z.B. der Wert 4 des Bildbereichs sowohl von 2 als auch von -2 getroffen wird: $f(-2) = f(2) = 4$. Sie ist nicht surjektiv, weil z.B. der Wert -1 des Bildbereichs überhaupt nicht getroffen wird.



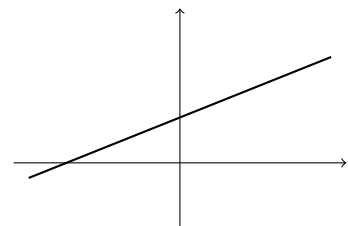
4. Die Funktion $f: \mathbb{R} \rightarrow \mathbb{R}$, $f(x) = x^3 - 10x$ ist surjektiv, aber nicht injektiv. Dass sie surjektiv ist, beweist man mit Techniken, die man in der Analysis lernt. Dass sie nicht injektiv ist, sieht man daran, dass z.B. der Wert 0 sowohl an der Stelle $x = 0$ als auch an den Stellen $\sqrt{10}$ und $-\sqrt{10}$ angenommen wird.



5. Die Funktion $f: \mathbb{R} \rightarrow \mathbb{R}$, $f(x) = e^x$ ist injektiv, aber nicht surjektiv. Dass sie injektiv ist, zeigt man wieder mit den Mitteln der Analysis. Dass sie nicht surjektiv ist, sieht man daran, dass z.B. der Wert -1 nirgends angenommen wird.



6. Die Funktion $f: \mathbb{R} \rightarrow \mathbb{R}$, $f(x) = \frac{1}{3}x + 1$ ist bijektiv.
 Beweis: (1) sie ist injektiv, denn wenn $x, y \in \mathbb{R}$ so sind, dass $f(x) = f(y)$ gilt, dann gilt $\frac{1}{3}x + 1 = \frac{1}{3}y + 1$, also $\frac{1}{3}x = \frac{1}{3}y$, also $x = y$, wie gefordert.
 (2) sie ist surjektiv, denn wenn $y \in \mathbb{R}$ beliebig ist, dann gibt es ein passendes $x \in \mathbb{R}$, nämlich $x = 3y - 3$, so dass $f(x) = \frac{1}{3}(3y - 3) + 1 = y$ ist.



7. Die Funktion $f: \mathbb{Z} \rightarrow \mathbb{Z}$, $f(x) = 3x + 2$ ist injektiv, aber nicht surjektiv. Dass sie injektiv ist, zeigt man ähnlich wie im vorherigen Beispiel. Dass sie nicht surjektiv ist, liegt daran, dass z.B. der Wert 2 des Bildbereiches nicht getroffen wird.

Definition 5. Seien $f: A \rightarrow B$, $g: B \rightarrow C$ zwei Funktionen. Dann heißt die Funktion $g \circ f: A \rightarrow C$, $(g \circ f)(x) := g(f(x))$ die *Verkettung* oder *Hintereinanderausführung* oder *Komposition* (engl. *composition*) von f und g .

Satz 1.

1. Die Verkettung injektiver Funktionen ist injektiv.
2. Die Verkettung surjektiver Funktionen ist surjektiv.
3. Die Verkettung bijektiver Funktionen ist bijektiv.

Beweis.

1. Seien $f: A \rightarrow B$ und $g: B \rightarrow C$ injektive Funktionen und $h = g \circ f$.

Zu zeigen: h ist injektiv, also $\forall x_1, x_2 \in A : h(x_1) = h(x_2) \Rightarrow x_1 = x_2$.

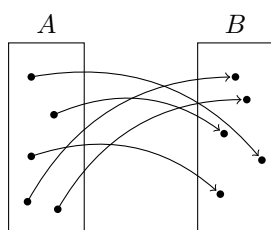
Seien also $x_1, x_2 \in A$ mit $h(x_1) = h(x_2)$, d. h. $g(f(x_1)) = g(f(x_2))$. Da g nach Annahme injektiv ist, folgt zunächst $f(x_1) = f(x_2)$. Und daraus folgt, da nach Annahme auch f injektiv ist, $x_1 = x_2$, was zu zeigen war.

2., 3. Übung. ■

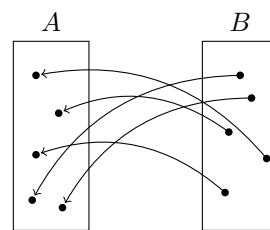
Satz 2. $f: A \rightarrow B$ ist genau dann bijektiv, wenn es eine Funktion $g: B \rightarrow A$ gibt mit $g \circ f = \text{id}_A$ und $f \circ g = \text{id}_B$.

Diese Funktion g ist dann eindeutig bestimmt und ihrerseits bijektiv.

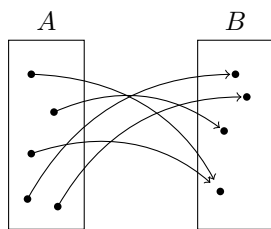
Dieser Satz besagt, dass eine Funktion genau dann bijektiv ist, wenn man durch umkehren der Abbildungsrichtung wieder eine Funktion bekommt. Man nennt die Funktion g deshalb auch die *Umkehrfunktion* oder *Inverse* (engl. *inverse*) von f , Notation: $f^{-1} := g$.



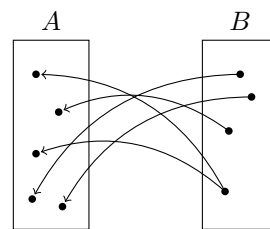
$f: A \rightarrow B$ ist bijektiv



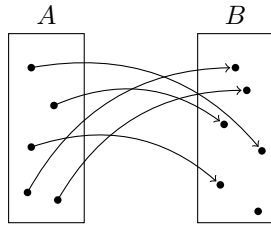
Umkehrfunktion $f^{-1}: B \rightarrow A$



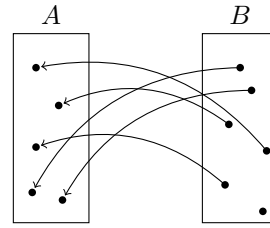
$f: A \rightarrow B$ ist nicht injektiv



Umkehrung ist keine Funktion!



$f: A \rightarrow B$ ist nicht surjektiv



Umkehrung ist keine Funktion!

Beispiel.

1. Sei $A = \{1, 2, 3, 4, 5\}$, $B = \{a, b, c, d, e\}$. Die unten links dargestellte Funktion $f: A \rightarrow B$ ist bijektiv. Ihre Umkehrfunktion $f^{-1}: B \rightarrow A$ (rechts dargestellt) ergibt sich durch Spiegelung der Tabelle an der Diagonalen.

e			●		
d	●				
c		●			
b					●
a				●	
	1	2	3	4	5

5		●			
4	●				
3					●
2			●		
1				●	
	a	b	c	d	e

Wenn A und B endliche Mengen sind, dann gibt es eine bijektive Funktion $f: A \rightarrow B$ offenbar genau dann wenn A und B gleich viele Elemente enthalten.

2. Die Funktion $f: \mathbb{R} \rightarrow \mathbb{R}$, $f(x) = \frac{1}{3}x + 1$ ist bijektiv. Die Umkehrfunktion ist $f^{-1}: \mathbb{R} \rightarrow \mathbb{R}$, $f^{-1}(y) = 3y - 3$. In der Tat gilt $f^{-1}(f(x)) = 3(\frac{1}{3}x + 1) - 3 = x$ für jedes $x \in \mathbb{R}$ sowie $f(f^{-1}(y)) = \frac{1}{3}(3y - 3) + 1 = y$ für jedes $y \in \mathbb{R}$.
3. Dr. T. hat den idealen Kompressionsalgorithmus gefunden: er fasst die zu komprimierende (Binär-)Datei als Wort über dem Alphabet $\Omega = \{0, 1\}$ auf und wendet die Funktion $f: \Omega^* \rightarrow \Omega^*$, $f(\omega) = \epsilon$ an. Das Ergebnis könnte kürzer nicht sein. Am Dekompressionsalgorithmus arbeitet er noch.

Er wird natürlich keinen finden, weil sein f nicht bijektiv ist. Die Kompressionsfunktion muss zumindest injektiv sein, damit eine Dekompression funktionieren kann, denn wenn es zwei Wörter $\omega_1 \neq \omega_2$ mit $f(\omega_1) = f(\omega_2)$ gibt, dann kann der Dekompressionsalgorithmus nicht wissen, ob er aus diesem Bild ω_1 oder ω_2 rekonstruieren soll.

Surjektivität ist dagegen verzichtbar, denn wenn $f: \Omega^* \rightarrow \Omega^*$ zwar injektiv aber nicht surjektiv ist, dann kann man die modifizierte Funktion

$$\tilde{f}: \Omega^* \rightarrow B := \{f(\omega) : \omega \in \Omega^*\} \subseteq \Omega^*$$

$$\tilde{f}(\omega) := f(\omega)$$

betrachten. Diese ist bijektiv, hat also eine Umkehrfunktion $\tilde{f}: B \rightarrow \Omega^*$. Wir können damit ein gegebenes Wort $\chi \in \Omega^*$ wie folgt dekomprimieren:

- 1 falls $\chi \notin B$
- 2 Fehlermeldung „Dies ist keine komprimierte Datei“
- 3 sonst
- 3 gib $\tilde{f}^{-1}(\chi)$ zurück.

4 Relationen

Für ein einzelnes Objekt x kann man sich fragen, ob es eine bestimmte Eigenschaft hat oder nicht, d.h. formal ob es zu einer bestimmten Menge gehört oder nicht (nämlich zur Menge aller Elemente, die die fragliche Eigenschaft haben). Bei zwei Objekten x, y kann man sich fragen, ob sie zueinander in einer bestimmten Beziehung stehen. Ob das der Fall ist, wird im allgemeinen nicht allein von x und nicht allein von y abhängen, sondern ist vielmehr eine Eigenschaft des Paares (x, y) . Das motiviert die folgende Definition:

Definition 6. Sei A eine Menge. Eine Teilmenge $R \subseteq A \times A$ heißt *Relation* (engl. *relation*) auf A . Statt $(x, y) \in R$ schreibt man $x R y$ und statt $(x, y) \notin R$ schreibt man $x \not R y$.

Beispiel.

1. \leq ist eine Relation auf \mathbb{Z} . Es gilt zum Beispiel $-5 \leq 3$ und $8 \leq 9$, aber $3 \not\leq 1$.
2. Teilbarkeit (engl. *divisibility*): $A = \mathbb{Z}$, $| = \{ (x, y) \in \mathbb{Z}^2 \mid \exists z \in \mathbb{Z} : xz = y \}$. Es gilt zum Beispiel $3 \mid 9$, $3 \nmid 10$, $5 \nmid 9$, $5 \nmid 10$.
3. Ist A eine Menge, so ist die Teilmengenbeziehung \subseteq eine Relation auf der Potenzmenge $\mathcal{P}(A)$.
4. Sei Ω ein Alphabet. Ein Wort $\omega \in \Omega^*$ heißt *Teilwort* (engl. *subword*) eines anderen Worts $\chi \in \Omega^*$, falls es zwei weitere Wörter $\sigma, \rho \in \Omega^*$ gibt mit $\chi = \sigma \circ \omega \circ \rho$. Dann ist

$$R := \{ (\omega, \chi) \in \Omega^* : \omega \text{ ist ein Teilwort von } \chi \}$$

eine Relation auf Ω^* .

5. Sei $\Omega = \{a, A, b, B, \dots, z, Z\}$. Dann ist

$$\leq_{\text{lex}} := \{ (\omega, \chi) \in \Omega^* : \omega \text{ kommt lexikographisch vor } \chi \}$$

eine Relation auf Ω^* . Es gilt zum Beispiel **Aachen** \leq_{lex} **Aal** \leq_{lex} \dots \leq_{lex} **Zwickau**.

Eine weitere Relation ist

$$\sim := \{ (\omega, \chi) \in \Omega^* : \omega \text{ und } \chi \text{ unterscheiden sich nur durch Groß- und Kleinschreibung} \}.$$

Dabei gilt zum Beispiel **wOrT** \sim **WoRt** $\not\sim$ **trow**.

6. Sei V die Menge aller Teilnehmer der Vorlesungsklausur und R die Menge aller $(x, y) \in V \times V$, so dass x und y dieselbe Note bekommen. Dann ist R eine Relation auf V .
7. Eine Relation auf der Menge M aller Menschen ist

$$\heartsuit := \{ (x, y) \in M^2 : x \text{ mag } y \}.$$
8. Auf der Menge F aller Formeln ist die Implikation \Rightarrow eine Relation.

Definition 7. Sei A eine Menge und $R \subseteq A \times A$ eine Relation auf A . Man nennt R eine (partielle) *Ordnungsrelation* oder *Halbordnung* (engl. *order*), falls gilt:

1. $\forall x \in A : x R x$ (Reflexivität)
2. $\forall x, y \in A : x R y \wedge y R x \Rightarrow x = y$ (Antisymmetrie)
3. $\forall x, y, z \in A : x R y \wedge y R z \Rightarrow x R z$ (Transitivität)

Man spricht von einer *Totalordnung* (engl. *total order*), wenn außerdem gilt

4. $\forall x, y \in A : x R y \vee y R x$.

Beispiel.

1. \leq ist eine Totalordnung auf \mathbb{Z}
2. \leq_{lex} ist eine Totalordnung auf $\{\mathbf{a}, \mathbf{A}, \dots, \mathbf{z}, \mathbf{Z}\}^*$.
Die ist-Teilwort-von-Relation ist eine Halbordnung, aber keine Totalordnung.
3. \subseteq ist eine Halbordnung auf $\mathcal{P}(A)$, aber im allgemeinen keine Totalordnung. Für $A = \{1, 2\}$ gilt zum Beispiel $\{1\} \in \mathcal{P}(A)$ und $\{2\} \in \mathcal{P}(A)$, aber weder $\{1\} \subseteq \{2\}$ noch $\{2\} \subseteq \{1\}$.
4. $|$ ist eine Halbordnung auf \mathbb{Z} , aber keine Totalordnung, denn es gilt zum Beispiel sowohl $3 \nmid 5$ als auch $5 \nmid 3$.
5. Auf $M = \mathbb{Z} \times \mathbb{Z}$ wird eine Ordnung \leq definiert durch

$$(a_1, a_2) \leq (b_1, b_2) :\iff a_1 \leq b_1 \wedge a_2 \leq b_2,$$

wobei mit \leq auf der rechten Seite die übliche Ordnung auf \mathbb{Z} gemeint ist.

Es gilt dann zum Beispiel $(3, 5) \leq (7, 8)$. Bei dieser Halbordnung handelt es sich nicht um eine Totalordnung, weil zum Beispiel die Elemente $(3, 7)$ und $(5, 3)$ nicht miteinander vergleichbar sind, d.h. es gilt weder $(3, 7) \leq (5, 3)$ noch $(5, 3) \leq (3, 7)$.

Definition 8. Sei A eine Menge und $R \subseteq A \times A$ eine Relation auf A . Man nennt R eine *Äquivalenzrelation* (engl. *equivalence relation*), falls gilt:

1. $\forall x \in A : x R x$ (Reflexivität)
2. $\forall x, y \in A : x R y \Rightarrow y R x$ (Symmetrie)
3. $\forall x, y, z \in A : x R y \wedge y R z \Rightarrow x R z$ (Transitivität)

Beispiel.

1. Für jede Menge A ist die Gleichheitsrelation $=$ eine Äquivalenzrelation, denn für alle Objekte x, y, z gilt $x = x$ (Reflexivität), $x = y \iff y = x$ (Symmetrie) und $x = y \wedge y = z \Rightarrow x = z$ (Transitivität).

Im allgemeinen darf man sich eine Äquivalenzrelation vorstellen als eine abgeschwächte Variante der Gleichheitsrelation, bei der bestimmte irrelevante Eigenschaften ignoriert werden.

2. Sei $A = \{\square, \blacksquare, \square, \blacksquare, \circ, \bullet, \bullet, \circ, \triangle, \blacktriangle, \blacktriangle, \triangle\}$. Man kann sich für diese Menge verschiedene Äquivalenzrelationen vorstellen. Hier sind ein paar Möglichkeiten:

- $x \sim y$, falls x und y die gleiche Form haben – dann gilt z. B. $\square \sim \blacksquare$ und $\square \not\sim \circ$.
- $x \sim y$, falls x und y die gleiche Farbe haben – dann gilt z. B. $\square \sim \circ$ und $\square \not\sim \blacksquare$.
- $x \sim y$, falls x und y die gleiche Höhe haben – dann gilt z. B. $\square \sim \blacktriangle$ und $\square \not\sim \triangle$.
- $x \sim y$, falls x und y sich höchstens in der Farbe unterscheiden – dann gilt z. B. $\square \sim \blacksquare$ und $\square \not\sim \bullet$.

3. Sei $\Omega = \{\mathbf{a}, \mathbf{A}, \mathbf{b}, \mathbf{B}, \dots, \mathbf{z}, \mathbf{Z}\}$. Die Relation \sim auf Ω^* , bei der $\omega \sim \chi$ gilt, wenn sich die Wörter nur durch Groß- und Kleinschreibung voneinander unterscheiden, ist eine Äquivalenzrelation.

4. Sei $m \in \mathbb{Z}$ und $\equiv_m := \{(x, y) \in \mathbb{Z}^2 : m \mid x - y\}$. Dann ist \equiv_m eine Äquivalenzrelation auf \mathbb{Z} . Es gilt zum Beispiel:

$$\begin{aligned} 0 &\equiv_3 3 \equiv_3 6 \equiv_3 9 \equiv_3 -3 \equiv_3 -6 \equiv_3 15 \equiv_3 \dots \\ 1 &\equiv_3 4 \equiv_3 7 \equiv_3 10 \equiv_3 -2 \equiv_3 -5 \equiv_3 16 \equiv_3 \dots \\ 2 &\equiv_3 5 \equiv_3 8 \equiv_3 11 \equiv_3 -1 \equiv_3 -4 \equiv_3 17 \equiv_3 \dots \end{aligned}$$

5. Ist A die Menge aller Schüler einer Volksschule und

$$R = \{(x, y) \in A \times A : x \text{ und } y \text{ gehen in dieselbe Klasse}\},$$

dann ist R eine Äquivalenzrelation auf A . Klassenbildung ist symptomatisch für Äquivalenzrelationen.

Definition 9. Ist \sim eine Äquivalenzrelation auf A und ist $x \in A$, so heißt

$$[x]_{\sim} := \{y \in A : x \sim y\}$$

die *Äquivalenzklasse* von x (bezüglich \sim). Man schreibt $A/\sim := \{[x]_{\sim} : x \in A\}$ für die Menge aller Äquivalenzklassen von Elementen von A .

Beispiel. Was sind bei den Äquivalenzrelationen aus dem vorherigen Beispiel die Äquivalenzklassen?

1. Bezüglich $=$ ist die Äquivalenzklasse eines Elementes $x \in A$ genau die Menge $\{x\}$, die nur dieses Element enthält.
2. Die genannten Äquivalenzrelationen auf $A = \{\square, \blacksquare, \square, \blacksquare, \circ, \bullet, \bullet, \circ, \triangle, \blacktriangle, \triangle, \triangle\}$ teilen A in folgende Äquivalenzklassen auf:

- gleiche Form: $\{\square, \blacksquare, \square, \blacksquare\}$, $\{\circ, \bullet, \bullet, \circ\}$, $\{\triangle, \blacktriangle, \triangle, \triangle\}$
- gleiche Farbe: $\{\square, \square, \circ, \triangle, \triangle\}$, $\{\blacksquare, \blacksquare, \bullet, \bullet, \blacktriangle\}$, $\{\blacksquare, \bullet, \blacktriangle\}$
- gleiche Höhe: $\{\square, \blacksquare, \square, \circ, \triangle, \blacktriangle, \blacktriangle\}$, $\{\square, \blacksquare, \bullet, \bullet, \triangle\}$
- gleich bis auf Farbe: $\{\square, \blacksquare, \blacksquare\}$, $\{\square, \blacksquare\}$, $\{\circ, \bullet\}$, $\{\bullet, \circ\}$, $\{\triangle, \blacktriangle, \blacktriangle\}$, $\{\triangle\}$

3. Die Äquivalenzrelation \sim auf Ω^* hat sehr viele Äquivalenzklassen. Ein Beispiel ist

$$[aBc]_{\sim} = \{abc, Abc, aBc, ABc, abC, AbC, aBC, ABC\}.$$

Eine weitere Äquivalenzklasse ist

$$[qM]_{\sim} = \{qm, qM, Qm, QM\}.$$

Die Äquivalenzklassen lassen sich auffassen als ein verallgemeinerter Typ von Wörtern, bei dem zwischen Groß- und Kleinschreibung nicht unterschieden wird. Es gilt zum Beispiel $[aBc]_{\sim} = [Abc]_{\sim} \neq [qM]_{\sim}$. Die Relation \sim auf Ω^* entspricht der Relation $=$ auf Ω^*/\sim .

4. Wir haben

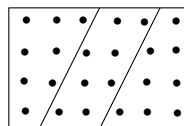
$$\begin{aligned} [0]_{\equiv_3} &= \{\dots, -3, 0, 3, 6, 9, \dots\} \\ [1]_{\equiv_3} &= \{\dots, -2, 1, 4, 7, 10, \dots\} \\ [2]_{\equiv_3} &= \{\dots, -1, 2, 5, 8, 11, \dots\} \\ [3]_{\equiv_3} &= [0]_{\equiv_3} \\ [4]_{\equiv_3} &= [1]_{\equiv_3} \end{aligned}$$

und so weiter.

5. In diesem Fall sind die Äquivalenzklassen genau die Schulklassen.

Satz 3. Es seien A eine Menge, \sim eine Äquivalenzrelation auf A , und $x, y \in A$. Dann gilt:

1. $x \sim y \iff [x]_{\sim} = [y]_{\sim}$
2. $x \not\sim y \iff [x]_{\sim} \cap [y]_{\sim} = \emptyset$
3. $A = \bigcup_{C \in A/\sim} C$



Beweis.

1. „ \Rightarrow “ Annahme $x \sim y$, zu zeigen: $[x]_{\sim} = [y]_{\sim}$.

Aus Symmetriegründen genügt es, $[x]_{\sim} \subseteq [y]_{\sim}$ zu zeigen.

Sei also $z \in [x]_{\sim}$. Dann gilt:

$$\begin{array}{lcl} z \in [x]_{\sim} & \xrightarrow{\text{Def.}} & x \sim z \\ & \xrightarrow{\text{Sym.}} & z \sim x \\ \text{Ann. + Trans.} & \xrightarrow{\quad} & z \sim y \\ & \xrightarrow{\text{Sym.}} & y \sim z \\ & \xrightarrow{\text{Def.}} & z \in [y]_{\sim}. \end{array}$$

„ \Leftarrow “ Annahme: $[x]_{\sim} = [y]_{\sim}$, zu zeigen: $x \sim y$.

Wegen Reflexivität gilt jedenfalls $x \in [x]_{\sim}$. Wegen $[x]_{\sim} = [y]_{\sim}$ also auch $y \in [x]_{\sim}$, und damit nach Definition auch $x \sim y$.

2. „ \Rightarrow “ Annahme: $x \not\sim y$, zu zeigen: $[x]_{\sim} \cap [y]_{\sim} = \emptyset$.

Widerspruchsbeweis: wäre $[x]_{\sim} \cap [y]_{\sim} \neq \emptyset$, so gäbe es ein $z \in [x]_{\sim} \cap [y]_{\sim}$. Für dieses z gilt dann $z \sim x$ und $z \sim y$, also wegen Symmetrie und Transitivität auch $x \sim y$, im Widerspruch zur Annahme $x \not\sim y$.

„ \Leftarrow “ Annahme: $[x]_{\sim} \cap [y]_{\sim} = \emptyset$, zu zeigen: $x \not\sim y$.

Wäre $x \sim y$, dann wäre $[x]_{\sim} = [y]_{\sim}$ nach Teil 1, also $[x]_{\sim} \cap [y]_{\sim} = [x]_{\sim} \neq \emptyset$, da zumindest $x \in [x]_{\sim}$ (wegen Reflexivität). Damit ist $x \sim y$ ausgeschlossen und es bleibt nur $x \not\sim y$.

3. „ \subseteq “ Sei $y \in A$. Zu zeigen: $y \in \bigcup_{[x]_{\sim} \in A/\sim} [x]_{\sim}$. Das folgt direkt aus $y \in [y]_{\sim} \in A/\sim$.

„ \supseteq “ Sei $y \in \bigcup_{[x]_{\sim} \in A/\sim} [x]_{\sim}$. Dann gibt es ein $[x]_{\sim} \in A/\sim$ mit $y \in [x]_{\sim}$. Wegen $[x]_{\sim} \subseteq A$ folgt $y \in A$. ■

Seien A, B Mengen und \sim eine Äquivalenzrelation auf A . Um eine Funktion $h: A/\sim \rightarrow B$ zu definieren, ist es verlockend, zunächst von einer Funktion $\tilde{h}: A \rightarrow B$ auszugehen und dann zu sagen $h([x]_{\sim}) := \tilde{h}(x)$. Damit das funktioniert, muss man entweder präzisieren, auf welches Element x der Äquivalenzklasse $[x]_{\sim}$ die Funktion \tilde{h} angewendet werden soll, oder man muss sich vergewissern, dass die Wahl auf das Ergebnis keinen Einfluss hat. Im letzteren Fall sagt man, die Definition ist „repräsentantenunabhängig“, oder „die Funktion h ist wohldefiniert“.

Beispiel. Wir betrachten wieder das Alphabet $\Omega = \{\mathbf{a}, \mathbf{A}, \dots, \mathbf{z}, \mathbf{Z}\}$ und die Äquivalenzrelation \sim , die Groß-/Kleinschreibung ignoriert.

1. Bei

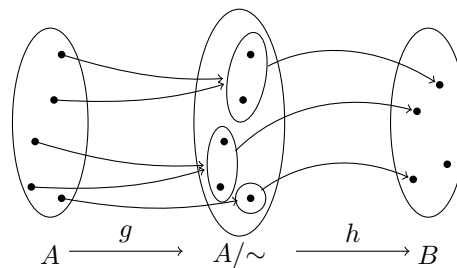
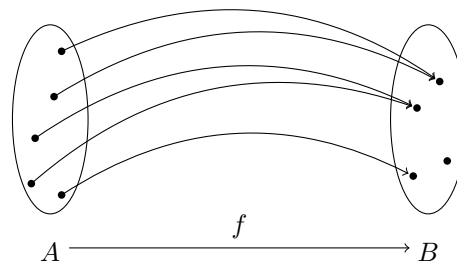
$$h: \Omega^*/\sim \setminus \{[\epsilon]_{\sim}\} \rightarrow \Omega, \quad h([\omega]_{\sim}) := \text{erster Buchstabe von } \omega$$

handelt es sich **nicht** um eine gültige Definition, denn einerseits gilt $h([\mathbf{aB}]_{\sim}) = \mathbf{a}$, und andererseits gilt wegen $\mathbf{aB} \sim \mathbf{Ab}$ auch $[\mathbf{aB}]_{\sim} = [\mathbf{Ab}]_{\sim}$, und daher muss auch $h([\mathbf{aB}]_{\sim}) = h([\mathbf{Ab}]_{\sim}) = \mathbf{A} \neq \mathbf{a}$ gelten. Das kann nicht sein.

2. Bei $h: \Omega^*/\sim \rightarrow \mathbb{N}$, $h([\omega]_{\sim}) := |\omega|$ handelt es sich um eine gültige Definition, denn wann immer $\omega \sim \chi$ gilt, gilt auch $|\omega| = |\chi|$. Die Definition ist also repräsentantenunabhängig.

Satz 4. (Homomorphiesatz für Mengen) Sei $f: A \rightarrow B$ eine Funktion.

1. Durch $x \sim y \iff f(x) = f(y)$ wird eine Äquivalenzrelation auf A definiert.
2. Die Funktion $g: A \rightarrow A/\sim$, $g(x) = [x]_\sim$ ist surjektiv.
3. Es gibt eine injektive Funktion $h: A/\sim \rightarrow B$ so dass $f = h \circ g$.
4. Diese Funktion h ist eindeutig bestimmt.
5. f ist genau dann surjektiv, wenn h bijektiv ist.



Beweis.

1., 2. Übung

3. Betrachte die Funktion $g: A \rightarrow A/\sim$ mit $g(x) = [x]_\sim$ für alle $x \in A$ und definiere die Funktion $h: A/\sim \rightarrow B$ durch $h([x]_\sim) = f(x)$. Diese Definition ist zulässig, weil $[x]_\sim = [y]_\sim \iff x \sim y \iff f(x) = f(y)$.

h ist injektiv, denn wenn $[x]_\sim, [y]_\sim \in A/\sim$ so sind, dass $h([x]_\sim) = h([y]_\sim)$ gilt, dann $f(x) = f(y)$, und dann $x \sim y$, und dann $[x]_\sim = [y]_\sim$.

Es gilt $f = h \circ g$, denn für alle $x \in A$ gilt $h(g(x)) = h([x]_\sim) = f(x)$.

4. Wenn $\bar{h}: A/\sim \rightarrow B$ eine andere Funktion mit $f = \bar{h} \circ g$ ist, müsste es ein $[x]_\sim \in A/\sim$ geben mit $h([x]_\sim) \neq \bar{h}([x]_\sim)$, obwohl doch $h([x]_\sim) = h(g(x)) = f(x) = \bar{h}(g(x)) = \bar{h}([x]_\sim)$ gelten soll.

5. Übung ■

Beispiel.

1. Seien $A = \{1, 2, 3, 4, 5, 6\}$, $B = \{a, b, c\}$ und sei f die folgende Funktion:

c				●	●	
b		●				
a	●		●			●
	1	2	3	4	5	6

Dann ist

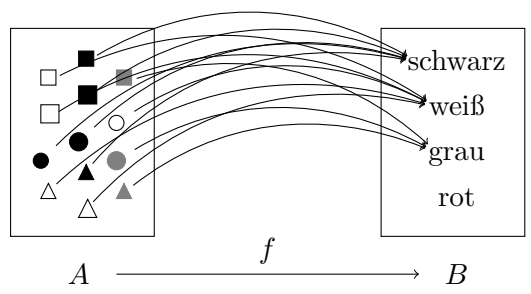
$$A/\sim = \{\{1, 3, 6\}, \{2\}, \{4, 5\}\}.$$

Die Funktionen g und h aus dem Satz sind gegeben durch

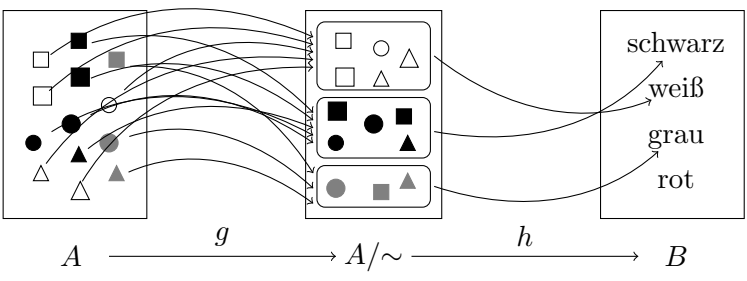
$$g: \begin{array}{c|c|c|c|c|c} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline \{1, 3, 6\} & \{2\} & \{1, 3, 6\} & \{4, 5\} & \{4, 5\} & \{1, 3, 6\} \end{array}$$

$$h: \begin{array}{c|c|c} \{1, 3, 6\} & \{2\} & \{4, 5\} \\ \hline a & b & c \end{array}$$

2. Seien $A = \{\square, \blacksquare, \square, \blacksquare, \square, \circ, \bullet, \bullet, \bullet, \triangle, \blacktriangle, \blacktriangle, \triangle\}$, $B = \{\text{schwarz}, \text{weiß}, \text{grau}, \text{rot}\}$, und sei $f: A \rightarrow B$ die Funktion, die jedem Element aus A dessen Farbe zuordnet:



Die Zerlegung von f in $f = h \circ g$ sieht dann wie folgt aus:



5 Graphen

Definition 10. Sei V eine endliche Menge und $E \subseteq V \times V$ eine Relation auf V . Dann heißt $G = (V, E)$ ein *Graph* (engl. *graph*). Die Elemente von V heißen *Knoten* (engl. *vertex*) und jene von E heißen *Kanten* (engl. *edge*) von G .

Graphen kann man sich graphisch veranschaulichen, indem man für jeden Knoten einen Punkt in der Ebene wählt und für jede Kante (u, v) einen Pfeil vom Punkt, der u darstellt, zum Punkt, der v darstellt, zeichnet. Natürlich ist eine solche Darstellung nicht eindeutig.

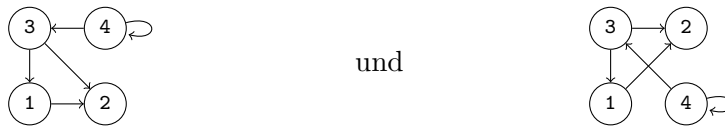
Wenn ein Graph $G = (V, E)$ so ist, dass für jedes $(u, v) \in E$ auch $(v, u) \in E$ ist, dann zeichnet man statt zweier Pfeile von u nach v und von v nach u einfach nur eine Verbindung ohne Pfeilspitzen. Man spricht in diesem Fall von einem *ungerichteten* (engl. *undirected*) Graph.

Beispiel.

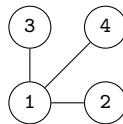
1. Zwei mögliche graphische Darstellungen des Graphen

$$G = (\{1, 2, 3, 4\}, \{(1, 2), (3, 1), (3, 2), (4, 3), (4, 4)\})$$

sind



2. Der Graph $G = (\{1, 2, 3, 4\}, \{(1, 2), (2, 1), (1, 3), (3, 1), (1, 4), (4, 1)\})$ ist ungerichtet und kann z.B. wie folgt visualisiert werden:

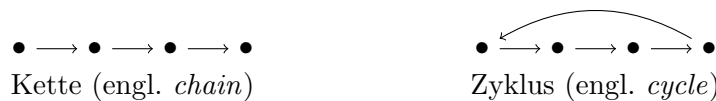


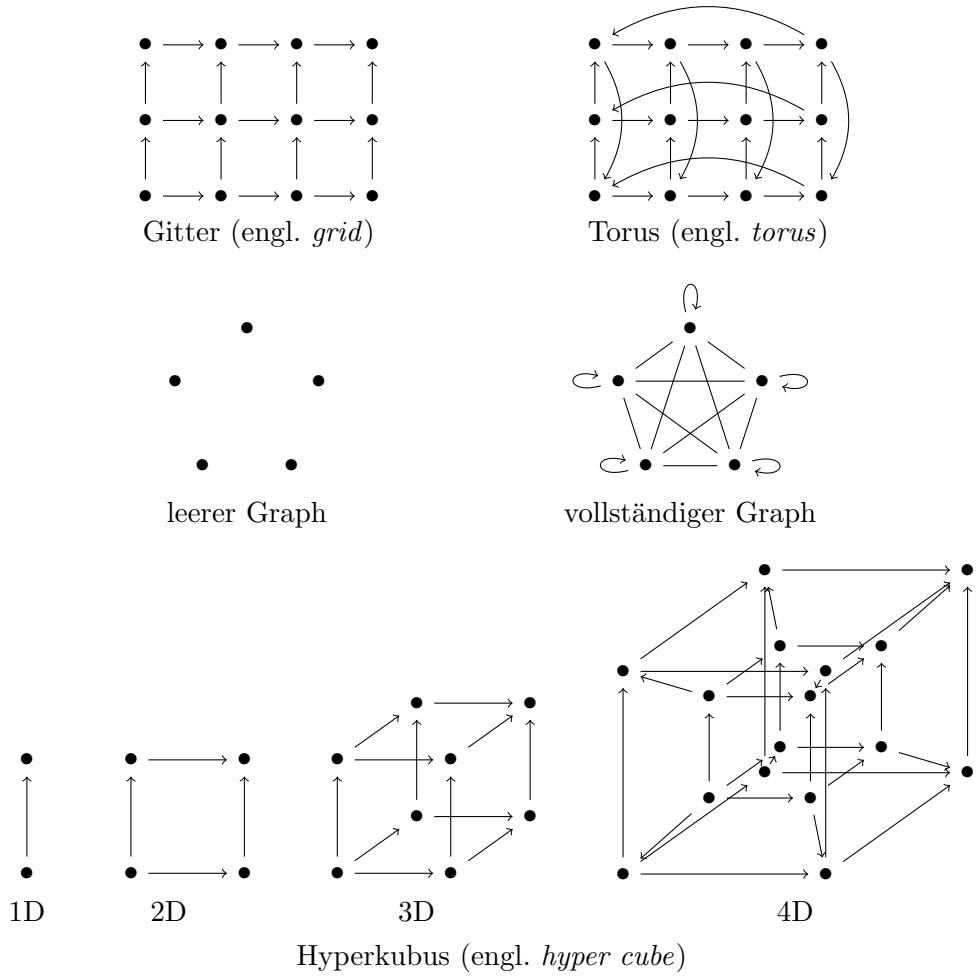
Abgesehen von dem Detail, dass wir V als endliche Menge voraussetzen, bezeichnet der Begriff Graph im wesentlichen dasselbe wie der Begriff Relation. Der neue Begriff ist also streng genommen überflüssig. Wir verwenden ihn hier trotzdem, um dem allgemeinen Sprachgebrauch Rechnung zu tragen. Man spricht typischerweise von einer Relation, wenn man Zusammenhänge zwischen bestimmten mathematischen Objekten (z.B. Rechengesetze für Zahlen) beschreiben will. Dagegen verwendet man den Begriff Graph, wenn nicht die mathematische Natur der Knoten im Vordergrund steht, sondern die wechselseitigen Beziehungen, die durch den Graph ausgedrückt werden.

Graphen spielen in der Informatik eine bedeutende Rolle, weil man mit ihnen Strukturen und Zusammenhänge modellieren kann.

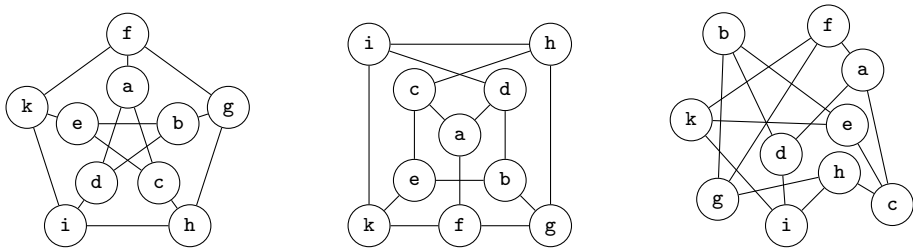
Beispiel.

1. Manche Graphen (oder Graphenfamilien) haben spezielle Namen. Dazu gehören:





2. Die folgenden drei Abbildungen stellen alle den gleichen Graphen dar (den sogenannten Petersen-Graph):



Wie man sieht, kann die Struktur eines Graphen mehr oder weniger deutlich mit dem Auge zu erkennen sein, je nach dem, wo man bei der Visualisierung seine Knoten positioniert.

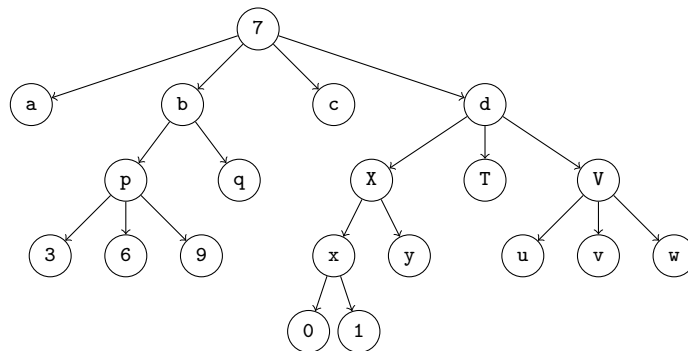
3. Eine besondere Familie von Graphen bilden die *Bäume* (engl. *tree*). Ein Graph ist ein Baum, wenn er sich durch wiederholte Anwendung der folgenden beiden Konstruktionsregeln bilden lässt:

- Jeder Graph mit nur einem Knoten und keinen Kanten ist ein Baum. Der Knoten ist zugleich die *Wurzel* (engl. *root*) des Baums.
- Wenn $T_1 = (V_1, E_1), \dots, T_n = (V_n, E_n)$ Bäume sind mit $V_i \cap V_j = \emptyset$ für $i \neq j$, wenn $t_1 \in V_1, \dots, t_n \in V_n$ die jeweiligen Wurzeln dieser Bäume sind, und wenn v ein Objekt mit $v \notin V_1 \cup \dots \cup V_n$ ist, dann ist auch der Graph

$$T = \left(\underbrace{\{v\} \cup V_1 \cup \dots \cup V_n}_{\text{Knoten}}, \underbrace{\{(v, t_1), \dots, (v, t_n)\} \cup E_1 \cup \dots \cup E_n}_{\text{Kanten}} \right)$$

ein Baum. Seine Wurzel ist v .

Der folgende Baum hat die Wurzel 7, sowie vier *Unterbäume* (engl. *subtree*). Die Wurzeln dieser Unterbäume sind a, b, c, d. Der Unterbaum mit der Wurzel d hat seinerseits wieder drei Unterbäume, deren Wurzeln sind X, T, V. Die Knoten v , für die es eine Kante (u, v) im Baum gibt, nennt man die *Kinder* (engl. *child*) von u . Die Kinder von d sind X, T, V. Die Knoten, die keine Kinder haben, nennt man *Blätter* (engl. *leaf*). Im Beispiel sind dies a, 3, 6, 9, q, c, 0, 1, y, T, u, v, w. Die Knoten, die keine Blätter sind, nennt man die *inneren Knoten* des Baums.

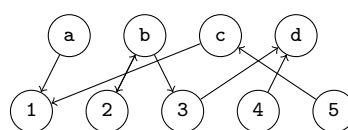


Dateisysteme werden typischerweise als Bäume organisiert. Darin sind Verzeichnisse und Dateien die Knoten, und a ist ein Kind von b wenn a in b liegt. Verzeichnisse die inneren Knoten und Dateien sind die Blätter des Baums.

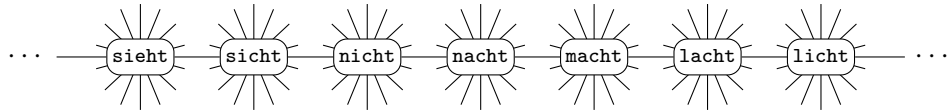
Auch in graphischen Benutzeroberflächen findet man Baumstrukturen, z.B. Menüs mit verschachtelten Untermenüs.

4. Ein Theaterstück hat verschiedene Szenen und verschiedene Figuren. Man kann den Graph betrachten, dessen Knoten die Szenen und Figuren des Stücks sind, und bei dem eine Kante von Figur a zu Szene b geht, wenn a in b auftritt.

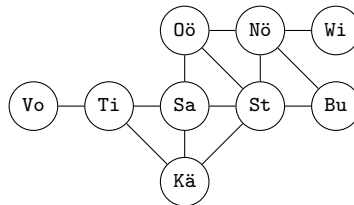
Einen solchen Graphen nennt man *bipartit* (engl. *bipartite*). Die charakteristische Eigenschaft eines bipartiten Graphen $G = (V, E)$ ist, dass sich die Knotenmenge so in $V = V_1 \cup V_2$ mit $V_1 \cap V_2 = \emptyset$ zerlegen lässt, dass jede Kante $(u, v) \in E$ einen Knoten einer der beiden Teilmengen mit einem Knoten aus der anderen verbindet. Es muss also $E \cap (V_1 \times V_1) = E \cap (V_2 \times V_2) = \emptyset$ gelten.



5. Die Menge aller deutschen Wörter mit fünf Buchstaben wird zu einem Graphen, wenn man definiert, dass zwischen zwei Wörtern a und b genau dann eine Kante ist, wenn a und b sich in genau einem Buchstaben unterscheiden. Es ist zweckmäßig, nicht zwischen Groß- und Kleinschreibung zu unterscheiden. Hier ist ein kleiner Ausschnitt dieses Graphen (die Strahlen sollen andeuten, dass die dargestellten Knoten noch mit vielen weiteren nicht dargestellten Knoten verbunden sind):

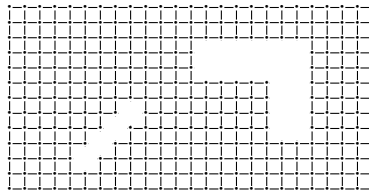


6. Die Menge der österreichischen Bundesländer kann man als Graph auffassen. Eine Kante von a nach b könnte angeben, dass a und b eine gemeinsame Grenze haben.

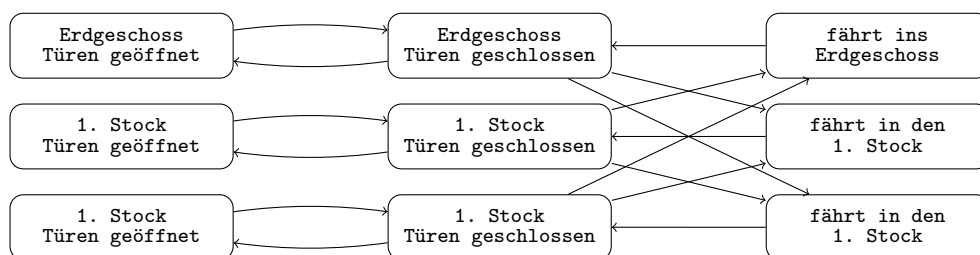


7. Die Struktur chemischer Moleküle kann man durch einen Graphen beschreiben. Dabei entsprechen die Knoten den Atomen und die Bindungen den Kanten.
8. Das WWW ist ein Graph, bei dem Webseiten die Knoten sind, und eine Kante von a nach b existiert, wenn die Seite a einen Link auf die Seite b enthält.
9. Die Menge der Klausurteilnehmer kann man als Graph auffassen. Eine Kante von a nach b könnte angeben, dass a von b abgeschrieben hat.
10. Die Menge aller Menschen kann man auf vielerlei Weise als Graph auffassen. Eine Kante von a nach b könnte z. B. bedeuten, dass a und b sich einmal die Hand gegeben haben (“*handshake graph*”).
11. Die Menge der Lehrveranstaltungen, die an einer Universität angeboten werden, kann man als Graph auffassen. Eine Kante von a nach b könnte angeben, dass der erfolgreiche Abschluss von a eine Voraussetzung für den Besuch von b ist.
12. Ein Kommunikationsnetzwerk ist ein Graph. Dabei sind die beteiligten Geräte (Mobiltelefone, Handymasten, Satelliten, etc.) die Knoten, und es gibt eine Kante von a nach b , wenn a eine direkte Verbindung zu b hat.
13. Das menschliche Skellet lässt sich durch einen Graph beschreiben, bei dem die Gelenke die Knoten darstellen und zwischen zwei Knoten eine Kante besteht, wenn die entsprechenden Gelenke durch einen Knochen verbunden sind.
14. Das Schachspiel ist ein Graph, bei dem die Knoten die möglichen Belegungen des Schachbretts mit Figuren sind, und eine Kante von a nach b existiert, wenn ein zulässiger Schachzug die Stellung a in die Stellung b überführt.

15. Ein Softwaresystem kann man als Graph auffassen. Dabei sind die Funktionen die Knoten und es besteht eine Kante von a nach b , wenn die Definition von a einen Aufruf der Funktion b enthält.
16. Ein Schaltkreis ist in natürlicher Weise ein Graph. Dabei sind die Bauelemente die Knoten und es gibt eine Kante von a nach b , wenn die Bauteile durch eine Leitung verbunden sind.
17. Das Straßennetz ist ein Graph. Hierbei sind die Knoten alle Kreuzungen und Einmündungen, und es besteht eine Kante von a nach b , wenn man von a nach b fahren kann, ohne dabei an einer weiteren Kreuzung oder Einmündung vorbeizukommen.
18. Ein Roboter soll sich auf einem freien Feld von einem Punkt zu einem anderen bewegen und dabei Hindernissen, die sich auf dem Feld befinden, ausweichen. Um ein solches Feld mit Hindernissen zu modellieren, kann man das Feld durch ein feines Gitter modellieren („diskretisieren“), und die Gitterpunkte, an denen sich Hindernisse befinden, aus dem Gitter löschen. Die übrigen Gitterpunkte bilden die Knoten, und je zwei benachbarte Knoten sind durch eine Kante verbunden. Der Roboter kann sich dann entlang der Kanten des Graphen bewegen.



19. Auch Verhalten kann sich unter Umständen durch Graphen beschreiben lassen. Die Knoten bezeichnen dann *Zustände* (engl. *state*), und es gibt eine Kante von a nach b , wenn das System vom Zustand a in den Zustand b wechseln kann. Zum Beispiel zur Beschreibung eines Aufzugs genügt ein solcher Graph:

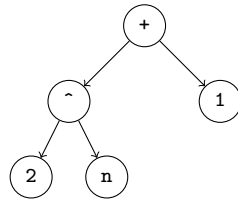


Je nach Anwendung kann es sinnvoll sein, Graphen mit weiteren Informationen anzureichern. Das ist auf mehrere Weise möglich. Wir beschränken uns hier auf zwei Beispiele.

1. *Orientierter Graphen.* Ist $G = (V, E)$ ein Graph, so kann man für jeden Knoten $v \in V$ die Menge $\{u \in V : (v, u) \in E\}$ aller Knoten betrachten, zu denen von v aus eine Kante führt. Auf dieser Menge kann man eine Ordnung definieren, so dass es Sinn macht, z.B. vom „dritten“ dieser Knoten zu sprechen. Die folgenden beiden Bäume sind z.B. identisch als gewöhnliche Graphen, aber verschieden als orientierte Graphen:



Besonders bei Bäumen ist es oft sinnvoll, den Kindern eines Knotens eine Reihenfolge zuzuweisen. Die Reihenfolge der Kinder ist zum Beispiel relevant, wenn man die syntaktische Struktur eines mathematischen Ausdrucks mit einem Baumen darstellen will. Dass zum Beispiel der folgende Baum den Ausdruck $2^n + 1$ darstellen soll, weiss man nur dann, wenn man bei den Kindern des Knotens \sim weiss, welcher das erste und welcher das zweite ist.



Jeder mathematische Ausdruck lässt sich auf diese Weise als Baum darstellen. Man spricht vom *Syntaxbaum* (engl. *syntax tree*) des Ausdrucks.

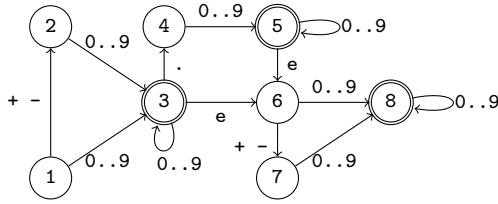
Eine ähnliche Variante sind *Binärbäume* (engl. *binary tree*). Das sind Bäume, bei denen jeder Knoten höchstens zwei Kinder hat, und zwar höchstens einen „linken“ und höchstens einen „rechten“. Die folgenden Binärbäume werden also als verschieden aufgefasst:



Binärbäume werden vor allem dazu verwendet, Daten so zu organisieren, dass man effizient darin suchen kann.

2. *Gewichtete und Gelabelte Graphen.* Bei einem gewichteten Graphen gibt es zusätzlich zu V und E eine Funktion $w: E \rightarrow X$, die jeder Kante $e \in E$ eine Zusatzinformation $w(e) \in X$ zuordnet. In einem Graphen, der ein Straßennetz codiert (mit Kreuzungen und Einmündungen als Knoten, s.o.), könnte eine solche Zusatzinformation zum Beispiel angeben, wie weit die Endpunkte u, v einer Kante $(u, v) \in E$ voneinander entfernt sind, oder wie lange man durchschnittlich braucht, um diese Entfernung mit dem Auto zurückzulegen.

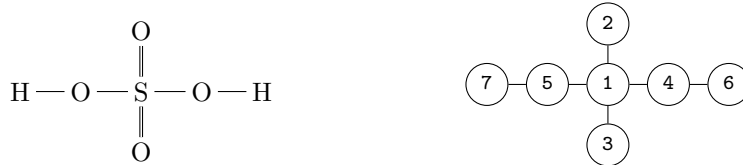
Im Fall $X = \mathbb{N}$ kann man das Gewicht $w(e)$ einer Kante $e = (u, v) \in E$ auch so interpretieren, dass zwischen u und v nicht nur eine, sondern $w(e)$ viele Kanten existieren sollen. Man spricht dann auch von einem *Multigraph*. Die Menge X der Gewichte muss aber nicht unbedingt eine Menge von Zahlen sein. Zum Beispiel verwendet man Teilmengen eines Alphabets Ω als Gewichte (also $X = \mathcal{P}(\Omega)$) bei Graphen, mit denen man definiert, welche Wörter für einen bestimmten Zweck zulässig sein sollen. Der folgende Graph definiert zum Beispiel, welche Wörter über dem Alphabet $\{0, \dots, 9, +, -, e, .\}$ zulässige Gleitkommazahlen sind. Der Graph ist wie folgt zu lesen: Beginnend im Knoten 1 (dem *Startzustand* (engl. *initial state*)) liest man das zu prüfende Wort Buchstabe für Buchstabe und folgt dabei in jedem Schritt einer Kante, die mit einer Buchstabenmenge gewichtet ist, die den aktuellen Buchstaben enthält. Das Wort ist zulässig, wenn es möglich ist, auf diese Weise am Ende des Wortes in einem doppelt umrandeten Knoten (einem *Endzustand* (engl. *final state*)) anzukommen. Zum Beispiel ist $3.5703e-20$ ein gültiges Wort (man besucht dabei nacheinander die Zustände 1, 3, 4, 5, 5, 5, 5, 6, 7, 8, 8). Dagegen ist $3.-e5.+$ kein gültiges Wort.



Die oben beschriebene Anwendung eines Graphen zur Unterscheidung von gültigen und ungültigen Wörtern bezeichnet man als *endlichen Automat* (engl. *finite automaton*), ein Begriff aus der Theorie der formalen Sprachen, über die Sie mehr in Vorlesungen über theoretische Informatik oder Compilerbau erfahren.

Von einem *gelabelten Graph* spricht man, wenn es zusätzlich zur Knotenmenge V und zur Kantenmenge E eine Funktion $q: V \rightarrow Y$ gibt, die jedem Knoten v ein Label $q(v) \in Y$ zuordnet. Natürlich kann ein Graph sowohl gewichtet als auch gelabelt sein. Ein Beispiel ist der endliche Automat von vorher, wenn man mit den Labels markiert, welche Zustände Endzustände sein sollen (im obigen Fall: $q: \{1, \dots, 8\} \rightarrow \{\text{final}, \text{non-final}\}$, $q(v) = \text{final}$ für $v \in \{3, 5, 8\}$ und $q(v) = \text{non-final}$ für $v \in \{1, 2, 4, 6, 7\}$).

Labels braucht man vor allem dann, wenn man verschiedene Knoten gleich bezeichnen will. Ein Beispiel ist die Beschreibung von Molekülen durch Graphen. Um zum Beispiel die Schwefelsäure (unten links) darzustellen, kann man nicht einfach die Knotenmenge $\{S, O, O, O, O, H, H\}$ nehmen, weil diese Menge identisch ist mit der Menge $\{S, O, H\}$, und drei Knoten genügen uns nicht. Stattdessen nimmt man als Knotenmenge $V = \{1, 2, 3, 4, 5, 6, 7\}$ und weist diesen Knoten die entsprechenden chemischen Elemente als Labels zu: $q: V \rightarrow \{S, O, H\}$, $q(1) = S$, $q(2) = q(3) = q(4) = q(5) = O$, $q(6) = q(7) = H$. Darüber hinaus kann man die Mehrfachbindungen des Moleküls durch gewichtete Kanten darstellen: $w: E \rightarrow \mathbb{N}$, $w((1, 2)) = w((2, 1)) = w((1, 3)) = w((3, 1)) = 2$ und $w((u, v)) = 1$ für alle anderen Kanten $(u, v) \in E$.

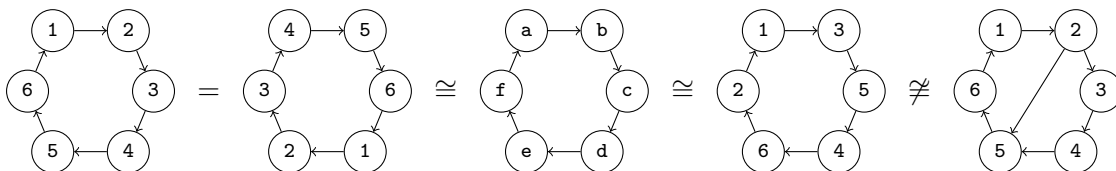


Für die *Struktur* eines Graphen ist es unerheblich, wie seine Knoten bezeichnet sind. Man bezeichnet zwei Graphen $G = (V, E)$ und $\tilde{G} = (\tilde{V}, \tilde{E})$ als *isomorph* (aus dem Griechischen für „von gleicher Gestalt“), wenn es eine bijektive Funktion $h: V \rightarrow \tilde{V}$ gibt, so dass gilt

$$\forall u, v \in V : (u, v) \in E \iff (h(u), h(v)) \in \tilde{E}.$$

In diesem Fall schreibt man $G \cong \tilde{G}$.

Beispiel.



Der zweite und der dritte Graph sind nicht identisch sondern (nur) isomorph. Auch der zweite und der vierte Graph sind (nur) isomorph und nicht identisch.

Allgemeiner:

Definition 11. Seien $G = (V, E)$ und $\tilde{G} = (\tilde{V}, \tilde{E})$ zwei Graphen.

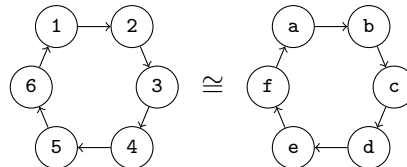
1. Eine (nicht notwendigerweise bijektive) Funktion $h: V \rightarrow \tilde{V}$ heißt (*Graphen-*)*Homomorphismus* (engl. *graph homomorphism*) von G nach \tilde{G} , falls gilt

$$\forall u, v \in V : (u, v) \in E \Rightarrow (h(u), h(v)) \in \tilde{E}.$$

2. G und \tilde{G} heißen (zueinander) *isomorph* (engl. *isomorphic*), falls es einen bijektiven Graphenhomomorphismus $h: V \rightarrow \tilde{V}$ gibt, dessen Umkehrfunktion $h^{-1}: \tilde{V} \rightarrow V$ ebenfalls ein Graphenhomomorphismus ist. Einen solchen Graphenhomomorphismus h nennt man dann *Graphenisomorphismus* für G und \tilde{G} .
3. G heißt *Teilgraph* oder *Untergraph* (engl. *subgraph*) von \tilde{G} , falls es einen injektiven Graphenhomomorphismus von G nach \tilde{G} gibt. Einen solchen Graphenhomomorphismus nennt man auch *Einbettung* (engl. *embedding*) von G in \tilde{G} .

Beispiel.

1. Im Beispiel



ist ein Isomorphismus gegeben durch die Funktion $h: \{1, \dots, 6\} \rightarrow \{a, \dots, f\}$ mit folgender Wertetabelle:

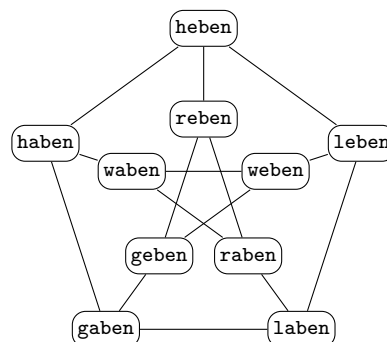
v	1	2	3	4	5	6
$h(v)$	a	b	c	d	e	f

Der Isomorphismus für zwei isomorphe Graphen ist im allgemeinen nicht eindeutig. Im vorliegenden Fall ist zum Beispiel auch die Funktion $\tilde{h}: \{1, \dots, 6\} \rightarrow \{a, \dots, f\}$ mit

v	1	2	3	4	5	6
$\tilde{h}(v)$	c	d	e	f	a	b

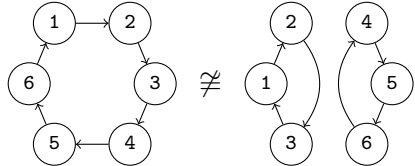
ein Isomorphismus.

2. Sei G der Petersen Graph und \tilde{G} der Graph, dessen Knoten die deutschen Wörter mit exakt fünf Buchstaben (ohne Unterscheidung von Groß- und Kleinschreibung) und der eine Kante von a nach b hat, wenn a und b sich an genau einer Stelle unterscheiden. Dann ist G ein Untergraph von \tilde{G} :



Gemäß unserer Definition macht es nichts aus, wenn es außer den erforderlichen Kanten noch weitere gibt, zum Beispiel die Kante von **gaben** zu **waben**.

3. Anfänger glauben oft, dass zwei Graphen $G = (V, E)$ und $\tilde{G} = (\tilde{V}, \tilde{E})$ dann isomorph sind, wenn sie die gleiche Anzahl von Knoten und Kanten haben, und in beiden Graphen die Zahl der Knoten mit einander entsprechender Anzahl von eingehenden bzw. abgehenden Kanten übereinstimmen. Das ist aber **falsch**. Zum Beispiel gilt sicher



obwohl beide Graphen sechs Knoten und sechs Kanten haben, und in beiden Graphen jeder Knoten genau eine eingehende und genau eine ausgehende Kante hat.

Definition 12. Sei $G = (V, E)$ ein Graph.

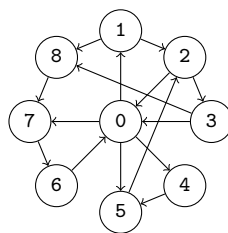
1. Für jedes $n \in \mathbb{N}$ heißt der Graph $P = (\{1, \dots, n\}, \{(1, 2), (2, 3), \dots, (n-1, n)\})$ eine *Kette* der Länge $n-1$.
2. Sei P eine Kette der Länge n und $h: \{1, \dots, n\} \rightarrow V$ ein Graphhomomorphismus von P nach G . Dann heißt $(h(1), h(2), \dots, h(n))$ ein *Pfad* (engl. *path*) in G der Länge n von $h(1)$ nach $h(n)$. Gilt außerdem $h(n) = h(1)$, so spricht man von einem *geschlossenen Pfad* (engl. *closed path*) oder *Zyklus* (engl. *cycle*).
3. Ein Knoten $q \in V$ heißt *Quelle* (engl. *source*) von G , falls für jedes $v \in V$ ein Pfad von q nach v existiert.
4. Ein Knoten $s \in V$ heißt *Senke* (engl. *sink*) von G , falls für jedes $v \in V$ ein Pfad von v nach s existiert.

Sei G nun ein ungerichteter Graph (d.h. es gelte $\forall (u, v) \in E : (v, u) \in E$).

4. G heißt *zusammenhängend* (engl. *connected*), falls für je zwei Knoten $v, w \in V$ ein Pfad von v nach w existiert.
5. Sei $V_0 \subseteq V$ nicht leer und $E_0 = \{(u, v) \in E : u, v \in V_0\} \subseteq E$. Der Graph $G_0 := (V_0, E_0)$ heißt *Zusammenhangskomponente* (engl. *connected component*) von G , falls G_0 zusammenhängend ist und für alle $v \in V \setminus V_0$ und alle $w \in V_0$ gilt $(v, w) \notin E$.

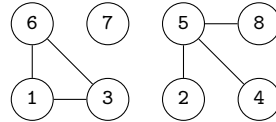
Beispiel.

1. Im Graphen



ist jeder Knoten sowohl Quelle als auch Senke. Ein Pfad von 1 nach 5 ist zum Beispiel $(1, 2, 0, 7, 6, 0, 4, 5)$. Beachten Sie, dass ein Pfad einen Knoten auch mehrfach besuchen darf, weil die Definition nicht verlangt, dass der Homomorphismus injektiv ist. Bei $(1, 2, 0, 6, 0, 4, 5)$ handelt es sich **nicht** um einen Pfad des Graphen, weil der Graph keine Kante von 0 nach 6 hat.

2. Der ungerichtete Graph



hat drei Zusammenhangskomponenten, nämlich

$$\begin{aligned} & \{\{1, 3, 6\}, \{(1, 3), (3, 1), (3, 6), (6, 3), (1, 6), (6, 1)\}\}, \\ & \{\{2, 4, 5, 8\}, \{(2, 5), (5, 2), (4, 5), (5, 4), (5, 8), (8, 5)\}\} \end{aligned}$$

und $(\{7\}, \emptyset)$.

Satz 5. Sei $G = (V, E)$ ein ungerichteter Graph. Dann gilt: E ist genau dann eine Äquivalenzrelation auf V , wenn alle Zusammenhangskomponenten von G vollständige Graphen sind (d.h. zwischen je zwei Knoten derselben Zusammenhangskomponente gibt es eine Kante).

Beweis. „ \Rightarrow “ E ist eine Äquivalenzrelation. Zu zeigen: die Zusammenhangskomponenten von G sind vollständig.

Sei (V_0, E_0) eine beliebige Zusammenhangskomponente. Dann ist also zu zeigen: $E_0 = V_0 \times V_0$, d.h. $\forall u, v \in V_0 : (u, v) \in E_0$.

Betrachte dazu zwei beliebige Knoten $u, v \in V_0$. Da (V_0, E_0) als Zusammenhangskomponente insbesondere zusammenhängend ist, gibt es einen Pfad von u nach v , etwa

$$(x_1, x_2, \dots, x_{n-1}, x_n)$$

mit $x_1 = u$ und $x_n = v$. Nach Voraussetzung ist E eine Äquivalenzrelation, darum ist E insbesondere transitiv. Aus $(x_i, x_{i+1}), (x_{i+1}, x_{i+2}) \in E$ folgt deshalb $(x_i, x_{i+2}) \in E$ ($i = 1, \dots, n-2$ beliebig). Mit der zweiten Bedingung aus der Definition der Zusammenhangskomponente folgt dann auch $(x_i, x_{i+2}) \in E_0$. Man kann also die beiden Kanten (x_i, x_{i+1}) und (x_{i+1}, x_{i+2}) des Pfades durch die einzelne Kante (x_i, x_{i+2}) ersetzen und erhält so einen kürzeren Pfad von u nach v . Durch wiederholte Anwendung des Arguments erhält man nach endlich vielen Schritten einen Pfad bestehend aus nur einer Kante: (u, v) . Damit ist gezeigt: für beliebige $u, v \in V_0$ gilt $(u, v) \in E_0$, wie gefordert.

„ \Leftarrow “ Die Zusammenhangskomponenten von G sind vollständig. Zu zeigen: E ist eine Äquivalenzrelation auf V .

- Symmetrie ist klar, weil G nach Voraussetzung ein ungerichteter Graph ist.
- Transitivität: Wenn $a, b, c \in V$ so sind, dass $(a, b), (b, c) \in E$ sind, dann gibt es einen Pfad von a nach c , d.h. a und c liegen in derselben Zusammenhangskomponente. Nach

Voraussetzung ist jede Zusammenhangskomponente vollständig. Die Zusammenhangskomponente, zu der a und c gehören, muss also die Kante (a, c) enthalten. Diese Kante muss dann auch schon zu E gehören. Damit ist gezeigt: $\forall a, b, c \in V : (a, b) \in E \wedge (b, c) \in E \Rightarrow (a, c) \in E$, wie gefordert.

- c) Reflexivität: Jeder Knoten $v \in V$ gehört zu einer bestimmten Zusammenhangskomponente, etwa zu (V_0, E_0) . Aus der Vollständigkeit dieser Zusammenhangskomponente folgt $(v, v) \in E_0 \subseteq E$, wie gefordert. ■

Im Fall des obigen Satzes entsprechen die Zusammenhangskomponenten des Graphen den Äquivalenzklassen.

Zur Darstellung eines Graphen $G = (V, E)$ in einem Computer gibt es mehrere Möglichkeiten.

1. Man kann wie in der Definition die Knotenmenge V und die Kantenmenge E mit Hilfe geeigneter Datenstrukturen für endliche Mengen abspeichern.

Zum Beispiel $G = (\{1, 2, 3, 4\}, \{(1, 2), (3, 1), (3, 2), (4, 1)\})$.

Diese Repräsentation ist zum Programmieren meistens nicht die beste Wahl. Sie eignet sich eher für theoretische Überlegungen.

2. Man kann auch die Knotenmenge in einer geordneten Liste abspeichern und die Kantenmenge als eine quadratische Tabelle mit $|V|$ Zeilen und $|V|$ Spalten, bei der in Zeile (i, j) eine 1 steht, wenn es eine Kante vom i ten Knoten zum j ten Knoten gibt, und sonst 0.

Zum Beispiel $[1, 2, 3, 4]$ und $\begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$.

Diese Repräsentation bietet sich vor allem bei sehr kleinen Graphen an.

3. Stattdessen kann man auch eine Liste von Paaren (v, S) speichern, wobei v ein Knoten ist und $S \subseteq V$ die Menge aller Knoten $w \in V$, so dass $(v, w) \in E$ ist. Die Liste muss für jeden Knoten $v \in V$ ein solches Paar enthalten.

Zum Beispiel $[(1, \{2\}), (2, \{\}), (3, \{1, 2\}), (4, \{1\})]$.

Diese Repräsentation hat den Vorteil, dass man den Graphen relativ einfach modifizieren kann (z.B. zusätzliche Knoten oder Kanten hinzufügen).

4. Bei sehr großen Graphen, die nicht auf einmal in den Speicher passen, kann unter Umständen auch eine implizite Repräsentation sinnvoll sein. Dabei programmiert man z.B. eine Funktion `is_vertex(-)`, die für jedes Objekt angibt, ob es zur Knotenmenge gehört oder nicht, sowie eine Funktion `is_edge(-, -)`, die für je zwei gegebene Objekte angibt, ob es sich um Knoten handelt, die durch eine Kante des Graphen verbunden sind. Der Graph ist dabei gewissermaßen im Programm codiert.

Für den folgenden Algorithmus, mit dem man herausfinden kann, ob zwei gegebene Knoten eines Graphen durch einen Pfad verbunden sind, verwendet man am besten die zweite oder dritte Datenstruktur.

INPUT: $G = (V, E)$, $u, v \in V$

OUTPUT: Ein Pfad von u nach v , falls es einen gibt, oder False, falls nicht.

- 1 falls $u = v$, gib den leeren Pfad als Ergebnis zurück und stop.
- 2 markiere den Knoten u als „besucht“.
- 3 für alle Knoten $w \in V$:
- 4 falls $(u, w) \in E$ und w ist noch nicht markiert:

- 5 suche einen Pfad von w nach v , indem dieser Algorithmus rekursiv aufgerufen wird.
- 6 falls ein Pfad (w, \dots, v) gefunden wurde:
- 7 gib (u, w, \dots, v) als Ergebnis zurück und stop.
- 8 gib False zurück.

Beachten Sie, dass dieser Algorithmus in Schritt 6 sich selbst aufruft. Sie sollten sich zur Übung überlegen, (a) warum der Algorithmus nur korrekte Ergebnisse liefert, und (b) warum der Algorithmus für jede mögliche Eingabe terminiert. Können Sie den jeweiligen Grund als formalen Beweis formulieren?

Es gibt clevere Varianten dieses Algorithmus, die mit nur geringem Mehraufwand den kürzesten Pfad in einem gewichteten Graph finden. Dabei sind die Gewichte positive reelle Zahlen und die Länge eines Pfades ist die Summe der Gewichte der in ihm enthaltenen Kanten. Mit etwas mehr Aufwand kann man sogar die kürzeste Verbindung für alle Knotenpaare auf einmal ausrechnen. Es gibt viele weitere interessante Algorithmen, mit denen man Fragestellungen über Graphen effizient lösen kann. Allerdings ist z.B. kein effizienter Algorithmus bekannt, mit dem man herausfinden könnte, ob ein Graph einen sogenannten hamiltonschen Pfad hat. Das ist ein Pfad, der jeden Knoten des Graphen genau einmal besucht. Es wird vermutet, dass es zur Lösung dieses Problems keinen effizienten Algorithmus gibt (wobei wir an dieser Stelle nicht erklären können, was genau in diesem Zusammenhang mit einem „effizienten Algorithmus“ gemeint ist). Auch zur Entscheidung der Graphisomorphie gibt es nach derzeitigem Kenntnisstand keine effizienten Algorithmen.

6 Gruppen

Definition 13.

1. Sei $X = \{x_1, \dots, x_n\}$ eine Menge mit n Elementen. Eine bijektive Funktion $f: X \rightarrow X$ heißt *Permutation* (engl. *permutation*) der Menge X . Man verwendet die Notation

$$f = \begin{pmatrix} x_1 & x_2 & \cdots & x_n \\ f(x_1) & f(x_2) & \cdots & f(x_n) \end{pmatrix}.$$

Die Menge aller Permutationen der Menge X wird mit S_X bezeichnet. Im Fall $X = \{1, \dots, n\}$ schreibt man auch einfach S_n statt $S_{\{1, \dots, n\}}$.

2. Eine Permutation $\pi \in S_n$ heißt *Zyklus* (engl. *cycle*), falls es paarweise verschiedene $k_1, \dots, k_m \in \{1, \dots, n\}$ gibt, so dass

$$\pi(k_1) = k_2, \quad \pi(k_2) = k_3, \quad \dots, \quad \pi(k_{m-1}) = k_m, \quad \pi(k_m) = k_1$$

sowie $\pi(k) = k$ für alle $k \in \{1, \dots, n\} \setminus \{k_1, \dots, k_m\}$ gilt.

Schreibweise: $\pi = (k_1 k_2 \dots k_m)$. Man nennt m die *Länge* (engl. *length*) des Zyklus.

Ein Zyklus der Länge zwei heißt *Transposition* (engl. *transposition*).

3. Zwei Permutationen π_1, π_2 heißen (zueinander) *disjunkt* (engl. *disjoint*) falls gilt

$$\forall k \in \{1, \dots, n\} : \pi_1(k) = k \vee \pi_2(k) = k.$$

4. Ein $k \in \{1, \dots, n\}$ mit $\pi(k) = k$ heißt *Fixpunkt* (engl. *fixed point*) von $\pi \in S_n$.

Beispiel.

1. $S_3 = \left\{ \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix} \right\}.$

2. Nach Satz 2 ist die Umkehrfunktion einer Permutation wieder eine Permutation.

Für $\pi = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 1 & 4 & 2 & 5 \end{pmatrix} \in S_5$ gilt zum Beispiel $\pi^{-1} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 4 & 1 & 3 & 5 \end{pmatrix} \in S_5.$

3. Nach Satz 1 ist auch die Verkettung zweier Permutationen wieder eine Permutation. Beachten Sie aber, dass es bei der Verkettung im allgemeinen auf die Reihenfolge ankommt:

$$\begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix} \circ \begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix}$$

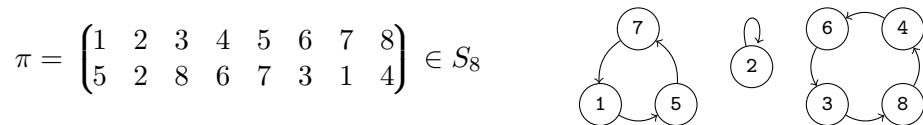
$$\begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{pmatrix} \circ \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{pmatrix}$$

4. Wenn (x_1, \dots, x_n) ein Tupel ist und $\pi \in S_n$ eine Permutation, dann hat das Tupel $(x_{\pi(1)}, \dots, x_{\pi(n)})$ genau die selben Komponenten, allerdings im Allgemeinen in einer anderen Reihenfolge.

5. $\begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 1 & 3 \end{pmatrix} = (1\ 2\ 4\ 3) = (3\ 1\ 2\ 4)$ ist ein Zyklus. Beachten Sie: Ein Zyklus lässt sich auf verschiedene Weise notieren.

6. $\begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix} = (1\ 3)$ ist eine Transposition.

7. Eine Permutation $\pi \in S_n$ lässt sich als Graph mit Knotenmenge $V = \{1, \dots, n\}$ veranschaulichen, bei der eine Kante (u, v) anzeigt, dass $\pi(u) = v$ gilt. Beispiel:



Diese Permutation hat den Fixpunkt 2.

Ein Graph mit Knotenmenge $\{1, \dots, n\}$ stellt genau dann in diesem Sinn eine Permutation dar, wenn jeder Knoten genau eine ausgehende Kante (wg. Funktionseigenschaft) und genau eine eingehende Kante (wg. Bijektivität: Surjektivität = mindestens eine; Injektivität = höchstens eine) hat.

Eine Permutation ist genau dann ein Zyklus im Sinn von Def. 13, wenn der zugehörige Graph als Graph ein Zyklus ist.

Wie das obige Beispiel 7 schon suggeriert, lässt sich jede Permutation in eindeutiger Weise in paarweise disjunkte Zyklen zerlegen. Für die Permutation in diesem Beispiel gilt nämlich $\pi = (1\ 5\ 7)(2)(3\ 4\ 8\ 6)$. In diesem Fall ist die Reihenfolge der Zyklen unbedeutend, denn wenn σ_1, σ_2 zwei disjunkte Zyklen sind, dann gilt $\sigma_1\sigma_2 = \sigma_2\sigma_1$. Fixpunkte sind Zyklen der Länge eins. Per Konvention braucht man diese bei der Zerlegung einer Permutation in Zyklen nicht zu notieren. Man kann also auch einfach $\pi = (1\ 5\ 7)(3\ 4\ 8\ 6)$ für die Permutation aus Beispiel 7 schreiben.

Um eine gegebene Permutation in disjunkte Zyklen zu zerlegen, geht man nach folgendem Algorithmus vor:

INPUT: $\pi \in S_n$

OUTPUT: Die Zerlegung von π in disjunkte Zyklen.

- 1 setze $B = \{1, \dots, n\}$
- 2 solange $B \neq \emptyset$:
- 3 wähle ein beliebiges $k \in B$
- 4 falls $\pi(k) = k$, dann
- 5 setze $B = B \setminus \{k\}$
- 6 anderenfalls
- 7 bestimme das kleinste $m \in \{1, \dots, n\}$ mit $\pi^m(k) = k$
- 8 notiere den Zyklus $(k \ \pi(k) \ \pi^2(k) \ \dots \ \pi^{m-1}(k))$
- 9 setze $B = B \setminus \{k, \pi(k), \dots, \pi^{m-1}(k)\}$.

Mit der Notation $\pi^i(k)$ ist die i -fache Anwendung von π auf k gemeint, also z.B. $\pi^3(7) = \pi(\pi(\pi(7)))$. Wir überlassen es wieder als Übung, sich davon zu überzeugen, dass der Algorithmus terminiert und die Ausgabe korrekt ist. (Überlegen Sie sich insbesondere, warum es in Schritt 7 immer ein $m \in \{1, \dots, n\}$ mit $\pi^m(k) = k$ geben muss.)

Wegen Satz 1 ist die Verkettung zweier Permutationen wieder eine Permutation, d.h. es gilt $\forall \pi, \sigma \in S_n : \pi \circ \sigma \in S_n$. Wir können die Verkettung also als eine Funktion $\circ : S_n \times S_n \rightarrow S_n$ auffassen. Allgemein nennt man eine Funktion $\circ : A \times A \rightarrow A$ eine *Verknüpfung* (engl. *operation*) auf der Menge A . Statt $\circ(x, y)$ schreibt man üblicherweise $x \circ y$. Solche Verknüpfungen können interessante Eigenschaften haben. Für die Verknüpfung von Permutationen gelten zum Beispiel folgende Regeln:

- $\forall \pi, \sigma, \tau \in S_n : (\pi \circ \sigma) \circ \tau = \pi \circ (\sigma \circ \tau)$
- $\forall \pi \in S_n : \pi \circ \text{id} = \text{id} \circ \pi = \pi$
- $\forall \pi \in S_n \exists \sigma \in S_n : \pi \circ \sigma = \sigma \circ \pi = \text{id}$

Ähnliche Regeln gelten für die bekannten Rechenoperationen $+$ (z.B. für $A = \mathbb{Z}$ oder $A = \mathbb{R}$) und \cdot (z.B. für $A = \mathbb{Q} \setminus \{0\}$ oder $A = \mathbb{R} \setminus \{0\}$). Das motiviert folgende Abstraktion:

Definition 14. Sei A eine Menge und $\circ : A \times A \rightarrow A$ eine Verknüpfung auf A .

1. \circ heißt *assoziativ* (engl. *associative*), falls gilt: $\forall x, y, z \in A : (x \circ y) \circ z = x \circ (y \circ z)$
2. \circ heißt *kommutativ* (engl. *commutative*), falls gilt: $\forall x, y \in A : x \circ y = y \circ x$
3. $e \in A$ heißt *Neutralelement* (engl. *neutral element*) (bezüglich \circ), falls gilt: $\forall x \in A : x \circ e = e \circ x = x$.
4. Ist $e \in A$ ein Neutralelement, so heißt $x \in A$ *invertierbar* (engl. *invertible*), falls gilt:

$$\exists y \in A : x \circ y = y \circ x = e.$$

Ein solches Element y heißt dann ein *Inverses* (engl. *inverse*) von x .

5. Das Paar (A, \circ) heißt *Gruppe* (engl. *group*), falls \circ assoziativ ist, A ein Neutralelement bezüglich \circ enthält, und es zu jedem Element x von A ein passendes Inverses in A gibt. Wenn die Verknüpfung aus dem Kontext klar ist, sagt man auch einfach „ A ist eine Gruppe“.
6. Ist (A, \circ) eine Gruppe und ist \circ auch kommutativ, so nennt man (A, \circ) eine *abelsche Gruppe* (engl. *abelian*).

Satz 6. Sei $\circ: A \times A \rightarrow A$ eine assoziative Verknüpfung.

1. Sind e_1, e_2 Neutralelemente von \circ , so gilt $e_1 = e_2$.
2. Ist e ein Neutralelement von \circ , $x \in A$ invertierbar, und sind y_1, y_2 Inverse von x , so gilt $y_1 = y_2$.
Notation: $x^{-1} := y_1 = y_2$.
3. Ist $x \in A$ invertierbar, so ist auch x^{-1} invertierbar und es gilt $(x^{-1})^{-1} = x$.
4. Sind $x, y \in A$ invertierbar, so ist auch $x \circ y$ invertierbar und es gilt $(x \circ y)^{-1} = y^{-1} \circ x^{-1}$.

Beweis.

1. Nach Definition gilt $\forall x \in A : x \circ e_1 = e_1 \circ x = x$ und $\forall x \in A : x \circ e_2 = e_2 \circ x = x$.
Aus dem ersten folgt mit $x = e_2$, dass $e_2 \circ e_1 = e_1 \circ e_2 = e_2$ ist, und aus dem zweiten folgt mit $x = e_1$, dass $e_1 \circ e_2 = e_2 \circ e_1 = e_1$ ist.
Aus beidem zusammen folgt $e_1 = e_1 \circ e_2 = e_2$, wie behauptet.
2. Es gilt $x \circ y_1 = e$, also $y_2 \circ (x \circ y_1) = y_2 \circ e$, also $(y_2 \circ x) \circ y_1 = y_2$, also $e \circ y_1 = y_2$, also $y_1 = y_2$.
- 3., 4. Übung.

Beispiel.

1. $(\mathbb{Z}, +)$, $(\mathbb{Q}, +)$, $(\mathbb{R}, +)$, $(\mathbb{Q} \setminus \{0\}, \cdot)$, $(\mathbb{R} \setminus \{0\}, \cdot)$ sind abelsche Gruppen. Das Neutralelement bezüglich $+$ ist jeweils 0, und das Neutralelement bezüglich \cdot ist jeweils 1.
2. (S_n, \circ) ist eine Gruppe, aber für $n \geq 3$ keine abelsche Gruppe. Das Neutralelement ist die Identitätsabbildung.
3. Bei endlichen Gruppen lässt sich die Verknüpfung als *Verknüpfungstabelle* aufschreiben. Zum Beispiel wird die Menge $G = \{1, 2, 3, 4, 5, 6\}$ mit der Verknüpfung $*$: $G \times G \rightarrow G$, die durch folgende Tabelle definiert ist, zu einer abelschen Gruppe.

$*$	1	2	3	4	5	6
1	1	2	3	4	5	6
2	2	4	6	1	3	5
3	3	6	2	5	1	4
4	4	1	5	2	6	3
5	5	3	1	6	4	2
6	6	5	4	3	2	1

4. Ist A eine Menge, so sind \cup und \cap Verknüpfungen auf $\mathcal{P}(A)$, aber weder $(\mathcal{P}(A), \cup)$ noch $(\mathcal{P}(A), \cap)$ ist eine Gruppe. Definiert man auf $\mathcal{P}(A)$ die Verknüpfung

$$\oplus: \mathcal{P}(A) \times \mathcal{P}(A) \rightarrow \mathcal{P}(A), \quad U \oplus V := (U \cup V) \setminus (U \cap V),$$

so ist $(\mathcal{P}(A), \oplus)$ eine abelsche Gruppe. Das Neutralelement ist dann die leere Menge \emptyset und jedes Element $U \in \mathcal{P}(A)$ ist zu sich selbst invers.

5. Betrachte die sechs Funktionen $f_i: \mathbb{R} \setminus \{0, 1\} \rightarrow \mathbb{R} \setminus \{0, 1\}$ definiert durch

$$\begin{aligned} f_1(x) &= x, & f_2(x) &= \frac{1}{1-x}, & f_3(x) &= \frac{1}{x}, \\ f_4(x) &= \frac{x-1}{x}, & f_5(x) &= 1-x, & f_6(x) &= \frac{x}{x-1}. \end{aligned}$$

Die Menge $G = \{f_1, \dots, f_6\}$ bildet zusammen mit der Komposition eine Gruppe.

6. Sei $G = \{x \in \mathbb{R} : -1 < x < 1\}$ und definiere

$$x \oplus y := \frac{x+y}{1+xy},$$

wobei die Symbole auf der rechten Seite die übliche Bedeutung haben. Dann ist (G, \oplus) eine abelsche Gruppe. Man sieht sofort, dass \oplus kommutativ ist, dass 0 ein neutrales Element ist, und dass $-x$ das Inverse von x ist. Assoziativität lässt sich leicht nachrechnen:

$$(x \oplus y) \oplus z = \frac{\frac{x+y}{1+xy} + z}{1 + \frac{x+y}{1+xy}z} = \frac{x+y+z+xyz}{1+xy+xz+yz} = \frac{x + \frac{y+z}{1+yz}}{1 + x\frac{y+z}{1+yz}} = x \oplus (y \oplus z).$$

Etwas weniger offensichtlich ist in diesem Beispiel, dass \oplus tatsächlich eine Verknüpfung ist, dass also für jede beliebige Wahl von $x, y \in G$ auch $x \oplus y$ in G liegt. Es lässt sich zeigen, dass dem so ist.

Die Gruppe (G, \oplus) erklärt die Addition von Geschwindigkeiten in der speziellen Relativitätstheorie.

7. Die Konkatenation \circ ist eine Verknüpfung auf der Menge Ω^* aller Wörter über einem Alphabet Ω . Diese Verknüpfung ist assoziativ und hat ein Neutralelement (nämlich das leere Wort), aber es handelt sich bei (Ω^*, \circ) nicht um eine Gruppe, weil es nicht zu jedem Wort ein Inverses gibt.

Um auf Wörtern eine Gruppe zu definieren, kann man folgendes tun. Betrachte zwei Alphabete $\Omega, \bar{\Omega}$ mit $\Omega \cap \bar{\Omega} = \emptyset$ und $|\Omega| = |\bar{\Omega}|$ sowie eine bijektive Funktion $i: \Omega \rightarrow \bar{\Omega}$. Wir betrachten Wörter über dem Alphabet $\Omega \cup \bar{\Omega}$ und wollen die Buchstaben in $\bar{\Omega}$ als Inverse der Buchstaben in Ω auffassen, und umgekehrt. Das Inverse von $x \in \Omega$ soll $i(x) \in \bar{\Omega}$ sein, und das von $y \in \bar{\Omega}$ sei $i^{-1}(y) \in \Omega$.

Es sei nun $W \subseteq (\Omega \cup \bar{\Omega})^*$ die Menge aller Wörter, die keine Teilwörter der Form $xi(x)$ oder $i(x)x$ mit $x \in \Omega$ enthalten. Die Verknüpfung $\circ: W \times W \rightarrow W$ sei so definiert, dass $\omega \circ \chi$ das Wort in W sei, das entsteht, wenn man aus der Konkatenation $\omega\chi$ so lange alle Teilwörter der Form $xi(x)$ und $i(x)x$ löscht, bis keine mehr übrig sind. Dann ist (W, \circ) eine Gruppe.

Beispiel: $\Omega = \{a, b, c\}$, $\bar{\Omega} = \{A, B, C\}$, $i(a) = A$, $i(b) = B$, $i(c) = C$. Dann gilt zum Beispiel

$$acBcACaaB \circ bAaCbbc = acBcACaaCbBc \quad \text{und} \quad bAaCbbc^{-1} = CBBcAaB.$$

8. Seien $(A, \circ), (B, *)$ zwei Gruppen und $G = A \times B$. Auf G wird durch

$$(a_1, b_1) \odot (a_2, b_2) := (a_1 \circ a_2, b_1 * b_2)$$

eine Verknüpfung $\odot: G \times G \rightarrow G$ definiert, mit der G zu einer Gruppe wird.

Definition 15. Sei (G, \circ) eine Gruppe.

1. $U \subseteq G$ heißt eine *Untergruppe* (engl. *subgroup*), falls gilt $U \neq \emptyset$ und $\forall u, v \in U : u \circ v \in U \wedge u^{-1} \in U$.
2. Sei $U \subseteq G$ eine Untergruppe und seien $u_1, \dots, u_m \in U$. Man sagt, U wird von u_1, \dots, u_m *erzeugt* (engl. *generated*), wenn es für jedes $u \in U$ Zahlen $i_1, \dots, i_k \in \{1, \dots, m\}$ und $e_1, \dots, e_k \in \{-1, 1\}$ gibt, so dass $u = u_{i_1}^{e_1} \circ u_{i_2}^{e_2} \circ \dots \circ u_{i_k}^{e_k}$. In diesem Fall schreibt man $U = \langle u_1, \dots, u_m \rangle$.

Beispiel.

1. $(\mathbb{Z}, +)$ ist eine Untergruppe von $(\mathbb{Q}, +)$; $(\mathbb{Q}, +)$ ist eine Untergruppe von $(\mathbb{R}, +)$.
2. $(\{x \in \mathbb{Q} : x > 0\}, \cdot)$ ist eine Untergruppe von $(\mathbb{Q} \setminus \{0\}, \cdot)$; $(\mathbb{Q} \setminus \{0\}, \cdot)$ und $(\{x \in \mathbb{R} : x > 0\}, \cdot)$ sind Untergruppen von $(\mathbb{R} \setminus \{0\}, \cdot)$.
3. S_3 lässt sich auffassen als Untergruppe von S_5 , wenn man sich jede Funktion

$$f: \{1, 2, 3\} \rightarrow \{1, 2, 3\}$$

aus S_3 zu einer Funktion $f: \{1, 2, 3, 4, 5\} \rightarrow \{1, 2, 3, 4, 5\}$ mit $f(4) = 4$ und $f(5) = 5$ fortgesetzt denkt.

4. S_6 ist eine Gruppe mit 720 Elementen. Die von $g_1 = (1\ 2)(3\ 4)(5\ 6)$, $g_2 = (1\ 5\ 3)(2\ 4\ 6)$, $g_3 = (1\ 2\ 3\ 4\ 5\ 6)$ erzeugte Untergruppe von S_6 besteht aus den folgenden 18 Elementen:

$$\begin{array}{ll} \text{id} = (1)(2)(3)(4)(5)(6) & g_1 = (1\ 2)(3\ 4)(5\ 6) \\ g_2 = (1\ 5\ 3)(2\ 4\ 6) & g_3 = (1\ 2\ 3\ 4\ 5\ 6) \\ g_1 \circ g_2 = (1\ 4)(2\ 5)(3\ 6) & g_1 \circ g_3 = (1\ 3\ 5)(2)(4)(6) \\ g_2 \circ g_1 = (1\ 6)(2\ 3)(4\ 5) & g_2 \circ g_2 = (1\ 3\ 5)(2\ 6\ 4) \\ g_2 \circ g_3 = (1\ 6\ 3\ 2\ 5\ 4) & g_3 \circ g_1 = (2\ 4\ 6)(1)(3)(5) \\ g_3 \circ g_2 = (1\ 4\ 3\ 6\ 5\ 2) & g_3 \circ g_3 = (1\ 3\ 5)(2\ 4\ 6) \\ g_1 \circ g_2 \circ g_3 = (1\ 5\ 3)(2\ 6\ 4) & g_1 \circ g_3 \circ g_3 = (1\ 4\ 5\ 2\ 3\ 6) \\ g_2 \circ g_3 \circ g_3 = (2\ 6\ 4)(1)(3)(5) & g_3 \circ g_1 \circ g_3 = (1\ 2\ 5\ 6\ 3\ 4) \\ g_3 \circ g_2 \circ g_3 = (1\ 5\ 3)(2)(4)(6) & g_1 \circ g_2 \circ g_3 \circ g_3 = (1\ 6\ 5\ 4\ 3\ 2) \end{array}$$

Weitere Elemente von S_6 enthält die Untergruppe $\langle g_1, g_2, g_3 \rangle$ nicht, denn wenn Sie irgend eines der obigen Elemente invertieren oder mit einem anderen verketteten, erhalten Sie immer ein Element, das schon in der Liste steht. Zum Beispiel gilt $(g_3 \circ g_1 \circ g_3)^{-1} = g_3 \circ g_2$.

5. Sei (W, \circ) die Gruppe aus dem vorherigen Beispiel, die aus Wörtern über dem Alphabet $\{\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{A}, \mathbf{B}, \mathbf{C}\}$ gebildet wird. Dann gilt $W = \langle \mathbf{a}, \mathbf{b}, \mathbf{c} \rangle$.

Definition 16. Seien (G_1, \circ) und $(G_2, *)$ Gruppen. Eine Funktion $h: G_1 \rightarrow G_2$ heißt *Homomorphismus*, falls gilt:

$$\forall x, y \in G_1 : h(x \circ y) = h(x) * h(y).$$

Sei e_2 das Neutralelement von G_2 . Die Menge

$$\ker h := \{x \in G_1 : h(x) = e_2\} \subseteq G_1$$

heißt der *Kern* (engl. *kernel*) und

$$\operatorname{im} h := \{h(x) : x \in G_1\} \subseteq G_2$$

heißt das *Bild* (engl. *image*) von h .

Ein bijektiver Homomorphismus heißt *Isomorphismus*. (engl. *isomorphism*) Wenn es einen Isomorphismus von G_1 nach G_2 gibt, sagt man, G_1 und G_2 sind (zueinander) *isomorph*. (engl. *isomorphic*) Notation in diesem Fall: $G_1 \cong G_2$.

Beispiel.

1. Die Abbildung $h: S_3 \rightarrow S_5$, die im vorigen Beispiel beschrieben wurde, ist ein Homomorphismus. Statt zu sagen, S_3 ist eine Untergruppe von S_5 , wäre es sauberer zu sagen $h(S_3)$ ist eine Untergruppe von S_5 .
2. Die Abbildung $f: \mathbb{R} \rightarrow \mathbb{R}$, $x \mapsto \exp(x)$ ist ein Homomorphismus zwischen $(\mathbb{R}, +)$ und $(\mathbb{R} \setminus \{0\}, \cdot)$.
3. Sind $(A, \circ), (B, *)$ zwei Gruppen und $G = A \times B$ zusammen mit

$$(a_1, b_1) \odot (a_2, b_2) := (a_1 \circ a_2, b_1 * b_2).$$

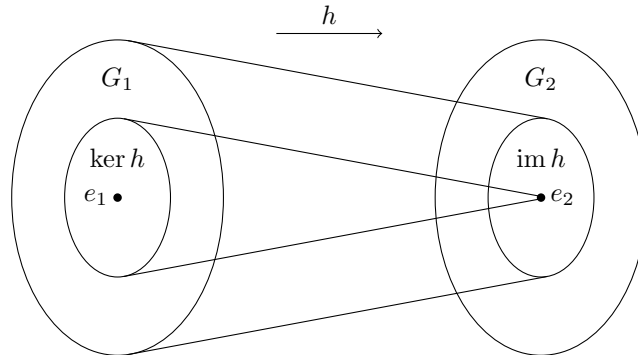
Dann ist $h: G \rightarrow A$, $(a, b) \mapsto a$ ein Homomorphismus.

4. Sei (W, \circ) die Gruppe aus dem vorherigen Beispiel, die aus Wörtern über dem Alphabet $\Omega \cup \bar{\Omega} = \{\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{A}, \mathbf{B}, \mathbf{C}\}$ gebildet wird. Sei $(G, *)$ irgendeine andere Gruppe und seien $g_1, g_2, g_3 \in G$ beliebige Elemente. Dann gibt es genau einen Gruppenhomomorphismus $h: W \rightarrow G$ mit $h(\mathbf{a}) = g_1$, $h(\mathbf{b}) = g_2$, $h(\mathbf{c}) = g_3$, denn aus diesen drei Werten und der Homomorphismeigenschaft folgt direkt der Wert $h(\omega)$ für jedes Wort $\omega \in (\Omega \cup \bar{\Omega})^*$. Zum Beispiel muss gelten:

$$\begin{aligned} h(\mathbf{aCbbbaBBAc}) &= h(\mathbf{a} \circ \mathbf{C} \circ \mathbf{b} \circ \mathbf{b} \circ \mathbf{a} \circ \mathbf{B} \circ \mathbf{B} \circ \mathbf{A} \circ \mathbf{c}) \\ &= g_1 * g_3^{-1} * g_2 * g_2 * g_1 * g_2^{-1} * g_2^{-1} * g_1^{-1} * g_3 \end{aligned}$$

Satz 7. Sei $h: G_1 \rightarrow G_2$ ein Homomorphismus. Dann gilt:

1. Ist e_1 das Neutralelement von G_1 und e_2 das Neutralelement von G_2 , so gilt $h(e_1) = e_2$.
2. $\ker h$ ist eine Untergruppe von G_1 .
3. $\operatorname{im} h$ ist eine Untergruppe von G_2 .



Beweis.

1. Es gilt $h(e_1) = h(e_1 \circ e_1) = h(e_1) * h(e_1)$. Multipliziert man diese Gleichung mit dem Inversen von $h(e_1)$ in G_2 , so erhält man $e_2 = h(e_1)$.
2. Nach Teil 1 gilt zunächst $e_1 \in \ker h$ und damit insbesondere $\ker h \neq \emptyset$. Darüber hinaus bleibt zu zeigen: für alle $u, v \in \ker h$ gilt $u \circ v \in \ker h$ und $u^{-1} \in \ker h$.

Seien $u, v \in \ker h$ beliebig. Es gilt $h(u) = h(v) = e_2$, weil $u, v \in \ker h$. Folglich gilt:

$$h(u \circ v) = h(u) * h(v) = e_2 * e_2 = e_2,$$

und also $u \circ v \in \ker h$.

Es gilt $e_2 = h(e_1) = h(u \circ u^{-1}) = h(u) * h(u^{-1}) = e_2 * h(u^{-1}) = h(u^{-1})$.

(Daraus folgt übrigens auch $h(u)^{-1} = h(u^{-1})$.)

3. Nach Teil 1 gilt zunächst $e_2 \in \text{im } h$. Darüber hinaus bleibt zu zeigen: für alle $u, v \in \text{im } h$ gilt $u * v \in \text{im } h$ und $u^{-1} \in \text{im } h$.

Seien $u, v \in \text{im } h$ beliebig. Dann gibt es $a, b \in G_1$ mit $u = h(a)$ und $v = h(b)$.

Es gilt $u * v = h(a) * h(b) = h(a \circ b) \in \text{im } h$.

Außerdem $u^{-1} = h(a)^{-1} = h(a^{-1}) \in \text{im } h$. ■

Beispiel. Betrachte $\pi = (1\ 3)(2\ 4\ 7)(5\ 6) \in S_7$. Dann ist

$$h: \mathbb{Z} \rightarrow S_7, \quad h(n) := \pi^n$$

ein Gruppenhomomorphismus von $(\mathbb{Z}, +)$ nach (S_7, \circ) . Dabei bezeichnet $\pi^n = \pi \circ \dots \circ \pi$ die Permutation, die sich durch n -fache Verkettung von π mit sich selbst gibt. Im Fall $n < 0$ ist die Definition im Sinn von $\pi^n := (\pi^{-1})^{|n|}$ gemeint, und für $n = 0$ soll $\pi^0 = \text{id}$ sein.

Wegen

$$\begin{aligned} \pi^0 &= \text{id} \\ \pi^1 &= (1\ 3)(2\ 4\ 7)(5\ 6) \\ \pi^2 &= (2\ 7\ 4) \\ \pi^3 &= (1\ 3)(5\ 6) \end{aligned}$$

$$\begin{aligned}\pi^4 &= (2\ 4\ 7) \\ \pi^5 &= (1\ 3)(2\ 7\ 4)(5\ 6) \\ \pi^6 &= \text{id}\end{aligned}$$

gilt $\pi^n = \pi^{n+6}$ für alle $n \in \mathbb{Z}$. Daraus folgt

$$\ker h = 6\mathbb{Z} = \{\dots, -12, -6, 0, 6, 12, \dots\} \subseteq \mathbb{Z}$$

und

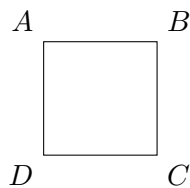
$$\text{im } h = \{\text{id}, \pi, \pi^2, \pi^3, \pi^4, \pi^5\} \subseteq S_7.$$

Beachten Sie, dass $6\mathbb{Z}$ mit $+$ eine Untergruppe von \mathbb{Z} und $\{\text{id}, \pi, \pi^2, \pi^3, \pi^4, \pi^5\}$ mit \circ eine Untergruppe von S_7 ist.

Gruppen kann man dazu verwenden, Symmetrien zu beschreiben.

Beispiel.

1. Betrachte ein Quadrat mit den Eckpunkten A, B, C, D .



Eine Symmetrie lässt sich auffassen als eine Funktion $\{A, B, C, D\} \rightarrow \{A, B, C, D\}$, die das Quadrat als Ganzes fest lässt.

Das Quadrat hat folgende Symmetrien:

- $\sigma = \begin{pmatrix} A & B & C & D \\ D & C & B & A \end{pmatrix}$ – das ist die Spiegelung an der horizontalen Achse
- $\rho = \begin{pmatrix} A & B & C & D \\ B & C & D & A \end{pmatrix}$ – das ist die Rotation um 90° entgegen dem Uhrzeigersinn

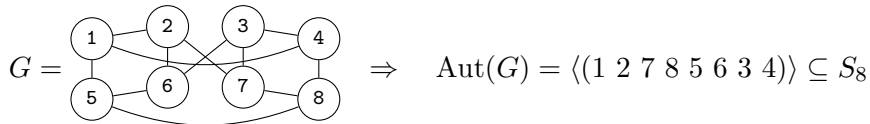
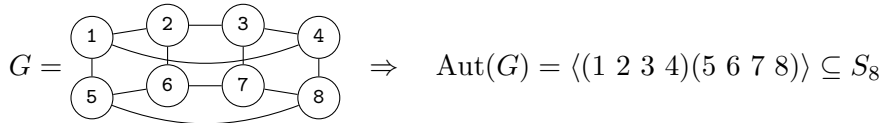
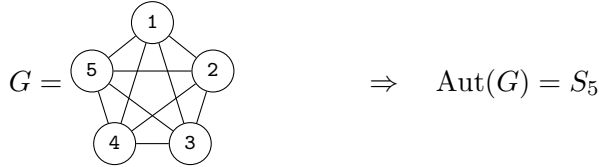
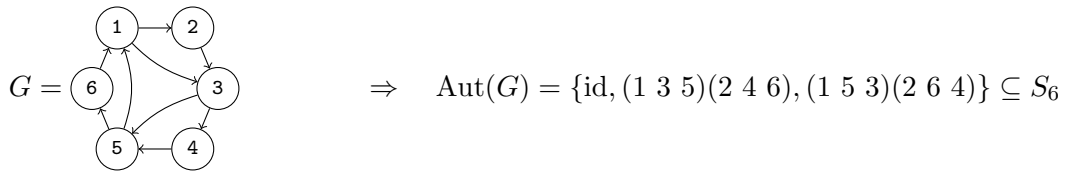
sowie einige weitere, die sich aus diesen durch Komposition bilden lassen. Die komplette Liste lautet:

$$G = \{\text{id}, \rho, \rho^2, \rho^3, \sigma, \sigma\rho, \sigma\rho^2, \sigma\rho^3\}.$$

Diese Menge G bildet zusammen mit der Komposition eine Gruppe, die sogenannte *Symmetriegruppe* (engl. *symmetry group*) des Quadrats. Die Gruppe ist nicht abelsch; es gilt aber zum Beispiel $\sigma\rho = \rho^3\sigma$.

2. Sei $G = (V, E)$ ein Graph. Die Menge aller Graphenisomorphismen von G auf sich selbst bildet mit der Komposition eine Gruppe, die sogenannte *Symmetriegruppe* (engl. *symmetry group*) oder *Automorphismengruppe* (engl. *automorphism group*) von G . Man bezeichnet sie mit $\text{Aut}(G)$. Diese Gruppe ist immer eine Untergruppe von S_V .

Beispiele:



Die Transformationen einer Symmetriegruppe lassen zwar die betreffende Struktur als Ganzes fest, aber nicht unbedingt ihre Einzelteile. Zum Beispiel werden die einzelnen Knoten eines Graphen von einer Symmetrie-Transformation typischerweise auf andere Knoten abgebildet. Andererseits ist es typischerweise nicht möglich, einen bestimmten Knoten durch eine Symmetrie-Transformation auf jeden beliebigen anderen Knoten abzubilden. Um genau zu beschreiben, was eine Gruppe (z.B. die Symmetriegruppe eines Graphen) mit den Elementen einer Menge (z.B. der Knotenmenge des Graphen) macht, verwendet man den Begriff der Gruppen-Operationen, der wie folgt definiert ist.

Definition 17. Sei (G, \circ) eine Gruppe, e das Neutralelement von G , und X eine Menge. Eine Funktion

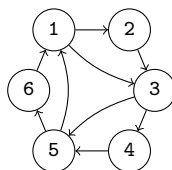
$$*: G \times X \rightarrow X$$

heißt *Gruppenoperation* (engl. *group action*) von G auf X , falls gilt:

1. $\forall x \in X : e * x = x$
2. $\forall g, h \in G \forall x \in X : (g \circ h) * x = g * (h * x)$

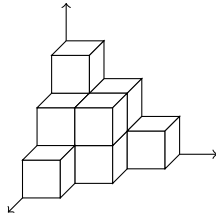
Beispiel.

1. $G = S_n$ operiert auf der Menge $X = \{1, \dots, n\}$, wenn man definiert $\pi * x := \pi(x)$ für $\pi \in S_n$ und $x \in X$.
2. Die Symmetriegruppe $\text{Aut}(G)$ eines Graphen $G = (V, E)$ definiert eine Gruppenoperation $\text{Aut}(G) \times V \rightarrow V$ auf der Knotenmenge sowie eine Gruppenoperation $\text{Aut}(G) \times E \rightarrow E$ auf der Kantenmenge des Graphen. Ist G zum Beispiel der Graph



von vorher, und ist $*$: $\text{Aut}(G) \times V \rightarrow V$ die Operation der Symmetriegruppe auf den Knoten von G , so gilt $(1\ 3\ 5)(2\ 4\ 6) * 4 = 6$, $(1\ 5\ 3)(2\ 6\ 4) * 5 = 3$, usw. Wie man sieht, kann der Knoten 1 durch Anwendung von Symmetrien auf die Knoten 3 und 5 abgebildet werden, nicht aber auf 2, 4 oder 6. Mit der entsprechenden Operation von $\text{Aut}(G)$ auf E kann die Kante $(1, 3)$ auf die Kante $(3, 5)$ oder die Kante $(5, 1)$ abgebildet werden, aber auf keine andere Kante.

3. Betrachten Sie die folgende Anordnung von 11 Würfeln in der Ecke eines Raumes:



Die Anordnung ist symmetrisch in dem Sinn, dass sich genau dann ein Würfel an Position (i, j, k) befindet, wenn es auch einen Würfel an jeder der Positionen (i, k, j) , (j, i, k) , (j, k, i) , (k, i, j) , (k, j, i) gibt. Wir können diese Symmetrie als eine Gruppenoperation von S_3 auf der Würfelmenge

$$X = \{(1, 1, 1), (2, 1, 1), (1, 2, 1), (1, 1, 2), (2, 2, 1), (2, 1, 2), \\ (1, 2, 2), (2, 2, 2), (1, 1, 3), (1, 3, 1), (3, 1, 1)\}$$

auffassen. Die Elemente von S_3 wirbeln die einzelnen Würfel in einer Weise durcheinander, dass dabei die Anordnung als Ganzes erhalten bleibt.

4. Sei (G, \circ) eine Gruppe und H eine Untergruppe von G . Dann wird durch

$$*: H \times G \rightarrow G, \quad h * g := h \circ g \circ h^{-1}$$

eine Gruppenoperation der Gruppe (H, \circ) auf der Menge G definiert.

Definition 18. Sei (G, \circ) eine Gruppe, X eine Menge und $*$: $G \times X \rightarrow X$ eine Gruppenoperation. Weiter sei $x \in X$.

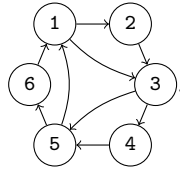
1. $G * x := \{g * x : g \in G\} \subseteq X$ heißt die *Bahn* (engl. *orbit*) von x (unter der Gruppenoperation $*$).
2. $\text{Stab}(x) := \{g \in G : g * x = x\} \subseteq G$ heißt der *Stabilisator* (engl. *stabilizer*) von x (bezüglich der Gruppenoperation $*$).

Beispiel.

1. Sei $G = S_4$ und $X = \{1, 2, 3, 4\}$ und sei $*$: $G \times X \rightarrow X$ definiert durch $\pi * x := \pi(x)$. Dann gilt $G * 3 = X$ und

$$\text{Stab}(3) = \{\text{id}, \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 4 & 3 & 2 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 3 & 4 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 3 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 1 & 3 & 2 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 2 & 3 & 1 \end{pmatrix}\}.$$

2. Sei nun $G = \langle (2\ 1), (3\ 4) \rangle \subseteq S_4$ und $X = \{1, 2, 3, 4\}$ und sei $*$: $G \times X \rightarrow X$ definiert durch $\pi * x := \pi(x)$. In diesem Fall gilt $G * 3 = \{3, 4\}$ und $\text{Stab}(3) = \{\text{id}, (2\ 1)\}$.
3. Sei $G = (V, E)$ der Graph



Die Symmetriegruppe $\text{Aut}(G)$ teilt die Knotenmenge des Graphen in zwei Bahnen auf, nämlich $\{1, 3, 5\}$ und $\{2, 4, 6\}$. Für jeden Knoten $v \in V$ gilt in diesem Beispiel $\text{Stab}(v) = \{\text{id}\}$.

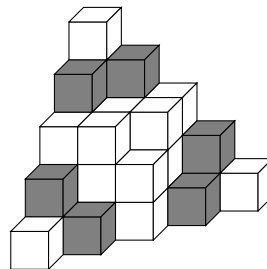
4. Sei $G = S_5$ und $X = \mathbb{N}^5$. Betrachte die Gruppenoperation

$$*: G \times X \rightarrow X, \quad \pi * (x_1, \dots, x_5) := (x_{\pi(1)}, \dots, x_{\pi(5)}).$$

Dann gilt unter anderem

- $\text{Stab}((0, 0, 0, 0, 0)) = G$
 $G * (0, 0, 0, 0, 0) = \{(0, 0, 0, 0, 0)\}$
- $\text{Stab}((0, 0, 0, 0, 1)) \cong S_4$
 $G * (0, 0, 0, 0, 1) = \{(0, 0, 0, 0, 1), (0, 0, 0, 1, 0), (0, 0, 1, 0, 0), (0, 1, 0, 0, 0), (1, 0, 0, 0, 0)\}$
- $\text{Stab}((0, 0, 0, 1, 1)) \cong S_3 \times S_2$ und
 $G * (0, 0, 0, 1, 1) = \{(0, 0, 0, 1, 1), (0, 0, 1, 0, 1), (0, 1, 0, 0, 1), (1, 0, 0, 0, 1), (0, 0, 1, 1, 0), (0, 1, 0, 1, 0), (1, 0, 0, 1, 0), (0, 1, 1, 0, 0), (1, 0, 1, 0, 0), (1, 1, 0, 0, 0)\}$
- $\text{Stab}((0, 0, 1, 1, 2)) \cong S_2 \times S_2$ und
 $G * (0, 0, 1, 1, 2) = \{(0, 0, 1, 1, 2), (0, 0, 1, 2, 1), (0, 0, 2, 1, 1), (0, 2, 0, 1, 1), (2, 0, 0, 1, 1), (0, 1, 0, 1, 2), (0, 1, 0, 2, 1), (0, 1, 2, 0, 1), (0, 2, 1, 0, 1), (2, 0, 1, 0, 1), (0, 1, 1, 0, 2), (0, 1, 1, 2, 0), (0, 1, 2, 1, 0), (0, 2, 1, 1, 0), (2, 0, 1, 1, 0), (1, 0, 0, 1, 2), (1, 0, 0, 2, 1), (1, 0, 2, 0, 1), (1, 2, 0, 0, 1), (2, 1, 0, 0, 1), (1, 0, 1, 0, 2), (1, 0, 1, 2, 0), (1, 0, 2, 1, 0), (1, 2, 0, 1, 0), (2, 1, 0, 1, 0), (1, 1, 0, 0, 2), (1, 1, 0, 2, 0), (1, 1, 2, 0, 0), (1, 2, 1, 0, 0), (2, 1, 1, 0, 0)\}$

5. Die unten stehende Anordnung von Würfeln ist im selben Sinn symmetrisch wie die Anordnung aus dem vorherigen Beispiel. Wir haben hier sechs Würfel grau markiert, die miteinander eine Bahn bilden.



Beachten Sie, dass jeder Würfel zu genau einer Bahn gehört. In diesem Beispiel hat jede Bahn entweder ein Element oder drei Elemente oder sechs Elemente.

Satz 8. Sei (G, \circ) eine Gruppe, X eine Menge und $*$: $G \times X \rightarrow X$ eine Gruppenoperation.

1. Durch $x \sim y \iff \exists g \in G : x = g * y$ wird auf X eine Äquivalenzrelation erklärt.
2. Für jedes $x \in X$ ist $[x]_{\sim}$ genau die Bahn von x .
3. Für jedes $x \in X$ ist $\text{Stab}(x)$ eine Untergruppe von G .

Beweis.

1. Reflexivität: $x \sim x$ gilt, weil jede Gruppe G ein Neutralelement e enthält (Def. 14) und für dieses nach Def. 17 gilt $x = e * x$. Es gibt also ein $g \in G$, nämlich $g = e$, mit $x = g * x$.

Symmetrie: Wenn $x \sim y$ gilt, dann $\exists g \in G : x = g * y$. Wähle ein solches g . Nach Def. 14 existiert ein Inverses g^{-1} von g in G . Damit gilt

$$g^{-1} * x = g^{-1} * (g * y) \underset{\text{Def. 17}}{=} (g^{-1} \circ g) * y \underset{\text{Def. 14}}{=} e * y \underset{\text{Def. 17}}{=} y.$$

Also gibt es ein Gruppenelement h , nämlich $h = g^{-1}$, mit $y = h * x$. Daraus folgt $y \sim x$.

Transitivität: Angenommen es gilt $x \sim y$ und $y \sim z$. Dann gibt es Gruppenelemente $g_1, g_2 \in G$ mit $x = g_1 * y$ und $y = g_2 * z$. Dann gilt auch

$$x = g_1 * y = g_1 * (g_2 * z) \underset{\text{Def. 17}}{=} (g_1 \circ g_2) * z.$$

Also gibt es ein Gruppenelement g , nämlich $g = g_1 \circ g_2$, mit $x = g * z$. Daraus folgt $x \sim z$.

2. Ist $x \in X$, so ist

$$\begin{aligned} [x]_{\sim} &= \{y \in X \mid x \sim y\} \\ &= \{y \in X \mid y \sim x\} \\ &= \{y \in X \mid \exists g \in G : y = g * x\} \\ &= \{g * x \mid g \in G\} = G * x. \end{aligned}$$

3. (a) Das Neutralelement e von G liegt in $\text{Stab}(x)$, weil $e * x = x$ nach Def. 17.

(b) Sind $g, h \in \text{Stab}(x)$, so gilt $g * x = h * x = x$. Damit gilt auch $(g \circ h) * x = g * (h * x) = g * x = x$, also ist $g \circ h \in \text{Stab}(x)$.

(c) Ist $g \in \text{Stab}(x)$, so gilt $g * x = x$. Dann gilt auch $g^{-1} * x = g^{-1} * (g * x) = (g^{-1} \circ g) * x = e * x = x$, also ist auch $g^{-1} \in \text{Stab}(x)$.

Aus (a), (b), (c) zusammen folgt, dass $\text{Stab}(x)$ eine Untergruppe von G ist.

7 Modulare Arithmetik

Wenn man in einem Computerprogramm mit reellen Zahlen hantiert, sollte man sich immer darüber im Klaren sein, dass diese Zahlen nur mit einer bestimmten beschränkten Genauigkeit im Computer dargestellt werden können. Addition und Multiplikation liefern für die meisten Inputs nicht das mathematisch exakte Ergebnis, sondern ein Ergebnis, das mit einem bestimmten Rundungsfehler behaftet ist. Zum Beispiel könnte man für die Division $1/3$ das nicht ganz korrekte Ergebnis 0.33333333 bekommen. Die Differenz zum korrekten Ergebnis mag zwar klein erscheinen, aber wenn man in einer umfangreichen Rechnung in jedem Schritt einen kleinen Fehler macht, ist es nicht ungewöhnlich, dass der Fehler im Laufe der Rechnung immer größer wird, und dass man besondere Vorkehrungen treffen muss, damit das Endergebnis überhaupt noch etwas mit dem korrekten Ergebnis zu tun hat. Wie sich Rundungsfehler in einer Rechnung fortpflanzen und was man dagegen tun kann, lernt man in Vorlesungen über Analysis oder Numerik.

Anders als bei reellen Zahlen kann man mit ganzen Zahlen komplett ohne Rundungsfehler rechnen. Allerdings hat eine ganze Zahl normalerweise kein Inverses bezüglich der Multiplikation, d.h. die Division zweier ganzer Zahlen ergibt im allgemeinen nicht wieder eine ganze Zahl. Ein zweites Problem ist, dass ganze Zahlen beliebig viele Ziffern haben können, die man alle abspeichern muss, wenn man nicht runden möchte. Es kann deshalb passieren, dass eine Zahl $a \in \mathbb{Z}$ gerade noch so in den Speicher passt, die Zahl a^2 , die etwa doppelt so viele Ziffern haben wird wie a , aber nicht mehr. Man spricht bei diesem Phänomen vom *expression swell*. Eines der Ziele dieses Abschnitts ist die Konstruktion von Zahlenräumen, in denen man Addition, Subtraktion, Multiplikation und Division exakt berechnen kann (d.h. ohne Rundungsfehler), und bei denen es keinen expression swell gibt.

Wir beginnen mit der Definition zweier Funktionen, die man dazu verwenden kann, reelle Zahlen auf ganze Zahlen zu runden.

Definition 19.

1. Die Funktion $\lfloor \cdot \rfloor : \mathbb{R} \rightarrow \mathbb{Z}$ ist definiert durch

$$\lfloor x \rfloor := \max\{m \in \mathbb{Z} : m \leq x\}$$

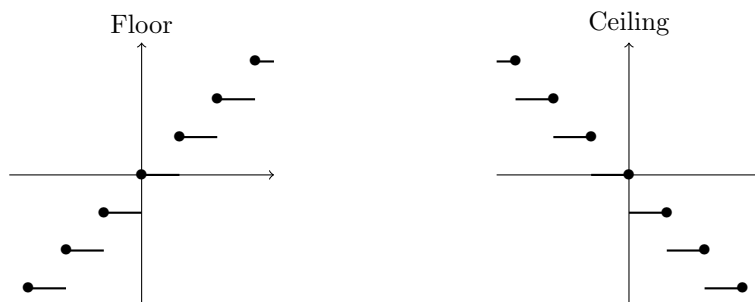
(gesprochen: „Floor x “).

2. Die Funktion $\lceil \cdot \rceil : \mathbb{R} \rightarrow \mathbb{Z}$ ist definiert durch

$$\lceil x \rceil := \min\{m \in \mathbb{Z} : m \geq x\}$$

(gesprochen: „Ceiling x “).

Beispiel. Es gilt $\lfloor 5 \rfloor = \lfloor 5.00000001 \rfloor = \lfloor 5.999999998 \rfloor = 5$, $\lfloor -\frac{1}{2} \rfloor = -1$, $\lceil 5 \rceil = \lceil 4.999998 \rceil = \lceil 4.00000001 \rceil = 5$, $\lceil -\frac{1}{2} \rceil = 0$, usw.



Durch die ausgefüllten Kreise soll verdeutlicht werden, welches der Funktionswert an den Stellen ist, wo ein Zweig beginnt und ein anderer endet.

Satz 9. Für alle $x \in \mathbb{R}$ gilt:

1. $\lfloor -x \rfloor = -\lceil x \rceil$
2. $\lfloor x \rfloor = \lceil x \rceil \iff x \in \mathbb{Z}$
3. $\lfloor x \rfloor = \lceil x \rceil - 1 \iff x \notin \mathbb{Z}$
4. $\forall n \in \mathbb{Z} : \lfloor x \rfloor = n \iff x - 1 < n \leq x \iff n \leq x < n + 1$
5. $\forall n \in \mathbb{Z} : \lceil x \rceil = n \iff x \leq n < x + 1 \iff n - 1 < x \leq n$
6. $x - 1 < \lfloor x \rfloor \leq x \leq \lceil x \rceil < x + 1$
7. $\forall n \in \mathbb{Z} : \lfloor x + n \rfloor = \lfloor x \rfloor + n$
8. $\forall n \in \mathbb{Z} : \lceil x + n \rceil = \lceil x \rceil + n$

Definition 20. Die Funktion $\text{mod} : \mathbb{R} \times (\mathbb{R} \setminus \{0\}) \rightarrow \mathbb{R}$ ist definiert durch

$$\text{mod}(x, y) := x - y \left\lfloor \frac{x}{y} \right\rfloor.$$

Statt $\text{mod}(x, y)$ schreibt man auch $x \bmod y$; Sprechweise: „ x modulo y “.

Beispiel.

1. $\text{mod}(7, 5) = 2$, $\text{mod}(7, -5) = -3$, $\text{mod}(-7, 5) = 3$, $\text{mod}(-7, -5) = -2$.
2. Eine Uhr mit einem gewöhnlichen analogen Ziffernblatt zählt die Stunden modulo 12.
3. Wenn x und y positive ganze Zahlen sind und man eine Division mit Rest von x durch y durchführt, dann ist $\text{mod}(x, y)$ genau der Rest, z.B. gilt $\text{mod}(12345, 52) = 21$, denn

$$\begin{array}{r} 12345 = 52 \cdot 237 + 21. \\ -\underline{104} \\ 1945 \\ -\underline{156} \\ 385 \\ -\underline{364} \\ 21 \end{array}$$

4. Für alle $x \in \mathbb{R}$ gilt $0 \leq \text{mod}(x, 1) < 1$ und $\text{mod}(x, 1) = 0 \iff x \in \mathbb{Z}$. Insbesondere gilt z.B. $\text{mod}(\pi, 1) = \pi - 3 \approx 0.1415926\dots$, $\text{mod}(\sqrt{2}, 1) = \sqrt{2} - 1 \approx 0.414\dots$, usw.
5. Für alle $x \in \mathbb{R}$ gilt $\sin(x) = \sin(\text{mod}(x, 2\pi))$
6. Für alle $x_1, x_2 \in \mathbb{R}$ und alle $y \in \mathbb{Z} \setminus \{0\}$ gilt

$$\text{mod}(x_1, y) = \text{mod}(x_2, y) \iff x_1 - x_2 \in \mathbb{Z} \wedge y \mid x_1 - x_2.$$

Beweis: „ \Rightarrow “ $\text{mod}(x_1, y) = \text{mod}(x_2, y) \Rightarrow x_1 - y\lfloor x_1/y \rfloor = x_2 - y\lfloor x_2/y \rfloor \Rightarrow x_1 - x_2 = y(\lfloor x_1/y \rfloor - \lfloor x_2/y \rfloor)$. Weil $\lfloor x_1/y \rfloor - \lfloor x_2/y \rfloor$ in \mathbb{Z} liegt, folgt die Behauptung.

„ \Leftarrow “ Aus $x_1 - x_2 \in \mathbb{Z}$ und $y \mid x_1 - x_2$ folgt, dass es ein $m \in \mathbb{Z}$ gibt mit $ym = x_1 - x_2$, also $x_1 = x_2 + my$. Darum gilt $\text{mod}(x_1, y) = \text{mod}(x_2 + my, y) = x_2 + my - y\lfloor (x_2 + my)/y \rfloor = x_2 + my - y\lfloor x_2/y + m \rfloor = x_2 + my - y(\lfloor x_2/y \rfloor + m) = x_2 - y\lfloor x_2/y \rfloor = \text{mod}(x_2, y)$, wie behauptet.

Im folgenden werden wir die Funktion mod nur noch auf ganze Zahlen anwenden. Eine Anwendung der Funktion ist die Berechnung des größten gemeinsamen Teilers zweier gegebener ganzer Zahlen. Wenn $a, b \in \mathbb{Z} \setminus \{0\}$ zwei ganze Zahlen mit $a \mid b$ sind, wenn also a ein Teiler von b ist, dann muss $|a| \leq |b|$ gelten. Die Menge $T(b)$ aller Teiler von b ist deshalb eine Teilmenge von $\{-|b|, \dots, -1, 0, 1, \dots, |b|\}$, und damit eine endliche Menge. Für je zwei ganze Zahlen $a, b \in \mathbb{Z}$ ist dann auch $T(a) \cap T(b)$ eine endliche Teilmenge von \mathbb{Z} . Das größte Element in dieser Menge nennt man den *größten gemeinsamen Teiler* (engl. *greatest common divisor*) von a und b , Notation: $\text{gcd}(a, b) := \max(T(a) \cap T(b))$.

Der größte gemeinsame Teiler zweier gegebener Zahlen $a, b \in \mathbb{Z}$ lässt sich mit dem *euklidischen Algorithmus* berechnen:

INPUT: $a, b \in \mathbb{Z}$

OUTPUT: $\text{gcd}(a, b)$

- 1 setze $(g, g') = (a, b)$
- 2 solange $g' \neq 0$:
- 3 ersetze (g, g') durch $(g', \text{mod}(g, g'))$
- 4 gib $|g|$ als Ergebnis zurück.

Dass dieser Algorithmus terminiert, folgt daraus, dass für alle $a, b \in \mathbb{Z}$ mit $b \neq 0$ gilt $|\text{mod}(a, b)| < |b|$ und deshalb bei jedem Durchlauf des Schritts 3 der Wert von g' durch einen Wert mit kleinerem Betrag ersetzt wird. Nach endlich vielen Schritten muss g' den Wert 0 erreichen, und das ist genau die Abbruchbedingung der Schleife.

Zur Korrektheit überlegt man sich, dass für alle $a, b \in \mathbb{Z}$ gilt $\text{gcd}(a, b) = \text{gcd}(b, \text{mod}(a, b))$. In der Tat gilt sogar $T(a) \cap T(b) = T(b) \cap T(\text{mod}(a, b))$: Ist nämlich d ein gemeinsamer Teiler von a und b , etwa $a = da_0, b = db_0$ für gewisse $a_0, b_0 \in \mathbb{Z}$, dann gilt $\text{mod}(a, b) = \text{mod}(da_0, db_0) = da_0 - db_0 \lfloor da_0/db_0 \rfloor = d \text{mod}(a_0, b_0)$, und somit ist d auch ein gemeinsamer Teiler von b und $\text{mod}(a, b)$. Damit gilt $T(a) \cap T(b) \subseteq T(b) \cap T(\text{mod}(a, b))$. Ist umgekehrt d ein gemeinsamer Teiler von b und $\text{mod}(a, b)$, dann ist d auch ein gemeinsamer Teiler von a und b , denn dann gilt $a = qb + \text{mod}(a, b)$ für ein gewisses $q \in \mathbb{Z}$. Wir haben also gezeigt $T(a) \cap T(b) = T(b) \cap T(\text{mod}(a, b))$, und damit müssen insbesondere die maximalen Elemente $\text{gcd}(a, b)$ und $\text{gcd}(b, \text{mod}(a, b))$ dieser Mengen übereinstimmen. Die Ersetzungen in Schritt 3 ändern den Wert des größten gemeinsamen Teilers also nicht. Wenn die Schleife zum Ende gekommen ist, gilt $g' = 0$, und wegen $\text{gcd}(g, 0) = |g|$ ist die Ausgabe des Algorithmus korrekt.

Beispiel. Betrachte $a = 19 \cdot 27 \cdot 47 \cdot 61 = 1470771$, $b = 19 \cdot 23 \cdot 43 \cdot 59 = 1108669$. Der euklidische Algorithmus berechnet eine Folge von Divisionsresten, $r_1 = a$, $r_2 = b$, und $r_k = \text{mod}(r_{k-2}, r_{k-1})$ für $k \geq 2$:

$$\begin{aligned}
 r_3 &= \text{mod}(1470771, 1108669) = 362102, \\
 r_4 &= \text{mod}(1108669, 362102) = 22363, \\
 r_5 &= \text{mod}(362102, 22363) = 4294, \\
 r_6 &= \text{mod}(22363, 4294) = 893, \\
 r_7 &= \text{mod}(4294, 893) = 722, \\
 r_8 &= \text{mod}(893, 722) = 171, \\
 r_9 &= \text{mod}(722, 171) = 38, \\
 r_{10} &= \text{mod}(171, 38) = 19, \\
 r_{11} &= \text{mod}(38, 19) = 0.
 \end{aligned}$$

Jeweils zwei aufeinanderfolgende Zahlen r_i haben den gleichen größten gemeinsamen Teiler. Insbesondere gilt $\text{gcd}(a, b) = \text{gcd}(19, 0) = 19$.

Man kann der Ausgabe des euklidischen Algorithmus zwar vertrauen, weil wir bewiesen haben, dass der Algorithmus korrekt ist, aber um zum Beispiel eine Implementierung des Algorithmus auf Fehler zu testen, wäre es praktisch, wenn man die Korrektheit eines Ergebnisses unabhängig überprüfen könnte. Es ist noch leicht zu überprüfen, ob das Ergebnis g ein gemeinsamer Teiler der Eingabezahlen a, b ist. Dazu braucht man nur überprüfen, dass $\text{mod}(a, g) = \text{mod}(b, g) = 0$ gilt. Aber es ist nicht ohne weiteres zu sehen, ob es sich tatsächlich um den größten gemeinsamen Teiler handelt.

Es gibt eine Erweiterung des euklidischen Algorithmus, die Zusatzdaten berechnet, mit denen man die Korrektheit der Ausgabe unabhängig zertifizieren kann. Es gibt nämlich für jede Wahl von $a, b \in \mathbb{Z}$ zwei passende Zahlen $u, v \in \mathbb{Z}$, so dass $\text{gcd}(a, b) = ua + vb$ gilt. Mehr noch: wenn $g, u, v \in \mathbb{Z}$ irgendwelche Zahlen sind, so dass $g = ua + vb$ gilt, dann muss $\text{gcd}(a, b) \mid g$ gelten. Man kann also, wenn man nicht nur g sondern auch u, v kennt, leicht unabhängig überprüfen, dass $g = \text{gcd}(a, b)$ gilt.

Der erweiterte euklidische Algorithmus berechnet u und v mit Hilfe von Zusatzvariablen, in denen er sich merkt, wie die aktuellen Werte von g und g' aus den ursprünglichen entstanden sind.

INPUT: $a, b \in \mathbb{Z}$

OUTPUT: $g = \text{gcd}(a, b)$ und $u, v \in \mathbb{Z}$ mit $g = ua + vb$.

- 1 setze $(g, u, v, g', u', v') = (a, 1, 0, b, 0, 1)$
- 2 solange $g' \neq 0$:
- 3 berechne $q = \lfloor g/g' \rfloor$
- 4 setze $(g, u, v, g', u', v') = (g', u', v', g - qg', u - qu', v - qv')$
- 5 gib $(|g|, \frac{|g|}{g}u, \frac{|g|}{g}v)$ als Ergebnis zurück.

Beispiel. Betrachte $a = 19 \cdot 27 \cdot 47 \cdot 61 = 1470771$, $b = 19 \cdot 23 \cdot 43 \cdot 59 = 1108669$. Die Anwendung des erweiterten euklidischen Algorithmus auf diese beiden Zahlen liefert folgende Zwischenergebnisse:

g	u	v	g'	u'	v'
1470771	1	0	1108669	0	1
1108669	0	1	362102	1	-1
362102	1	-1	22363	-3	4
22363	-3	4	4294	49	-65
4294	49	-65	893	-248	329
893	-248	329	722	1041	-1381
722	1041	-1381	171	-1289	1710
171	-1289	1710	38	6197	-8221
38	6197	-8221	19	-26077	34594
19	-26077	34594	0	58351	-77409

Daraus folgt $\text{gcd}(a, b) = 19 = -26077a + 34594b$.

Definition 21. Sei $m \in \mathbb{N} \setminus \{0, 1\}$. Auf der Menge \mathbb{Z} wird die Äquivalenzrelation \equiv_m definiert durch

$$x \equiv_m y \iff \text{mod}(x, m) = \text{mod}(y, m).$$

Die Menge \mathbb{Z}/\equiv_m der Äquivalenzklassen wird mit \mathbb{Z}_m bezeichnet.

Auf \mathbb{Z}_m sind die Verknüpfungen $+$ und \cdot definiert durch

$$\begin{aligned} + : \mathbb{Z}_m \times \mathbb{Z}_m &\rightarrow \mathbb{Z}_m, & [x]_{\equiv_m} + [y]_{\equiv_m} &:= [x + y]_{\equiv_m} \\ \cdot : \mathbb{Z}_m \times \mathbb{Z}_m &\rightarrow \mathbb{Z}_m, & [x]_{\equiv_m} \cdot [y]_{\equiv_m} &:= [xy]_{\equiv_m}. \end{aligned}$$

Die Definition von $+$ und \cdot sind natürlich repräsentantenunabhängig. Gilt nämlich $x_1 \equiv_m x_2$ und $y_1 \equiv_m y_2$, so gilt nach Definition $\text{mod}(x_1, m) = \text{mod}(x_2, m)$ und $\text{mod}(y_1, m) = \text{mod}(y_2, m)$. Da es sich um ganze Zahlen handelt, folgt $m \mid x_1 - x_2$ und $m \mid y_1 - y_2$, also $x_1 - x_2 = x_0 m$ und $y_1 - y_2 = y_0 m$ für gewisse $x_0, y_0 \in \mathbb{Z}$. Addition dieser Gleichungen liefert $(x_1 + y_1) - (x_2 + y_2) = (x_0 + y_0)m$, also $m \mid (x_1 + y_1) - (x_2 + y_2)$, also $x_1 + y_1 \equiv_m x_2 + y_2$. Damit ist die Definition von $+$ repräsentantenunabhängig. Für die Multiplikation zeigt die Rechnung

$$\begin{aligned} x_1 - x_2 &= x_0 m \\ \Rightarrow (x_1 - x_2)y_1 &= x_0 y_1 m \\ \Rightarrow x_1 y_1 - x_2 y_1 + x_2(y_1 - y_2) &= x_0 y_1 m + x_2 y_0 m \\ \Rightarrow x_1 y_1 - x_2 y_2 &= (x_0 y_1 + x_2 y_0)m, \end{aligned}$$

dass $x_1 y_1 \equiv_m x_2 y_2$ gilt. Also ist auch diese Definition zulässig.

Beispiel. Es gilt z.B.

$$\begin{aligned} [5]_{\equiv_7} \cdot [3]_{\equiv_7} &= [15]_{\equiv_7} = [1]_{\equiv_7} \\ [5]_{\equiv_7} + [3]_{\equiv_7} &= [8]_{\equiv_7} = [1]_{\equiv_7} \\ [4]_{\equiv_6} \cdot [3]_{\equiv_6} &= [12]_{\equiv_6} = [0]_{\equiv_6} \\ [4]_{\equiv_6} + [5]_{\equiv_6} &= [9]_{\equiv_6} = [3]_{\equiv_6}, \end{aligned}$$

usw. Wenn aus dem Kontext klar ist, in welchem Zahlenraum man sich befindet, kann man die Äquivalenzklassenklammern $[\cdot]_{\equiv_m}$ auch weglassen und einfach z.B. $5 \cdot 3 = 1$, $5 + 3 = 1$, $25 = 32$ (für $m = 7$) oder $4 \cdot 3 = 0$, $4 + 5 = 3$, $25 = 31$ (für $m = 6$) schreiben.

Satz 10. Sei $m \in \mathbb{N} \setminus \{0, 1\}$. Dann ist $(\mathbb{Z}_m, +)$ eine abelsche Gruppe, und für die Verknüpfung $\cdot: \mathbb{Z}_m \times \mathbb{Z}_m \rightarrow \mathbb{Z}_m$ gilt:

1. $\forall A, B, C \in \mathbb{Z}_m : (A \cdot B) \cdot C = A \cdot (B \cdot C)$
2. $\forall A, B \in \mathbb{Z}_m : A \cdot B = B \cdot A$
3. $\forall A \in \mathbb{Z}_m : A \cdot [1]_{\equiv_m} = A$
4. $\forall A \in \mathbb{Z}_m : A \cdot [0]_{\equiv_m} = [0]_{\equiv_m}$
5. $\forall A, B, C \in \mathbb{Z}_m : (A + B) \cdot C = (A \cdot C) + (B \cdot C)$.

Beweis.

2. Sind A, B zwei beliebige Elemente von \mathbb{Z}_m , dann gibt es $a, b \in \mathbb{Z}$ mit $A = [a]_{\equiv_m}$, $B = [b]_{\equiv_m}$. Daher gilt

$$A \cdot B = [a]_{\equiv_m} \cdot [b]_{\equiv_m} = [a \cdot b]_{\equiv_m} = [b \cdot a]_{\equiv_m} = [b]_{\equiv_m} \cdot [a]_{\equiv_m} = B \cdot A,$$

wobei im dritten Schritt die Kommutativität der üblichen Multiplikation in \mathbb{Z} verwendet wurde.

Die Beweise für die restlichen Aussagen gehen ähnlich und werden zur Übung empfohlen. ■

\mathbb{Z}_m kann mit \cdot sicher keine Gruppe bilden, denn wegen Punkt 3 kommt nur $[1]_{\equiv_m}$ als Neutralement in Frage, aber dann hat wegen Punkt 4 das Element $[0]_{\equiv_m}$ kein Inverses bezüglich \cdot . Das ist ganz ähnlich wie bei \mathbb{Q} oder \mathbb{R} , die aus dem gleichen Grund mit der Multiplikation keine Gruppe bilden. Allerdings sind $\mathbb{Q} \setminus \{0\}$ und $\mathbb{R} \setminus \{0\}$ zusammen mit der Multiplikation eine Gruppe. Ob das für \mathbb{Z}_m auch gilt, hängt von m ab.

Beispiel.

1. $\mathbb{Z}_6 \setminus \{[0]_{\equiv_6}\}$ ist keine Gruppe. Tatsächlich handelt es sich bei \cdot gar nicht um eine Verknüpfung auf $\mathbb{Z}_6 \setminus \{[0]_{\equiv_6}\}$, denn für die beiden Elemente $[2]_{\equiv_6}, [3]_{\equiv_6}$ von $\mathbb{Z}_6 \setminus \{[0]_{\equiv_6}\}$ gilt $[2]_{\equiv_6} \cdot [3]_{\equiv_6} = [0]_{\equiv_6} \notin \mathbb{Z}_6 \setminus \{[0]_{\equiv_6}\}$, d.h. die Menge $\mathbb{Z}_6 \setminus \{[0]_{\equiv_6}\}$ ist gar nicht abgeschlossen unter \cdot .
2. $\mathbb{Z}_7 \setminus \{[0]_{\equiv_7}\}$ bildet eine Gruppe. Die Verknüpfungstabelle lautet

\cdot	1	2	3	4	5	6
1	1	2	3	4	5	6
2	2	4	6	1	3	5
3	3	6	2	5	1	4
4	4	1	5	2	6	3
5	5	3	1	6	4	2
6	6	5	4	3	2	1

wobei wir jeweils die Äquivalenzklassen durch ihre kleinsten nichtnegativen Elemente dargestellt haben, z.B. steht 6 für $[6]_{\equiv_7}$ usw. Dass es zu jedem Element ein passendes Inverses gibt, sieht man daran, dass in jeder Zeile und jeder Spalte der Verknüpfungstafel jedes Element von $\mathbb{Z}_7 \setminus \{0\}$ genau einmal vorkommt. Es gilt zum Beispiel $4 \cdot 2 = 1$ (genauer gesagt $4 \cdot 2 \equiv_7 1$ bzw. $[4]_{\equiv_7} \cdot [2]_{\equiv_7} = [1]_{\equiv_7}$). Dass 4 in \mathbb{Z}_7 das multiplikative Inverse von 2 ist, kann man auch durch die Gleichungen $4^{-1} = 2$ oder gar durch $\frac{1}{4} = 2$ ausdrücken. Aber Vorsicht mit solchen Schreibweisen: das Objekt $4^{-1} = [4]_{\equiv_7}^{-1} = [2]_{\equiv_7}$ hat nichts mit dem bekannten Objekt $4^{-1} = 0.25 \in \mathbb{Q}$ zu tun.

Satz 11. $(\mathbb{Z}_m \setminus \{[0]_{\equiv_m}\}, \cdot)$ ist genau dann eine Gruppe, wenn m eine Primzahl ist.

Beweis. „ \Rightarrow “ Wenn m keine Primzahl ist, dann gibt es $u, v \in \{2, \dots, m-1\}$ mit $uv = m$. Dann gilt $\text{mod}(u, m) = u \neq 0$ und $\text{mod}(v, m) = v \neq 0$, also $[u]_{\equiv_m} \neq [0]_{\equiv_m}$ und $[v]_{\equiv_m} \neq [0]_{\equiv_m}$. Also sind $[u]_{\equiv_m}$ und $[v]_{\equiv_m}$ Elemente von $\mathbb{Z}_m \setminus \{[0]_{\equiv_m}\}$. Für deren Produkt gilt aber

$$[u]_{\equiv_m} \cdot [v]_{\equiv_m} = [uv]_{\equiv_m} = [m]_{\equiv_m} = [0]_{\equiv_m} \notin \mathbb{Z}_m \setminus \{[0]_{\equiv_m}\},$$

also $\mathbb{Z}_m \setminus \{[0]_{\equiv_m}\}$ nicht abgeschlossen unter \cdot , und damit erstreckt keine Gruppe.

„ \Leftarrow “ m ist eine Primzahl. Es genügt zu zeigen, dass es dann zu jedem Element von $\mathbb{Z}_m \setminus \{[0]_{\equiv_m}\}$ ein passendes Inverses bezüglich \cdot gibt. Alles andere ist schon in Satz 10 enthalten.

Sei also $A = [a]_m \in \mathbb{Z} \setminus \{[0]_{\equiv_m}\}$ beliebig. Wegen $[a]_{\equiv_m} = [\text{mod}(a, m)]_{\equiv_m}$ und $0 \leq \text{mod}(a, m) < m$ können wir annehmen $0 \leq a < m$ (denn wenn das nicht gilt, ersetzen wir einfach a durch $\text{mod}(a, m)$, und dann gilt es). Wegen $A \neq [0]_{\equiv_m}$ gilt $a \neq 0$, also $0 < a < m$. Da m eine Primzahl ist, hat m nur die Teiler $-m, -1, 1, m$. Welche Teiler a hat, wissen wir nicht, aber wegen $a \neq 0$ bilden die Teiler eine Teilmenge von $\{-a, -a+1, \dots, a-1, a\}$. Wegen $0 < a < m$ kann jedenfalls m kein Teiler von a sein. Deshalb muss $\text{gcd}(a, m) = 1$ gelten, und aus dem erweiterten Euklidischen Algorithmus folgt dann, dass es $u, v \in \mathbb{Z}$ gibt mit $1 = au + mv$. Dann aber gilt

$$[1]_{\equiv_m} = [au + mv]_{\equiv_m} = [a]_{\equiv_m} [u]_{\equiv_m} + \underbrace{[mv]_{\equiv_m}}_{=[0]_{\equiv_m}} = [a]_{\equiv_m} [u]_{\equiv_m}.$$

Wegen Teil 4 von Satz 10 und $[0]_{\equiv_m} \neq [1]_{\equiv_m}$ muss $[u]_{\equiv_m} \neq [0]_{\equiv_m}$ gelten, das Element $[u]_{\equiv_m}$ gehört also zu $\mathbb{Z}_m \setminus \{[0]_{\equiv_m}\}$ und ist das gesuchte multiplikative Inverse von $[a]_{\equiv_m}$. ■

Beispiel.

1. Eine Anwendung von Addition und Multiplikation in \mathbb{Z}_m ist die Berechnung von Zufallszahlen. Natürlich kann man „echte“ Zufallszahlen nur mit spezieller Hardware bekommen, die z.B. Quanteneffekte ausnutzt. Für manche Anwendungen, insbesondere in der Kryptologie, ist das auch nötig. Für andere Anwendungen genügt es, wenn statt einer echten Folge von Zufallszahlen eine Folge von Zahlen verwendet, für die ein Zusammenhang nicht ohne weiteres erkennbar ist. Man spricht dann von *Pseudozufallszahlen*. Solche Folgen verwendet man zum Beispiel zum Testen von Software.

Eine einfache Art, eine Zahlenfolge zu generieren, ist durch eine Funktion $f: \mathbb{Z}_m \rightarrow \mathbb{Z}_m$ und einen Startwert (engl. *seed*) $a_0 \in \mathbb{Z}_m$. Man definiert dann $a_1 = f(a_0)$, $a_2 = f(a_1)$, $a_3 = f(a_2)$, usw. Es ist klar, dass sich eine solche Funktion nach spätestens m Schritten wiederholen muss, weil \mathbb{Z}_m nur m Elemente hat, also spätestens nach m Schritten ein Element auftauchen muss, das schon einmal da gewesen ist. Da jede Zahl in der Folge nur von ihrem unmittelbaren Vorgänger abhängt, muss sich die Folge dann wiederholen.

Ein periodisches Verhalten möchte man bei einer Zufallszahlenfolge eher nicht beobachten. Man wählt deshalb m viel größer als die Anzahl der Zufallszahlen, die man braucht. Man muss dann aber noch berücksichtigen, dass die Periodizität der Folge zwar nicht größer als m sein kann, aber durchaus kleiner. Man wird deshalb die Funktion f so wählen wollen, dass die maximal mögliche Periodizität erreicht wird.

Für die meisten Funktion wird die Periodizität enttäuschend klein sein. Man kann aber zeigen, mit einer Funktion der Form

$$f: \mathbb{Z}_m \rightarrow \mathbb{Z}_m, \quad f(X) = [u]_{\equiv m} \cdot X + [v]_{\equiv m}$$

die maximale Periodizität m erreicht wird, wenn $m \in \mathbb{Z}$ und $u, v \in \{0, \dots, m-1\}$ so gewählt werden, dass gilt: (1) $\gcd(m, v) = 1$, (2) $p \mid u-1$ für jede Primzahl p mit $p \mid m$, (3) wenn $4 \mid m$, dann auch $4 \mid u-1$.

Für $m = 1024$, $u = 101$, $v = 97$ und den Anfangswert $a_0 = 0$ erhält man zum Beispiel

0, 97, 678, 991, 860, 941, 930, 843, 248, 569, 222, 1015, 212, 5, 602, 483, 752, 273, ...

Das ist gar nicht so schlecht. Wählt man dagegen $u = 100$ statt $u = 101$, wodurch die zweite Bedingung verletzt wird, so bekommt man

0, 97, 581, 853, 405, 661, 661, 661, 661, 661, 661, 661, 661, 661, 661, 661, 661, 661, ...

Das ist nicht brauchbar.

2. Dass die Addition und Multiplikation die Elemente von \mathbb{Z}_m heftig durcheinanderwirbeln, nutzt man auch in der Kryptologie aus. Wir skizzieren hier als ein Beispiel das RSA-Verfahren (benannt nach seinen Erfindern Rivest, Shamir, Adleman). Dabei wird die zu übertragende Nachricht als Element von \mathbb{Z}_m codiert. Die allgemeine Idee des Verfahrens besteht darin, sich eine bijektive Funktion $f: \mathbb{Z}_m \rightarrow \mathbb{Z}_m$ zu konstruieren, für die man auch die Umkehrfunktion $f^{-1}: \mathbb{Z}_m \rightarrow \mathbb{Z}_m$ kennt, für die diese Umkehrfunktion aber nur sehr schwer zu bestimmen ist, wenn man nur f kennt. Man kann dann die Funktion f als sogenannten *öffentlichen Schlüssel* (engl. *public key*) öffentlich bekannt machen.

Die Umkehrfunktion f^{-1} ist der *geheime Schlüssel* (engl. *private key*), den man unter Verschluss hält.

Wenn Alice dann die Nachricht $X \in \mathbb{Z}_m$ an Bob verschicken will, berechnet sie $f(X)$ mit dem öffentlich bekannten Schlüssel f von Bob und schickt ihm diesen Wert. Nur Bob hat den passenden privaten Schlüssel f^{-1} , mit dem er die Nachricht $X = f^{-1}(f(X))$ entschlüsseln kann.

Beim RSA-Verfahren konstruiert man ein Schlüsselpaar f, f^{-1} wie folgt.

- Wähle zwei (große) Primzahlen p, q mit $p \neq q$
- Berechne $m = pq$ und $\phi = (p - 1)(q - 1)$
- Wähle $e \in \{2, 3, \dots, \phi - 1\}$ mit $\gcd(e, \phi) = 1$
- Berechne $d \in \{1, \dots, \phi - 1\}$ so dass $de \equiv_{\phi} 1$
- Der öffentliche Schlüssel ist dann die Funktion $f: \mathbb{Z}_m \rightarrow \mathbb{Z}_m, f(X) = X^e$
- Der geheime Schlüssel ist die Funktion $f^{-1}: \mathbb{Z}_m \rightarrow \mathbb{Z}_m, f^{-1}(X) = X^d$.

Man veröffentlicht also die beiden Zahlen m und e und hält d geheim. Ohne die Kenntnis der beiden Faktoren p, q von m ist nicht klar, wie ein Angreifer sich d verschaffen könnte. Das Verfahren beruht also auf der Annahme, dass es keinen effizienten Faktorisierungsalgorithmus für ganze Zahlen gibt. Nach allem, was wir wissen, ist das eine berechnete Annahme.

Es ist nicht ganz offensichtlich, dass die oben beschriebene Funktion f^{-1} tatsächlich die Umkehrfunktion von f ist. Um das zu zeigen, muss man auf ein Resultat aus der Zahlentheorie zurückgreifen, das besagt, dass $X^{(p-1)(q-1)} = X$ für alle $X \in \mathbb{Z}_{pq}$ und alle Primzahlen p, q mit $p \neq q$ gilt. Die Bedingung $de \equiv_{\phi} 1$ bedeutet, dass es eine Zahl $k \in \mathbb{Z}$ mit $de = 1 + k\phi = 1 + k(p - 1)(q - 1)$ gibt. Für alle $X \in \mathbb{Z}_{pq}$ gilt deshalb

$$f^{-1}(f(X)) = (X^e)^d = X^{de} = X^{1+k(p-1)(q-1)} = X \underbrace{(X^{(p-1)(q-1)})^k}_{=1} = X,$$

wie gefordert.

Es gilt dann übrigens auch $f(f^{-1}(X)) = X$ für alle $X \in \mathbb{Z}_m$. Damit kann man eine elektronische Signatur konstruieren. Wenn z.B. Bob eine Nachricht $X \in \mathbb{Z}_m$ an Alice schicken will, und Alice überprüfen können soll, dass die Nachricht wirklich von Bob kommt, kann er ihr zusätzlich zu X den Wert $f^{-1}(X)$ schicken. Alice kennt den öffentlichen Schlüssel f von Bob und kann damit $f(f^{-1}(X))$ berechnen und mit X vergleichen. Wenn die beiden Elemente übereinstimmen, ist die Nachricht ziemlich sicher von Bob, denn niemand außer ihm kann niemand $f^{-1}(X)$ berechnen.

3. Ein dritter Anwendungsbereich für modulare Arithmetik ist der Entwurf von effizienten Algorithmen. Die Idee dabei ist, eine Rechnung in \mathbb{Z} auf eine Rechnung in \mathbb{Z}_m zurückzuführen, für ein geeignet gewähltes $m \in \mathbb{Z}$. Da \mathbb{Z}_m nur m Elemente hat, können alle Zwischenergebnisse der Rechnung durch die Zahlen $\{0, 1, \dots, m - 1\}$ dargestellt werden, während bei einer Rechnung in \mathbb{Z} die Zwischenergebnisse unter Umständen viel größer werden können.

Beispiel: nehmen wir als einfaches Beispiel an, wir sollen die Zahl $(2^9 - 1)(3^9 - 1) - 6^9 \in \mathbb{Z}$ berechnen (ohne den Ausdruck vorher zu vereinfachen). Direkte Rechnung in \mathbb{Z} liefert

$$\begin{array}{r} \underbrace{(2^9 - 1)}_{512} \underbrace{(3^9 - 1)}_{19683} - \underbrace{6^9}_{10077696} = -20194. \\ \hline \underbrace{\quad\quad\quad}_{511} \quad \underbrace{\quad\quad\quad}_{19682} \\ \hline \underbrace{\quad\quad\quad}_{10057502} \\ \hline \underbrace{\quad\quad\quad}_{-20194} \end{array}$$

Man sieht, dass schon in diesem einfachen Beispiel einige Zwischenergebnisse deutlich länger sind als das Endergebnis. In realistischen Situation fällt der Unterschied oft wesentlich dramatischer aus. Wenn wir dieselbe Rechnung in \mathbb{Z}_{50000} durchführen, werden die Zwischenergebnisse am Wachstum gehindert:

$$\begin{array}{r} \underbrace{(2^9 - 1)}_{512} \underbrace{(3^9 - 1)}_{19683} - \underbrace{6^9}_{27696} \equiv_{50000} 29806. \\ \hline \underbrace{\quad\quad\quad}_{511} \quad \underbrace{\quad\quad\quad}_{19682} \\ \hline \underbrace{\quad\quad\quad}_{7502} \\ \hline \underbrace{\quad\quad\quad}_{29806} \end{array}$$

Daraus folgt

$$\begin{aligned} [(2^9 - 1)(3^9 - 1) - 6^9]_{\equiv_{50000}} &= [29806]_{\equiv_{50000}} \\ &= \{29806 + 50000k : k \in \mathbb{Z}\} \\ &= \{\dots, -70194, -20194, 29806, 79806, 129806, \dots\}, \end{aligned}$$

d.h. wir bekommen als Ergebnis der Rechnung in \mathbb{Z}_m genau die Äquivalenzklasse, die das Ergebnis der Rechnung in \mathbb{Z} enthält. Wenn man jetzt vielleicht noch weiß, dass das Ergebnis in \mathbb{Z} im Betrag kleiner als 25000 ist (z.B. weil man die Rechnung schon einmal mit grob gerundeten Näherungszahlen durchgeführt hat), dann kann es sich nur noch um -20194 handeln.

Wenn wir also wissen (oder erwarten), dass für das Ergebnis $x \in \mathbb{Z}$ gilt $|x| < M$, dann kann man die Rechnung statt in \mathbb{Z} in \mathbb{Z}_m für ein beliebiges $m > 2M$ durchführen, weil dann $[x]_m \cap \{-M, \dots, M\} = \{x\}$ gilt, d.h. die Äquivalenzklasse des Ergebnisses enthält nur ein einziges Element, das auch die Abschätzung erfüllt, und dieses muss dann die Zahl sein, die bei einer direkten Rechnung in \mathbb{Z} herauskommen würde. Die Größe der Zwischenergebnisse der Rechnung in \mathbb{Z} sind irrelevant.

Jetzt könnte es noch sein, dass das kleinstmögliche m , für das wir das Ergebnis aus der Äquivalenzklasse direkt ablesen können, so groß ist, dass wir in \mathbb{Z}_m auch nicht rechnen möchten. In diesem Fall kann man die Rechnung für mehrere kleine Zahlen m_1, \dots, m_k durchführen. Man bekommt dann die Äquivalenzklassen des Ergebnisses bezüglich $\equiv_{m_1}, \dots, \equiv_{m_k}$. Jede dieser Klassen ist eine Teilmenge von \mathbb{Z} , und die gesuchte Zahl muss im Schnitt dieser Klassen liegen.

Im Beispiel von oben können wir z.B. $m_1 = 35$, $m_2 = 36$, $m_3 = 37$ nehmen. Aus

$$(2^9 - 1)(3^9 - 1) - 6^9 \equiv_{35} 1$$

$$(2^9 - 1)(3^9 - 1) - 6^9 \equiv_{36} 2$$

$$(2^9 - 1)(3^9 - 1) - 6^9 \equiv_{37} 8$$

folgt

$$\begin{aligned} (2^9 - 1)(3^9 - 1) - 6^9 &\in [1]_{\equiv_{35}} \cap [2]_{\equiv_{36}} \cap [8]_{\equiv_{37}} \\ &= \{\dots, -20229, -20194, -20159, \dots, -34, 1, 36, \dots\} \\ &\cap \{\dots, -20230, -20194, -20158, \dots, -34, 2, 37, \dots\} \\ &\cap \{\dots, -20231, -20194, -20157, \dots, -31, 8, 45, \dots\}. \end{aligned}$$

Beachten Sie, dass die drei Rechnungen in \mathbb{Z}_{35} , \mathbb{Z}_{36} , \mathbb{Z}_{37} völlig unabhängig voneinander sind und deshalb auch zur gleichen Zeit parallel auf mehreren Rechnern durchgeführt werden können.

Tatsächlich ist -20194 die Zahl mit dem kleinsten Betrag, die im Schnitt dieser drei Klassen liegt. Wie man einen solchen Schnitt berechnet, ist im folgenden Satz erklärt.

Satz 12. (Chinesischer Restsatz) Seien $m_1, m_2 \in \mathbb{Z}$ mit $\gcd(m_1, m_2) = 1$, und seien $u_1, u_2 \in \mathbb{Z}$. Weiter seien $q_1, q_2 \in \mathbb{Z}$ so, dass $q_1 m_1 + q_2 m_2 = 1$. Dann gilt

$$[u_1]_{\equiv_{m_1}} \cap [u_2]_{\equiv_{m_2}} = [u_1 + m_1 \operatorname{mod}(q_1(u_2 - u_1), m_2)]_{\equiv_{m_1 m_2}}.$$

Beweis. Zunächst gilt

$$\begin{aligned} u_1 + m_1 \operatorname{mod}(q_1(u_2 - u_1), m_2) &\equiv_{m_1 m_2} u_1 + m_1 q_1(u_2 - u_1) \\ &= (1 - m_1 q_1)u_1 + m_1 q_1 u_2 \\ &= m_2 q_2 u_1 + m_1 q_1 u_2. \end{aligned}$$

Wir zeigen deshalb $[u_1]_{\equiv_{m_1}} \cap [u_2]_{\equiv_{m_2}} = [m_2 q_2 u_1 + m_1 q_1 u_2]_{\equiv_{m_1 m_2}}$.

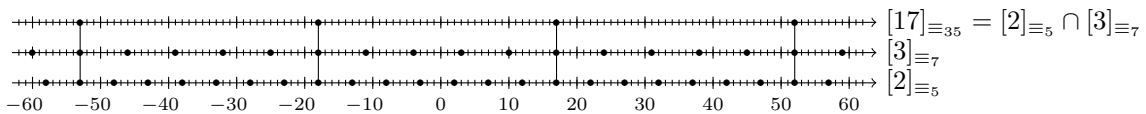
„ \subseteq “ Ist $x \in [u_1]_{\equiv_{m_1}} \cap [u_2]_{\equiv_{m_2}}$, so gilt $x = u_1 + p_1 m_1 = u_2 + p_2 m_2$ für gewisse $p_1, p_2 \in \mathbb{Z}$. Dann gilt

$$\begin{aligned} x &= (m_2 q_2 + m_1 q_1)x \\ &= m_2 q_2(u_1 + p_1 m_1) + m_1 q_1(u_2 + p_2 m_2) \\ &= m_2 q_2 u_1 + m_1 q_1 u_2 + m_1 m_2(p_1 + p_2) \in [m_2 q_2 u_1 + m_1 q_1 u_2]_{\equiv_{m_1 m_2}}. \end{aligned}$$

„ \supseteq “ Ist $x \in [m_2 q_2 u_1 + m_1 q_1 u_2]_{\equiv_{m_1 m_2}}$, etwa $x = m_2 q_2 u_1 + m_1 q_1 u_2 + p m_1 m_2$ für ein $p \in \mathbb{Z}$, so gilt

$$\begin{aligned} x &= (1 - m_1 q_1)u_1 + m_1 q_1 u_2 + p m_1 m_2 \equiv_{m_1} u_1 \\ x &= m_2 q_2 u_1 + (1 - m_2 q_2)u_2 + p m_1 m_2 \equiv_{m_2} u_2, \end{aligned}$$

also $x \in [u_1]_{\equiv_{m_1}}$ und $x \in [u_2]_{\equiv_{m_2}}$, wie behauptet. ■



Beispiel. Wir bestimmen $[3]_{\equiv 7} \cap [2]_{\equiv 5}$. In den Bezeichnungen des Satzes gilt also $u_1 = 3$, $u_2 = 2$, $m_1 = 7$, $m_2 = 5$. Mit dem erweiterten euklidischen Algorithmus (oder durch probieren) findet man $(-2) \cdot 7 + 3 \cdot 5 = 1$, also ist $q_1 = -2$ und $q_2 = 3$. Mit der Formel des Satzes erhält man dann

$$[3]_{\equiv 7} \cap [2]_{\equiv 5} = [3 + 7 \operatorname{mod}((-2)(2 - 3), 5)]_{\equiv 35} = [17]_{\equiv 35}.$$

8 Kombinatorik

9 Summen und Rekurrenzen

Index

- abelian*, 37
- abelsch, 37
- abschreiben, 27
- Addition, 48
- Adleman, 54
- Algorithmus, 11
- Alphabet, 8
- Analysis, 14, 48
- Antisymmetrie, 18
- Äquivalenzklasse, 19
- Äquivalenzrelation, 18
- associative*, 37
- assoziativ, 37
- Atom, 27
- Aufzug, 28
- Ausdruck, 5, 29
- automorphism group*, 43
- Automorphismengroup, 43

- Bahn, 45
- Baum, 25
- beliebig aber fest, 6
- Benutzeroberfläche, 26
- Beweis, 6
- bijective*, 13
- Bild, 41
- Binärbaum, 29
- binary tree*, 29
- bipartit, 26
- bipartite*, 26
- Blatt, 26
- Bundesland, 27

- Ceiling-Funktion, 48
- chain*, 24
- child*, 26
- Chinesischer Restsatz, 57
- closed path*, 32
- commutative*, 37

- Compilerbau, 30
- composition*, 14
- concatenation*, 8
- connected*, 32
- connected component*, 32
- cycle*, 24, 32, 35

- Dateisystem, 26
- disjoint*, 35
- disjunkt, 35
- diskretisieren, 28
- divisibility*, 17
- Division mit Rest, 49

- Ecke, 45
- edge*, 24
- effiziente Algorithmen, 55
- Einbettung, 31
- elektronische Signatur, 55
- Element, 4
- element*, 4
- embedding*, 31
- endlicher Automat, 30
- Endlosschleife, 11
- Endzustand, 29
- equivalence relation*, 18
- erweiterter euklidischer Algorithmus, 51
- erzeugen, 40
- euklidischer Algorithmus, 50
- expression*, 5
- expression swell, 48

- Fehler, 11
- Figur, 26
- final state*, 29
- finite automaton*, 30
- fixed point*, 35
- Fixpunkt, 35
- Floor-Funktion, 48
- formale Sprache, 30

Formel, 5
formula, 5
function, 9
 Funktion, 9

ganze Zahlen, 48
 Gegenteil, 5
 geheimer Schlüssel, 54
 gelabelter Graph, 29
 Gelenk, 27
 Genauigkeit, 48
generated, 40
 geschlossener Pfad, 32
 Geschwindigkeit, 39
 gewichteter Graph, 29
 Gitter, 25
 größter gemeinsamer Teiler, 50
 Graph, 24
graph, 24
graph homomorphism, 31
 Graphenhomomorphismus, 31
greatest common divisor, 50
grid, 25
 Großschreibung, 17
group, 37
group action, 44
 Gruppe, 37
 Gruppenoperation, 44

Halbordnung, 18
 hamiltonscher Pfad, 35
handshake graph, 27
 Hindernis, 28
 Hintereinanderausführung, 14
 Homomorphiesatz, 22
 Homomorphismus, 31, 41
hyper cube, 25
 Hyperkubus, 25

image, 41
 Implikation, 17
 Induktion, 7
initial state, 29
injective, 13
 Inverse, 15, 37
inverse, 15, 37
invertible, 37
 Invertierbarkeit, 37

isomorph, 31, 41
isomorphic, 31, 41
isomorphism, 41
 Isomorphismus, 41

Kante, 24, 27
 Kern, 41
kernel, 41
 Kette, 24, 32
 Kind, 26
 Klausurteilnehmer, 27
 Kleinschreibung, 17
 Knochen, 27
 Knoten, 24
 kommutativ, 37
 Komposition, 14
 Konkatenation, 8
 Konstruktionsregel, 25
 Kryptologie, 54
 kürzester Pfad, 35

Länge, 35
 Länge, 8
leaf, 26
 leere Menge, 4
 leerer Graph, 25
 leeres Wort, 8
 Lehrveranstaltung, 27
length, 35
 lexikographisch, 17

mag, 17
 Mehrfachbindung, 30
 Menge, 4
 Menschen, 17
 Menü, 26
 modulo, 49
 Molekül, 27
 Multigraph, 29
 Multiplikation, 48

Negation, 5
neutral element, 37
 Neutralelement, 37
 Numerik, 48

öffentlicher Schlüssel, 54
operation, 37

orbit, 45
order, 18
 Ordnungsrelation, 18
 orientierter Graph, 28

 Paar, 7
 paralleler Algorithmus, 57
path, 32
 Periodizität, 54
 Permutation, 35
permutation, 35
 Petersen-Graph, 25
 Pfad, 32
 Potenzmenge, 4
power set, 4
private key, 54
 Pseudozufallszahlen, 54
public key, 54

quantifier, 5
 Quantor, 5
 Quelle, 32

 reelle Zahlen, 48
 Reflexivität, 18
 Relation, 17
relation, 17
 Relativitätstheorie, 39
 repräsentantenunabhängig, 21
 Restsatz, 57
 Rivest, 54
 Roboter, 28
root, 26
 RSA, 54
 Rundungsfehler, 48

 Schach, 27
 Schaltkreis, 28
 Schwefelsäure, 30
seed, 54
set, 4
 Shamir, 54
 Sink, 32
sink, 32
 Skellet, 27
 Softwaresystem, 28
source, 32
 Stabilisator, 45

stabilizer, 45
 Startwert, 54
 Startzustand, 29
state, 28
 Straßennetz, 28
string, 8
 Struktur, 30
subgraph, 31
subgroup, 40
subset, 4
subtree, 26
subword, 17
surjective, 13
 Symmetrie, 18, 43
 Symmetriegruppe, 43
symmetry group, 43
syntax tree, 29
 Syntaxbaum, 29
 Szene, 26

 Teilbarkeit, 17
 Teilgraph, 31
 Teilmenge, 4
 Teilwort, 17
 Testen, 54
 Theater, 26
 theoretische Informatik, 30
 Torus, 25
torus, 25
total order, 18
 Totalordnung, 18
 Transitivität, 18
 Transposition, 35
transposition, 35
tree, 25
 Tupel, 8

 Uhr, 49
 Umkehrfunktion, 15
undirected, 24
 ungerichtet, 24
 Unterbaum, 26
 Untergraph, 31
 Untergruppe, 40

 Verhalten, 28
 Verkettung, 14
 Verknüpfung, 37

Verknüpfungstabelle, 38

vertex, 24

Verzeichnis, 26

vollständige Induktion, 7

Webseite, 27

wohldefiniert, 21

Wort, 8

Worthomomorphismus, 11

Wurzel, 26

WWW, 27

Zahlentheorie, 55

Zeichenkette, 8

Ziffernblatt, 49

Zufallszahlen, 54

zusammenhängend, 32

Zusammenhangskomponente, 32

Zustand, 28

Zyklus, 24, 32, 35