Diskrete Strukturen

Manuel Kauers

Institut für Algebra
Johannes Kepler Universität

Version: 16. Dezember 2016

Inhalt

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 $0, 1, 2, \ldots$.
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 $p\colon A \to \{\text{True}, \text{False}\}$ 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.

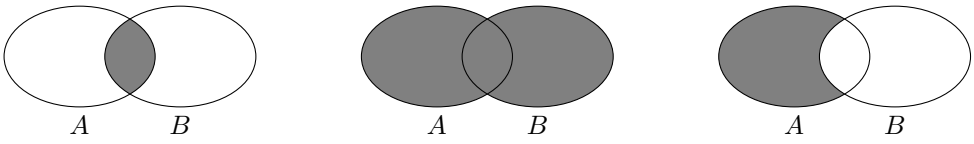
1. $\{\, x \in \mathbb{Z} : \text{die letzte Ziffer von } x \text{ ist } 3 \,\} = \{\ldots, -23, -13, -3, 3, 13, 23, \ldots\}$
2. $\{\, p \in \mathbb{N} : p \text{ ist eine Primzahl} \,\} = \{2, 3, 5, 7, 11, 13, 17, \ldots\}$
3. $\{\, x \in \mathbb{R} : x > 0 \,\}$ ist die Menge aller positiven reellen 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 $f\colon A \to B$ (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 $\{\, x^2 : x \in \mathbb{N}, n \text{ prim} \,\} = \{4, 9, 25, 49, 121, 169, \ldots\}$.

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

$$\begin{aligned}
&\forall\, x : \big(x \in A \cap B \iff x \in A \wedge x \in B\big) \\
&\forall\, x : \big(x \in A \cup B \iff x \in A \vee x \in B\big) \\
&\forall\, x : \big(x \in A \setminus B \iff x \in A \wedge x \notin B\big)
\end{aligned}$$



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

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

- $\neg\big(\forall\, x : p(x)\big) \iff \exists\, x : \neg p(x)$
- $\neg(A \Rightarrow B) \iff (A \wedge \neg B)$
- $\neg(A \wedge B) \iff (\neg A) \vee (\neg B)$, 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 $x = \ldots$“. 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:

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

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 $n \to n+1$: 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. $\Omega = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$. 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 $A_1, \ldots, A_n$ Mengen, so ist $A_1 \times \cdots \times A_n$ die Menge aller *Tupel* $(a_1, \ldots, a_n)$ mit $a_1 \in A_1, \ldots, a_n \in A_n$. Man schreibt A^n statt $A \times A \times \cdots \times A$. 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

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

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 $(a_1, a_2, \ldots, a_n)$ schreibt man in diesem Zusammenhang einfach $a_1 a_2 \cdots a_n$.

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

Definition 2. Seien $\omega = a_1 a_2 \ldots a_n$, $\chi = b_1 b_2 \ldots b_m$ zwei Wörter über einem Alphabet Ω. Dann heißt $\omega \circ \chi := a_1 a_2 \ldots a_n b_1 b_2 \ldots b_m \in \Omega^*$ 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 $\mathtt{abcca} = \mathtt{a} \circ \mathtt{b} \circ \mathtt{c} \circ \mathtt{c} \circ \mathtt{a}$.
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:

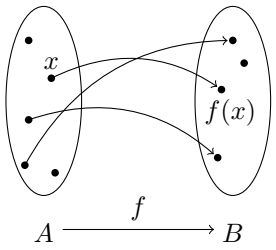
- Kann man mit Wörtern rechnen wie mit Zahlen?
- Wenn man Funktionen auf Wörtern definiert, wie verhalten sich diese zur Konkatenation?
- Wie kann man Gleichungen für Wörter lösen?
- Was, wenn man bestimmte Wörter als „im wesentlichen gleich“ auffassen will (z.B. gleich bis auf Groß-Kleinschreibung)?

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



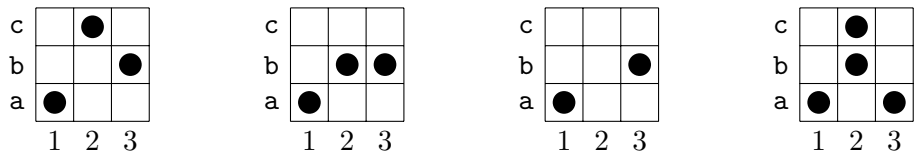
Definition 3. Seien A, B zwei Mengen. Eine Teilmenge $f \subseteq A \times B$ heißt *Funktion* (engl. *function*) von A nach B, falls gilt:

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 $f\colon A \to B$ 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.

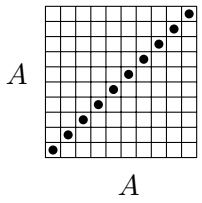
Beispiel.

1. Seien $A = \{1, 2, 3\}$ und $B = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$. 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?

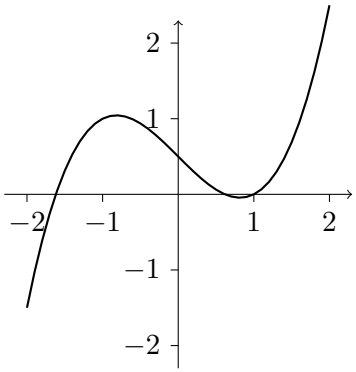


   Die ersten beiden sind Funktionen, die anderen nicht. Die dritte verletzt die erste Bedingung der Definition und die vierte die zweite Bedingung der Definition. Nur wenn beide Bedingungen erfüllt sind, handelt es sich um eine Funktion.

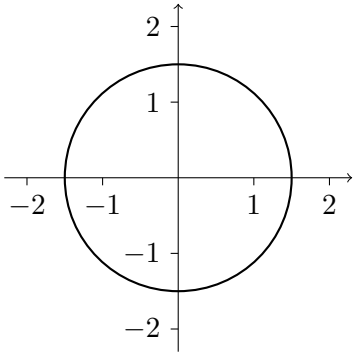
2. Sei A irgendeine Menge. Dann ist $\mathrm{id}_A\colon A \to A$, $\mathrm{id}_A(x) = x$ eine Funktion, die sogenannte *Identitätsfunktion*.



3. Bei $f\colon \mathbb{R} \to \mathbb{R}$, $f(x) = \frac{1}{2}x^3 - x + \frac{1}{2}$ 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. $f\colon \Omega^* \to \mathbb{N}$, $f(\omega) = |\omega|$ ist eine Funktion.

5. Manche Funktionen lassen sich am leichtesten durch eine Fallunterscheidung beschreiben, z.B. die Funktion

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

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

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

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

Man sollte in solchen Fällen erst gar nicht die Funktionsnotation $f\colon A \to B$ 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

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

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

gültige Funktionsdefinitionen.

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

Dann ist auch $F\colon \Omega^* \to \bar{\Omega}^*$ 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 ≠ 0 ist:
6     falls x ungerade ist:
7        setze y = y + 1 und x = x − 1
8     setze x = x/2.
9  gib z · y als Ergebnis zurück.
```

Dieser Algorithmus definiert eine Funktion $f\colon \mathbb{Z} \to \mathbb{Z}$. 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).

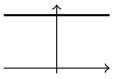
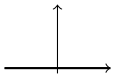
Intuitiv kann man sich eine Funktion grundsätzlich vorstellen als etwas, das einen Input nimmt und dazu einen passenden Output produziert. Aber Vorsicht: nicht jede (mathematische) Funktion lässt sich durch einen Algorithmus berechnen. Es gibt sogenannte

„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

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

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

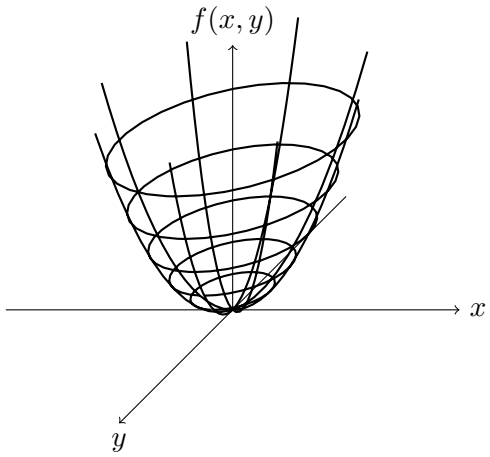
$g\colon \mathbb{R} \to \mathbb{R}, g(x) = 1$  oder $h\colon \mathbb{R} \to \mathbb{R}, h(x) = 0$ 

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

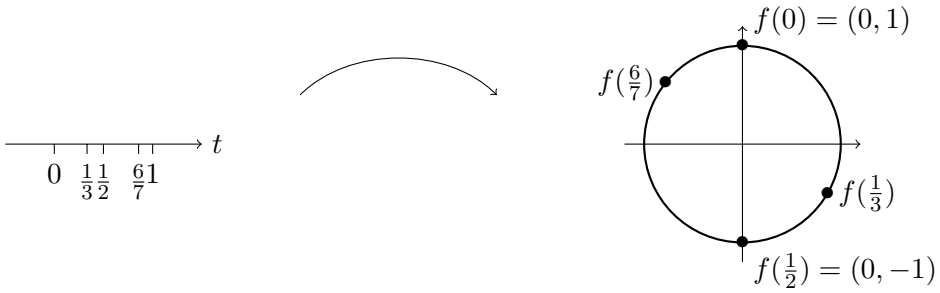
$$f\colon \mathbb{R}^2 \to \mathbb{R}, \quad f(x, y) = x^2 + y^2,$$

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 $f\colon [0, 1] \to \mathbb{R}^3$ die Flugbahn eines Teilches 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 $f\colon [0,1] \to \mathbb{R}^2$, $f(t) = \big(\sin(2\pi t), \cos(2\pi t)\big)$. Bei dieser Funktion wird der Kreis beginnend beim Punkt $(0,1)$ im Uhrzeigersinn durchlaufen:



Definition 4. Sei $f\colon A \to B$ 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.

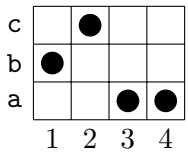
Injektiv bedeutet, dass jedes Element des Bildbereichs *höchstens* einmal getroffen wird. Surjektiv bedeutet, dass jedes Element des Bildbereichs *mindestens* einmal getroffen wird. Bijektiv bedeutet, dass jedes Element des Bildbereichs *genau* einmal getroffen wird. Man vergleiche die beiden Formeln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

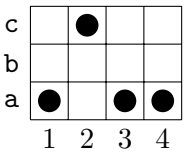
Beispiel.

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

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

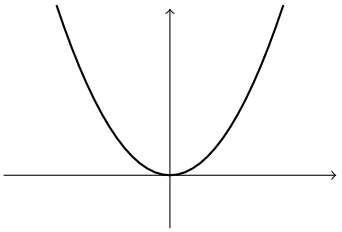
2. $A = \{1,2,3,4\}$, $B = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$. Von den folgenden beiden Funktionen ist die linke surjektiv, die rechte nicht (weil der Wert `b` nicht getroffen wird).



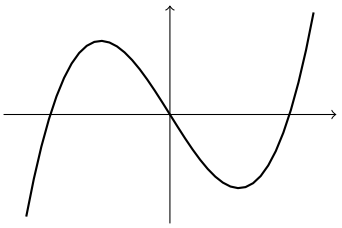


Eine injektive Funktion von A nach B kann es in diesem Fall offenbar nicht geben, weil die Menge A mehr Elemente hat als B.

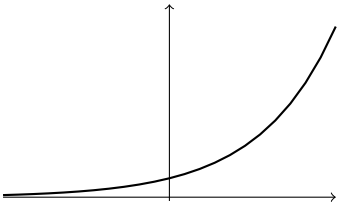
3. Die Funktion $f\colon \mathbb{R} \to \mathbb{R}$, $f(x) = x^2$ ist weder injektiv noch surjektiv. Sie ist nicht injektiv, weil z.B. der Wert 4 des Bildbereichs sowohl von 2 als auch von -2 getroffen wird: $f(-2) = f(2) = 4$. Sie ist nicht surjektiv, weil z.B. der Wert -1 des Bildbereichs überhaupt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



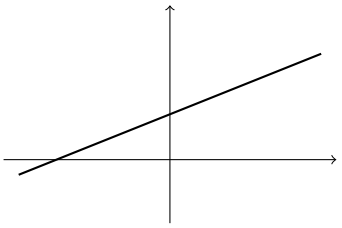
4. Die Funktion $f\colon \mathbb{R} \to \mathbb{R}$, $f(x) = x^3 - 10x$ ist surjektiv, aber nicht injektiv. Dass sie surjektiv ist, beweist man mit Techniken, die man in der Analysis lernt. Dass sie nicht injektiv ist, sieht man daran, dass z.B. der Wert 0 sowohl an der Stelle $x = 0$ als auch an den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



5. Die Funktion $f\colon \mathbb{R} \to \mathbb{R}$, $f(x) = \mathrm{e}^x$ 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 $f\colon \mathbb{R} \to \mathbb{R}$, $f(x) = \frac{1}{3}x + 1$ 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 $f\colon \mathbb{Z} \to \mathbb{Z}$, $f(x) = 3x+2$ 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 1 des Bildbereiches nicht getroffen wird.

Definition 5. Seien $f\colon A \to B$, $g\colon B \to C$ zwei Funktionen. Dann heißt die Funktion $g \circ f\colon A \to C$, $(g \circ f)(x) := g(f(x))$ 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 $f\colon A \to B$ und $g\colon B \to C$ injektive Funktionen und $h = g \circ f$.

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

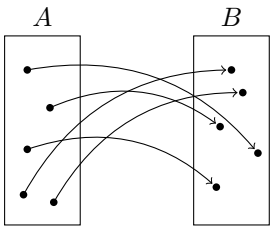
   Seien also $x_1, x_2 \in A$ mit $h(x_1) = h(x_2)$, d. h. $g(f(x_1)) = g(f(x_2))$. Da g nach Annahme injektiv ist, folgt zunächst $f(x_1) = f(x_2)$. Und daraus folgt, da nach Annahme auch f injektiv ist, $x_1 = x_2$, was zu zeigen war.

2., 3. Übung. ∎

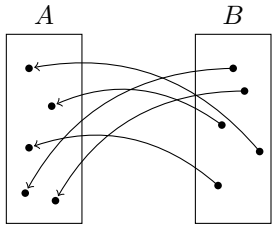
Satz 2. $f\colon A \to B$ ist genau dann bijektiv, wenn es eine Funktion $g\colon B \to A$ gibt mit $g \circ f = \mathrm{id}_A$ und $f \circ g = \mathrm{id}_B$.

Diese Funktion g ist dann eindeutig bestimmt und ihrerseits bijektiv.

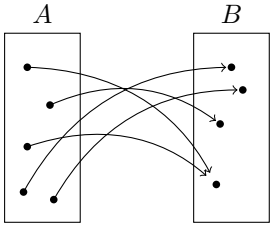
Dieser Satz besagt, dass eine Funktion genau dann bijektiv ist, wenn man durch umkehren der Abbildungsrichtung wieder eine Funktion bekommt. Man nennt die Funktion g deshalb auch die *Umkehrfunktion* oder *Inverse* (engl. *inverse*) von f, Notation: $f^{-1} := g$.



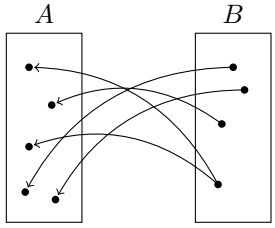
$f\colon A \to B$ ist bijektiv



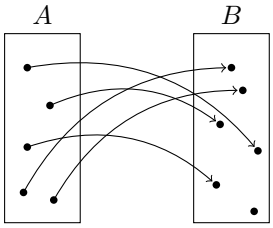
Umkehrfunktion $f^{-1}\colon B \to A$



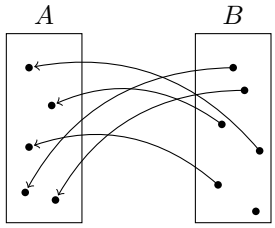
$f\colon A \to B$ ist nicht injektiv



Umkehrung ist keine Funktion!



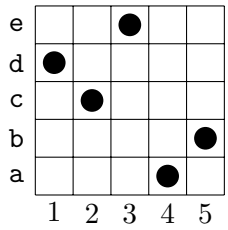
$f\colon A \to B$ ist nicht surjektiv

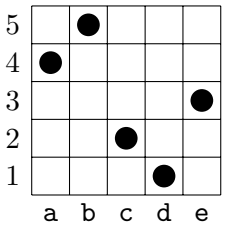


Umkehrung ist keine Funktion!

Beispiel.

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





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

2. Die Funktion $f\colon \mathbb{R} \to \mathbb{R}$, $f(x) = \frac{1}{3}x+1$ ist bijektiv. Die Umkehrfunktion ist $f^{-1}\colon \mathbb{R} \to \mathbb{R}$, $f^{-1}(y) = 3y - 3$. 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 $f\colon \Omega^* \to \Omega^*$, $f(\omega) = \epsilon$ 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 $f\colon \Omega^* \to \Omega^*$ zwar injektiv aber nicht surjektiv ist, dann kann man die modifizierte Funktion

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

   betrachten. Diese ist bijektiv, hat also eine Umkehrfunktion $\tilde{f}\colon B \to \Omega^*$. 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 Paars (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 $x \mathrel{R} y$ und statt $(x, y) \notin R$ schreibt man $x \mathrel{\not R} y$.

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 $3 \mid 9$, $3 \nmid 10$, $5 \nmid 9$, $5 \mid 10$.

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 $\Omega = \{\mathtt{a}, \mathtt{A}, \mathtt{b}, \mathtt{B}, \ldots, \mathtt{z}, \mathtt{Z}\}$. Dann ist

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

   eine Relation auf Ω^*. Es gilt zum Beispiel $\texttt{Aachen} \leq_{\text{lex}} \texttt{Aal} \leq_{\text{lex}} \cdots \leq_{\text{lex}} \texttt{Zwickau}$.

   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 $\texttt{wOrT} \sim \texttt{WoRt} \not\sim \texttt{trow}$.

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. $\leq_{\text{lex}}$ ist eine Totalordnung auf $\{\mathtt{a}, \mathtt{A}, \ldots, \mathtt{z}, \mathtt{Z}\}^*$.

   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

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

   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 $A = \{\square, \blacksquare, \square, \square, \blacksquare, \circ, \bullet, \bullet, \bullet, \triangle, \blacktriangle, \blacktriangle, \triangle\}$. 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 $\Omega = \{\mathtt{a}, \mathtt{A}, \mathtt{b}, \mathtt{B}, \ldots, \mathtt{z}, \mathtt{Z}\}$. 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:

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

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 $A = \{\square, \blacksquare, \blacksquare, \square, \blacksquare, \circ, \bullet, \bullet, \bullet, \triangle, \blacktriangle, \blacktriangle, \triangle\}$ teilen A in folgende Äquivalenzklassen auf:

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

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

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

   Eine weitere Äquivalenzklasse ist

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

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

4. Wir haben

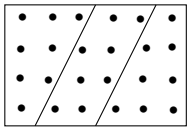
   $$\begin{aligned}
   [0]_{\equiv_3} &= \{\ldots, -3, 0, 3, 6, 9, \ldots\} \\
   [1]_{\equiv_3} &= \{\ldots, -2, 1, 4, 7, 10, \ldots\} \\
   [2]_{\equiv_3} &= \{\ldots, -1, 2, 5, 8, 11, \ldots\} \\
   [3]_{\equiv_3} &= [0]_{\equiv_3} \\
   [4]_{\equiv_3} &= [1]_{\equiv_3}
   \end{aligned}$$

   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. „⇒“ 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:

$$\begin{aligned} z \in [x]_\sim &\overset{\text{Def.}}{\Longrightarrow} x \sim z \\ &\overset{\text{Sym.}}{\Longrightarrow} z \sim x \\ &\overset{\text{Ann. + Trans.}}{\Longrightarrow} z \sim y \\ &\overset{\text{Sym.}}{\Longrightarrow} y \sim z \\ &\overset{\text{Def.}}{\Longrightarrow} z \in [y]_\sim. \end{aligned}$$

   „⇐“ 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. „⇒“ 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 $x \not\sim y$.

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

   „⊇“ 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 $h\colon A/\sim \to B$ zu definieren, ist es verlockend, zunächst von einer Funktion $\tilde{h}\colon A \to B$ 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 $\Omega = \{\mathtt{a}, \mathtt{A}, \ldots, \mathtt{z}, \mathtt{Z}\}$ und die Äquivalenzrelation $\sim$, die Groß-/Kleinschreibung ignoriert.

1. Bei

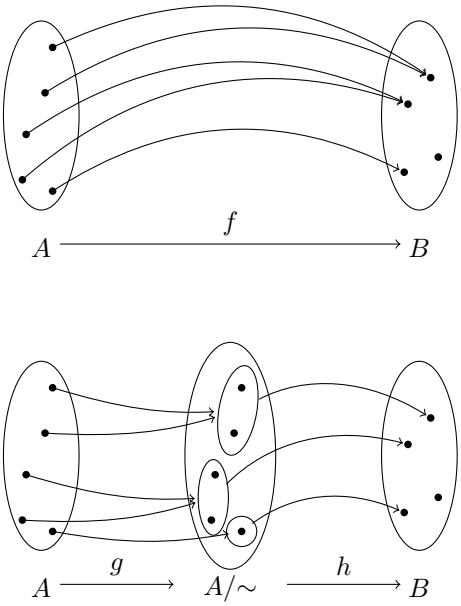
$$h\colon \Omega^*/\sim \setminus \{[\epsilon]_\sim\} \to \Omega, \quad h([\omega]_\sim) := \text{erster Buchstabe von } \omega$$

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

2. Bei $h\colon \Omega^*/\sim \to \mathbb{N}$, $h([\omega]_\sim) := |\omega|$ 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 $f\colon A \to B$ eine Funktion.

1. Durch $x \sim y \iff f(x) = f(y)$ wird eine Äquivalenzrelation auf A definiert.
2. Die Funktion $g\colon A \to A/\sim$, $g(x) = [x]_\sim$ ist surjektiv.
3. Es gibt eine injektive Funktion $h\colon A/\sim \to B$ 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 $g\colon A \to A/\sim$ mit $g(x) = [x]_\sim$ für alle $x \in A$ und definiere die Funktion $h\colon A/\sim \to B$ 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 $[x]_\sim = [y]_\sim$.

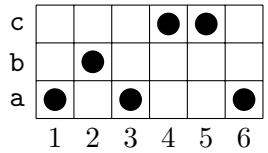
   Es gilt $f = h \circ g$, denn für alle $x \in A$ gilt $h(g(x)) = h([x]_\sim) = f(x)$.

4. Wenn $\bar{h}\colon A/\sim \to B$ 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 $A = \{1, 2, 3, 4, 5, 6\}$, $B = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ und sei f die folgende Funktion:



Dann ist

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

Die Funktionen g und h aus dem Satz sind gegeben durch

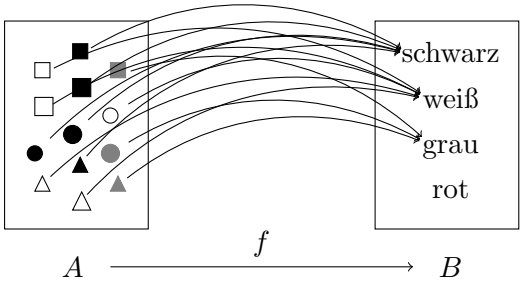
g:

1	2	3	4	5	6
$\{1,3,6\}$	$\{2\}$	$\{1,3,6\}$	$\{4,5\}$	$\{4,5\}$	$\{1,3,6\}$

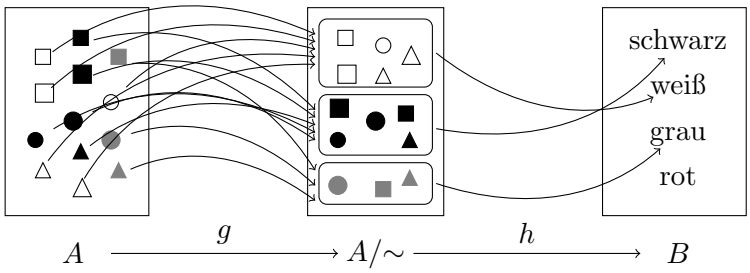
h:

$\{1,3,6\}$	$\{2\}$	$\{4,5\}$
a	b	c

2. Seien $A = \{\square, \blacksquare, \blacksquare, \square, \blacksquare, \circ, \bullet, \bullet, \bullet, \triangle, \blacktriangle, \blacktriangle, \triangle\}$, $B = \{\text{schwarz}, \text{weiß}, \text{grau}, \text{rot}\}$, und sei $f\colon A \to B$ 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.

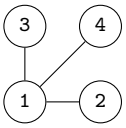
Beispiel.

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:

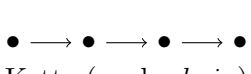


Abgesehen von dem Detail, dass wir V als endliche Menge voraussetzen, bezeichnet der Begriff Graph im wesentlichen dasselbe wie der Begriff Relation. Der neue Begriff ist also streng genommen überflüssig. Wir verwenden ihn hier trotzdem, um dem allgemeinen Sprachgebrauch Rechnung zu tragen. Man spricht 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 modelieren kann.

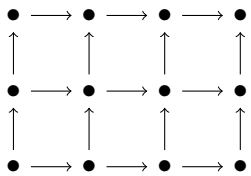
Beispiel.

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

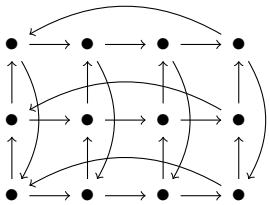


Kette (engl. *chain*)

Zyklus (engl. *cycle*)



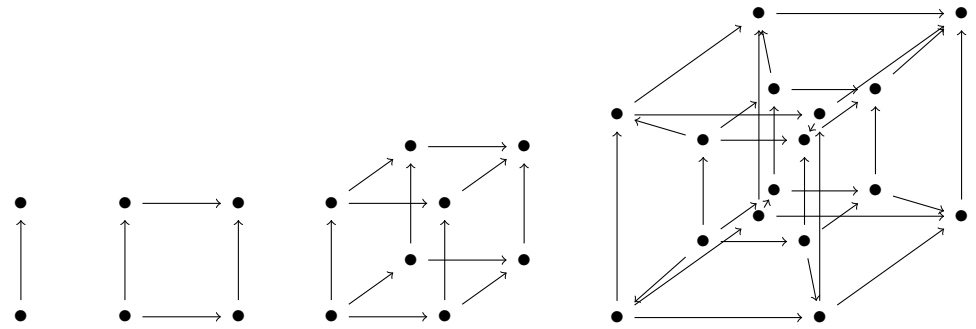
Gitter (engl. *grid*)



Torus (engl. *torus*)

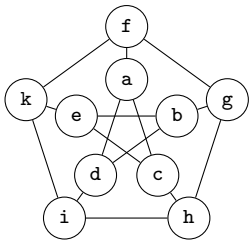
leerer Graph

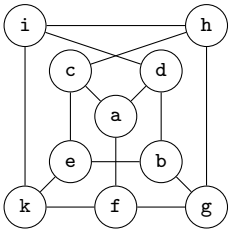
vollständiger Graph

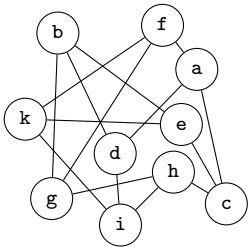


Hyperkubus (engl. *hyper cube*)

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.

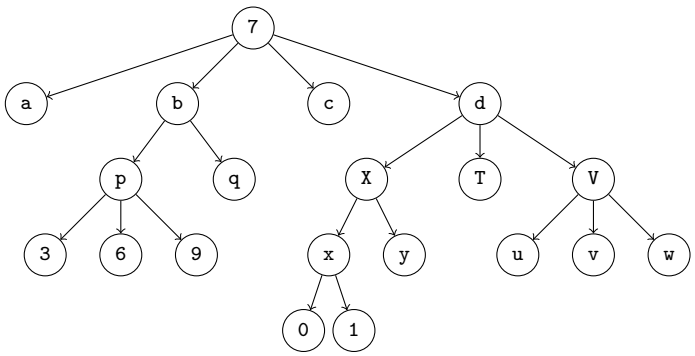
3. Eine besondere Familie von Graphen bilden die *Bäume* (engl. *tree*). Ein Graph ist ein Baum, wenn er sich durch wiederholte Anwendung der folgenden beiden Konstruktionsregeln bilden lässt:

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

$$T = \big(\underbrace{\{v\} \cup V_1 \cup \cdots \cup V_n}_{\text{Knoten}}, \underbrace{\{(v, t_1), \ldots, (v, t_n)\} \cup E_1 \cup \cdots \cup E_n}_{\text{Kanten}} \big)$$

     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.

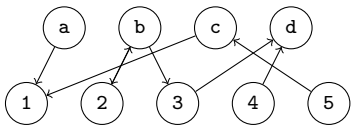


Dateisysteme werden typischerweise als Bäume organisiert. Darin sind Verzeichnisse und Dateien die Knoten, und a ist ein Kind von b wenn a in b liegt. Verzeichnisse die inneren Knoten und Dateien sind die Blätter des Baums.

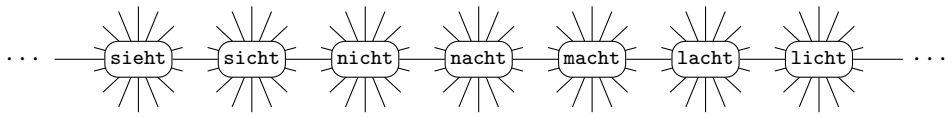
Auch in graphischen Benutzeroberflächen findet man Baumstrukturen, z.B. Menüs mit verschachtelten Untermenüs.

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.

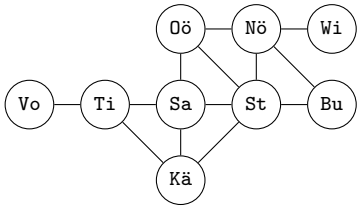
   Einen solchen Graphen nennt man *bipartit* (engl. *bipartite*). Die charakteristische Eigenschaft eines bipartiten Graphen $G = (V, E)$ ist, dass sich die Knotenmenge so in $V = V_1 \cup V_2$ mit $V_1 \cap V_2 = \emptyset$ zerlegen lässt, dass jede Kante $(u, v) \in E$ einen Knoten einer der beiden Teilmengen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

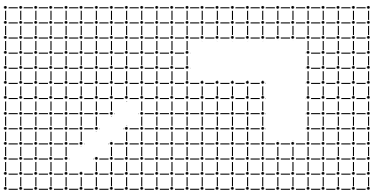
14. Das Schachspiel ist ein Graph, bei dem die Knoten die möglichen Belegungen des Schachbretts mit Figuren sind, und eine Kante von a nach b existiert, wenn ein zulässiger Schachzug die Stellung a in die Stellung b überführt.

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.

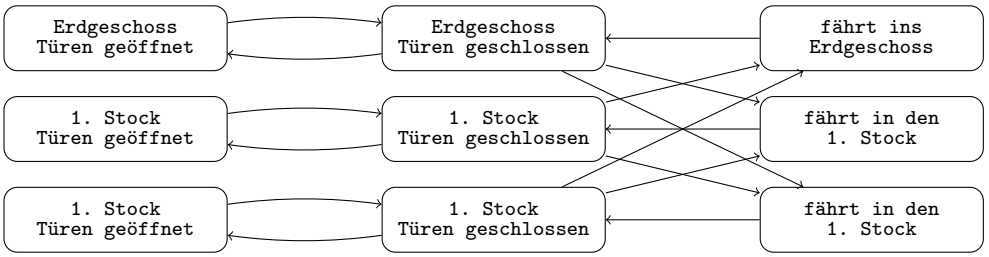
16. Ein Schaltkreis ist in natürlicher Weise ein Graph. Dabei sind die Bauelemente die Knoten und es gibt eine Kante von a nach b, wenn die Bauteile durch eine Leitung verbunden sind.

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 weiterer Kreuzung oder Einmündung vorbeizukommen.

18. Ein Roboter soll ich auf einem freien Feld von einen 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.



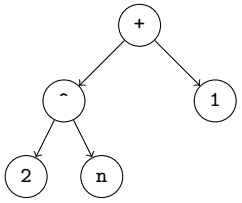
19. Auch Verhalten kann sich unter Umständen durch Graphen beschreiben lassen. Die Knoten bezeichnen dann *Zustände* (engl. *state*), und es gibt eine Kante von a nach b, wenn das System vom Zustand a in den Zustand b welchseln kann. Zum Beispiel zur Beschreibung eines Aufzugs genügt ein solcher Graph:



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.

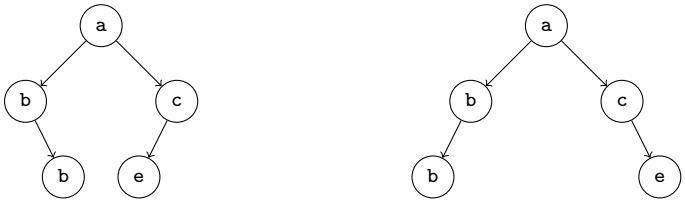
1. *Orientierter Graphen.* Ist $G = (V, E)$ ein Graph, so kann man für jeden Knoten $v \in V$ die Menge $\{\, u \in V : (v, u) \in E \,\}$ aller Knoten betrachten, zu denen von v aus eine Kante führt. Auf dieser Menge kann man eine Ordnung definieren, so dass es Sinn macht, z.B. vom „dritten“ dieser 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 Kontens ^ 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.

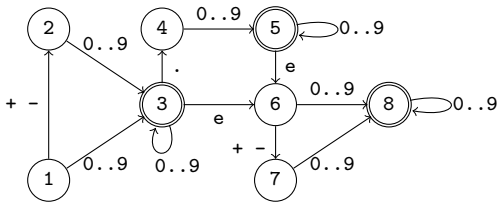
Eine ähnliche Variante sind *Binärbäume* (engl. *binary tree*). Das sind Bäume, bei denen jeder Knoten höchstens zwei Kinder hat, und zwar höchstens einen „linken“ und höchstens einen „rechten“. Die folgenden Binärbäume werden also als verschieden aufgefasst:



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 $w\colon E \to X$, 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, sonder $w(e)$ 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 $\{\texttt{0}, \ldots, \texttt{9}, \texttt{+}, \texttt{-}, \texttt{e}, \texttt{.}\}$ 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 $q\colon V \to Y$ 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: $q\colon \{1,\dots,8\} \to \{\text{final}, \text{non-final}\}$, $q(v) = \text{final}$ für $v \in \{3,5,8\}$ und $q(v) = \text{non-final}$ für $v \in \{1,2,4,6,7\}$).

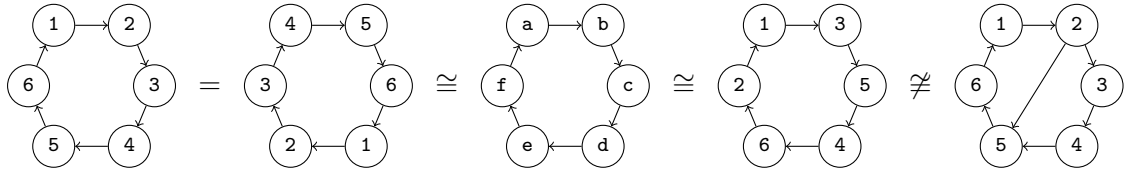
Labels braucht man vor allem dann, wenn man verschiedene Knoten gleich bezeichnen will. Ein Beispiel ist die Beschreibung von Molekülen durch Graphen. Um zum Beispiel die Schwefelsäure (unten links) darzustellen, kann man nicht einfach die Knotenmenge $\{\mathrm{S},\mathrm{O},\mathrm{O},\mathrm{O},\mathrm{O},\mathrm{H},\mathrm{H}\}$ nehmen, weil diese Menge identisch ist mit der Menge $\{\mathrm{S},\mathrm{O},\mathrm{H}\}$, 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: $q\colon V \to \{\mathrm{S},\mathrm{O},\mathrm{H}\}$, $q(1) = \mathrm{S}$, $q(2) = q(3) = q(4) = q(5) = \mathrm{O}$, $q(6) = q(7) = \mathrm{H}$. Darüber hinaus kann man die Mehrfachbindungen des Moleküls durch gewichtete Kanten darstellen: $w\colon E \to \mathbb{N}$, $w((1,2)) = w((2,1)) = w((1,3)) = w((3,1)) = 2$ 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 $h\colon V \to \tilde{V}$ 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 $h\colon V \to \tilde{V}$ heißt *(Graphen-)Homomorphismus* (engl. *graph homomorphism*) von G nach $\tilde{G}$, falls gilt

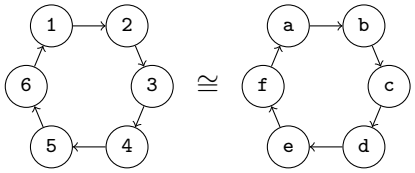
$$\forall\, u,v \in V : (u,v) \in E \Rightarrow (h(u),h(v)) \in \tilde{E}.$$

2. G und $\tilde{G}$ heißen (zueinander) *isomorph* (engl. *isomorphic*), falls es einen bijektiven Graphenhomomorphismus $h\colon V \to \tilde{V}$ gibt, dessen Umkehrfunktion $h^{-1}\colon \tilde{V} \to V$ 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 $h\colon \{1,\dots,6\} \to \{\mathtt{a},\dots,\mathtt{f}\}$ mit folgender Wertetabelle:

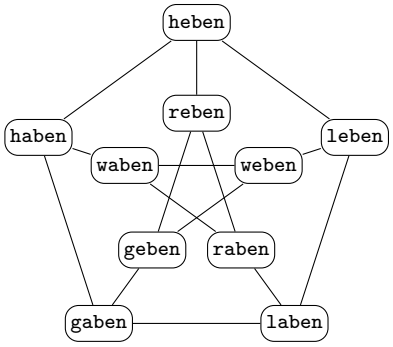
v	1	2	3	4	5	6
$h(v)$	a	b	c	d	e	f

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

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

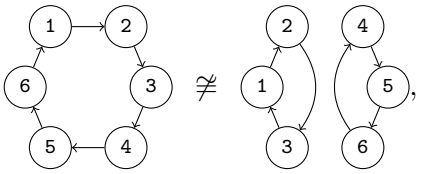
ein Isomorphismus.

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 $P = (\{1, \ldots, n\}, \{(1, 2), (2, 3), \ldots, (n-1, n)\})$ eine *Kette* der Länge $n - 1$.

2. Sei P eine Kette der Länge n und $h\colon \{1, \ldots, n\} \to V$ ein Graphhomomorphismus von P nach G. Dann heißt $(h(1), h(2), \ldots, h(n))$ 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.

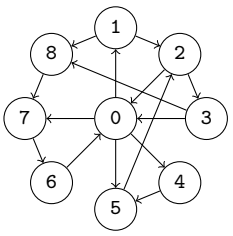
Sei G nun ein ungerichteter Graph (d.h. es gelte $\forall\ (u, v) \in E : (v, u) \in E$).

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 $(v, w) \notin E$.

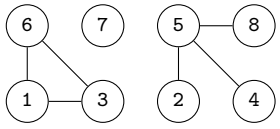
Beispiel.

1. Im Graphen



ist jeder Knoten sowohl Quelle als auch Senke. Ein Pfad von `1` nach `5` ist zum Beispiel $(\texttt{1},\texttt{2},\texttt{0},\texttt{7},\texttt{6},\texttt{0},\texttt{4},\texttt{5})$. Beachten Sie, dass ein Pfad einen Knoten auch mehrfach besuchen darf, weil die Definition nicht verlangt, dass der Homomorphismus injektiv ist. Bei $(\texttt{1},\texttt{2},\texttt{0},\texttt{6},\texttt{0},\texttt{4},\texttt{5})$ 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

$$(\{1,3,6\},\{(1,3),(3,1),(3,6),(6,3),(1,6),(6,1)\}),$$
$$(\{2,4,5,8\},\{(2,5),(5,2),(4,5),(5,4),(5,8),(8,5)\})$$

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. „⇒“ 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

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

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 $(x_i,x_{i+2}) \in E$ ($i=1,\ldots,n-2$ 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.

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

a) Symmetrie ist klar, weil G nach Voraussetzung ein ungerichteter Graph ist.

b) 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 geigneter 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 Kantemenge 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 $(w, \ldots, v)$ gefunden wurde:

7 gib $(u, w, \ldots, v)$ 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 $X = \{x_1, \ldots, x_n\}$ eine Menge mit n Elementen. Eine bijektive Funktion $f\colon X \to X$ 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 $X = \{1, \ldots, n\}$ schreibt man auch einfach S_n statt $S_{\{1,\ldots,n\}}$.

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

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

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

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

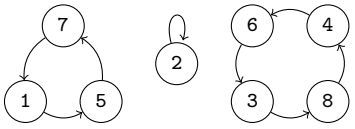
   Ein Zyklus der Länge zwei heißt *Transposition* (engl. *transposition*).

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

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

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

Beispiel.

1. $S_3 = \{\begin{pmatrix}1&2&3\\1&2&3\end{pmatrix}, \begin{pmatrix}1&2&3\\1&3&2\end{pmatrix}, \begin{pmatrix}1&2&3\\2&1&3\end{pmatrix}, \begin{pmatrix}1&2&3\\2&3&1\end{pmatrix}, \begin{pmatrix}1&2&3\\3&1&2\end{pmatrix}, \begin{pmatrix}1&2&3\\3&2&1\end{pmatrix}\}$.
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 $(x_1, \ldots, x_n)$ ein Tupel ist und $\pi \in S_n$ eine Permutation, dann hat das Tupel $(x_{\pi(1)}, \ldots, x_{\pi(n)})$ 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 $V = \{1, \ldots, n\}$ veranschaulichen, bei der eine Kante (u, v) anzeigt, dass $\pi(u) = v$ gilt. Beispiel:
$$\pi = \begin{pmatrix}1&2&3&4&5&6&7&8\\5&2&8&6&7&3&1&4\end{pmatrix} \in S_8$$


   Diese Permutation hat den Fixpunkt 2.

   Ein Graph mit Knotenmenge $\{1, \ldots, n\}$ 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 $\pi = (1\ 5\ 7)(2)(3\ 8\ 4\ 6)$. 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 $\pi = (1\ 5\ 7)(3\ 8\ 4\ 6)$ 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 $B = \{1, \ldots, n\}$
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 $m \in \{1, \ldots, n\}$ mit $\pi^m(k) = k$
8 notiere den Zyklus $(k\ \pi(k)\ \pi^2(k)\ \ldots\ \pi^{m-1}(k))$
9 setze $B = B \setminus \{k, \pi(k), \ldots, \pi^{m-1}(k)\}$.

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 $m \in \{1, \ldots, n\}$ 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 $\circ\colon S_n \times S_n \to S_n$ auffassen. Allgemein nennt man eine Funktion $\circ\colon A \times A \to A$ 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:

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

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 $\circ\colon A \times A \to A$ 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 $\circ\colon A \times A \to A$ 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. $(\mathbb{Z}, +), (\mathbb{Q}, +), (\mathbb{R}, +), (\mathbb{Q}\backslash\{0\}, \cdot), (\mathbb{R}\backslash\{0\}, \cdot)$ 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 $*\colon G \times G \to G$, 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

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

   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 $f_i\colon \mathbb{R}\setminus\{0,1\} \to \mathbb{R}\setminus\{0,1\}$ definiert durch

$$f_1(x) = x, \qquad f_2(x) = \frac{1}{1-x}, \quad f_3(x) = \frac{1}{x},$$
$$f_4(x) = \frac{x-1}{x}, \quad f_5(x) = 1-x, \quad f_6(x) = \frac{x}{x-1}.$$

   Die Menge $G = \{f_1, \ldots, f_6\}$ 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 $(\Omega^*, \circ)$ 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 $i\colon \Omega \to \bar{\Omega}$. 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 $\circ\colon W \times W \to W$ 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: $\Omega = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$, $\bar{\Omega} = \{\mathtt{A}, \mathtt{B}, \mathtt{C}\}$, $i(\mathtt{a}) = \mathtt{A}$, $i(\mathtt{b}) = \mathtt{B}$, $i(\mathtt{c}) = \mathtt{C}$. Dann gilt zum Beispiel

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

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 $\odot\colon G \times G \to G$ 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 $u_1, \ldots, u_m \in U$. Man sagt, U wird von $u_1, \ldots, u_m$ *erzeugt* (engl. *generated*), wenn es für jedes $u \in U$ Zahlen $i_1, \ldots, i_k \in \{1, \ldots, m\}$ und $e_1, \ldots, e_k \in \{-1, 1\}$ gibt, so dass $u = u_{i_1}^{e_1} \circ u_{i_2}^{e_1} \circ \cdots \circ u_{i_k}^{e_1}$. In diesem Fall schreibt man $U = \langle u_1, \ldots, u_m \rangle$.

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

$$f\colon \{1, 2, 3\} \to \{1, 2, 3\}$$

aus S_3 zu einer Funktion $f\colon \{1, 2, 3, 4, 5\} \to \{1, 2, 3, 4, 5\}$ 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:

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

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 verketten, 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 $\{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{A}, \mathtt{B}, \mathtt{C}\}$ gebildet wird. Dann gilt $W = \langle \mathtt{a}, \mathtt{b}, \mathtt{c} \rangle$.

Definition 16. Seien $(G_1, \circ)$ und $(G_2, *)$ Gruppen. Eine Funktion $h\colon G_1 \to G_2$ 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 $h\colon S_3 \to S_5$, 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 $f\colon \mathbb{R} \to \mathbb{R}$, $x \mapsto \exp(x)$ 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 $h\colon G \to A$, $(a, b) \mapsto a$ ein Homomorphismus.

4. Sei $(W, \circ)$ die Gruppe aus dem vorherigen Beispiel, die aus Wörtern über dem Alphabet $\Omega \cup \bar{\Omega} = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{A}, \mathtt{B}, \mathtt{C}\}$ 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 $h\colon W \to G$ mit $h(\mathtt{a}) = g_1$, $h(\mathtt{b}) = g_2$, $h(\mathtt{c}) = g_3$, denn aus diesen drei Werten und der Homomorphismuseigenschaft folgt direkt der Wert $h(\omega)$ für jedes Wort $\omega \in (\Omega \cup \bar{\Omega})^*$. Zum Beispiel muss gelten:

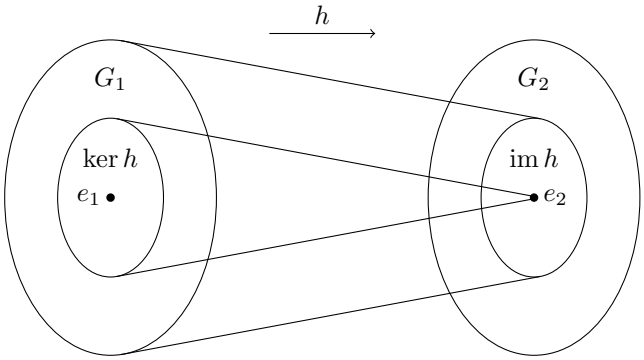
   $$\begin{aligned} h(\mathtt{aCbbaBBAc}) &= h(\mathtt{a} \circ \mathtt{C} \circ \mathtt{b} \circ \mathtt{b} \circ \mathtt{a} \circ \mathtt{B} \circ \mathtt{B} \circ \mathtt{A} \circ \mathtt{c}) \\ &= g_1 * g_3^{-1} * g_2 * g_2 * g_1 * g_2^{-1} * g_2^{-1} * g_1^{-1} * g_3 \end{aligned}$$

Satz 7. Sei $h\colon G_1 \to G_2$ 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 $e_2 \in \operatorname{im} h$. Darüber hinaus bleibt zu zeigen: für alle $u, v \in \operatorname{im} h$ gilt $u * v \in \operatorname{im} h$ und $u^{-1} \in \operatorname{im} h$.

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

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

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

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

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

ein Gruppenhomomorphismus von $(\mathbb{Z}, +)$ ach $(S_7, \circ)$. Dabei bezeichnet $\pi^n = \pi \circ \cdots \circ \pi$ 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 $\pi^0 = \mathrm{id}$ sein.
Wegen

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

$$\pi^4 = (2\ 4\ 7)$$
$$\pi^5 = (1\ 3)(2\ 7\ 4)(5\ 6)$$
$$\pi^6 = \mathrm{id}$$

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

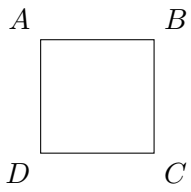
$$\operatorname{im} h = \{\mathrm{id}, \pi, \pi^2, \pi^3, \pi^4, \pi^5\} \subseteq S_7.$$

Beachten Sie, dass $6\mathbb{Z}$ mit $+$ eine Untergruppe von $\mathbb{Z}$ und $\{\mathrm{id}, \pi, \pi^2, \pi^3, \pi^4, \pi^5\}$ 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 $\{A, B, C, D\} \to \{A, B, C, D\}$, 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

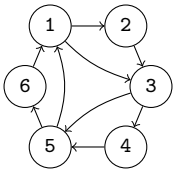
   sowie einige weitere, die sich aus diesen durch Komposition bilden lassen. Die komplette Liste lautet:

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

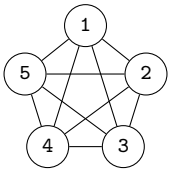
   Diese Menge G bildet zusammen mit der Komposition eine Gruppe, die sogenannte *Symmetriegruppe* (engl. *symmetry group*) des Quadrats. Die Gruppe ist nicht abelsch; es gilt aber zum Beispiel $\sigma\rho = \rho^3\sigma$.

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 $\mathrm{Aut}(G)$. Diese Gruppe ist immer eine Untergruppe von S_V.

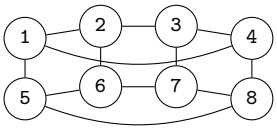
Beispiele:

$G =$ 

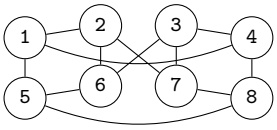
$\Rightarrow \quad \mathrm{Aut}(G) = \{\mathrm{id}, (1\ 3\ 5)(2\ 4\ 6), (1\ 5\ 3)(2\ 6\ 4)\} \subseteq S_6$

$G =$ 

$\Rightarrow \quad \mathrm{Aut}(G) = S_5$

$G =$ 

$\Rightarrow \quad \mathrm{Aut}(G) = \langle (1\ 2\ 3\ 4)(5\ 6\ 7\ 8), (1\ 5)(2\ 6)(3\ 7)(4\ 8) \rangle \subseteq S_8$

$G =$ 

$\Rightarrow \quad \mathrm{Aut}(G) = \langle (1\ 2\ 7\ 8\ 5\ 6\ 3\ 4), (1\ 5)(2\ 6)(3\ 7)(4\ 8) \rangle \subseteq S_8$

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 Gruppenoperationen, der wie folgt definiert ist.

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

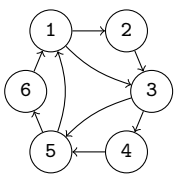
$$* \colon G \times X \to X$$

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

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

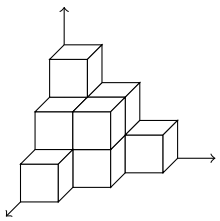
Beispiel.

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



von vorher, und ist $*\colon \mathrm{Aut}(G) \times V \to V$ 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 Konten 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 $\mathrm{Aut}(G)$ 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

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

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

Definition 18. Sei $(G, \circ)$ eine Gruppe, X eine Menge und $*\colon G \times X \to X$ 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. $\mathrm{Stab}(x) := \{\, g \in G : g * x = x \,\} \subseteq G$ heißt der *Stabilisator* (engl. *stabilizer*) von x (bezüglich der Gruppenoperation $*$).

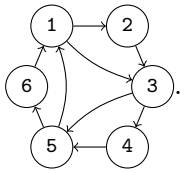
Beispiel.

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

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

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

3. Sei $G = (V, E)$ der Graph



Die Symmetriegruppe $\mathrm{Aut}(G)$ 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 $\mathrm{Stab}(v) = \{\mathrm{id}\}$.

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

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

Dann gilt unter anderem

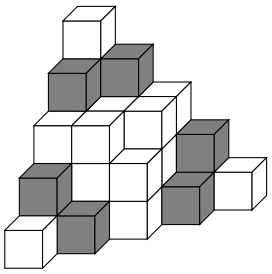
- $\mathrm{Stab}((0,0,0,0,0)) = G$
  $G * (0,0,0,0,0) = \{(0,0,0,0,0)\}$
- $\mathrm{Stab}((0,0,0,0,1)) \cong S_4$
  $G*(0,0,0,0,1) = \{(0,0,0,0,1), (0,0,0,1,0), (0,0,1,0,0), (0,1,0,0,0), (1,0,0,0,0)\}$
- $\mathrm{Stab}((0,0,0,1,1)) \cong S_3 \times S_2$ und

  $G * (0,0,0,1,1) = \{(0,0,0,1,1), (0,0,1,0,1), (0,1,0,0,1), (1,0,0,0,1), (0,0,1,1,0),$
  $(0,1,0,1,0), (1,0,0,1,0), (0,1,1,0,0), (1,0,1,0,0), (1,1,0,0,0)\}$

- $\mathrm{Stab}((0,0,1,1,2)) \cong S_2 \times S_2$ 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 $*\colon G \times X \to X$ 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 $\mathrm{Stab}(x)$ 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 $\mathrm{Stab}(x)$, weil $e * x = x$ nach Def. 17.

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

   (c) Ist $g \in \mathrm{Stab}(x)$, 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 $g^{-1} \in \mathrm{Stab}(x)$.

   Aus (a), (b), (c) zusammen folgt, dass $\mathrm{Stab}(x)$ 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 einen bestimmten Rundungsfehler behaftet ist. Zum Beispiel könnte man für die Division 1/3 das nicht ganz korrekte Ergebnis 0.333333333 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 $\lfloor\cdot\rfloor\colon \mathbb{R} \to \mathbb{Z}$ ist definiert durch

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

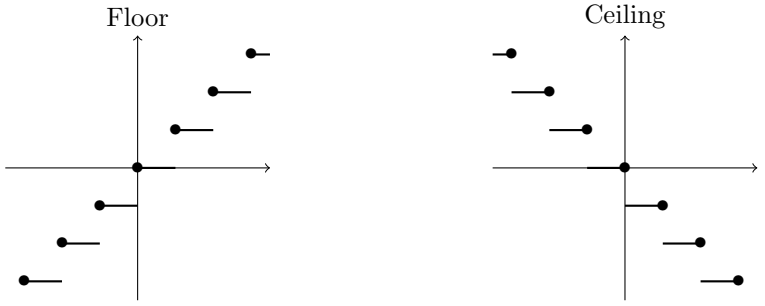
   (gesprochen: „Floor x“).

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

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

   (gesprochen: „Ceiling x“).

Beispiel. Es gilt $\lfloor 5 \rfloor = \lfloor 5.00000001 \rfloor = \lfloor 5.9999999998 \rfloor = 5$, $\lfloor -\frac{1}{2} \rfloor = -1$, $\lceil 5 \rceil = \lceil 4.999998 \rceil = \lceil 4.00000001 \rceil = 5$, $\lceil -\frac{1}{2} \rceil = 0$, 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 $\mathrm{mod} \colon \mathbb{R} \times (\mathbb{R} \setminus \{0\}) \to \mathbb{R}$ ist definiert durch

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

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

Beispiel.

1. $\mathrm{mod}(7, 5) = 2$, $\mathrm{mod}(7, -5) = -3$, $\mathrm{mod}(-7, 5) = 3$, $\mathrm{mod}(-7, -5) = -2$.
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 $\mathrm{mod}(x, y)$ genau der Rest, z.B. gilt $\mathrm{mod}(12345, 52) = 21$, 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 $0 \leq \mathrm{mod}(x, 1) < 1$ und $\mathrm{mod}(x, 1) = 0 \iff x \in \mathbb{Z}$. Insbesondere gilt z.B. $\mathrm{mod}(\pi, 1) = \pi - 3 \approx 0.1415926\ldots$, $\mathrm{mod}(\sqrt{2}, 1) = \sqrt{2} - 1 \approx 0.414\ldots$, usw.
5. Für alle $x \in \mathbb{R}$ gilt $\sin(x) = \sin(\mathrm{mod}(x, 2\pi))$
6. Für alle $x_1, x_2 \in \mathbb{R}$ und alle $y \in \mathbb{Z} \setminus \{0\}$ gilt

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

Beweis: „$\Rightarrow$“ $\mathrm{mod}(x_1, y) = \mathrm{mod}(x_2, y) \Rightarrow x_1 - y\lfloor x_1/y \rfloor = x_2 - y\lfloor x_2/y \rfloor \Rightarrow x_1 - x_2 = y(\lfloor x_1/y \rfloor - \lfloor x_2/y \rfloor)$. 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 $m \in \mathbb{Z}$ gilt mit $ym = x_1 - x_2$, also $x_1 = x_2 + my$. Darum gilt $\mathrm{mod}(x_1, y) = \mathrm{mod}(x_2 + my, y) = x_2 + my - y\lfloor (x_2 + my)/y \rfloor = x_2 + my - y\lfloor x_2/y + m \rfloor = x_2 + my - y(\lfloor x_2/y \rfloor + m) = x_2 - y\lfloor x_2/y \rfloor = \mathrm{mod}(x_2, y)$, 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 $\{-|b|, \ldots, -1, 0, 1, \ldots, |b|\}$, 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: $\gcd(a, b) := \max(T(a) \cap T(b))$.

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: $\gcd(a, b)$

1 setze $(g, g') = (a, b)$

2 solange $g' \neq 0$:

3 ersetze (g, g') durch $(g', \mathrm{mod}(g, g'))$

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 $|\mathrm{mod}(a, b)| < |b|$ 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 Korrekheit überlegt man sich, dass für alle $a, b \in \mathbb{Z}$ gilt $\gcd(a, b) = \gcd(b, \mathrm{mod}(a, b))$. In der Tat gilt sogar $T(a) \cap T(b) = T(b) \cap T(\mathrm{mod}(a, b))$: 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 $\mathrm{mod}(a, b) = \mathrm{mod}(da_0, db_0) = da_0 - db_0 \lfloor da_0/db_0 \rfloor = d\,\mathrm{mod}(a_0, b_0)$, und somit ist d auch ein gemeinsamer Teiler von b und $\mathrm{mod}(a, b)$. Damit gilt $T(a) \cap T(b) \subseteq T(b) \cap T(\mathrm{mod}(a, b))$. Ist umgekehrt d ein gemeinsamer Teiler von b und $\mathrm{mod}(a, b)$, dann ist d auch ein gemeinsamer Teiler von a und b, denn dann gilt $a = qb + \mathrm{mod}(a, b)$ für ein gewisses $q \in \mathbb{Z}$. Wir haben also gezeigt $T(a) \cap T(b) = T(b) \cap T(\mathrm{mod}(a, b))$, und damit müssen insbesondere die maximalen Elemente $\gcd(a, b)$ und $\gcd(b, \mathrm{mod}(a, b))$ 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 $\gcd(g, 0) = |g|$ 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 $r_k = \mathrm{mod}(r_{k-2}, r_{k-1})$ für $k \geq 2$:

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

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

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 $\mathrm{mod}(a, g) = \mathrm{mod}(b, g) = 0$ 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 $\gcd(a, b) = ua + vb$ gilt. Mehr noch: wenn $g, u, v \in \mathbb{Z}$ irgendwelche Zahlen sind, so dass $g = ua + vb$ gilt, dann muss $\gcd(a, b) \mid g$ gelten. Man kann also, wenn man nicht nur g sondern auch u, v kennt, leicht unabhängig überprüfen, dass $g = \gcd(a, b)$ 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: $g = \gcd(a, b)$ 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 $\gcd(a, b) = 19 = -26077a + 34594b$.

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

$$x \equiv_m y \quad \Longleftrightarrow \quad \mathrm{mod}(x, m) = \mathrm{mod}(y, m).$$

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

$$+ \colon \mathbb{Z}_m \times \mathbb{Z}_m \to \mathbb{Z}_m, \quad [x]_{\equiv_m} + [y]_{\equiv_m} := [x + y]_{\equiv_m}$$
$$\cdot \colon \mathbb{Z}_m \times \mathbb{Z}_m \to \mathbb{Z}_m, \quad [x]_{\equiv_m} \cdot [y]_{\equiv_m} := [xy]_{\equiv_m}.$$

Die Definitionen von + und · sind natürlich repräsentantenunabhängig. Gilt nämlich $x_1 \equiv_m x_2$ und $y_1 \equiv_m y_2$, so gilt nach Definition $\mathrm{mod}(x_1, m) = \mathrm{mod}(x_2, m)$ und $\mathrm{mod}(y_1, m) = \mathrm{mod}(y_2, m)$. 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

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

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 $\cdot \colon \mathbb{Z}_m \times \mathbb{Z}_m \to \mathbb{Z}_m$ 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 · sicher keine Gruppe bilden, denn wegen Punkt 3 kommt nur $[1]_{\equiv_m}$ als Neutralelement in Frage, aber dann hat wegen Punkt 4 das Element $[0]_{\equiv_m}$ kein Inverses bezüglich ·. 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 $\mathrm{mod}(u, m) = u \neq 0$ und $\mathrm{mod}(v, m) = v \neq 0$, 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 erstrecht 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 $[a]_{\equiv_m} = [\mathrm{mod}(a, m)]_{\equiv_m}$ und $0 \leq \mathrm{mod}(a, m) < m$ können wir annehmen $0 \leq a < m$ (denn wenn das nicht gilt, ersetzen wir einfach a durch $\mathrm{mod}(a, m)$, 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 $\gcd(a, m) = 1$ 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 $f\colon \mathbb{Z}_m \to \mathbb{Z}_m$ 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

   $$f\colon \mathbb{Z}_m \to \mathbb{Z}_m, \quad f(X) = [u]_{\equiv_m} \cdot X + [v]_{\equiv_m}$$

   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

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

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

   $$0, 97, 581, 853, 405, 661, 661, 661, 661, 661, 661, 661, 661, 661, 661, 661, 661, 661, \dots$$

   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 $f\colon \mathbb{Z}_m \to \mathbb{Z}_m$ zu konstruieren, für die man auch die Umkehrfunktion $f^{-1}\colon \mathbb{Z}_m \to \mathbb{Z}_m$ 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 $e \in \{2, 3, \ldots, \phi - 1\}$ mit $\gcd(e, \phi) = 1$
- Berechne $d \in \{1, \ldots, \phi - 1\}$ so dass $de \equiv_\phi 1$
- Der öffentliche Schlüssel ist dann die Funktion $f\colon \mathbb{Z}_m \to \mathbb{Z}_m$, $f(X) = X^e$
- Der geheime Schlüssel ist die Funktion $f^{-1}\colon \mathbb{Z}_m \to \mathbb{Z}_m$, $f^{-1}(X) = X^d$.

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 berechtigte 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 $X \in \mathbb{Z}_n$. 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 $\{0, 1, \ldots, m-1\}$ 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

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

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:

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

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

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

Beweis. Zunächst gilt

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

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

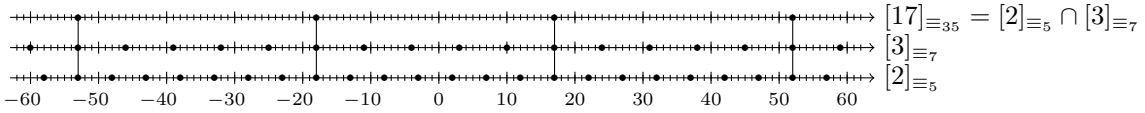
„$\subseteq$“ Ist $x \in [u_1]_{\equiv_{m_1}} \cap [u_2]_{\equiv_{m_2}}$, so gilt $x = u_1 + p_1 m_1 = u_2 + p_2 m_2$ für gewisse $p_1, p_2 \in \mathbb{Z}$. Dann gilt

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

„$\supseteq$“ 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

Eine charakteristische Eigenschaft von diskreten Strukturen ist, dass man sie zählen kann. Während man zum Beispiel keine vernünftige Antwort auf die Frage geben kann, wie viele reelle Zahlen es zwischen 0 und 1 gibt, kann man sich leicht davon überzeugen, dass z.B. die Gruppe $\mathbb{Z}_4$ genau vier Klassen, die Gruppe S_3 genau sechs Permutationen, die Menge von $\{1, 2, 3\}$ genau acht Teilmengen hat, und dass es z.B. genau 512 verschiedene Graphen mit der Knotenmenge $\{1, 2, 3\}$ gibt. Von all diesen Objekten gibt es jeweils nur endlich viele. Die Kombinatorik beschäftigt sich mit der Frage wie viele.

Bei den genannten Beispielen handelt es sich um Instanzen allgemeinerer Familien von diskreten Strukturen. Statt zu fragen, wie viele Permutation die Gruppe S_3 enthält, könnte man auch allgemeiner fragen, wie viele Permutationen die Gruppe S_n enthält, und zwar in Abhängigkeit vom Parameter n. Die Frage ist also, was sich über die Funktion $f\colon \mathbb{N} \to \mathbb{N}$, $f(n) := |S_n|$ sagen lässt. Analog kann man statt nach der Anzahl der Teilmengen von $\{1, 2, 3\}$ auch nach der Anzahl der Teilmengen von $\{1, 2, \ldots, n\}$ für beliebiges $n \in \mathbb{N}$ fragen. Diese beiden Fragen lassen sich noch relativ leicht beantworten.

Satz 13.

1. Für alle $n \in \mathbb{N}$ gilt $|\mathcal{P}(\{1, \ldots, n\})| = 2^n := \underbrace{2 \cdot 2 \cdots 2}_{n \text{ Faktoren}}$.
2. Für alle $n \in \mathbb{N}$ mit $n \geq 1$ gilt $|S_n| = n! := n \cdot (n-1) \cdots 3 \cdot 2 \cdot 1$.

Beweis. Wir verwenden das Prinzip der *vollständigen Induktion*.

1. Induktionsanfang: Für $n = 0$ gilt $\{1, \ldots, n\} = \emptyset$ und $|\mathcal{P}(\emptyset)| = |\{\emptyset\}| = 1 = 2^0$.

   Induktionsvoraussetzung: Angenommen, $n \in \mathbb{N}$ ist so, dass $|\mathcal{P}(\{1, \ldots, n\})| = 2^n$ gilt.

   Induktionsschritt: Wir zeigen, dass dann auch $|\mathcal{P}(\{1, \ldots, n, n+1\})| = 2^{n+1}$ gilt.

   Die Teilmengen von $\{1, \ldots, n, n+1\}$, die $n+1$ nicht enthalten, sind offenbar genau die Teilmengen von $\{1, \ldots, n\}$. Davon gibt es nach Induktionsvoraussetzung 2^n viele. Für jede dieser Teilmengen von $\{1, \ldots, n\}$ erhält man durch Hinzufügen von $n+1$ genau eine Teilmenge von $\{1, \ldots, n, n+1\}$, die nicht schon Teilmenge von $\{1, \ldots, n\}$ ist. Es gibt also von $\{1, \ldots, n, n+1\}$ auch genau 2^n viele Teilmengen, die $n+1$ enthalten. Da jede Teilmenge von $\{1, \ldots, n, n+1\}$ das Element $n+1$ entweder enthält oder nicht enthält, gibt es insgesamt genau $2^n + 2^n = 2 \cdot 2^n = 2^{n+1}$ viele Teilmengen von $\{1, \ldots, n, n+1\}$, wie behauptet.

2. Induktionsanfang: Für $n = 1$ gilt $S_1 = \{\mathrm{id}\}$, also $|S_1| = 1 = 1!$.

   Induktionsvoraussetzung: Angenommen, $n \in \mathbb{N} \setminus \{0\}$ ist so, dass $|S_n| = n!$ gilt.

   Induktionsschluss: Wir zeigen, dass dann auch $|S_{n+1}| = (n+1)!$ gilt.

Zunächst ist klar, dass die Permutationen von $\{1,\dots,n\}$ genauso zahlreich sind wie die Permutationen von $\{1,\dots,n+1\}$, die $n+1$ als Fixpunkt haben. Nach Induktionsannahme gibt es $n!$ viele Permutationen von $\{1,\dots,n\}$. Für jede solche Permutation π und jede Wahl von $k \in \{1,\dots,n+1\}$ ist $(n+1\ k)\circ\pi$ eine Permutation von $\{1,\dots,n+1\}$. Da all diese Permutationen paarweise verschieden sind, folgt $|S_{n+1}| \geq (n+1)|S_n|$.

Umgekehrt gilt: ist $\sigma \in S_{n+1}$ beliebig, so ist $\pi := (n+1\ \sigma(n+1))\circ\sigma$ eine Permutation, die $n+1$ als Fixpunkt hat. Da Transpositionen selbstinvers sind, gilt auch $\sigma = (n+1\ \sigma(n+1))\circ\pi$, so dass sich also jedes Element von S_{n+1} als Produkt einer Permutation mit Fixpunkt $n+1$ und einer Transposition $(n+1\ k)$ schreiben lässt. Daher werden mit der vorher beschriebenen Konstruktion alle Elemente von S_{n+1} erreicht und es folgt $|S_{n+1}| = (n+1)|S_n|$.

Solche einfache Formeln wie im vorigen Satz gibt es nicht immer.

Beispiel.

1. Eine *Partition* (engl. *partition*) der Menge $\{1,\dots,n\}$ ist eine Teilmenge $\{u_1,\dots,u_k\}$ von $\mathcal{P}(\{1,\dots,n\})$ (d.h. jedes u_i ist eine Teilmenge von $\{1,\dots,n\}$), so dass (a) $u_i \neq \emptyset$ für alle i; (b) $u_i \cap u_j = \emptyset$ für alle $i \neq j$, (c) $u_1 \cup \dots \cup u_k = \{1,\dots,n\}$.

   Zum Beispiel hat die Menge $\{1,2,3,4\}$ die folgenden 15 Partitionen:

   $$\begin{array}{lll}
   \{\{1\},\{2\},\{3\},\{4\}\}, & \{\{1\},\{2\},\{3,4\}\}, & \{\{1\},\{3\},\{2,4\}\},\\
   \{\{1\},\{4\},\{2,3\}\}, & \{\{2\},\{3\},\{1,4\}\}, & \{\{2\},\{4\},\{1,3\}\},\\
   \{\{3\},\{4\},\{1,2\}\}, & \{\{1,2\},\{3,4\}\}, & \{\{1,3\},\{2,4\}\},\\
   \{\{1,4\},\{2,3\}\}, & \{\{1\},\{2,3,4\}\}, & \{\{2\},\{1,3,4\}\},\\
   \{\{3\},\{1,2,4\}\}, & \{\{4\},\{1,2,3\}\}, & \{\{1,2,3,4\}\}.
   \end{array}$$

   Die Anzahl der Partitionen der Menge $\{1,\dots,n\}$ heißt die $n$te *Bell-Zahl* und wird mit $\mathbf{B}_n$ bezeichnet. Es gilt also $\mathbf{B}_4 = 15$. Einige weitere Werte kann man der folgenden Tabelle entnehmen:

n	1	2	3	4	5	6	7	8	9	10
$\mathbf{B}_n$	1	2	5	15	52	203	877	4140	21147	115975

   Es ist keine einfache Formel für $\mathbf{B}_n$ bekannt.

2. Eine *Partition* (engl. *partition*) einer Zahl $n \in \mathbb{N}$ ist ein Tupel $(a_1,a_2,\dots,a_k) \in \mathbb{N}^k$ mit $a_1+a_2+\dots+a_k = n$ und $a_1 \geq a_2 \geq \dots \geq a_k \geq 1$. Zum Beispiel hat die Zahl $7 \in \mathbb{N}$ die folgenden 15 Partitionen:

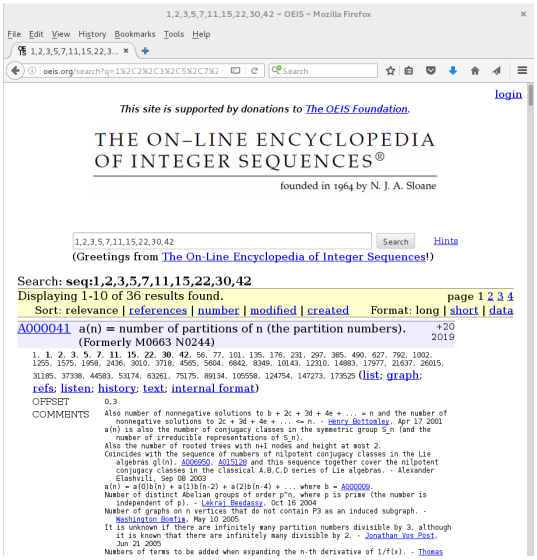
   $$\begin{array}{lllll}
   (1,1,1,1,1,1,1), & (2,1,1,1,1,1), & (2,2,1,1,1), & (2,2,2,1), & (3,1,1,1,1),\\
   (3,2,1,1), & (3,2,2), & (3,3,1), & (4,1,1,1), & (4,2,1),\\
   (4,3), & (5,1,1), & (5,2), & (6,1), & (7).
   \end{array}$$

   Mit $p(n)$ wird die Anzahl der Partitionen der Zahl n bezeichnet. Es gilt also $p(7) = 15$. Einige weitere Werte kann man der folgenden Tabelle entnehmen.

n	1	2	3	4	5	6	7	8	9	10
$p(n)$	1	2	3	5	7	11	15	22	30	42

Es ist keine einfache Formel für $p(n)$ bekannt.

Selbst wenn man, wie im obigen Beispiel, keine allgemeine Formel für die Anzahl bestimmter Objekte in Abhängigkeit von ihrer Größe n finden kann, ist es meistens zumindest möglich, die Anzahlen für einige kleine Werte von n zu bestimmen. Dazu muss man nur die Objekte der entsprechenden Größe alle auflisten. Aber Vorsicht, dass man dabei keines vergisst oder mehrfach zählt! Meist ist es am besten, sich ein kleines Programm dafür zu schreiben. Sobald man die ersten Werte der Zahlenfolge kennt, kann man in Online-Datenbanken danach suchen. Für Folgen von ganzen Zahlen bietet sich zum Beispiel die frei zugängliche *Online Encyclopedia of Integer Sequences* (OEIS) an, die man auf `http://www.oeis.org/` findet. Gibt man dort die Zahlen $1, 2, 3, 5, 7, 11, 15, 22, 30, 42$ ein, wird man auf eine Seite geführt, die allerlei Information über die Folge der Partitionszahlen enthält:



Wie konnte die Seite wissen, dass wir diese Folge meinen und nicht irgendeine andere, die mit den gleichen Zahlen beginnt? Die Antwort lautet: sie weiss es gar nicht sondern listet wie eine Suchmaschine alle Einträge der Datenbank auf, die mit den angegebenen Zahlen beginnen. Im obigen Beispiel sind es 36 Einträge, von denen der relevanteste Treffer als erstes angezeigt wird. Daraus, dass eine Zahlenreihe mit $1, 2, 3, 4, 5, 6, 7, 8, 9, 10$, darf man natürlich nicht ohne weiteres schließen, dass als nächstes die Zahl 11 kommt. Tatsächlich enthält die OEIS weit über tausend Einträge von verschiedenen Folgen, die alle mit $1, 2, 3, 4, 5, 6, 7, 8, 9, 10$ beginnen. Welche die gesuchte ist, oder ob die gesuchte womöglich gar nicht in der Datenbank enthalten ist, muss man als Benutzer selbst herausfinden.

Dabei ist zu beachten, dass dieselbe Zahlenfolge unter Umständen unterschiedliche diskrete Strukturen abzählen kann.

Beispiel.

1. Für jedes $n \in \mathbb{N}$ gibt es genau so viele Teilmengen von $\{1, \dots, n\}$ wie es Wörter der Länge n über dem Alphabet $\Omega = \{\mathtt{a}, \mathtt{b}\}$ gibt. Um das zu zeigen, nutzt man aus, dass zwei Mengen A, B genau dann gleich groß sind, wenn es eine bijektive Funktion $f\colon A \to B$ gibt.

   Betrachte für fixes $n \in \mathbb{N}$ die Funktion $f\colon \mathcal{P}(\{1, \dots, n\}) \to \Omega^n$, die jeder Teilmenge $A \subseteq \{1, \dots, n\}$ das Wort $f(A) \in \Omega^n$ zuordnet, das wie folgt definiert ist: Für jedes

$i = 1, \ldots, n$ sei der $i$te Buchstabe $\mathtt{a}$, falls $i \in A$, und $\mathtt{b}$ falls $i \notin A$. Im Fall $n = 10$ wäre also zum Beispiel $f(\{1, 4, 5, 6\}) = \mathtt{abbaaabbbb}$ und $f(\{3, 6, 9, 10\}) = \mathtt{bbabbabbaa}$.

Dass f bijektiv ist, sieht man leicht ein, denn die Funktion $g\colon \Omega^n \to \mathcal{P}(\{1, \ldots, n\})$, die jedem Wort $\omega = u_1 \ldots u_{10} \in \Omega^n$ die Menge $\{\, i \in \{1, \ldots, n\} : u_i = \mathtt{a} \,\} \in \mathcal{P}(\{1, \ldots, n\})$ zuordnet, ist offensichtlich eine Umkehrfunktion zu f.

2. Für jedes $n \in \mathbb{N}$ betrachte wie oben die Menge Ω^n aller Wörter der Länge n über dem Alphabet $\Omega = \{\mathtt{a}, \mathtt{b}\}$ sowie die Menge $\{0, 1, \ldots, 2^n - 1\}$. Dass diese Mengen für jede Wahl von $n \in \mathbb{N}$ gleich groß sind, kann man auch mit Hilfe einer bijektiven Abbildung zeigen.

   Betrachte die Funktion

   $$f\colon \Omega^n \to \{0, 1, \ldots, 2^n - 1\}, \quad f(u_1 \ldots u_{10}) := \sum_{i=1}^{10} [\![u_i = \mathtt{a}]\!] 2^{i-1}.$$

   Dabei ist die sogenannte Iverson-Klammer $[\![X]\!]$ eine Notation, die einen Wahrheitswert in einen Zahlwert umwandelt: Wenn X wahr ist, ist $[\![X]\!] = 1$, und wenn X falsch ist, ist $[\![X]\!] = 0$. Es gilt also zum Beispiel

   $$f(\mathtt{aababba}) = 1 \cdot 1 + 1 \cdot 2 + 0 \cdot 4 + 1 \cdot 8 + 0 \cdot 16 + 0 \cdot 32 + 1 \cdot 64 = 75.$$

   Die zu f inverse Abbildung $g\colon \{0, \ldots, 2^n - 1\} \to \Omega^n$ konstruiert aus einem gegebenen $x \in \{0, \ldots, 2^n - 1\}$ ein Wort, an dessen i-ter Stelle ($i = 1, \ldots, n$) der Buchstabe $\mathtt{a}$ steht, wenn $\lfloor x/2^{i-1} \rfloor$ eine ungerade Zahl ist, und ansonsten $\mathtt{b}$. Wegen

   $$\lfloor 75/2^0 \rfloor = 75, \quad \lfloor 75/2^1 \rfloor = 37, \quad \lfloor 75/2^2 \rfloor = 18, \quad \lfloor 75/2^3 \rfloor = 9,$$
   $$\lfloor 75/2^4 \rfloor = 4, \quad \lfloor 75/2^5 \rfloor = 2, \quad \lfloor 75/2^6 \rfloor = 1,$$

   ist zum Beispiel $g(75) = \mathtt{aababba}$.

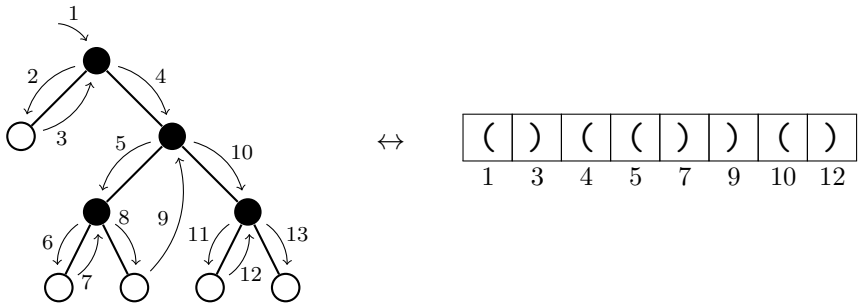
   Allgemein gilt: Für jede endliche Menge M gibt es eine bijektive Abbildung von M nach $\{0, \ldots, |M| - 1\}$. Es ist aber nicht immer einfach, eine solche Abbildung explizit anzugeben.

3. Es sei K die Menge aller Wörter über dem Alphabet $\Omega = \{\mathtt{(}, \mathtt{)}\}$, die korrekt geklammert sind. Es gilt also zum Beispiel $\mathtt{(()(()()))} \in K$ und $\mathtt{)(()(} \notin K$.

   Weiter sei B die Menge aller Isomorphieklassen von Binärbäumen (d.h. Binärbäume, bei denen man die Bezeichnungen der Knoten ignoriert und nur die Struktur betrachtet), bei denen jeder innere Knoten genau zwei Nachfolger hat. Die Elemente von B heißen *vollständige Binärbäume.*

   Wir wissen zwar nicht, wie viele Wörter der Länge $2n$ in K enthalten sind oder wie viele Bäume mit n inneren Knoten in B enthalten sind. Aber es ist leicht einzusehen, dass es sich um die gleiche Anzahl handeln muss. Dazu betrachtet man die folgende Bijektion: beginnend mit der Wurzel wandern wir einmal komplett durch den Baum, und zwar besuchen wir nach jedem Knoten v alle Knoten des linken Unterbaums von v, dann nochmals v, und dann alle Knoten des rechten Unterbaums von v. Die Knoten der Unterbäume werden nach diesem Schema durchalufen: zuerst die Wurzel des Unterbaums, dann die Knoten des linken Unterunterbaums (nach demselben Schema), dann wieder

die Wurzel, dann die Knoten des rechten Unterunterbaums (nach demselben Schema). Die Prozedur ist in der folgenden Abbildung veranschaulicht. Jeder innere Knoten wird zweimal besucht. Beim ersten Besuch notieren wir eine öffnende Klammer, beim zweiten Besuch eine schließende. So entspricht jeder innere Knoten einem Klammernpaar.

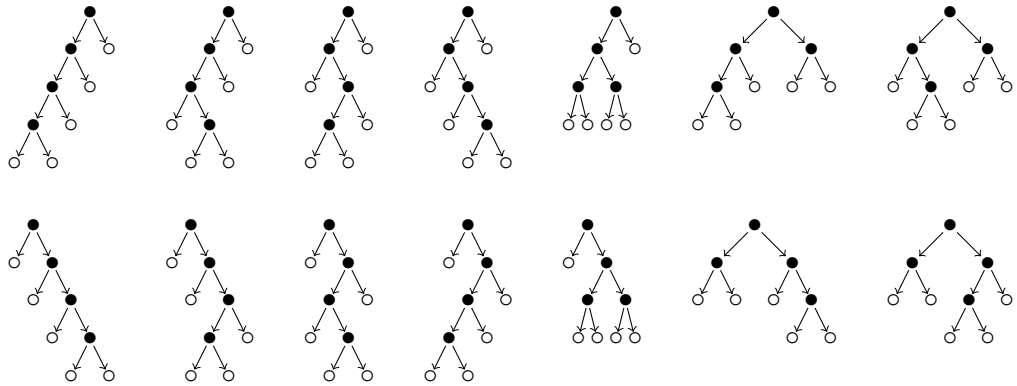


Die Anzahl der Binärbäume mit n inneren Knoten wird mit C_n bezeichnet und die n-te *Catalan-Zahl* (engl. *catalan number*) genannt.

Die ersten Catalan-Zahlen lauten wie folgt:

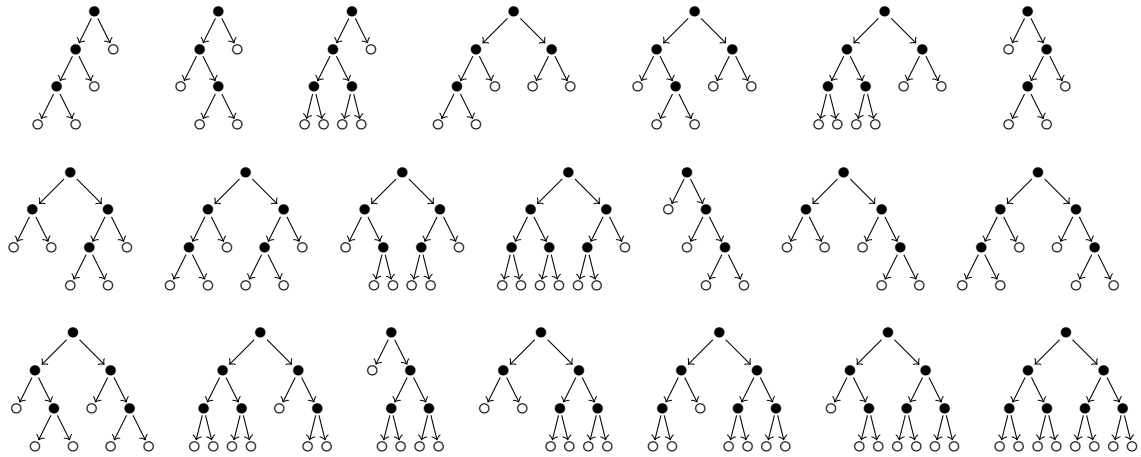
n	1	2	3	4	5	6	7	8	9	10
C_n	1	2	5	14	42	132	429	1430	4862	16796

Zum Beispiel gilt $C_4 = 14$, weil es genau die folgenden vierzehn vollständigen Binärbäume mit vier inneren Knoten gibt:



Es gibt zahlreiche weitere und zum Teil sehr verschiedenartige diskrete Strukturen, die alle von den Catalan-Zahlen abgezählt werden.

Die Abzählung der vollständige Binärbäume ist ein typisches Szenario in der Kombinatorik: Obwohl es insgesamt unendlich viele solcher Bäume gibt, gibt es doch für jede gegebene Größe $n \in \mathbb{N}$ gibt nur endlich viele, wenn man die Zahl der inneren Knoten als die Größe eines vollständigen Binärbaums betrachtet. Man kann die Größe eines vollständigen Binärbaumes auch mit anderen Kennzahlen messen, zum Beispiel mit der Höhe. Die *Höhe* (engl. *height*) eines Baumes ist definiert als die Länge des längsten Pfades im Baum. Der längste Pfad beginnt immer an der Wurzel und endet an einem Blatt. Es gibt zum Beispiel 21 vollständige Binärbäume der Höhe 4:

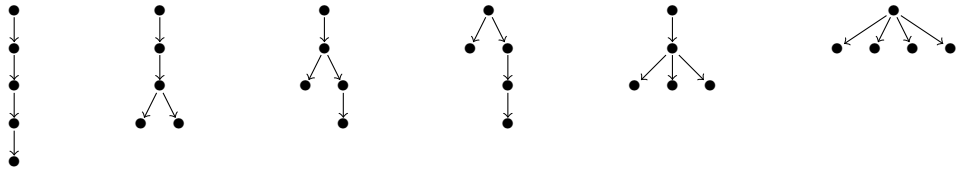


Wenn H_k die Zahl der vollständigen Binärbäume der Höhe k ist, dann lauten die ersten Werte der Folge H_k wie folgt:

k	1	2	3	4	5	6	7	8
H_k	1	1	3	21	651	457653	210065930571	44127887745696109598901

Wie viele Strukturen einer bestimmten Größe es gibt, hängt also nicht nur vom Typ der betrachteten Struktur ab, sondern auch davon, was man als deren Größe auffasst.

Nicht jede Definition von „Größe“ ist für jeden Typ von Struktur brauchbar. Wenn man zum Beispiel nicht Binärbäume sondern beliebige Bäume (genauer gesagt Isomorphieklassen von beliebigen Bäumen) zählt, kann man immer noch die Zahl T_n aller Bäume mit n Knoten betrachten. Zum Beispiel gilt $T_5 = 6$:



Dagegen ist es nicht sinnvoll, von der Zahl aller Bäume der Höhe k zu sprechen, denn davon gibt schon für $k = 2$ unendlich viele:

Wenn man trotzdem etwas darüber wissen will, wie häufig Bäume mit bestimmter Höhe sind, kann man untersuchen, wie viele Bäume mit einer vorgegebenen Anzahl n von Knoten *und* einer vorgegebenen Höhe k es gibt. Für jede Wahl von $n, k \in \mathbb{N}$ ist dies eine endliche Zahl, nennen wir sie $T_{n,k}$. Es gilt $T_{n,k} = 0$ für $k > n$, weil ein Baum mit Höhe k mindestens k Knoten haben muss. Ausserdem gilt

$$T_n = T_{n,0} + T_{n,1} + \cdots + T_{n,n},$$

weil jeder Baum mit n Knoten entweder ein Baum mit n Knoten der Höhe 0 oder ein Baum mit n Knoten der Höhe 1 oder … oder ein Baum mit n Knoten der Höhe n ist.

In der folgenden Definition werden einige Größen eingeführt, die die Abzählresultate aus Satz 13 in ähnlicher Weise verfeinern.

Definition 22. Seien $n, k \in \mathbb{N}$.

1. Mit $\binom{n}{k}$ (sprich: „n über k“; engl. „n choose k“) wird die Anzahl der Teilmengen von $\{1, \ldots, n\}$ mit genau k Elementen bezeichnet. Man nennt $\binom{n}{k}$ den *Binomialkoeffizient* (engl. *binomial coefficient*).

2. Sei $c(n, k)$ die Anzahl der Permutationen $\pi \in S_n$ mit genau k Zyklen. Dann heißt $\left[\begin{smallmatrix} n \\ k \end{smallmatrix}\right] := (-1)^{n+k} c(n, k)$ die *Stirling-Zahl erster Art* zum Index (n, k).

3. Mit $\left\{\begin{smallmatrix} n \\ k \end{smallmatrix}\right\}$ wird die Menge aller Partitionen U von $\{1, \ldots, n\}$ mit $|U| = k$ bezeichnet. Man nennt $\left\{\begin{smallmatrix} n \\ k \end{smallmatrix}\right\}$ die *Stirling-Zahl zweiter Art* zum Index (n, k).

4. Für die Anzahl der Partitionen $(p_1, \ldots, p_k)$ von $n \in \mathbb{N}$ mit genau k Komponenten schreibt man $p_k(n)$.

Beispiel.

1. Es gilt $\binom{5}{3} = 10$, denn die Menge $\{1, 2, 3, 4, 5\}$ hat die folgenden zehn Teilmengen der Größe drei:

$$\begin{array}{lllll} \{1,2,3\}, & \{1,2,4\}, & \{1,2,5\}, & \{1,3,4\}, & \{1,3,5\}, \\ \{1,4,5\}, & \{2,3,4\}, & \{2,3,5\}, & \{2,4,5\}, & \{3,4,5\}. \end{array}$$

2. Es gilt $\left[\begin{smallmatrix} 4 \\ 2 \end{smallmatrix}\right] = 11$, denn es gibt nur die folgenden elf Permutationen von $\{1, 2, 3, 4\}$ mit genau zwei Zyklen:

$$\begin{array}{llllll} (1\ 2)(3\ 4), & (1\ 3)(2\ 4), & (1\ 4)(2\ 3), & (1)(2\ 3\ 4), & (1)(2\ 4\ 3), & (2)(1\ 3\ 4), \\ (2)(1\ 4\ 3), & (3)(1\ 2\ 4), & (3)(1\ 4\ 2), & (4)(1\ 2\ 3), & (4)(1\ 3\ 2). \end{array}$$

3. Es gilt $\left\{\begin{smallmatrix} 4 \\ 3 \end{smallmatrix}\right\} = 6$, denn es gibt nur die folgenden sechs Partitionen der Menge $\{1, 2, 3, 4\}$ in genau drei Teilmengen:

$$\begin{array}{lll} \{\{1\},\{2\},\{3,4\}\}, & \{\{1\},\{3\},\{2,4\}\}, & \{\{1\},\{4\},\{2,3\}\}, \\ \{\{2\},\{3\},\{1,4\}\}, & \{\{2\},\{4\},\{1,3\}\}, & \{\{3\},\{4\},\{1,2\}\}. \end{array}$$

4. Es gilt $p_5(10) = 7$, denn es gibt nur die folgenden sieben Partitionen der Zahl 10 mit genau fünf Komponenten:

$$\begin{array}{llll} (6,1,1,1,1), & (5,2,1,1,1), & (4,3,1,1,1), & (4,2,2,1,1), \\ (3,3,2,1,1), & (3,2,2,2,1), & (2,2,2,2,2). & \end{array}$$

Da jede Teilmenge von $\{1, \ldots, n\}$ eine Teilmenge von einer bestimmten Größe k ist, ergibt sich mit Satz 13 sofort die binomische Formel

$$\sum_{k=0}^{n} \binom{n}{k} = 2^n \qquad (n \in \mathbb{N})$$

Aus dem gleichen Grund gilt

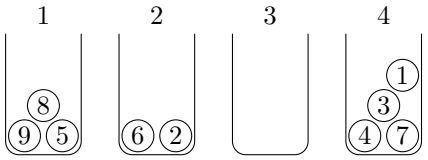
$$\sum_{k=0}^{n} \left|\begin{bmatrix} n \\ k \end{bmatrix}\right| = n! \quad \text{und} \quad \sum_{k=0}^{n} \begin{Bmatrix} n \\ k \end{Bmatrix} = \mathbf{B}_n$$

für alle $n \in \mathbb{N}$.

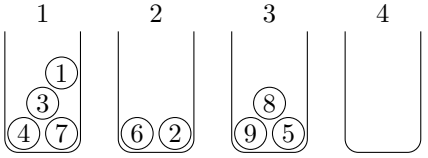
Die Größen aus Definition 22 treten in vielen weiteren kombinatorischen Zusammenhängen auf. Ein Beispiel ist das Abzählen von Symmetrieklassen von Funktionen. Sei $K = \{1, \dots, k\}$, $N = \{1, \dots, n\}$. Wir betrachten die Menge K^N aller Funktionen $f\colon N \to K$. Zunächst ist leicht einzusehen, dass es insgesamt k^n solcher Funktionen gibt, denn es gibt n mögliche Argumente aus N, und für jedes dieser n Argumente gibt es k mögliche Funktionswerte.

Man kann sich eine Funktion $f\colon N \to K$ veranschaulichen als eine bestimmte Aufteilung von n verschiedenen Kugeln auf k verschiedene Töpfe:

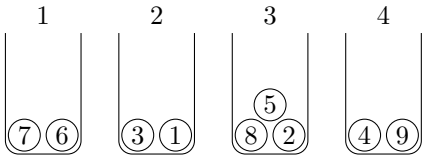
n	1	2	3	4	5	6	7	8	9
$f(n)$	4	2	4	4	1	2	4	1	1



n	1	2	3	4	5	6	7	8	9
$g(n)$	1	2	1	1	3	2	1	3	3



n	1	2	3	4	5	6	7	8	9
$h(n)$	2	3	2	4	3	1	1	3	4



Es handelt sich bei diesen Beispielen um drei verschiedene Funktionen. Hätten wir es dagegen statt mit durchnumerierten Töpfen mit vier völlig identischen und ununterscheidbaren Töpfen zu tun, dann wären auch die Funktionen f und g nicht voneinander zu unterscheiden; h wäre dann aber immer noch eine andere Funktion. Die Ununterscheidbarkeit von Töpfen wird durch eine Äquivalenzrelation auf der Menge aller Funktionen ausgedrückt: Zwei Funktionen $f, g\colon N \to K$ sind zueinander äquivalent, wenn es eine Permutation $\pi \in S_k$ gibt, so dass $f(x) = \pi(g(x))$ für alle $x \in \{1, \dots, n\}$ gilt. Die Permutation π entspricht einer Umnumerierung der Töpfe. Die Äquivalenzklassen bezüglich dieser Äquivalenzrelation kann man sich vorstellen als Zuordnungen von n Kugeln auf k ununterscheidbare Töpfe.

Analog kann man auch eine Ununterscheidbarkeit von Kugeln modellieren. In diesem Fall wären zwei Funktionen $f, g\colon N \to K$ zueinander äquivalent, wenn es eine Permutation $\sigma \in S_n$ gibt, so dass $f(x) = g(\sigma(x))$ für alle $x \in \{1, \dots, n\}$ gilt. Die Permutation σ entspricht dann einer Umnumerierung der Kugeln, und die Äquivalenzklassen bezüglich dieser Äquivalenzrelation sind die möglichen Zuordnungen von n ununterscheidbaren Kugeln auf k (unterscheidbare) Töpfe.

Natürlich ist es auch möglich, sowohl ununterscheidbare Kugeln als auch ununterscheidbare Töpfe zu nehmen. In diesem Fall würde man $f, g\colon N \to K$ als äquivalent betrachten, wenn es ein Paar $(\pi, \sigma) \in S_k \times S_n$ gibt, so dass $f(x) = \pi(g(\sigma(x)))$ für alle $x \in \{1, \dots, n\}$ gilt.

All diese Fälle lassen sich auch mit Hilfe von Gruppenoperationen beschreiben. Dazu betrachtet man eine Untergruppe G von $S_k \times S_n$ und definiert die Gruppenoperation

$$*\colon G \times K^N \to K^N \quad (\pi, \sigma) * f := \pi \circ f \circ \sigma^{-1}.$$

(Die Invertierung von σ ist nötig, damit die Gesetze aus Def. 17 erfüllt sind, spielt für das Verständnis aber keine große Rolle.) Die Bahnen der Gruppenoperation sind dann genau die Äquivalenzklassen (vgl. Satz 8). Die verschiedenen Fälle entsprechen verschiedenen Wahlen von Untergruppen von G: für unterscheidbare Kugeln und unterscheidbare Töpfe nimmt man $G = \{\mathrm{id}\} \times \{\mathrm{id}\}$; für unterscheidbare

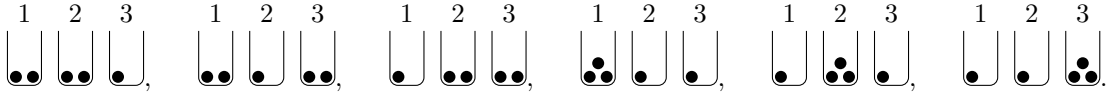
Kugeln und ununterscheidbare Töpfe nimmt man $G = S_k \times \{\mathrm{id}\}$; für ununterscheidbare Kugeln und unterscheidbare Töpfe nimmt man $G = \{\mathrm{id}\} \times S_n$; für ununterscheidbare Kugeln und ununterscheidbare Töpfe nimmt man $G = S_k \times S_n$.

Natürlich gibt es viele weitere Möglichkeiten. Zum Beispiel kann man mit einer geeigneten Untergruppe von $S_k \times S_n$ auch den Fall modellieren, dass es vier grüne Kugeln und drei blaue Kugeln gibt, wobei jede grüne Kugel von einer blauen unterscheidbar ist, gleichfarbige Kugeln untereinander aber ununterscheidbar sind. Mit einer anderen Untergruppe von $S_k \times S_n$ kann man modellieren, dass es insgesamt zehn Töpfe gibt, davon fünf durchnumerierte und fünf weitere, die ununterscheidbar sind.

Für jede Untergruppe G von $S_k \times S_n$ kann man fragen, wie viele Funktionen es gibt, und wie viele dieser Funktionen injektiv, surjektiv oder bijektiv sind. Die folgende Tabelle enthält die Antworten für die vier Gruppen $\{\mathrm{id}\} \times \{\mathrm{id}\}$, $S_k \times \{\mathrm{id}\}$, $\{\mathrm{id}\} \times S_n$, $S_k \times S_n$:

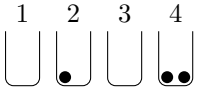
	beliebig	injektiv	surjektiv	bijektiv
$\lvert K^N \rvert$	k^n	$n!\binom{k}{n}$	$k!\left\{{n \atop k}\right\}$	$[\![k = n]\!]n!$
$\lvert K^N/(\{\mathrm{id}\} \times S_n) \rvert$	$\binom{k+n-1}{k-1}$	$\binom{k}{n}$	$\binom{n-1}{n-k}$	$[\![k = n]\!]$
$\lvert K^N/(S_k \times \{\mathrm{id}\}) \rvert$	$\sum_{i=0}^{k} \left\{{n \atop i}\right\}$	$[\![n \le k]\!]$	$\left\{{n \atop k}\right\}$	$[\![k = n]\!]$
$\lvert K^N/(S_k \times S_n) \rvert$	$p_k(n+k)$	$[\![n \le k]\!]$	$p_k(n)$	$[\![k = n]\!]$

Es gibt also zum Beispiel $\binom{5-1}{3-1} = 6$ verschiedene surjektive Zuordnungen von fünf ununterscheidbaren Kugeln auf drei unterscheidbare Töpfe. Sie lauten:

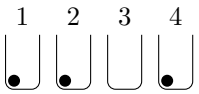


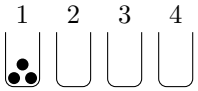
Für die meisten Einträge der Tabelle gibt es sehr anschauliche Beweise. Betrachten wir als Beispiel den Fall $G = \{\mathrm{id}\} \times S_n$, also n ununterscheidbare Kugeln und k unterscheidbare Töpfe. Die Behauptung ist, dass es dann $\binom{k+n-1}{n}$ Äquivalenzklassen von Funktionen gibt. Nach Definition ist $\binom{k+n-1}{k-1}$ die Anzahl der Teilmengen von $\{1, \ldots, k+n-1\}$ mit genau $k-1$ Elementen. Um zu zeigen, dass es auch die Zahl der Äquivalenzklassen ist, konstruieren wir eine bijektive Funktion, die jeder Teilmenge eine solche Klasse zuordnet. Die Idee ist, zunächst eine Bijektion zwischen den Teilmengen mit $k-1$ Elementen und den Wörtern der Länge $k+n-1$ über dem Alphabet $\{\mathtt{o}, \mathtt{|}\}$ mit genau $k-1$ Symbolen $\mathtt{|}$ anzugeben, und dann eine weitere Bijektion von dort zu den Äquivalenzklassen. Eine gegebene Teilmenge $\{u_1, \ldots, u_k\} \subseteq \{1, \ldots, k+n-1\}$ mit $u_i \neq u_j$ für $i \neq j$ soll dabei abgebildet werden auf das Wort $\omega \in \{\mathtt{o}, \mathtt{|}\}^{k+n-1}$, das an den Stellen $u_1, \ldots, u_k$ das Symbol $\mathtt{|}$ und an allen anderen Stellen das Symbol $\mathtt{o}$ hat. Aus jedem solchen Wort lässt sich als nächstes eine Funktion konstruieren, indem man in den $i$ten Topf genau so viele Kugeln gibt, wie es Symbole $\mathtt{o}$ zwischen dem $(i-1)$ten und dem $i$ten Symbol $\mathtt{|}$ gibt. Da ω ein Wort der Länge $k+n-1$ mit genau $k-1$ vielen Symbolen $\mathtt{|}$ ist, enthält es $(k+n-1)-(k-1) = n$ viele Symbole $\mathtt{o}$, und da die Kugeln alle ununterscheidbar sind, kommt es nur auf ihre Anzahl an. Beispiele für $n = 3$, $k = 4$:

$\{1,3,4\} \subseteq \{1,2,3,4,5,6\}$ $\leftrightarrow$ $\mathtt{|o||oo}$ $\leftrightarrow$



$\{2,4,5\} \subseteq \{1,2,3,4,5,6\}$ $\leftrightarrow$ $\mathtt{o|o||o}$ $\leftrightarrow$



$\{4,5,6\} \subseteq \{1,2,3,4,5,6\}$ $\leftrightarrow$ ooo||| $\leftrightarrow$ 

Da man die Konstruktion auch in die umgekehrte Richtung durchführen kann, handelt es sich um eine Bijektion.

9 Rekurrenzen

Beispiel.

1. Die Größen $a_n = 2^n$ und $b_n = n!$ aus Satz 13 erfüllen die Gleichungen

$$a_{n+1} = 2a_n \qquad \text{und} \qquad b_{n+1} = (n+1)b_n$$

für alle $n \in \mathbb{N}$. Solche Gleichungen nennt man *Rekurrenzen* (engl. *recurrence*). Wenn man die Werte a_0 und b_0 kennt (die sogenannten *Anfangswerte* (engl. *initial values*)), kann man mit diesen Gleichungen nacheinander alle weiteren Werte von a_n und b_n berechnen:

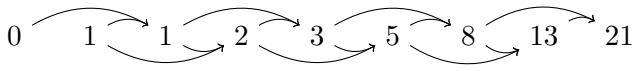
$$a_0 = 1 \Rightarrow a_1 = 2a_0 = 2 \Rightarrow a_2 = 2a_1 = 4 \Rightarrow a_3 = 2a_2 = 8 \Rightarrow a_4 = 2a_3 = 16 \cdots$$
$$b_0 = 1 \Rightarrow b_1 = 1b_0 = 1 \Rightarrow b_2 = 2b_1 = 2 \Rightarrow b_3 = 3b_2 = 6 \Rightarrow b_4 = 4b_3 = 24 \cdots$$

Auch $c_n = 2^{2^n}$ erfüllt eine Rekurrenz, nämlich $c_{n+1} = c_n^2$ für $n \in \mathbb{N}$.

2. Für die Fibonacci-Zahlen F_n gilt die Rekurrenz

$$F_{n+2} = F_n + F_{n+1}$$

für alle $n \in \mathbb{N}$. In diesem Fall werden die Zahlen F_n eindeutig durch die Rekurrenz festgelegt, sobald man die beiden Anfangswerte $F_0 = 0$ und $F_1 = 1$ kennt. Jede Zahl F_n ergibt sich aus den beiden vorherigen:



Um F_n für ein bestimmtes $n \in \mathbb{N}$ zu berechnen, könnte man die Rekurrenz in folgenden Algorithmus übersetzen:

INPUT: $n \in \mathbb{N}$
OUTPUT: F_n

1 falls $n = 0$, gib 0 zurück und stop.

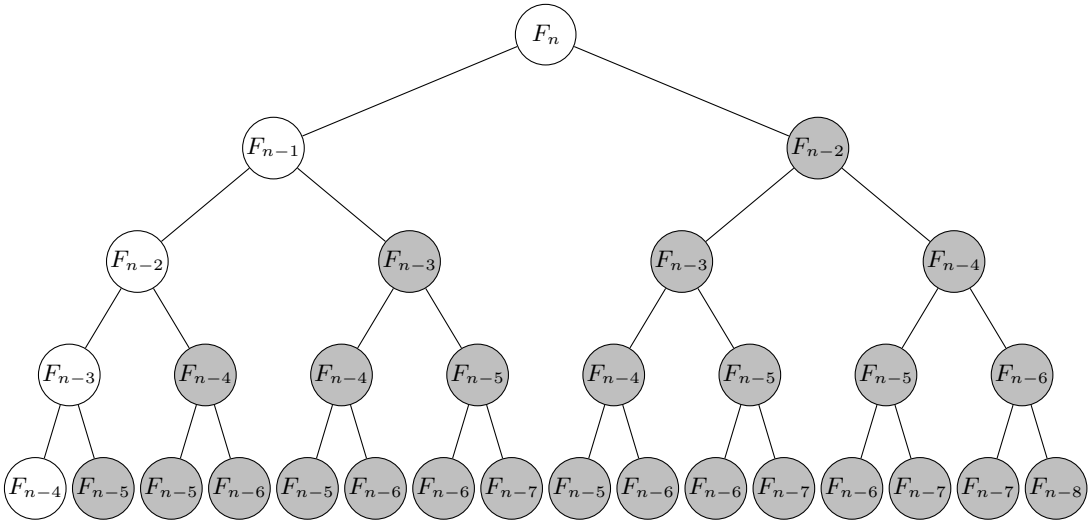
2 falls $n = 1$, gib 1 zurück und stop.

3 berechne F_{n-1} mit diesem Algorithmus

4 berechne F_{n-2} mit diesem Algorithmus

5 gib $F_{n-1} + F_{n-2}$ zurück.

Das ist allerdings ein sehr ineffizienter Algorithmus, weil dabei sehr viele Teilrechnungen sehr oft wiederholt werden. Zum Beispiel wird bei der Berechnung von F_{n-1} in Schritt 3 schon F_{n-2} und F_{n-3} berechnet, und es wäre besser, sich den Wert von F_{n-2} zu merken, statt ihn in Schritt 4 noch einmal frisch zu berechnen. Beachten Sie, wie viel Rechenleistung wir dadurch einsparen:



Alle grau markierten Knoten entsprechen einer überflüssigen Addition. Deren Anteil ist signifikant: Wenn man 10^9 Additionen pro Sekunde durchführen kann, bräuchte der naive Algorithmus immer noch fast 7000 Jahre, um den Wert

$$F_{100} = 354224848179261915075$$

auszurechnen. Wenn man dagegen die Zwischenergebnisse abspeichert und wiederverwertet, hat man insgesamt nur 100 Additionen durchzuführen und braucht nur 10^{-7} Sekunden.

3. Die Catalan-Zahlen C_n erfüllen auch eine Rekurrenz, die man leicht aus der Struktur der Binärbäume herleiten kann: Ein Binärbaum mit $n+1$ inneren Knoten hat eine Wurzel und zwei Unterbäume mit insgesamt n inneren Knoten. Diese Knoten können sich irgendwie auf den linken und den rechten Unterbaum aufteilen. Wenn der linke Unterbaum k innere Knoten hat, dann muss der rechte $n-k$ haben. Es gibt C_k Binärbäume mit k inneren Knoten und C_{n-k} Binärbäume mit $n-k$ inneren Knoten. Es gibt dann $C_k C_{n-k}$ viele Kombinationen von einem Baum mit k Knoten und einem Baum mit $n-k$ Knoten. (Beachte: $|A \times B| = |A| \cdot |B|$.) Weil k jeden beliebigen Wert aus $\{0, 1, \ldots, n\}$ annehmen kann, erhalten wir die Rekurrenz

$$C_{n+1} = \sum_{k=0}^{n} C_k C_{n-k},$$

die für alle $n \in \mathbb{N}$ gilt. Auch diese Rekurrenz legt zusammen mit dem Startwert $C_0 = 1$ alle Werte C_n fest:

$$C_1 = C_0 C_0 = 1$$

$$
\begin{aligned}
C_2 &= C_1C_0 + C_0C_1 = 2\\
C_3 &= C_2C_0 + C_1C_1 + C_0C_2 = 5\\
C_4 &= C_3C_0 + C_2C_1 + C_1C_2 + C_0C_3 = 14\\
&\vdots
\end{aligned}
$$

4. Die Bell-Zahlen erfüllen ebenfalls eine Rekurrenz, bei der in die Berechnung eines Werts alle vorherigen Werte eingehen. Es gilt

$$\mathbf{B}_{n+1} = \sum_{k=0}^{n} \binom{n}{k} \mathbf{B}_k$$

für alle $n \in \mathbb{N}$.

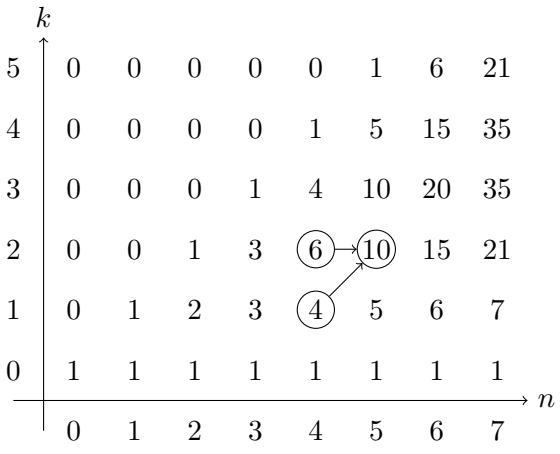
5. Für den Binomialkoeffizienten $\binom{n}{k}$ gibt es eine Rekurrenz, in der beide Variablen variiert werden. Es gilt

$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$$

für alle positiven Zahlen $n, k \in \mathbb{N}$.

Die Gleichung hat folgende kombinatorische Erklärung. Für eine k-elementige Teilmenge von $\{1, \ldots, n\}$ gibt es zwei Möglichkeiten: entweder sie enthält n oder sie enthält n nicht. Es gibt $\binom{n-1}{k}$ viele Teilmengen von $\{1, \ldots, n-1\}$ mit k Elementen, und jede davon ist auch eine Teilmenge von $\{1, \ldots, n-1, n\}$, die n nicht enthält. Darüber hinaus gibt es $\binom{n-1}{k-1}$ viele Teilmengen von $\{1, \ldots, n-1\}$ mit $k-1$ Elementen, und jede davon ergibt durch Hinzufügen von n eine Teilmenge von $\{1, \ldots, n\}$ mit k Elementen.

Mit der obigen Rekurrenz lässt sich jeder Wert von $\binom{n}{k}$ bequem berechnen. Da nun aber zwei Variablen im Spiel sind, brauchen wir mehr Anfangswerte. Die wiederholte Anwendung der Rekurrenz führt jeden Wert $\binom{n}{k}$ auf Werte der Form $\binom{i}{0}$ oder $\binom{0}{i}$ für gewisse $i \in \mathbb{N}$ zurück. Für $i > 0$ ist klarerweise $\binom{i}{0} = 1$ und $\binom{0}{i} = 0$. Es ist zweckmässig, außerdem $\binom{0}{0} := 1$ zu definieren.



Wie man sieht, liegen alle Paare (n, k), für die $\binom{n}{k}$ nicht Null ist, in dem dreieckigen Gebiet, das nach unten von der horizontalen Achse und nach oben von der Diagonalen begrenzt wird. Man spricht vom Pascal-Dreieck.

6. Die Stirling-Zahlen erfüllen ähnliche Rekurrenzen wie der Binomialkoeffizient. Es gilt

$$\begin{bmatrix} n \\ k \end{bmatrix} = (1-n)\begin{bmatrix} n-1 \\ k \end{bmatrix} + \begin{bmatrix} n-1 \\ k-1 \end{bmatrix} \quad \text{und} \quad \begin{Bmatrix} n \\ k \end{Bmatrix} = k\begin{Bmatrix} n-1 \\ k \end{Bmatrix} + \begin{Bmatrix} n-1 \\ k-1 \end{Bmatrix}$$

für alle positiven $n, k \in \mathbb{N}$. Als Anfangswerte kann man $\begin{bmatrix} i \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ i \end{bmatrix} = 0$ für $i > 0$ und $\begin{bmatrix} 0 \\ 0 \end{bmatrix} = 1$, sowie $\begin{Bmatrix} i \\ 0 \end{Bmatrix} = \begin{Bmatrix} 0 \\ i \end{Bmatrix} = 0$ für $i > 0$ und $\begin{Bmatrix} 0 \\ 0 \end{Bmatrix} = 1$ nehmen. Dann lassen sich alle weiteren Zahlen nacheinander mit den Rekurrenzen berechnen.

k \ n	0	1	2	3	4	5	6
5	0	0	0	0	0	1	−15
4	0	0	0	0	1	−10	85
3	0	0	0	1	−6	35	−225
2	0	0	1	−3	11	−50	274
1	0	1	−1	2	−6	24	−120
0	1	0	0	0	0	0	0

Stirlingzahlen erster Art

k \ n	0	1	2	3	4	5	6
5	0	0	0	0	0	1	15
4	0	0	0	0	1	10	65
3	0	0	0	1	6	25	90
2	0	0	1	3	7	15	31
1	0	1	1	1	1	1	1
0	1	0	0	0	0	0	0

Stirlingzahlen zweiter Art

7. Die Partitionszahlen $p_k(n)$ erfüllen eine Rekurrenz, die von wieder etwas anderer Gestalt ist; es gilt

$$p_k(n) = p_k(n-k) + p_{k-1}(n-1)$$

für alle $n, k \in \mathbb{N}$, wenn man $p_0(0) = 1$ und ansonsten $p_k(n) = 0$ für $k \leq 0$ oder $n \leq 0$ definiert. Hier greift der erste Term auf der rechten Seite nicht auf einen Wert in der unmittelbaren Nachbarschaft zu, sondern auf einen Wert, der k Schritte zurück liegt. Auch damit kann man alle Werte von $p_k(n)$ nacheinander berechnen.

k \ n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
5	0	0	0	0	0	1	1	2	3	5	7	10	13	18	23	30
4	0	0	0	0	1	1	2	3	5	6	9	11	15	18	23	27
3	0	0	0	1	1	2	3	4	5	7	8	10	12	14	16	19
2	0	0	1	1	2	2	3	3	4	4	5	5	6	6	7	7
1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8. Sei $\Omega = \{\mathtt{a}, \mathtt{b}\}$ und sei $L \subseteq \Omega^*$ die Menge aller Wörter, die nicht `aba` als Teilwort enthalten. Sei u_n die Anzahl der Wörter der Länge n in L. Es gilt also

$$\begin{aligned}
u_0 &= 1 && \{\epsilon\} \\
u_1 &= 2 && \{\mathtt{a}, \mathtt{b}\} \\
u_2 &= 4 && \{\mathtt{aa}, \mathtt{ab}, \mathtt{ba}, \mathtt{bb}\} \\
u_3 &= 7 && \{\mathtt{aaa}, \mathtt{aab}, \mathtt{abb}, \mathtt{baa}, \mathtt{bab}, \mathtt{bba}, \mathtt{bbb}\} \\
u_4 &= 12 && \{\mathtt{aaaa}, \mathtt{aaab}, \mathtt{aabb}, \mathtt{abba}, \mathtt{abbb}, \mathtt{baaa}, \\
&\vdots && \quad \mathtt{baab}, \mathtt{babb}, \mathtt{bbaa}, \mathtt{bbab}, \mathtt{bbba}, \mathtt{bbbb}\}
\end{aligned}$$

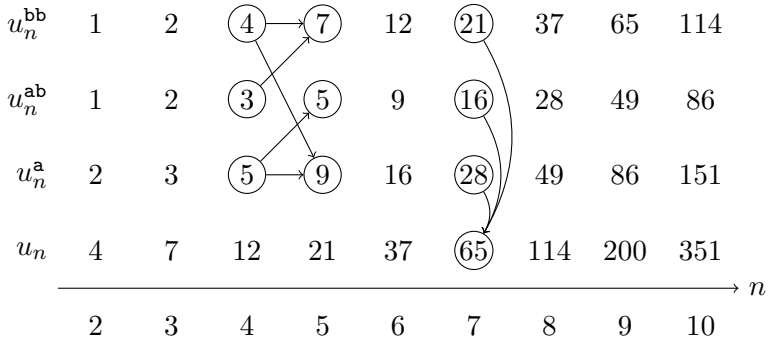
Eine Rekurrenz für die Zahlen u_n drängt sich nicht unmittelbar auf. Man kann sich helfen, indem man geeignete Hilfsgrößen betrachtet. Seien $u_n^{\mathtt{a}}, u_n^{\mathtt{ab}}, u_n^{\mathtt{bb}}$ die Anzahlen der Wörter der Länge n in L, die mit dem Teilwort enden, das im oberen Index angegeben ist. Für $n \geq 2$ gilt dann $u_n = u_n^{\mathtt{a}} + u_n^{\mathtt{ab}} + u_n^{\mathtt{bb}}$.

Die Hilfsgrößen erfüllen für $n \geq 2$ die folgenden Gleichungen:

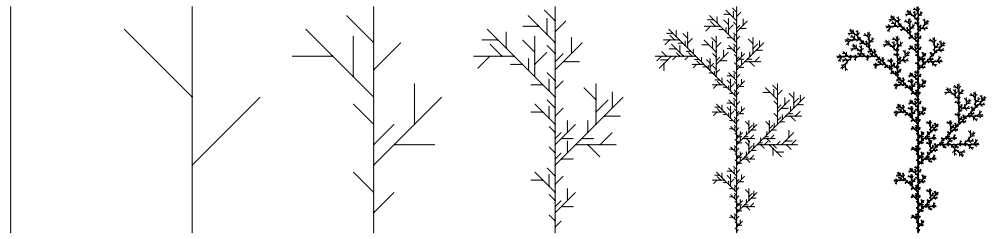
$$\begin{aligned}
u_{n+1}^{\mathtt{bb}} &= u_n^{\mathtt{ab}} + u_n^{\mathtt{bb}} \\
u_{n+1}^{\mathtt{ab}} &= u_n^{\mathtt{a}} \\
u_{n+1}^{\mathtt{a}} &= u_n^{\mathtt{a}} + u_n^{\mathtt{bb}}
\end{aligned}$$

Zum Beispiel ergeben sich alle Wörter der Länge $n+1$, die auf `aa` enden, aus den Wörtern der Länge n, die auf `aa` oder `ba` enden, indem man ein `a` hinten anfügt. Bei der dritten Gleichung ist auf der rechten Seite der Term $u_n^{\mathtt{ab}}$ weggelassen, weil durch Anhängen eines `a` an ein Wort, das auf `ab` endet, ein Wort mit dem verbotenen Teilwort `aba` entstehen würde.

Die drei Gleichungen für $u_n^{\mathtt{a}}, u_n^{\mathtt{ab}}, u_n^{\mathtt{bb}}$ bilden ein sogenanntes *Rekurrenzensystem*. Obwohl die drei Größen darin wechselseitig voneinander abhängen, lassen sich die Zahlen damit nacheinander ausrechnen.



9. Rekurrenzen müssen sich nicht unbedingt auf Zahlen beziehen. Es kann zum Beispiel auch um eine Abfolge von geometrischen Figuren gehen, bei der jede Figur sich nach einem bestimmten festen Bildungsgesetz aus der vorherigen konstruieren lässt. Man könnte zum Beispiel mit einem einzelnen Liniensegment beginnen (Anfangswert) und in jedem Schritt jedes Liniensegment der vorherigen Figur durch die Teilfigur ⅄ ersetzen (Rekurrenz):



Mit nur geringfügig komplizierteren Rekurrenzen kann man verblüffend realistische Darstellungen von Pflanzen, Flammen, Wolken oder Gebirgsreliefs erzeugen.

Rekurrenzen sind also ein bestimmter Typ von Gleichungen. Wegen der Vielfalt, mit der Rekurrenzen auftreten, ist es nicht ganz einfach, formal zu definieren, was genau eine Gleichung zu einer Rekurrenz macht. Eine wesentliche Eigenschaft einer Rekurrenz ist jedenfalls, dass man mit ihr eine Berechnung auf eine oder mehrere kleinere Berechnungen des gleichen Typs zurückführen kann, und zwar so, dass man nach endlich vielen Schritten bei Instanzen ankommt, für die die Berechnung einfach ist. Rekurrenzen sind auch nützlich, um Induktionsbeweise zu führen.

Beispiel. Wir zeigen die Identität von Cassini über Fibonacci-Zahlen: Für alle positiven $n \in \mathbb{N}$ gilt $F_{n+1}F_{n-1} - F_n^2 = (-1)^n$.

Induktionsanfang: Für $n = 1$ gilt $F_2F_0 - F_1^2 = -1 = (-1)^1$.

Induktionshypothese: Die Gleichung stimmt für ein $n \in \mathbb{N}$.

Induktionsschluss: Wir zeigen, dass die Gleichung dann auch für $n + 1$ anstelle von n gilt.

$$\begin{aligned} F_{(n+1)+1}F_{(n+1)-1} - F_{n+1}^2 &= F_{n+2}F_n - F_{n+1}F_{n+1} \\ &= (F_{n+1} + F_n)F_n - (F_n + F_{n-1})F_{n+1} \\ &= F_{n+1}F_n + F_n^2 - F_nF_{n+1}F_{n-1}F_{n+1} \\ &= -(F_{n+1}F_{n-1} - F_n^2) = (-1)^{n+1}, \end{aligned}$$

wie behauptet. Im zweiten Schritt wurde die Rekurrenz der Fibonacci-Zahlen verwendet, und im vierten Schritt die Induktionshypothese.

In ähnlicher Weise lassen sich auch die folgenden Aussagen über den Binomialkoeffizienten beweisen. Für manche dieser Aussagen gibt es auch anschauliche kombinatorische Beweise.

Satz 14. Für alle $n, k \in \mathbb{N}$ gilt:

1. $(a+b)^n = \sum_{k=0}^{n} \binom{n}{k} a^k b^{n-k}$ (a, b beliebig)
2. $\binom{a+b}{n} = \sum_{k=0}^{n} \binom{a}{k}\binom{b}{n-k}$ (für $a, b \in \mathbb{N}$)
3. $\binom{n}{k} = \frac{n!}{k!(n-k)!}$ (falls $k \leq n$)
4. $\binom{n+1}{k} = \frac{n+1}{n+1-k}\binom{n}{k}$ (falls $n + 1 - k \neq 0$)
5. $\binom{n}{k+1} = \frac{n-k}{k+1}\binom{n}{k}$
6. $\binom{n+1}{k+1} = \frac{n+1}{k+1}\binom{n}{k}$
7. $\binom{n}{k} = \binom{n}{n-k}$ (falls $k \leq n$)
8. $\binom{n}{1} = \binom{n}{n-1} = 1$ (falls $n \geq 1$)
9. $\binom{n}{2} = \frac{1}{2}n(n-1)$
10. $\binom{n}{k} = \sum_{i=0}^{k} \left[{k \atop i}\right] \frac{n^i}{k!}$
11. $C_n = \frac{1}{n+1}\binom{2n}{n}$
12. $F_n = \sum_{k=0}^{n} \binom{n-k}{k}$

Beweis.

1. Induktionsanfang $n = 0$: $(a+b)^0 = 1 = \sum_{k=0}^{0} \binom{0}{k} a^k b^{0-k}$ stimmt.

   Induktionshypothese: Die Gleichung stimmt für ein bestimmtes $n \in \mathbb{N}$.

   Induktionsschluss: Wir zeigen, dass sie dann auch für $n+1$ anstelle von n gilt.

$$\begin{aligned}\sum_{k=0}^{n+1} \binom{n+1}{k} a^k b^{(n+1)-k} &= \sum_{k=0}^{n+1} \binom{n}{k} a^k b^{(n+1)-k} + \sum_{k=0}^{n+1} \binom{n}{k-1} a^k b^{(n+1)-k} \\ &= \sum_{k=0}^{n} \binom{n}{k} a^k b^{(n+1)-k} + \sum_{k=0}^{n} \binom{n}{k} a^{k+1} b^{n-k} \\ &= b(a+b)^n + a(a+b)^n = (a+b)^{n+1}.\end{aligned}$$

   Im ersten Schritt wurde die Rekurrenz des Pascal-Dreiecks angewendet. Im zweiten Schritt wurde verwendet $\binom{n}{n+1} = \binom{n}{-1} = 0$, und bei der zweiten Summe wurde der Index k um eins verschoben.

2. Wir geben einen kombinatorischen Beweis. Die linke Seite ist die Anzahl der Teilmengen von $\{1, \ldots, a+b\}$ mit genau n Elementen. Jede solche Teilmenge kann man zerlegen in eine k-elementige Teilmenge von $\{1, \ldots, a\}$ und eine $(n-k)$-elementige Teilmenge von $\{a+1, \ldots, b\}$. Für jedes k gibt es $\binom{a}{k}\binom{b}{n-k}$ solcher Paare. Weil k jeden Wert von 0 bis n annehmen kann, gibt es insgesamt $\sum_{k=0}^{n} \binom{a}{k}\binom{b}{n-k}$ Teilmengen von $\{1, \ldots, a+b\}$ mit genau $k + (n-k) = n$ Elementen.

3. Kombinatorischer Beweis: Um eine k-elementige Teilmenge von $\{1, \ldots, n\}$ zu konstruieren, kann man nacheinander k Elemente wählen. Für das erste gibt es n Möglichkeiten, nachdem eine Wahl getroffen ist und das entsprechende Element entfernt wurde, bleiben $n-1$ Möglichkeiten für die Wahl des zweiten Elements, und so weiter, bis am Ende $n-k+1$ Möglichkeiten für die Wahl des k-ten Elements bleiben. Insgesamt gibt es also $n(n-1)\cdots(n-k+1) = n!/(n-k)!$ viele Möglichkeiten, k Elemente nacheinander aus der Menge $\{1, \ldots, n\}$ zu entfernen. Da dabei die Reihenfolge berücksichtigt wird, haben wir jede k-elementige Teilmenge $|S_k| = k!$ mal gezählt. Die Zahl der Teilmengen ist deshalb nur $\frac{n!}{k!(n-k)!}$, wie behauptet.

   Alternativer algebraischer Beweis: Induktion über das Pascal-Dreieck.

   Induktionsanfang: Es gilt $\binom{n}{n} = 1 = \frac{n!}{n!(n-n)!}$ und $\binom{n}{0} = 1 = \frac{n!}{0!(n-0)!}$ für alle $n \in \mathbb{N}$ (beachte: nach Definition gilt $0! = \binom{0}{0} = 1$). Damit gilt die Aussage an den Seiten des Pascal-Dreiecks.

   Induktionshypothese: $n \in \mathbb{N}$ sei so, dass die Aussage für $n-1$ und beliebige $k \leq n-1$ anstelle von n und k gilt.

   Induktionsschluss: Wir zeigen, dass sie dann auch für n und beliebige $k \leq n-1$ gilt.

$$\begin{aligned}\binom{n}{k} &= \binom{n-1}{k} + \binom{n-1}{k-1} \\ &= \frac{(n-1)!}{k!((n-1)-k)!} + \frac{(n-1)!}{(k-1)!((n-1)-(k-1))!}\end{aligned}$$

$$= \frac{(n-1)!(n-k)}{k!(n-k)!} + \frac{(n-1)!k}{k!(n-k)!}$$
$$= \frac{(n-1)!(n-k+k)}{k!(n-k)!} = \frac{n!}{k!(n-k)!},$$

wie behauptet. Im zweiten Schritt wurde die Induktionshypothese verwendet. Im dritten Schritt wurde der erste Bruch mit $n-k$ und der zweite mit k erweitert.

4.–12. Übung. ∎

Die Identität in Punkt 1 nennt man die *binomische Formel* (engl. *binomial theorem*). Punkt 2 nennt man die *Vandermonde-Identität*.

In der Informatik begegnet man Rekurrenzen vor allem im Zusammenhang mit der Komplexitätsabschätzung von Algorithmen. Die Komplexität eines Algorithmus gibt an, wie viele Grundoperationen der Algorithmus in Abhängigkeit von der Größe der Eingabe braucht, um die Ausgabe zu berechnen. Was dabei als Grundoperation betrachtet wird und wie die Größe der Eingabe gemessen wird, kommt auf das vorliegende Problem an.

Beispiel. (Binärsuche) Betrachten Sie den folgenden Algorithmus, der feststellt, ob eine gegebene Zahl in einem gegebenen Bereich einer geordneten Liste $(a_0, \ldots, a_{n-1}) \in \mathbb{N}^n$ mit $a_0 \leq a_1 \leq \cdots \leq a_{n-1}$ enthalten ist.

INPUT: $b \in \mathbb{N}$ und $l, r \in \{0, \ldots, n\}$ mit $l < r$.
OUTPUT: $k \in \{l, \ldots, r-1\}$ so dass $b = a_k$, oder -1, falls es kein solches k gibt.

1 wenn $l = r$ ist,
2 gib $k = -1$ zurück.
3 setze $m = \lfloor (l+r)/2 \rfloor$
4 wenn $a_m = b$ ist
5 gib $k = m$ zurück.
6 wenn $b < a_m$ ist
7 wende den Algorithmus mit l, m statt l, r an und gib das Ergebnis zurück.
8 anderenfalls
9 wende den Algorithmus auf m, r statt l, r an und gib das Ergebnis zurück.

Bei diesem Algorithmus bietet es sich an, n als Problemgröße aufzufassen, und die Anzahl der Vergleichsoperation $b < a_m$, die in Schritt 6 durchgeführt werden, als Maß für die Komplexität zu nehmen. Wie viele solcher Vergleiche führt der Algorithmus durch, wenn er mit $l = 0$ und $r = n$ gestartet wird? Die Antwort wird nicht nur von der Problemgröße abhängen, sondern z.B. auch davon, ob b in der Liste $(a_0, \ldots, a_{n-1})$ enthalten ist oder nicht. Im besten Fall haben wir gleich im ersten Versuch $a_m = b$ und brauchen Schritt 6 gar nicht durchzuführen. Der schlechteste Fall tritt ein, wenn b nicht in der Liste enthalten ist. Dann grenzt der Algorithmus den Bereich, in dem b liegen müsste, immer weiter ein, bis der in Frage kommende Bereich leer ist. Betrachten wir diesen sogenannten *worst case* etwas genauer. Wenn $T(n)$ die Zahl der Vergleiche ist, die durchgeführt werden, dann gilt die Abschätzung

$$T(n) \leq 1 + T(\lceil n/2 \rceil)$$

für alle $n \in \mathbb{N}$, denn in jedem Rekursionsschritt reduziert der Algorithmus das Problem im Wesentlichen auf ein Problem der halben Größe. Wenn wir zur Vereinfachung annehmen, dass n eine Potenz von 2 ist, also $n = 2^m$ für ein $m \in \mathbb{N}$, dann gilt $T(2^m) \leq 1 + T(2^{m-1})$ und $T(2^0) = 1$. Mit Induktion kann man jetzt leicht zeigen, dass $T(2^m) \leq m + 1$ gilt, also $T(n) \leq 1 + \log_2(n)$.

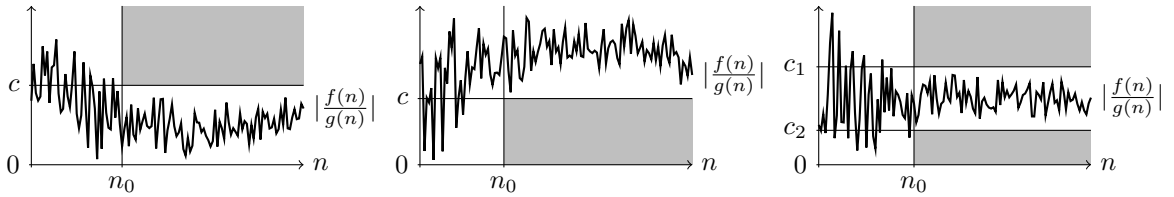
Die Analyse in obigem Beispiel gibt Anlass zu folgenden Beobachtungen:

- Die Komplexität hängt im allgemeinen nicht nur von der Größe des Problems ab, sondern von der konkreten Instanz. Für jede Problemgröße kann man aber die maximale Anzahl der Operationen zählen, die für Instanzen dieser Größe nötig sind. Das ist die *worst case* Komplexität des Algorithmus.
- Selbst diese *worst case* Komplexität lässt sich oft nicht exakt bestimmen. Das ist aber meistens auch gar nicht nötig. Es genügt schon, wenn man eine gute Abschätzung der Kosten angeben kann, denn ob ein Algorithmus z.B. $\log_2(n)$ oder $\log_2(n) + 1$ Operationen braucht, ist ein Unterschied, der für große n nicht ins Gewicht fällt.

Um grobe Abschätzungen der Komplexität trotzdem präzise formulieren zu können, hat man die folgende Notation erfunden.

Definition 23. Seien $f, g \colon \mathbb{N} \to \mathbb{N}$ zwei Funktionen. Man schreibt

1. $f(n) = \mathrm{O}(g(n))$, falls gilt: $\exists\, c > 0\ \exists\, n_0 \in \mathbb{N}\ \forall\, n \geq n_0 : |f(n)| \leq c|g(n)|$
2. $f(n) = \Omega(g(n))$, falls gilt: $g(n) = \mathrm{O}(f(n))$
3. $f(n) = \Theta(g(n))$, falls gilt: $f(n) = \mathrm{O}(g(n))$ und $f(n) = \Omega(g(n))$



$f(n) = \mathrm{O}(g(n))$ $f(n) = \Omega(g(n))$ $f(n) = \Theta(g(n))$

Beispiel.

1. $f(n) = \mathrm{O}(g(n))$ heißt, dass die Funktion f für große Werte von n höchstens so schnell wächst wie ein konstantes Vielfaches von g. Zum Beispiel gilt $3n^2+7n+\sqrt{n}-5 = \mathrm{O}(n^2)$, $n^{1000000000} = \mathrm{O}(2^n)$, $\log(n)^{10000000} = \mathrm{O}(n)$, $2^n = \mathrm{O}(3^n)$, $\frac{1}{n} = \mathrm{O}(1)$, etc.

   Beachten Sie insbesondere, dass zum Beispiel neben $n^2 = \mathrm{O}(n^2)$ auch $n^2 = \mathrm{O}(n^3)$ gilt, weil durch die Notation $= \mathrm{O}(\cdot)$ nur eine Beschränkung nach oben ausgedrückt wird.

2. $f(n) = \Omega(g(n))$ heißt, dass die Funktion f für große Werte von n mindestens so schnell wächst wie ein konstantes Vielfaches von g. Zum Beispiel gilt $3n^2+7n+\sqrt{n}-5 = \Omega(n^2)$, $2^n = \Omega(n^{1000000000})$, $n = \Omega(\log(n)^{100000000})$, $3^n = \Omega(2^n)$, $1 = \Omega(\frac{1}{n})$, etc.

3. $f(n) = \Theta(g(n))$ heißt, dass die Funktion f für große Werte von n genau so schnell wächst wie ein konstantes Vielfaches von g. Zum Beispiel gilt $3n^2 + 7n + \sqrt{n} - 5 = \Theta(n^2)$, aber es gilt weder $n^2 = \Theta(n)$ noch $n^2 = \Theta(n^3)$.

4. Es gilt $2n^3 = \mathrm{O}(n^3)$ und $2n^3 = \mathrm{O}(n^4)$ und $5n^3 = \mathrm{O}(n^3)$, aber daraus folgt natürlich weder, dass $\mathrm{O}(n^3) = \mathrm{O}(n^4)$ ist, noch dass $2n^3 = 5n^3$ gilt. Vielmehr ist das Symbol „=“ hier nicht als die übliche Gleichheitsrelation zu verstehen, sondern als Bestandteil der Notation „ $= \mathrm{O}(\cdot)$“, für die typischen Eigenschaften von Äquivalenzrelatioen nicht unbedingt gelten. Wenn man ganz präzise sein will, sollte man $\mathrm{O}(g)$ als eine Menge von Funktionen auffassen, und $f \in \mathrm{O}(g)$ statt $f(n) = \mathrm{O}(g(n))$ usw. schreiben.

5. Die Schreibweise $f(n) = \mathrm{O}(n^3)$ lässt sich lesen als „$f(n)$ wächst (höchstens) von der Ordnung n^3“. Mit der Schreibeweise $f(n) = u(n) + \mathrm{O}(g(n))$ meint man $f(n) - u(n) = \mathrm{O}(g(n))$ im Sinne obiger Definition, z.B. lässt sich $f(n) = 5n^3 + \mathrm{O}(n^2)$ lesen als „$f(n)$ wächst wie $5n^3$ plus Terme von (höchstens) der Ordnung n^2“.

6. Angenommen, wir wollen zwei Algorithmen vergleichen. Algorithmus A habe die Laufzeit $T_1(n) = \mathrm{O}(n^2)$ und Algorithmus B die Laufzeit $T_2(n) = \mathrm{O}(n^3)$. Daraus lässt sich **nicht** schließen, dass Algorithmus A effizienter ist als Algorithmus B. Es könnte nämlich sein, dass z.B. $T_1(n) = n^2 = \mathrm{O}(n^2)$ und $T_2(n) = n = \mathrm{O}(n^3)$ gilt. Noch einmal: Durch die Notation $= \mathrm{O}(\cdot)$ wird nur eine Beschränkung nach oben ausgedrückt. Wenn Sie zeigen wollen, dass Algorithmus A für große Inputs besser ist als Algorithmus B, brauchen Sie eine Abschätzung der Form $T_1(n) = \mathrm{O}(n^2)$, $T_2(n) = \Omega(n^3)$.

7. Abschätzungen, die mit der Notation aus Def. 23 angegeben werden, beziehen sich immer ausschließlich auf große Werte von n. Das ist sinnvoll, weil ein Computer kleine Problemgrößen typischerweise sehr schnell lösen kann und erst für sehr große Problemgrößen die Frage relevant wird, wie lange eine Rechnung dauern wird. Wenn eine Rechnung 100 Millisekunden braucht, werden wir es selten für nötig halten, den Code so weit zu optimieren, dass er nur 10 Millisekunden braucht. Bei einer großen Rechnung, die z.B. 10 Jahre Rechenzeit braucht, wäre dagegen durchaus zu fragen, ob sich dasselbe Problem nicht auch mit nur einem Jahr Rechenzeit lösen lässt.

8. In der Notation von Def. 23 werden auch multiplikative Vielfache ignoriert, weil diese für große Werte von n nicht ins Gewicht fallen. Vergleichen Sie dazu die Werte der folgenden Funktionen für steigende Funktionswerte: Obwohl die Konstanten für die unteren Funktionen sehr vorteilhaft gewählt sind, werden sie doch früher oder später von den oberen Funktionen unterboten.

n	1	10	100	1000	10000	100000
1	1	1	1	1	1	1
$\frac{1}{10}\log_2(n)$	0	0.33	0.66	1	1.33	1.66
$\frac{1}{100}n$	0.01	0.1	1	10	100	1000
$\frac{1}{1000}n\log_2(n)$	0	0.033	0.66	9.97	132.9	1660.96
$\frac{1}{10000}n^2$	0.0001	0.01	1	100	10000	1000000
$\frac{1}{100000}2^n$	0.00002	0.01	$1.27 \cdot 10^{25}$	$1.07 \cdot 10^{296}$	$2.00 \cdot 10^{3005}$	$9.99 \cdot 10^{30097}$

Beachten Sie vor allem, wie schnell die letzte Funktion wächst. Zum Beispiel sind $1.27 \cdot 10^{25}$ Millisekunden eine sehr lange Zeit, nämlich $4 \cdot 10^{14}$ Jahre. Zum Vergleich: das

geschätzte Alter des Universums beträgt „nur“ $13.8 \cdot 10^9$ Jahre. Ein Algorithmus mit einer Laufzeit von $O(n^c)$ ist deshalb fast immer einem Algorithmus mit einer Laufzeit von $\Omega(u^n)$ vorzuziehen.

Die oben vorgestellte Binärsuche ist ein Beispiel für einen *divide-and-conquer-Algorithmus.* Divide and conquer ist ein bewährtes Prinzip zum Entwurf effizienter Algorithmen. Die Grundidee besteht darin, ein Problem in mehrere kleinere Probleme desselben Typs zu zerlegen, diese rekursiv mit dem gleichen Algorithmus zu lösen, und dann die einzelnen Ergebnisse zum Ergebnis des ursprünglichen Problems zusammenzusetzen. Die Komplexitätsanalyse solcher Algorithmen führt immer auf eine Rekurrenz der folgenden Form:

$$T(n) = a\,T(n/b) + f(n).$$

Dabei ist $T(n)$ die Laufzeit des Algorithmus, a ist die Anzahl der Teilprobleme (normalerweise ist a unabhängig von n), in die das Problem zerlegt wird, n/b ist die Größe dieser Teilprobleme (normalerweise ist b unabhängig von n), und $f(n)$ ist die Zeit, die zur Zerlegung des Problems in die Teilprobleme sowie zur Kombination der Ergebnisse der Teilprobleme zum Gesamtergebnis benötigt wird. Mit Hilfe des folgenden Satzes kann man viele dieser Rekurrenzen direkt lösen.

Satz 15. (Master Theorem) Seien $T\colon \mathbb{N} \to \mathbb{N}$ eine Funktion, $a \geq 1$ und $b > 1$ Konstanten, $f\colon \mathbb{N} \to \mathbb{N}$ eine Funktion, und es gelte

$$T(n) = a\,T(n/b) + f(n)$$

für alle $n \in \mathbb{N}$. Seien $k \geq 0$ und $\epsilon > 0$ Konstanten und schreibe $c = \log_b(a)$.

1. Wenn $f(n) = O(n^{c-\epsilon})$, dann gilt $T(n) = \Theta(n^c)$.
2. Wenn $f(n) = O(n^c \log(n)^k)$, dann gilt $T(n) = O(n^c \log(n)^{k+1})$.
3. Wenn $f(n) = \Omega(n^c \log(n)^k)$, dann gilt $T(n) = \Omega(n^c \log(n)^{k+1})$.
4. Wenn $f(n) = \Omega(n^{c+\epsilon})$, dann gilt $T(n) = \Omega(n^{c+\epsilon})$.

Beispiel.

1. Für die Komplexität $T(n)$ der Binärsuche hatten wir oben die Rekurrenz $T(n) = T(n/2) + \Theta(1)$ hergeleitet. Hier ist $a = 1$, $b = 2$, $c = \log_2(1) = 0$. Wir sind in den Fällen 2/3 mit $k = 0$ und bekommen $T(n) = \Theta(\log(n))$.
2. Ein klassischer Sortieralgorithmus sortiert eine gegebene Liste $(a_1, \ldots, a_n)$ von Zahlen, indem er zunächst die rechte Hälfte und die linke Hälfte der Liste rekursiv sortiert und dann die beiden separat sortierten Halblisten zu einer vollständig sortierten Liste verschmelzt. Das Verschmelzen lässt sich mit einer Komplexität von $O(n)$ realisieren. Für die Komplexität $T(n)$ des Sortieralgorithmus erhalten wir also $T(n) = 2T(n/2) + O(n)$. In diesem Fall ist $a = 2$, $b = 2$, $c = \log_2(2) = 1$. Wir sind wieder in den Fällen 2/3 mit $k = 0$ und bekommen $T(n) = \Theta(n \log(n))$.
3. Der Standardalgorithmus zum Multiplizieren zweier ganzer Zahlen mit je n Ziffern hat die Komplexität $\Omega(n^2)$. Ein effizienterer Algorithmus wurde von Karatsuba gefunden. Dieser teilt die beiden gegebenen Zahlen mit n Ziffern in vier Zahlen mit jeweils nur $n/2$ Ziffern auf, führt mit diesen Zahlen drei Multiplikationen durch, und rekonstruiert aus

den Ergebnissen das Produkt der beiden ursprünglichen Zahlen. Die Rekonstruktion kostet $O(n)$ Schritte. Für die Komplexität des Multiplikationsalgorithmus gilt deshalb $T(n) = 3T(n/2) + O(n)$. Jetzt ist $a = 3$, $b = 2$, $c = \log_2(3) \approx 1.58496$. Wir sind im Fall 1 und erhalten $T(n) = \Theta(n^{\log_2(3)}) = O(n^{1.585})$.

4. Für die Rekurrenz $T(n) = 2T(n/2) + \Omega(n^2)$ ist $a = 2$, $b = 2$, $c = \log_2(2) = 1$. Wir sind im Fall 4 und erhalten $T(n) = \Omega(n^2)$.

5. Auf eine Rekurrenz der Form $T(n) = 2T(\sqrt{n}) + O(n)$ ist der Satz nicht anwendbar.

10 Summation

11 Wahrscheinlichkeit

Index