

Diskrete Mathematik für Informatikstudien

505.006

SS 2020

© Sophie FRISCH, Mihyun KANG, Franz LEHNER,
Philipp SPRÜSSEL, Wolfgang WOESS

Inhaltsverzeichnis

Kapitel A. Zahlen, Kongruenzen, Verschlüsselung	A.1
1. Zahlen	A.3
2. Beweise	A.3
3. Teilbarkeit und der euklidische Algorithmus	A.6
4. Primzahlen	A.9
5. Äquivalenzrelationen und Partitionen	A.10
6. Rechnen mit Kongruenzklassen	A.12
7. Diophantische Gleichungen und der chinesische Restsatz	A.14
8. Halbgruppen und Gruppen	A.17
9. Der Satz von Euler-Fermat	A.23
10. Kryptographie	A.24
11. Das RSA-Verfahren	A.27
 Kapitel B. Grundlagen der Logik	 B.1
1. Aussagen und Junktoren	B.3
2. Die Sprache der Aussagenlogik	B.5
3. Exkurs: Induktive Strukturen	B.7
4. Aussagenlogik	B.9
5. Äquivalenz von aussagenlogischen Formeln	B.11
6. Konjunktive und disjunktive Normalform	B.17
7. Prädikatenlogik	B.23
 Kapitel C. Erzeugende Funktionen	 C.1
1. Kombinatorik	C.3
2. Abzählende Potenzreihen	C.7
3. Rekursionen	C.16
 Kapitel D. Graphen und Bäume	 D.1
1. Ungerichtete und gerichtete Graphen	D.3
2. Wege und Kreise in Graphen	D.4
3. Bäume	D.13
4. Matchings	D.19

- | | |
|--------------------|------|
| 5. Färbungen | D.22 |
| 6. Planare Graphen | D.24 |

Literatur:

- G. Baron – P. Kirschenhofer: *Einführung in die Mathematik für Informatiker*, Bände 1 und 3, Springer, Wien.
- N.L. Biggs, *Discrete Mathematics*. Oxford University Press.
- R. Diestel, *Graphentheorie*, 5. Aufl., Springer 2017.
- T. Emden-Weinert, S. Hougardy, B. Kreuter, H. J. Prömel, A. Steger, *Einführung in Graphen und Algorithmen*.
<http://www.or.uni-bonn.de/~hougardy/paper/ga.pdf>
- R. L. Graham, D. E. Knuth, O. Patashnik, *Concrete mathematics*. Addison-Wesley.
- D. E. Knuth, *The Art of Computer Programming*, Volumes 1–3, Addison - Wesley [für Fortgeschrittene!]
- S. B. Maurer, A. Ralston. *Discrete Algorithmic Mathematics*. A K Peters Ltd.
- K. H. Rosen. *Discrete mathematics and its applications*. McGraw-Hill.
- S. Singh, *The Code Book*, 2000.
- G. und S. Teschl, *Mathematik für Informatiker 1*, 4.Aufl., Springer 2013.
- W. D. Wallis, *A Beginner's Guide to Discrete Mathematics*. Birkhäuser, 2003.
- H. Wilf, *Generatingfunctionology*, Academic Press, 1990.

KAPITEL A

Zahlen, Kongruenzen, Verschlüsselung

Inhaltsangabe

1.	Zahlen	A.3
2.	Beweise	A.3
3.	Teilbarkeit und der euklidische Algorithmus	A.6
4.	Primzahlen	A.9
5.	Äquivalenzrelationen und Partitionen	A.10
6.	Rechnen mit Kongruenzklassen	A.12
7.	Diophantische Gleichungen und der chinesische Restsatz	A.14
8.	Halbgruppen und Gruppen	A.17
9.	Der Satz von Euler-Fermat	A.23
10.	Kryptographie	A.24
11.	Das RSA-Verfahren	A.27

1. Zahlen

(1.1) Zahlenbereiche. $\mathbb{N} = \{1, 2, 3, \dots\}$ natürliche Zahlen.

$$\mathbb{N}_0 = \mathbb{N} \cup \{0\}$$

$$\mathbb{Z} = \mathbb{N}_0 \cup \{-1, -2, -3, \dots\} \quad \text{ganze Zahlen.}$$

$$\mathbb{Q} = \left\{ \frac{n}{m} : m \in \mathbb{N}, n \in \mathbb{Z} \right\} \quad \text{rationale Zahlen.}$$

$$\mathbb{R} \quad \text{reelle Zahlen}$$

Eine ganze Zahl heißt *gerade*, wenn sie von der Form $2a$ ist mit $a \in \mathbb{Z}$. Zahlen von der Form $2a + 1$ (mit $a \in \mathbb{Z}$) heißen *ungerade*.

(1.2) Bemerkung. Wenn man es genau nimmt, sind die natürlichen Zahlen \mathbb{N} rekursiv bzw. induktiv durch die folgenden Axiome von Peano¹ (1889) charakterisiert:

- (1) $1 \in \mathbb{N}$
- (2) jede Zahl $n \in \mathbb{N}$ hat einen Nachfolger $n' \in \mathbb{N}$
- (3) wenn $n \in \mathbb{N}$, dann ist ihr Nachfolger $n' \neq 1$
- (4) wenn $m' = n'$, dann ist $m = n$
- (5) Sei $M \subseteq \mathbb{N}$ eine Teilmenge mit den folgenden Eigenschaften.
 - (i) $1 \in M$
 - (ii) Wenn $n \in M$, dann ist auch $n' \in M$
 dann ist $M = \mathbb{N}$

Statt n' schreibt man üblicherweise $n + 1$.

2. Beweise

Unter einem *direkten Beweis* versteht man Methoden, die Wahrheit einer Behauptung zu zeigen, indem man bereits bekannte Fakten und Sätze kombiniert, ohne weitere Annahmen zu treffen.

¹Giuseppe Peano (1858–1932)

(2.1) Beispiel. Man zeige: Das Produkt zweier ungerader Zahlen ist ebenfalls ungerade.

Betrachte zwei ungerade Zahlen x, y und schreibe sie in der Form $x = 2a + 1$ und $y = 2b + 1$. Dann erhält man als Produkt der beiden

$$x \cdot y = (2a + 1)(2b + 1) = 4ab + 2a + 2b + 1 = 2(2ab + a + b) + 1,$$

also wieder eine ungerade Zahl. \square

Eine der Methoden, direkte Beweise zu führen, ist der Beweis durch *vollständige Induktion*, welche sich aus dem fünften Axiom in Bemerkung 1.2 ergibt.

(2.2) Prinzip der vollständigen Induktion, „Dominoprinzip“. Für jedes $n \in \mathbb{N}$ sei $A(n)$ eine Aussage. Um zu zeigen, dass $A(n)$ für jedes $n \in \mathbb{N}$ wahr ist, kann man die folgenden zwei Schritte verwenden.

- (1) **Induktionsanfang.** Man zeigt, dass $A(1)$ wahr ist.
- (2) **Induktionsschritt** $n \rightarrow n + 1$. Man zeigt: Ist $A(n)$ wahr für ein bestimmtes n (*Induktionsannahme*), dann ist auch $A(n + 1)$ wahr.

(2.3) Beispiel. Man zeige für alle natürlichen Zahlen n

$$1 + 2 + \cdots + n = \frac{n(n + 1)}{2}.$$

Der Beweis erfolgt mittels vollständiger Induktion.

- Induktionsanfang: Für $n = 1$ lautet die Aussage

$$1 = \frac{1 \cdot 2}{2},$$

diese ist offensichtlich wahr.

- Induktionsschritt: Wir setzen vorraus (*Induktionsannahme*), dass

$$1 + 2 + \cdots + n = \frac{n(n + 1)}{2}$$

für ein bestimmtes n erfüllt ist und wollen zeigen, dass dann auch

$$1 + 2 + \cdots + (n + 1) = \frac{(n + 1)(n + 2)}{2}$$

gilt. Es ist

$$1 + 2 + \cdots + (n + 1) = (1 + 2 + \cdots + n) + (n + 1)$$

und damit wegen der Induktionsannahme

$$(1 + 2 + \cdots + n) + (n + 1) = \frac{n(n + 1)}{2} + (n + 1) = \frac{n(n + 1) + 2(n + 1)}{2}.$$

Herausheben von $n + 1$ führt zum gewünschten Resultat:

$$1 + 2 + \cdots + (n + 1) = \frac{(n + 1)(n + 2)}{2}.$$

□

Im Gegensatz zum direkten Beweis können zu Beginn eines *indirekten Beweises* zusätzliche Annahmen stehen. Die entstandenen Unsicherheiten werden eliminiert, bis nur mehr ein einziger gültiger Schluss übrig bleibt.

(2.4) Beispiel. Man zeige: Ist das Produkt zweier natürlicher Zahlen x, y gerade, dann ist mindestens eine der beiden Zahlen x, y gerade.

Wir beweisen die Aussage indirekt: Wären x, y beide ungerade, dann wäre auch $x \cdot y$ ungerade, wie wir in Beispiel 2.1 gesehen haben. Da das Produkt aber gerade ist, können nicht beide Zahlen ungerade sein, d.h., mindestens eine von ihnen ist gerade.

Beweis durch Widerspruch oder *Reductio ad absurdum* ist eine indirekte Beweisvariante, bei der das Gegenteil der zu beweisenden Aussage angenommen wird und daraus ein logischer Widerspruch konstruiert wird. Damit ist das Gegenteil der Aussage widerlegt und die Aussage dadurch bewiesen.

(2.5) Beispiel. Man zeige, dass $\sqrt{2}$ keine rationale Zahl ist.

Wir nehmen an, $\sqrt{2}$ wäre eine rationale Zahl, wir könnten sie also als Bruch schreiben. Durch Kürzen des Bruches könnte man erreichen, dass

$$\sqrt{2} = \frac{p}{q}$$

mit ganzen Zahlen p, q , von denen mindestens eine ungerade ist.

Nun gilt aber

$$2 = \frac{p^2}{q^2}$$

und damit $p^2 = 2q^2$. Insbesondere ist p^2 eine gerade Zahl. Dann ist aber auch p gerade (wegen Beispiel 2.4) und kann in der Form $p = 2r$ mit $r \in \mathbb{Z}$ geschrieben werden.

Setzt man dies in $p^2 = 2q^2$ ein, ergibt sich $4r^2 = 2q^2$, also $2r^2 = q^2$. Wie zuvor für p können wir nun folgern, dass auch q gerade ist. Dies ist ein Widerspruch dazu, dass mindestens eine der beiden Zahlen p, q ungerade ist. Also war unsere ursprüngliche Annahme, dass $\sqrt{2}$ rational ist, falsch. \square

3. Teilbarkeit und der euklidische Algorithmus

(3.1) Definition. Seien $m \in \mathbb{N}, n \in \mathbb{Z}$.

m teilt n , in Zeichen $m | n$ oder $m \backslash n$, wenn eine ganze Zahl k existiert, sodass

$$n = k m .$$

Es gilt klarerweise $m | 0$ und $m | m$ für alle $m \in \mathbb{N}$, sowie $1 | n$ für alle $n \in \mathbb{Z}$. Weiters gilt für $m, n \in \mathbb{N}$ (also beide positiv)

Falls $m | n$ und $n | m$, dann ist $m = n$.

(3.2) Satz. (Divisionssatz) Für $m \in \mathbb{N}$ und $n \in \mathbb{Z}$ gibt es eindeutig bestimmte Zahlen $q \in \mathbb{Z}$ und $r \in \{0, 1, \dots, m - 1\}$ (Rest), sodass

$$n = q m + r .$$

Für $x \in \mathbb{R}$ bezeichnen wir mit $\lfloor x \rfloor$ (bzw. $\lceil x \rceil$) die nächstkleinere (bzw. nächstgrößere) ganze Zahl:

$$\lfloor x \rfloor = n \in \mathbb{Z} \text{ genau dann, wenn } n \leq x < n + 1 .$$

$$\lceil x \rceil = n \in \mathbb{Z} \text{ genau dann, wenn } n - 1 < x \leq n .$$

Im Divisionssatz ist daher

$$q = \lfloor n/m \rfloor \text{ und } r = n - q m .$$

(3.3) Definition. Seien $m, n \in \mathbb{N}_0$, nicht beide gleichzeitig = 0. Der *größte gemeinsame Teiler* von m und n ist

$$\text{ggT}(m, n) = \max\{k \in \mathbb{N} : k \mid m \text{ und } k \mid n\}.$$

Triviale Beziehungen: für $m \in \mathbb{N}$, $n \in \mathbb{N}_0$ und $\ell \in \mathbb{N}$ ist

$$(3.4) \quad \begin{aligned} \text{ggT}(m, 0) &= m, & \text{ggT}(m, 1) &= 1, & \text{ggT}(m, m) &= m, \text{ und} \\ && && \text{ggT}(\ell m, \ell n) &= \ell \text{ ggT}(m, n). \end{aligned}$$

Aus (3.4) folgt auch:

$$\text{Falls } \text{ggT}(m, n) = d, \text{ dann ist } \text{ggT}\left(\frac{m}{d}, \frac{n}{d}\right) = 1.$$

Wenn $\text{ggT}(m, n) = 1$, dann heißen m und n teilerfremd oder relativ prim.

(3.5) Definition. Das *kleinste gemeinsame Vielfache* von m und n ist

$$\text{kgV}(m, n) = \min\{\ell \in \mathbb{N} : m \mid \ell \text{ und } n \mid \ell\}.$$

Für $m, n \in \mathbb{N}$ besteht ein einfacher Zusammenhang zwischen $\text{kgV}(m, n)$ und $\text{ggT}(m, n)$. (\rightarrow Übungen. Hinweis: Produkt bilden!)

Im Allgemeinen stellen wir fest, dass folgendes gilt:

$$(3.6) \quad \text{Falls } n = qm + r, \text{ dann ist } \text{ggT}(n, m) = \text{ggT}(m, r).$$

Beweis. Zunächst ist klar, dass

$$\text{falls } k \mid a \text{ und } k \mid b, \text{ dann gilt auch } k \mid a + b \quad \text{und} \quad k \mid a - b$$

- (1) Für $k = \text{ggT}(m, n)$ gilt $k \mid m$ und $k \mid n$, also auch $k \mid (n - qm) = r$ und $k \mid m$ und somit

$$\text{ggT}(m, n) \leq \text{ggT}(r, m).$$

- (2) Für $\ell = \text{ggT}(r, m)$ gilt $\ell \mid r$ und $\ell \mid m$. Daher gilt auch $\ell \mid (qm + r) = n$, also

$$\text{ggT}(r, m) \leq \text{ggT}(m, n).$$

□

Auf der Beobachtung (3.6) beruht der *euklidische*² Algorithmus zur Berechnung des ggT von $m, n \in \mathbb{N}$.

Wir dividieren zunächst $n = q_1 m + r_1$ durch m , wobei $0 \leq r_1 < m$. Dann ist

$$\text{ggT}(m, n) = \text{ggT}(r_1, m).$$

Falls $r_1 = 0$, so ist $\text{ggT}(m, n) = m$.

Andernfalls dividieren wir $m = q_2 r_1 + r_2$ mit $0 \leq r_2 < r_1$. Dann ist

$$\text{ggT}(m, n) = \text{ggT}(r_1, m) = \text{ggT}(r_2, r_1).$$

Falls $r_2 = 0$, so ist $\text{ggT}(m, n) = r_1$.

Andernfalls dividieren wir $r_1 = q_3 r_2 + r_3$ mit $0 \leq r_3 < r_2$, und so weiter. Wir erhalten eine absteigende Folge

$$m > r_1 > r_2 > r_3 > \dots \geq 0$$

von Zahlen aus \mathbb{N}_0 . Diese Folge muss irgendwann bei $r_j = 0$ abbrechen, und daher gilt

$$\text{ggT}(m, n) = \text{ggT}(r_1, m) = \text{ggT}(r_2, r_1) = \dots = \text{ggT}(r_{j-1}, 0) = r_{j-1}.$$

Wir behaupten, dass im euklidischen Algorithmus jedes r_i (bis zum Abbruch des Algorithmus) von der folgenden Form ist:

$$r_i = a_i m + b_i n \quad \text{mit} \quad a_i, b_i \in \mathbb{Z}$$

Beweis. Wir arbeiten mit vollständiger Induktion. Setzen wir $r_{-1} = n$ und $r_0 = m$, so ist diese Aussage richtig für $i = -1$ und $i = 0$. Nehmen wir an, dass die Aussage für $i - 1$ und i richtig ist. Dann gilt nach dem Divisionssatz (Satz 3.2) $r_{i-1} = q_{i+1} r_i + r_{i+1}$ und daher

$$r_{i+1} = r_{i-1} - q_{i+1} r_i = (a_{i-1} - q_{i+1} a_i) m + (b_{i-1} - q_{i+1} b_i) n,$$

die Aussage ist also auch für $i + 1$ richtig, und wir erhalten rekursiv

$$a_{i+1} = a_{i-1} - q_{i+1} a_i \quad \text{und} \quad b_{i+1} = b_{i-1} - q_{i+1} b_i.$$

²Euklid (ca. 365-300 v.Chr.), *Elemente*, Buch VII, Prop. 2

Wenn $r_j = 0$, dann ist

$$\text{ggT}(m, n) = r_{j-1} = a_{j-1} m + b_{j-1} n.$$

□

Als Konsequenz erhalten wir den folgenden wichtigen Satz:

(3.7) Satz. Für $m, n \in \mathbb{N}$ gibt es Zahlen $a, b \in \mathbb{Z}$ sodass

$$\text{ggT}(m, n) = a m + b n.$$

(3.8) Beispiel. $m = 1098, n = 3405$. Beginnend mit den Anfangswerten $r_{-1} = 3405, a_{-1} = 0, b_{-1} = 1$, sowie $r_0 = 1098, a_0 = 1, b_0 = 0$ erstellen wir eine Tabelle:

i	-1	0	1	2	3	4	5
a_i	0	1	-3	28	-31	276	
b_i	1	0	1	-9	10	-89	
q_i			3	9	1	8	4
r_i	3405	1098	111	99	12	3	0

$r_5 = 0$, also $\text{ggT}(m, n) = r_4 = 3 = a_4 m + b_4 n = 276 \cdot 1098 - 89 \cdot 3405$.

4. Primzahlen

(4.1) Definition. $p \in \mathbb{N}$ ($p \geq 2$) heißt *Primzahl*, wenn die einzigen natürlichen Teiler von p die Zahlen 1 und p sind.

Liste der ersten Primzahlen:

$$2, 3, 5, 7, 11, 13, 17, 19, 23, 29, \dots$$

Auch $2^{216091} - 1$ ist Primzahl. Diese Zahl hat 65050 Dezimalstellen. Es gibt unendlich viele Primzahlen. L. Euler hat gezeigt, dass sogar $\sum_p 1/p = \infty$ (vgl. $\sum_n 1/n^2 < \infty$). Für großes n gibt es etwa $\log n$ Primzahlen, die kleiner sind als n .

Um festzustellen, ob ein gegebenes $n \in \mathbb{N}$ in \mathbb{P} (Menge aller Primzahlen) ist, genügt es, für jede Primzahl $p \leq \lfloor \sqrt{n} \rfloor$ ($p \geq 2$) zu überprüfen, ob $p \mid n$.

Warum? → Übungen!

(4.2) Satz. Jede natürliche Zahl ≥ 2 ist durch wenigstens eine Primzahl teilbar.

Beweis → Übungen! (Vollständige Induktion)

Primzahlen sind auch durch folgende Eigenschaft charakterisiert:

(4.3) Satz. Für alle Primzahlen $p \in \mathbb{P}$ und natürlichen Zahlen $a, b \in \mathbb{N}$ gilt:
Wenn p ein Teiler von $a \cdot b$ ist, dann gilt $p \mid a$ oder $p \mid b$.

Hieraus folgt:

(4.4) Satz. (Primfaktoren) Jedes $n \in \mathbb{N}$, $n \geq 2$ kann man (eindeutig) in der folgenden Form schreiben

$$n = p_1^{k_1} p_2^{k_2} \dots p_m^{k_m},$$

wobei $m \geq 1$, $p_1, \dots, p_m \in \mathbb{P}$ (verschieden) und $k_1, \dots, k_m \in \mathbb{N}$.

Für $n \in \mathbb{N}$ und $p \in \mathbb{P}$ definieren wir $\nu_n(p) = k$ wenn $p^k \mid n$, aber nicht $p^{k+1} \mid n$. Falls p kein Teiler von n ist, gilt also $\nu_n(p) = 0$. Dann kann man die Zerlegung von n in Primfaktoren auch so schreiben:

$$n = \prod_{p \in \mathbb{P}} p^{\nu_n(p)}.$$

In diesem (scheinbar) unendlichen Produkt sind nur endlich viele Faktoren von 1 verschieden – nur diese „zählen“. Die Formel gilt auch für $n = 1$, es ist dann $\nu_n(p) = 0$ für alle $p \in \mathbb{P}$.

Es gilt (warum?)

$$\text{ggT}(m, n) = \prod_{p \in \mathbb{P}} p^{\min\{\nu_m(p), \nu_n(p)\}}.$$

5. Äquivalenzrelationen und Partitionen

(5.1) Definition. Sei X eine Menge. Eine Relation R ist eine Beziehung, die zwischen zwei Elementen von X bestehen kann oder nicht. Wir schreiben

$$x R y$$

wenn x zu y in Beziehung steht. Formal ist

$$R \subset X \times X = \{(x, y) : x, y \in X\}$$

(Menge von geordneten Paaren), und $x R y$ ist gleichbedeutend mit $(x, y) \in R$.

(5.2) Beispiel.

- (a) Menge $X = \{ \text{Österreicher} \}$, Relation „ x ist mit y verheiratet“.
- (b) Menge $X = \text{Erdbevölkerung}$, Relation „ x und y haben den gleichen Geburtsort“.
- (c) Menge $X = \mathbb{R}$, Relation $x \leq y$.
- (d) Menge $X = \{(a_n)_{n \in \mathbb{N}} : a_n > 0 \text{ für alle } n\}$, Relation $(a_n) R (b_n)$, falls $\lim_{n \rightarrow \infty} a_n/b_n = 1$.
- (e) Beliebige Menge X , Gleichheitsrelation $x = y$.

(5.3) Definition. Mögliche Eigenschaften von Relationen:

- (i) **Reflexiv:** Für alle $x \in X$ gilt $x R x$
- (ii) **Symmetrisch:** Für alle $x, y \in X$ mit $x R y$ gilt auch $y R x$
- (iii) **Antisymmetrisch:** Für alle $x, y \in X$ gilt: Falls $x R y$ und $y R x$, dann ist $x = y$
- (iv) **Transitiv:** Für alle $x, y, z \in X$ mit $x R y$ und $y R z$ gilt auch $x R z$

(5.4) Definition. Eine Äquivalenzrelation auf (in) der Menge X ist eine Relation \sim (übliche allgemeine Notation), die (i), (ii) und (iv) erfüllt.

Eine *Ordnungsrelation* (auch: *Halbordnung*) auf (in) der Menge X ist eine Relation \leq (übliche allgemeine Notation), die (i), (iii) und (iv) erfüllt.

(5.5) Definition. Sei $m \in \mathbb{N}$. Zwei Zahlen $a, b \in \mathbb{Z}$ heißen *kongruent modulo m* , in Zeichen

$$a \equiv b \pmod{m} \quad \text{oder} \quad a \equiv_m b$$

wenn $m | (b - a)$, d.h., es gibt ein $k \in \mathbb{Z} : b = a + k m$.

Kongruenz modulo m ist eine Äquivalenzrelation.

(5.6) Definition. Eine *Partition* (Zerlegung) der Menge X ist eine Familie $\{X_i : i \in I\}$ (Indexmenge) von nichtleeren Teilmengen von X mit den Eigenschaften:

$$X_i \cap X_j = \emptyset \text{ wann immer } i \neq j \quad \text{und} \quad \bigcup_{i \in I} X_i = X.$$

(5.7) Satz. (a) Sei \sim eine Äquivalenzrelation auf X . Wir definieren die Äquivalenzklasse von $x \in X$:

$$C(x) = \{y \in X : y \sim x\}.$$

Dann ist die Familie $\{C(x) : x \in X\}$ eine Partition von X , und

$$x \sim y \text{ genau dann, wenn } C(x) = C(y).$$

(b) Sei $\{X_i : i \in I\}$, eine Partition von X . Dann ist durch

$$x \sim y, \text{ falls es ein } i \in I \text{ gibt mit } x, y \in X_i$$

eine Äquivalenzrelation auf X definiert, und

$$\{C(x) : x \in X\} = \{X_i : i \in I\}.$$

(5.8) Definition. Ein Repräsentantensystem einer Äquivalenzrelation \sim in der Menge X ist eine Teilmenge $S \subseteq X$ mit der Eigenschaft:

Für jedes $y \in X$ existiert genau ein $x \in S$ mit $y \sim x$.

In jeder Äquivalenzklasse liegt also genau ein Element von S .

6. Rechnen mit Kongruenzklassen

Die Äquivalenzklasse von $a \in \mathbb{Z}$ bezüglich der Relation $\equiv \bmod m$ aus Definition 5.5 ist die Menge

$$[a]_m = \{q m + a : q \in \mathbb{Z}\}.$$

und heißt *Restklasse modulo m* . Das „natürliche“ Repräsentantsystem ist

$$S = \{0, 1, \dots, m - 1\}.$$

Für $k \in S$ ist $[k]_m$ die Menge aller Zahlen, die bei ganzzahliger Division durch m den Rest k haben.

Wir stellen fest, dass für $a, b, c, d \in \mathbb{Z}$ mit $a \equiv b \pmod{m}$ und $c \equiv d \pmod{m}$ gilt:

$$(6.1) \quad a + c \equiv (b + d) \pmod{m} \quad \text{und} \quad a \cdot c \equiv b \cdot d \pmod{m}.$$

Daher können wir Restklassen addieren und multiplizieren:

$$[a]_m + [b]_m = [a + b]_m \quad \text{und} \quad [a]_m \cdot [b]_m = [a \cdot b]_m$$

und diese Operationen hängen nicht von der Wahl des Repräsentanten ab. Zum Beispiel, mit $m = 5$:

$$[3]_5 + [4]_5 = [7]_5 = \{q5 + 7 : q \in \mathbb{Z}\} = \{\tilde{q}5 + 2 : \tilde{q} \in \mathbb{Z}\} = [2]_5$$

Das ist gleichbedeutend mit $3 + 4 = 7 \equiv 2 \pmod{5}$.

Wir schreiben \mathbb{Z}_m für die Menge aller Restklassen modulo m . Typischerweise identifiziert man \mathbb{Z}_m mit $\{0, 1, \dots, m - 1\}$. Beim Rechnen mit diesen Zahlen in \mathbb{Z}_m muss man modulo m reduzieren, z.B. $4 \cdot 4 = 16 \equiv 1 \pmod{5}$.

(6.2) Beispiel. Die Additions- und Multiplikationstafeln in \mathbb{Z}_6 .

+	0	1	2	3	4	5	·	0	1	2	3	4	5
0	0	1	2	3	4	5	0	0	0	0	0	0	0
1	1	2	3	4	5	0	1	0	1	2	3	4	5
2	2	3	4	5	0	1	2	0	2	4	0	2	4
3	3	4	5	0	1	2	3	0	3	0	3	0	3
4	4	5	0	1	2	3	4	0	4	2	0	4	2
5	5	0	1	2	3	4	5	0	5	4	3	2	1

(6.3) Satz. (Multiplikative Inverse in \mathbb{Z}_m) Ein Element $[x]_m \in \mathbb{Z}_m$ heißt invertierbar, wenn es ein $[y]_m$ gibt, sodass $[x]_m \cdot [y]_m = [1]_m$. Das geht genau dann, wenn $\text{ggT}(x, m) = 1$ ist. Notation $[x]_m^{-1} = [y]_m$ oder $x^{-1} \equiv y \pmod{m}$.

Beweis. $[x]_m \in \mathbb{Z}_m$ zu invertieren heißt, ein $y \in \mathbb{N}$ zu finden, sodass $x \cdot y \equiv 1 \pmod{m}$. Das heißt aber

$$x \cdot y = 1 + k \cdot m \quad \text{für ein } k \in \mathbb{Z}$$

und wegen (3.6) muss gelten $\text{ggT}(x, m) = 1$.

Umgekehrt sei $\text{ggT}(x, m) = 1$. Wegen Satz 3.7 existieren $y, k \in \mathbb{Z}$ sodass $y x + k m = 1$. Daher ist $y x \equiv 1 \pmod{m}$ und $[x]_m^{-1} = [y]_m$. \square

(6.4) Beispiel. Wir bestimmen $[10]_{17}^{-1}$, d.h., wir suchen eine Zahl y , sodass $[10]_{17} \cdot [y]_{17} = [1]_{17}$. Es muss also gelten

$$10 \cdot y \equiv 1 \pmod{17}$$

und es sind Zahlen y, k so zu bestimmen, dass $10y + 17k = 1$. Dabei hilft der euklidische Algorithmus:

a_i	1	1	-1	3
b_i		1	-1	2
q_i			1	2
r_i	17	10	7	3

Es ist also $3 \cdot 17 - 5 \cdot 10 = 1$ und daher $[10]_{17}^{-1} = [-5]_{17} = [12]_{17}$.

7. Diophantische Gleichungen und der chinesische Restsatz

Eine *diophantische*³ Gleichung ist eine Gleichung, bei der ganzzahlige Lösungen gesucht sind. Als Beispiel betrachten wir lineare diophantische Gleichungen.

(7.1) Satz. Seien $a, b, c \in \mathbb{N}$. Die diophantische Gleichung

$$(7.2) \quad ax + by = c$$

besitzt eine Lösung $x, y \in \mathbb{Z}$ genau dann, wenn $\text{ggT}(a, b) | c$. Erfüllen x_0, y_0 die Gleichung

$$ax_0 + by_0 = c$$

dann sind alle ganzzahligen Lösungen der Gleichung (7.2) gegeben durch

$$\left\{ \left(x_0 + \frac{bk}{g}, y_0 - \frac{ak}{g} \right) : k \in \mathbb{Z} \right\}$$

wobei $g = \text{ggT}(a, b)$.

³Diophantos von Alexandria, zw. 100 v.Chr. und 350 n.Chr.

Beweis. Im dem Fall, wo $\text{ggT}(a, b) = 1$ ist, ist leicht einzusehen, dass die Lösungsmenge die angegebene Form hat; der allgemeine Fall folgt daraus, indem man die Gleichung

$$\frac{a}{g}x + \frac{b}{g}x = \frac{c}{g}$$

betrachtet, die die gleiche Lösungsmenge hat.

(7.3) Beispiel. Wir bestimmen alle ganzzahligen Lösungen der Gleichung

$$12x + 21y = 15.$$

Zunächst der bekannte euklidische Algorithmus.

a_i	1	1	-1	
b_i		1	-1	2
q_i			1	1
	21	12	9	3

also $\text{ggT}(21, 12) = 3 = 2 \cdot 12 - 1 \cdot 21$. Wir multiplizieren beide Seiten mit 5 und erhalten

$$15 = 10 \cdot 12 - 5 \cdot 21,$$

die Grundlösung der Gleichung lautet daher $(x_0, y_0) = (10, -5)$. Die Menge aller Lösungen ist

$$\begin{aligned} \{(10 + \frac{21k}{3}, -5 - \frac{12k}{3}) : k \in \mathbb{Z}\} &= \{(10 + k \cdot 7, -5 - k \cdot 4) : k \in \mathbb{Z}\} \\ &= \{\dots, (3, -1), (10, -5), (17, -9), \dots\}. \end{aligned}$$

(7.4) Der chinesische Restsatz. Wir wollen nun ein „Gleichungssystem“ von Kongruenzen lösen. Gegeben seien

$$\begin{aligned} m_1, \dots, m_s \in \mathbb{N} \quad \text{mit} \quad \text{ggT}(m_i, m_j) = 1 \quad \text{für alle } i \neq j \quad (\text{„relativ prim“}), \\ c_1, \dots, c_s \in \mathbb{N} \quad (\text{bzw. } \mathbb{Z}). \end{aligned}$$

Gesucht ist $x \in \mathbb{Z}$ (bzw. $\in \mathbb{N}$), sodass

$$x \equiv c_i \pmod{m_i} \quad \text{für } i = 1, \dots, s.$$

Zur Lösung setzen wir $m = m_1 \cdots m_s$. Wir stellen fest, dass $m/m_i \in \mathbb{N}$ und dass $\text{ggT}(m_i, m/m_i) = 1$. Nach Satz 3.7 (vgl. Beispiel 3.8) können wir

mit Hilfe des euklidischen Algorithmus konstruktiv Zahlen $a_i, b_i \in \mathbb{Z}$ finden, sodass

$$(7.5) \quad a_i \frac{m}{m_i} + b_i m_i = 1, \quad i = 1, \dots, s.$$

Daher ist

$$c_i a_i \frac{m}{m_i} = c_i - c_i b_i m_i \equiv c_i \pmod{m_i} \text{ und } c_i a_i \frac{m}{m_i} \equiv 0 \pmod{m_j} \text{ für alle } j \neq i.$$

Setzen wir also

$$(7.6) \quad x = \sum_{i=1}^s c_i a_i \frac{m}{m_i},$$

dann haben wir eine Lösung gefunden. Jedes y mit $y \equiv x \pmod{m}$ ist gleichfalls Lösung, wir können also auch positive Lösungen finden. Der Satz, die wir soeben bewiesen haben (Lösung simultaner Kongruenzen), ist der sogenannte *chinesische Restsatz*⁴.

(7.7) Beispiel. Wir vollziehen den durch Satz 3.7, (7.5) und (7.6) vorgegebenen Algorithmus nach, um die folgenden simultanen Kongruenzen zu lösen.

$$x \equiv 3 \pmod{7}, \quad x \equiv 2 \pmod{8}, \quad x \equiv 1 \pmod{15},$$

$$\text{Wir haben } m_1 = 7, m_2 = 8, m_3 = 15, \frac{m}{m_1} = 120, \frac{m}{m_2} = 105, \frac{m}{m_3} = 56,$$

$$1 \cdot \frac{m}{m_1} - 17 \cdot m_1 = 1, \quad 1 \cdot \frac{m}{m_2} - 13 \cdot m_2 = 1, \quad -4 \cdot \frac{m}{m_3} + 15 \cdot m_3 = 1,$$

also $a_1 = a_2 = 1$ und $a_3 = -4$. Wir erhalten

$$x = 3 \cdot 1 \cdot 120 + 2 \cdot 1 \cdot 105 + 1 \cdot (-4) \cdot 56 = 346. \quad \square$$

(7.8) Bemerkung. Es gilt für $m = \text{kgV}(m_1, m_2)$

Falls $a \equiv b \pmod{m_1}$ und $a \equiv b \pmod{m_2}$, dann ist auch $a \equiv b \pmod{m}$.

Ist insbesondere $\text{ggT}(m_1, m_2) = 1$, dann gilt $a \equiv b \pmod{m_1 m_2}$.

⁴so benannt nach dem Vorkommen im *Sunzi Suan Jing* aus dem 5.Jh

8. Halbgruppen und Gruppen

Wir wollen die Beobachtungen aus (6.1) und Beispiel 6.2 formalisieren.

(8.1) Definition. Sei X eine nichtleere Menge.

- (a) Eine *binäre Operation* auf X ist eine Vorschrift, die je zwei Elementen $x, y \in X$ ein Element $x \circ y$ (allgemeine Notation) in X zuordnet.
- (b) Das Paar (X, \circ) heißt *Halbgruppe*, wenn \circ assoziativ ist, d.h.,

$$(x \circ y) \circ z = x \circ (y \circ z).$$

Assoziativ bedeutet, dass es auf spezifische Klammerung nicht ankommt. Es gilt dann auch z.B.

$$(w \circ x) \circ (y \circ z) = w \circ (x \circ (y \circ z))$$

(8.2) Beispiel. 1.) $(\mathbb{N}, +)$, (\mathbb{N}, \cdot) , $(\mathbb{Z}, +)$, $(\mathbb{Z}, \cdot), \dots$ sind Halbgruppen.

2.) $(\mathbb{Z}_m, +)$, (\mathbb{Z}_m, \cdot) sind Halbgruppen.

3.) Sei Σ eine endliche Menge von Symbolen („Buchstaben“). Ein (nichtleeres) Wort über Σ ist eine endliche Sequenz

$$v = a_1 a_2 \dots a_m, \quad m \in \mathbb{N}, \quad a_i \in \Sigma.$$

Die Zahl m heißt die *Länge* von v .

Man schreibt Σ^+ für die Menge aller (nichtleeren) Worte über Σ . Seien $v = a_1 a_2 \dots a_m$, $w = b_1 b_2 \dots b_n$ Worte über Σ . Dann definieren wir $v \circ w$, die Konkatenation von v und w , als

$$v \circ w = a_1 a_2 \dots a_m b_1 b_2 \dots b_n.$$

Üblicherweise schreibt man einfach vw anstatt $v \circ w$.

Auf diese Art wird Σ^+ zu einer Halbgruppe.

4.) Sei $X = \mathbb{N}$, und definieren wir $x \circ y = x^y$ (Potenz). Diese Operation ist *nicht* assoziativ, da z.B.

$$(2 \circ 3) \circ 4 = 8^4 = 2^{12} \text{ und } 2 \circ (3 \circ 4) = 2^{81}.$$

(8.3) Definition. Sei (X, \circ) eine Halbgruppe.

(a) Die Halbgruppe heißt *kommutativ* (auch: *abelsch*), wenn

$$x \circ y = y \circ x \quad \text{für alle } x, y \in X .$$

(b) Ein *neutrales Element* oder *Einselement* ist ein Element $e \in X$ mit

$$e \circ x = x \circ e = x \quad \text{für alle } x \in X .$$

Eine Halbgruppe mit neutralem Element heißt *Monoid*.

(8.4) Beispiel. 1.) $(\mathbb{N}_0, +)$, (\mathbb{N}, \cdot) , $(\mathbb{Z}, +)$, (\mathbb{Z}, \cdot) sind kommutative Monoide mit neutralem Element 0 bzw. 1.

2.) $(Z_m, +)$ und (Z_m, \cdot) sind kommutative Monoide mit neutralem Element $[0]_m$ bzw. $[1]_m$.

3.) Die Halbgruppe (Σ^+, \circ) aus (4.2.1) ist nicht kommutativ (außer wenn $|\Sigma| = 1$) und hat kein neutrales Element. Wir können aber ein solches hinzufügen, das sogenannte *leere Wort* ε (Wort ohne Buchstaben).

Man schreibt $\Sigma^* = \Sigma^+ \cup \{\varepsilon\}$.

Es ist natürlich $w \circ \varepsilon = \varepsilon \circ w = w$ für alle $w \in \Sigma^*$.

Somit ist (Σ^*, \circ) ein Monoid, das *freie Monoid* über Σ . Es spielt in der theoretischen Informatik (Theorie der Programmiersprachen!) eine wichtige Rolle: eine *formale Sprache* ist eine Teilmenge von Σ^* .

(8.5) Bemerkung. In einem Monoid kann es nicht mehr als ein neutrales Element geben.

In der Tat, nehmen wir an, dass e' ein weiteres Einselement sei. Dann ist $e \circ e' = e$ und $e \circ e' = e'$, also $e = e'$. \square

(8.6) Definition. Sei (X, \circ) ein Monoid mit neutralem Element e , und sei $x \in X$.

(a) Ein Element $y \in X$ heißt *inverses Element* von x , wenn

$$x \circ y = y \circ x = e .$$

(b) Ein Monoid, in dem jedes Element ein Inverses besitzt, heißt *Gruppe*.

(8.7) Bemerkung. (a) In einem Monoid kann kein Element x mehr als ein inverses Element y haben.

Man schreibt dann $y = x^{-1}$. Es gilt $(x^{-1})^{-1} = x$.

(b) Sind x_1 und x_2 invertierbar, dann hat auch $x_1 \circ x_2$ ein inverses Element, nämlich

$$(x_1 \circ x_2)^{-1} = x_2^{-1} \circ x_1^{-1}.$$

Weiters gilt $(x^{-1})^{-1} = x$.

In der Tat, nehmen wir an, dass y, y' inverse Elemente von $x \in X$ seien.

Dann ist

$$y = y \circ e = y \circ (x \circ y') = (y \circ x) \circ y' = e \circ y' = y'. \quad \square$$

Wenn die Operation mit $+$ bezeichnet wird, schreibt man $-x$ statt x^{-1} .

Folgerung aus Bemerkung 8.7:

(8.8) Satz. Die Menge G der invertierbaren Elemente eines Monoids (X, \circ) ist bezüglich \circ eine Gruppe.

(8.9) Beispiel. 1.) In (Σ^*, \circ) hat kein Element außer dem neutralen Element ε ein inverses.

2.) $(\mathbb{Z}, +)$ ist eine kommutative Gruppe mit neutralem Element 0.

3.) Überprüfen Sie Schritt für Schritt, dass $(\mathbb{Z}_m, +)$ für jedes $m \in \mathbb{N}$ eine kommutative Gruppe ist.

4.) (\mathbb{Z}_m, \cdot) mit Einselement 1 (genauer: $[1]_m$) ist keine Gruppe, da 0 (genauer: $[0]_m$) kein inverses Element hat.

(Ist vielleicht $\mathbb{Z}_m \setminus \{0\}$ bezüglich der Restklassenmultiplikation eine Gruppe?)

5.) $(\mathbb{Z}_7 \setminus \{0\}, \cdot)$ hat folgende Multiplikationstafel.

.	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

Dies ist eine Gruppe.

- 6.) Die Tafel aus Beispiel 6.2 zeigt, dass $(\mathbb{Z}_6 \setminus \{0\}, \cdot)$ nicht einmal eine Halbgruppe ist, da z.B. $2 \cdot 3 \equiv 0 \pmod{6}$ nicht in der Grundmenge $X = \mathbb{Z}_6 \setminus \{0\}$ liegt. Streichen wir aber alle Zeilen und Spalten, die eine 0 enthalten, so verbleibt

.	1	5
1	1	5
5	5	1

Wir sehen, dass $(\mathbb{Z}_6 \setminus \{0, 2, 3, 4\}, \cdot)$ eine Gruppe ist. □

Wegen der Sätze 8.8 und 6.3 gilt:

(8.10) Satz. Die Menge $\mathbb{G}_m := \{[x]_m \in \mathbb{Z}_m : \text{ggT}(x, m) = 1\}$ bildet bezüglich der Multiplikation modulo m eine Gruppe. □

Häufig verwendet man das Symbol \cdot für die Gruppenoperation oder schreibt einfach $x y$ statt $x \cdot y$. Man sagt auch oft: „sei G eine Gruppe“, anstatt (G, \cdot) .

(8.11) Bemerkung. Sei (G, \circ) eine Gruppe mit neutralem Element e , und sei $y \in G$ beliebig. Dann ist die Abbildung

$$f_y : G \rightarrow G, \quad f_y(x) = y \circ x$$

bijektiv, d.h.: injektiv und surjektiv, siehe ANALYSIS T1. (\rightarrow Übungen !) □

(8.12) Definition. Sei (G, \circ) eine Gruppe und $H \subset G$ (nichtleer). Dann heißt H Untergruppe von G , wenn auch H bezüglich der gleichen Operation \circ eine Gruppe ist.

D.h., man muss das Folgende überprüfen:

- (i) Für $x, y \in H$ liegt auch $x \circ y$ in H und
- (ii) für $x \in H$ liegt auch x^{-1} in H .

Es folgt, dass auch $e \in H$.

(8.13) Beispiel. (1) Sei $m \in \mathbb{N}$, $m \geq 2$. Dann ist

$$m\mathbb{Z} := \{m k : k \in \mathbb{Z}\}$$

eine Untergruppe von $(\mathbb{Z}, +)$.

- (2) Sei (G, \circ) eine beliebige Gruppe. Für $x \in G$ und $k \in \mathbb{N}$ definieren wir $x^k = x \cdots x$ (k -faches Produkt) und $x^{-k} = (x^{-1})^k$. Weiters definieren wir $x^0 = e$ (das neutrale Element).

Dann ist

$$\langle x \rangle := \{x^k : k \in \mathbb{Z}\}$$

eine Untergruppe von G (die von x erzeugte Untergruppe).

(8.14) Definition. (1) Sei (G, \circ) eine Gruppe und H eine Untergruppe. Wir definieren eine Relation *Kongruenz modulo H* auf G ,

$$x \equiv y \ (H), \text{ falls } x^{-1}y \in H.$$

Dies ist eine Äquivalenzrelation. (\rightarrow Übungen!)

- (2) Die Äquivalenzklasse von $x \in G$ ist die Menge

$$xH = \{xh : h \in H\}.$$

(Die Mengen xH , $x \in G$, heißen *Linksnebenklassen* von H in G .)

Wir nehmen nun an, dass G eine endliche Gruppe ist, mit $|G|$ Elementen.

Wegen Bemerkung 8.11 ist $|xH| = |H|$ für alle $x \in G$. Wegen Definition 8.14 können wir ein Repräsentantsystem $\{x_1, \dots, x_m\}$ für $\equiv \ (H)$ finden, und nach Satz 5.7 ist

$$G = \bigcup_{j=1}^m x_j H, \quad (\text{Vereinigung disjunkter Mengen!}).$$

Daher ist

$$|G| = m |H|$$

Also:

(8.15) Satz. (Lagrange⁵) Ist G eine endliche Gruppe und H eine Untergruppe, dann gilt $|H| \mid |G|$. \square

Nun sei $x \in G$ und $\langle x \rangle = \{x^k : k \in \mathbb{Z}\}$ wie in Beispiel 8.13(2). Die Elemente x^k , $k \in \mathbb{N}_0$, können nicht alle verschieden sein. Daher muss es $k \in \mathbb{N}_0$ und $n \in \mathbb{N}$ geben, sodass $x^{k+n} = x^k$. Multiplizieren mit x^{-k} ergibt

$$x^n = e.$$

(8.16) Definition. Die *Ordnung* von $x \in G$ (endlich) ist

$$o_G(x) := o(x) := \min\{n \in \mathbb{N} : x^n = e\}.$$

Wenn $o(x) = m$, gilt also

$$\{x^k : k \in \mathbb{Z}\} = \{e = x^0, x_1, \dots, x^{m-1}\},$$

letztere Elemente sind alle verschieden (warum?), und die Gruppenoperation in $\langle x \rangle$ ist – mittels der Exponenten – die gleiche wie in \mathbb{Z}_m . Aus Satz 8.15 folgern wir nun

(8.17) Satz. Ist G eine endliche Gruppe und $x \in G$, dann gilt $o(x) \mid |G|$.

Wenn $o(x) = m$, dann ist also $|G| = m k$ mit $k \in \mathbb{N}$. Daher

$$x^{|G|} = x^{mk} = (x^m)^k = e^k = e.$$

(8.18) Folgerung. Für eine endliche Gruppe G und $x \in G$ gilt $x^{|G|} = e$.

⁵Joseph-Louis Lagrange (1736–1813)

9. Der Satz von Euler-Fermat

Betrachten wir nun die Gruppe $\mathbb{G}_m := \{[x]_m \in \mathbb{Z}_m : \text{ggT}(x, m) = 1\}$ aus Satz 8.10. Das natürliche Repräsentantensystem von \mathbb{Z}_m ist $\{0, 1, \dots, m-1\}$. Daher können wir \mathbb{G}_m mit der Menge

$$\left\{ k \in \{1, \dots, m-1\} : \text{ggT}(k, m) = 1 \right\}$$

identifizieren. Wir definieren die *Eulersche⁶ Phi-Funktion*

$$\varphi(m) := |\mathbb{G}_m| = \left| \left\{ k \in \{1, \dots, m-1\} : \text{ggT}(k, m) = 1 \right\} \right|.$$

Ist $p \in \mathbb{P}$, dann ist $\text{ggT}(k, p) = 1$ für alle $k \in \{1, \dots, p-1\}$, also

$$\varphi(p) = p - 1, \quad p \in \mathbb{P}.$$

Allgemein gilt

(9.1) Satz. Hat $m \in \mathbb{N}$, $m \geq 2$ gemäß Satz 4.4 die Primfaktorenzerlegung

$$m = p_1^{k_1} p_2^{k_2} \dots p_r^{k_r}$$

mit $p_i \in \mathbb{P}$ und $k_i \in \mathbb{N}$, so ist

$$\varphi(m) = p_1^{k_1-1} (p_1 - 1) p_2^{k_2-1} (p_2 - 1) \dots p_r^{k_r-1} (p_r - 1).$$

Aus 8.18 folgt

(9.2) Satz. (Euler-Fermat⁷)

Falls $\text{ggT}(a, m) = 1$, dann gilt $a^{\varphi(m)} \equiv 1 \pmod{m}$.

Beweis für $p \in \mathbb{P}$. Sei $\text{ggT}(a, p) = 1$, d.h., p kein Teiler von a . Zu zeigen ist $a^{p-1} \equiv 1 \pmod{p}$. Dann ist die Abbildung

$$\begin{aligned} f : \mathbb{Z}_p \setminus \{0\} &\rightarrow \mathbb{Z}_p \setminus \{0\} \\ [x] &\mapsto [a \cdot x] \end{aligned}$$

⁶Leonhard Euler (1707-1783)

⁷Pierre de Fermat (1607–1665)

bijektiv, denn wenn $[a \cdot x]_p = [a \cdot y]_p$ dann ist nach Multiplikation mit $[a]_p^{-1}$ auch $[x]_p = [y]_p$. Daher gilt die Gleichheit

$$\{[a]_p, [2a]_p, [3a]_p, \dots, [(p-1)a]_p\} = \{[1]_p, [2]_p, [3]_p, \dots, [p-1]_p\}$$

und folglich

$$[a]_p \cdot [2a]_p \cdot [3a]_p \cdots [(p-1)a]_p = [1]_p \cdot [2]_p \cdot [3]_p \cdots [p-1]_p$$

d.h.,

$$[a]_p^{p-1} [1]_p \cdot [2]_p \cdot [3]_p \cdots [p-1]_p = [1]_p \cdot [2]_p \cdot [3]_p \cdots [p-1]_p$$

und da die rechte Seite invertierbar ist (warum?), muss $[a]_p^{p-1} = [1]_p$ sein.

Für beliebige abelsche Gruppen kann man den gleichen Beweis führen. \square

(9.3) Beispiel. Arithmetik in \mathbb{Z}_m Der Satz von Euler-Fermat ist nützlich bei der Berechnung hoher Potenzen in \mathbb{Z}_m . Um z.B. $7^{256} \bmod 13$ zu berechnen, bestimmt man $\varphi(13) = 12$ und zerlegt $256 = 21 \cdot 12 + 4$ und weiß wegen Satz 9.2, dass $7^{12} \equiv 1 \bmod 13$ gilt und daher

$$7^{256} = (7^{12})^{21} \cdot 7^4 \equiv 7^4 = 49^2 \equiv 10^2 \equiv (-3)^2 = 9 \bmod 13.$$

10. Kryptographie

Verschlüsselung wird angewendet, um Nachrichten über einen unsicheren Kommunikationskanal zu übermitteln. Als Modellschema dient meist das folgende: A(lice) möchte eine Nachricht an B(ob) übermitteln, E(ve) (=Eavesdropper) hört mit. Daher muss die Botschaft so verändert werden, dass

- (1) Eve die Nachricht nicht verstehen kann.
- (2) Bob die Nachricht trotzdem wieder lesbar machen kann.

(10.1) Beispiel. (Caesar⁸) Dies ist eines der einfachsten und unsichersten Verfahren. Zunächst wird jedem Buchstaben eine Zahl zugeordnet:

⁸Gaius Iulius Caesar (100-44 v.Chr.) hat dieses Verfahren nach Angaben von Sueton verwendet.

„	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26

Ein Beispiel:

$$\begin{array}{ccccccccccccccccccccc} A & L & E & A & & I & A & C & T & A & & E & S & T \\ 1 & 12 & 5 & 1 & 0 & 9 & 1 & 3 & 20 & 1 & 0 & 5 & 19 & 20 \end{array}$$

Zu den so erhaltenen Ziffern wird dann eine fixe Zahl, z.B. 11, modulo 27 addiert:

$$\begin{array}{ccccccccccccccccccccc} 1 & 12 & 5 & 1 & 0 & 9 & 1 & 3 & 20 & 1 & 0 & 5 & 19 & 20 \\ \downarrow & \downarrow \\ 12 & 23 & 16 & 12 & 11 & 20 & 12 & 14 & 4 & 12 & 11 & 16 & 3 & 4 \\ L & W & P & L & K & T & L & N & D & L & K & P & C & D \end{array}$$

Der so erhaltene Kryptotext „LWPMLTLOELKPCD“ wird verschickt und der Empfänger braucht nur den Vorgang umzukehren, um den Klartext lesen zu können.

Dieses Beispiel zeigt die grundsätzliche Vorgangsweise jeglicher Verschlüsselungsverfahren. Alice verwendet eine Verschlüsselungsfunktion f , um eine Nachricht (=Zeichenkette) $x_1x_2\dots x_m$ in eine verschlüsselte Botschaft $y_1y_2\dots y_n = f(x_1x_2\dots x_m)$ umzuwandeln.

Bob besitzt eine Entschlüsselungsfunktion g , die invers zu f ist, und empfangene Botschaften rückverwandelt: $g(y_1y_2\dots y_n) = x_1x_2\dots x_m$. Es ist also $g \circ f$ die identische Funktion.

Im obigen Beispiel ist die Verschlüsselungsfunktion f gegeben durch

$$f(x_1x_2\dots x_m) = f_k(x_1)f_k(x_2)\dots f_k(x_m)$$

wobei k zwischen A und B vorher vereinbart wird und $f_k(x) = x + k \bmod 27$. Die Entschlüsselungsfunktion g ist analog gegeben durch zeichenweise Anwendung der Funktion $g_k(x) = x - k \bmod 27$.

An diesem Beispiel sind folgende Probleme erkennbar.

- (1) Wenn jeder Buchstabe immer gleich verschlüsselt wird, kann die Verschlüsselung durch Häufigkeitsanalyse sehr leicht geknackt werden.

Das wird vermieden, wenn Gruppen von Buchstaben verschlüsselt werden und zusätzlich die Nachricht mit Störzeichen verändert wird. Sei z.B. $x_1x_2\dots x_m$ die Nachricht und $s_1s_2\dots s_m$ eine zufällig gewählte Zeichenfolge, dann kann man die veränderte Nachricht

$$x_1s_1x_2s_2\dots x_ms_m$$

blockweise verschlüsseln.

- (2) Bevor Caesar seine Nachricht verschlüsselt, muss er sich vorher mit Bob getroffen haben, um einen geheimen Schlüssel zu vereinbaren.

Lösung 1: Geheime Schlüsselübergabe, z.B. mit dem *Diffie-Hellman-Verfahren* (10.2).

Lösung 2: Verwendung von asymmetrischen Systemen mit öffentlichem Schlüssel, z.B. dem RSA-Verfahren (nächster Abschnitt).

(10.2) Diffie-Hellman-Schlüsselvereinbarung⁹

Ziel: Schlüsselübergabe über einen unsicheren Kanal.

- (1) Alice und Bob vereinbaren öffentlich eine natürliche Zahl $g \in \mathbb{N}$ und eine Primzahl $p \in \mathbb{P}$.
- (2) Alice wählt eine geheime Zahl $a < p$ und berechnet $m = g^a \bmod p$
- (3) Bob wählt eine geheime Zahl $b < p$ und berechnet $n = g^b \bmod p$
- (4) Die Zahlen m und n werden öffentlich an den jeweils anderen übermittelt.
- (5) Alice berechnet den Schlüssel $r = n^a \bmod p$.
- (6) Bob berechnet den Schlüssel $s = m^b \bmod p$.

Der Witz dabei ist, dass

$$r = n^a = (g^b)^a = g^{ab} = (g^a)^b = m^b = s$$

ist. Beide haben also jetzt den gleichen Schlüssel in der Hand, der aber aus den öffentlichen Daten nur mit viel Aufwand eruierbar ist.

⁹Beschrieben von W. Diffie und M. Hellman 1976.

11. Das RSA-Verfahren

Ziel: Zeichenfolgen („Botschaften“) so zu verschlüsseln, dass sie sehr schwer (de facto unmöglich) zu entschlüsseln sind.

Zerlegung einer Zahl $m \in \mathbb{N}$ in Primfaktoren ist ein sehr aufwendiger Algorithmus, für großes m mit großen Primfaktoren faktisch unmöglich (zu langsam).

Sei also $m = pq$, wobei $p, q \in \mathbb{P}$ zwei große, verschiedene Primzahlen sind. (Wir kennen sie, aber unsere „Gegner“ werden nur m kennen.)

Die Zeichen unserer Botschaften interpretieren wir als Elemente von \mathbb{Z}_m , oder besser $\mathbb{Z}_m \setminus \{0\}$, d.h., als Elemente von $\{1, \dots, m-1\}$. Die Verschlüsselung erfolgt mittels einer bijektiven Abbildung

$$(11.1) \quad f : \{1, \dots, m-1\} \rightarrow \{1, \dots, m-1\}.$$

Eine Zeichenkette $k_1 k_2 \dots k_n$ könnten wir z.B. mit

$$(11.2) \quad f(k_1) \cdots f(k_n)$$

verschlüsseln. (Achtung: bei langen Zeichenketten mit vielen Wiederholungen ist das natürlich riskant, „knacken“ leichter möglich! vgl. Turing/Enigma.)

Was ist eine geeignete Wahl der Funktion f ?

Aus dem Satz von Euler-Fermat folgt:

(11.3) Satz. Für $m = pq$ mit $p, q \in \mathbb{P}$, $p \neq q$ gilt

$$a^{k(p-1)(q-1)+1} \equiv a \pmod{m} \quad \text{für alle } a \in \mathbb{Z}, k \in \mathbb{N}.$$

Beweis. Es ist $\varphi(m) = (p-1)(q-1)$ nach Satz 9.1.

Es gilt $\text{ggT}(a, m) \in \{1, p, q, m\}$.

Fall 1: Wenn $\text{ggT}(a, m) = 1$ dann gilt nach Satz 9.2

$$a^{(p-1)(q-1)} \equiv 1 \pmod{m}, \quad \text{daher} \quad a^{k(p-1)(q-1)+1} = (a^{(p-1)(q-1)})^k a \equiv a \pmod{m}.$$

Fall 2: Wenn $\text{ggT}(a, m) = p$ dann gilt $p \mid a$ und $\text{ggT}(a, q) = 1$, also

$$a^{k(p-1)(q-1)+1} \equiv 0 \equiv a \pmod{p} \quad \text{und}$$

$$a^{k(p-1)(q-1)+1} = (a^{q-1})^{k(p-1)} a \equiv 1^{k(p-1)} a \equiv a \pmod{q}$$

und daher auch

$$a^{k(p-1)(q-1)+1} \equiv a \pmod{pq}.$$

Fall 3: Wenn $\text{ggT}(a, m) = q$ dann gilt $q \mid a$ und $\text{ggT}(a, p) = 1$, also genau so wie Fall 2.

Fall 4: Wenn $\text{ggT}(a, m) = m$ dann gilt $a \equiv 0 \pmod{m}$ und die Aussage ist klar. \square

Das RSA-Verfahren¹⁰ geht nun wie folgt vor:

Wir wählen $r, s \in \mathbb{N}$ so, dass für $\varphi(m) = (p-1)(q-1)$

$$(11.4) \quad \text{ggT}(r, \varphi(m)) = 1 \text{ und } rs \equiv 1 \pmod{\varphi(m)}$$

Wenn r gewählt wurde, kann s gemäß Satz 3.7 aus dem euklidischen Algorithmus bestimmt werden: da $\text{ggT}(r, \varphi(m)) = 1$ gibt es $s, n \in \mathbb{N}$ sodass $sr - n\varphi(m) = 1$ (siehe Übungen).

Wir definieren

$$(11.5) \quad f(k) = k^r \pmod{m},$$

also k^r modulo m reduziert zu einem Element von $\{0, \dots, m-1\}$. Analog definieren wir

$$(11.6) \quad g(k) = k^s \pmod{m}.$$

Dann gilt nach Satz 11.3 für alle $k \in \mathbb{Z}$

$$g \circ f(k) = k^{rs} = k^{n\varphi(m)+1} \equiv k \pmod{m}$$

und analog $f \circ g(k) = k$. Insbesondere muss f bijektiv sein, und g ist die inverse Abbildung.

¹⁰Beschrieben 1978 von R. Rivest, A. Shamir und L. Adleman

Die Ver- und Entschlüsselung verläuft also wie folgt:

- Die Zahlen m und r werden öffentlich bekanntgegeben.
- Nur der Empfänger der Nachricht kennt die Zahl s .
- Die Verschlüsselung erfolgt gemäß (11.1), (11.2) und (11.5). Jede beliebige Person kann mithilfe der öffentlichen Schlüssel m und r eine verschlüsselte Nachricht an den Empfänger schicken.
- Die Entschlüsselung erfolgt gemäß (11.6).
- Die Faktorisierung $m = p q$ ist üblicherweise (z.B. PGP, SSH) nur dem Empfänger der verschlüsselten Nachricht bekannt.

Außer dem Empfänger kann de facto niemand die Verschlüsselungsmethode „knacken“, da dazu die Faktorisierung $m = p q$ gebraucht wird, deren Auffinden für große p, q zu aufwendig ist. Andererseits braucht selbst der Empfänger die Faktorisierung p und q nicht zu kennen, um erfolgreich entschlüsseln zu können.

Probleme:

- (1) Bei schlechter Wahl von r kann $f(k) = k$ für viele k sein, also praktisch keine wirkliche Verschlüsselung.
- (2) Wenn man Buchstaben für Buchstaben verschlüsselt, so kann der „Gegner“ eine Häufigkeitsanalyse der vorkommenden Symbole durchführen und daraus die Verschlüsselung „knacken“.

Um letzteres Problem zu umgehen, kann man z.B. wie folgt vorgehen: zu je zwei aufeinanderfolgenden Buchstaben (Zeichen) fügt man zwei zufällig generierte hinzu. Dann verwendet man eine Verschlüsselung, wo jedem Viererblock ein $k \in \{2, \dots, m - 1\}$ entspricht (warum werden 0 und 1 ausgeschlossen?), das gemäß (11.5) umgeformt wird. Häufigkeitsanalyse ist dann nicht mehr möglich. Der Empfänger weiß, dass nur die ersten zwei Zeichen jedes von ihm nach (11.6) entschlüsselten Viererblocks relevant sind.

(11.7) Beispiel. Wir wählen die zwei Primzahlen $p = 41$, $q = 53$, also

$$m = 2173 \quad \text{und} \quad (p-1)(q-1) = 2080 = 2^5 \cdot 5 \cdot 13.$$

Wir wählen nun $r = 19$ und berechnen s .

i	-1	0	1	2	3
a_i	0	1	-109	219	
b_i	1	0	1	-2	
q_i			109	2	9
r_i	2080	19	9	1	0

Also $219 \cdot 19 - 2 \cdot 2080 = 1$ und somit $s = 219$.

Wir verwenden eine vereinfachte Variante des obigen Punktes (2):

Wir werden Buchstabenpaare verschlüsseln, und zwar der Einfachheit halber nur Kleinbuchstaben, also

$$\begin{aligned} \text{aa} &\leftrightarrow 0, \text{ ab} \leftrightarrow 1, \dots, \text{az} \leftrightarrow 25 \\ \text{ba} &\leftrightarrow 26, \text{ bb} \leftrightarrow 27, \dots, \text{bz} \leftrightarrow 51 \\ &\vdots \\ \text{ka} &\leftrightarrow 260, \text{ kb} \leftrightarrow 261, \dots, \text{kz} \leftrightarrow 285 \\ &\vdots \\ \text{za} &\leftrightarrow 650, \text{ zb} \leftrightarrow 651, \dots, \text{zz} \leftrightarrow 675 \end{aligned}$$

- Ist α der ℓ_1 -te und β der ℓ_2 -te Buchstabe ($\ell_1, \ell_2 \in \{0, 1, \dots, 25\}$), dann entspricht das Buchstabenpaar $\alpha\beta$ der Zahl $\ell_1 \cdot 26 + \ell_2$.
- Ist umgekehrt $n \in \{0, \dots, 675 = 26^2 - 1\}$, dann dividieren wir ganz-zahlig: $n = \ell_1 \cdot 26 + \ell_2$, und n entspricht dem Buchstabenpaar $\alpha\beta$, wo α der ℓ_1 -te und β der ℓ_2 -te Buchstabe ist.

Wir müssen nun die Zahlen $k \in \{0, \dots, 675\}$ mittels $f(k) = k^r \bmod m$ verschlüsseln.

Wollen wir etwa das Wort *graz* verschlüsseln, so erzeugen wir zu jedem seiner Buchstaben einen zufälligen zweiten, also z.B. *wien*. Wir verschlüsseln nun die vier Buchstabenpaare *(gw)(ri)(ae)(zn)*. (Der Empfänger muss beim Entschlüsseln von den jeweiligen Buchstabenpaaren immer nur den ersten berücksichtigen.)

$$\begin{array}{lll}
 \text{gw} \leftrightarrow k_1 = 6 \cdot 26 + 22 = 178 & f(178) = 178^{19} \equiv 899 \pmod{2173} \\
 \text{ri} \leftrightarrow k_2 = 17 \cdot 26 + 8 = 450 & \mapsto f(450) = 450^{19} \equiv 1188 \pmod{2173} \\
 \text{ae} \leftrightarrow k_3 = 0 \cdot 26 + 4 = 4 & f(4) = 4^{19} \equiv 1999 \pmod{2173} \\
 \text{zn} \leftrightarrow k_4 = 25 \cdot 26 + 13 = 663 & f(663) = 663^{19} \equiv 773 \pmod{2173}
 \end{array}$$

Die Zahlen $m = 2173$ und $r = 19$ sind öffentlich.

Dem Empfänger, der allein den Schlüssel $s = 219$ kennt, wird die verschlüsselte Zahlenfolge $(899, 1188, 1999, 773)$ übermittelt.

Er muss dann jede der Zahlen $(899^{219}, 1188^{219}, 1999^{219}, 773^{219})$ modulo 2173 reduzieren.

Die Zahlen, die dabei herauskommen, müssen $(178, 450, 4, 663)$ sein.

Diesen entsprechen die Buchstabenpaare gw, ri, ae, zn.

Die jeweils ersten Buchstaben ergeben zusammengesetzt graz. □

KAPITEL B

Grundlagen der Logik

Inhaltsangabe

- | | |
|---|------|
| 1. Aussagen und Junktoren | B.3 |
| 2. Die Sprache der Aussagenlogik | B.5 |
| 3. Exkurs: Induktive Strukturen | B.7 |
| 4. Aussagenlogik | B.9 |
| 5. Äquivalenz von aussagenlogischen Formeln | B.11 |
| 6. Konjunktive und disjunktive Normalform | B.17 |
| 7. Prädikatenlogik | B.23 |
-

1. Aussagen und Junktoren

(1.1) Definition. Eine Aussage ist ein sprachlicher Satz, der entweder wahr (W) oder falsch (F) ist. Nicht erlaubt sind also Sätze, die weder wahr noch falsch oder solche, die beides sind („Tertium non datur“).

Die *formale Logik* befasst sich mit dem Wahrheitsgehalt von Aussagen und Verknüpfungen von Aussagen unabhängig von ihrem Inhalt. Es stehen die folgenden Verknüpfungsoperatoren (*Junktoren*) zur Verfügung:

(1.2) Definition. Wir definieren den Wahrheitswert von Verknüpfungen als Funktionen Φ_* auf $\{W, F\}$ bzw. $\{W, F\}^2$ wie folgt.

(1) **Negation.** Der NOT-Operator.

Der Wahrheitswert einer negierten Aussage ist der dem ursprünglichen entgegengesetzte:

$$\begin{aligned}\neg W &= \Phi_{\neg}(W) = F \\ \neg F &= \Phi_{\neg}(F) = W\end{aligned}$$

(2) **Konjunktion.**

Die AND-Verknüpfung zweier Aussagen ist wahr, wenn beide Aussagen wahr sind.

$$x \wedge y = \Phi_{\wedge}(x, y) = \begin{cases} W & \text{wenn } x = W \text{ und } y = W \\ F & \text{sonst} \end{cases}$$

(3) **Disjunktion.**

Die OR-Verknüpfung zweier Aussagen ist wahr, wenn zumindest eine der beiden Aussagen wahr ist.

$$x \vee y = \Phi_{\vee}(x, y) = \begin{cases} F & \text{wenn } x = F \text{ und } y = F \\ W & \text{sonst} \end{cases}$$

(4) **Subjunktion.**

Die Subjunktion

$$x \rightarrow y = \Phi_{\rightarrow}(x, y) = \begin{cases} F & \text{wenn } x = W \text{ und } y = F \\ W & \text{sonst} \end{cases}$$

drückt den Sachverhalt der Implikation aus, siehe 1.4.

(5) **Bijunktion.**

Die Bijunktion

$$x \leftrightarrow y = \Phi_{\leftrightarrow}(x, y) = \begin{cases} W & \text{wenn } x = y \\ F & \text{sonst} \end{cases}$$

drückt die logische Äquivalenz zweier Aussagen aus.

(6) **Schefferscher Strich.** Bekannt auch unter dem Namen NAND-Verknüpfung

$$x|y = \Phi_{|}(x, y) = \begin{cases} F & \text{wenn } x = W \text{ und } y = W \\ W & \text{sonst} \end{cases}$$

(7) Die XOR-Verknüpfung zweier Aussagen ist wahr, wenn genau eine der Aussagen wahr ist.

$$x \dot{\vee} y = \Phi_{\dot{\vee}}(x, y) = \begin{cases} F & \text{wenn } x = y \\ W & \text{wenn } x \neq y \end{cases}$$

(1.3) Die folgende Auflistung, eine sogenannte *Wahrheitstafel*, gibt einen Überblick über die Wahrheitsfunktionen.

x	y	$x \wedge y$	$x \vee y$	$x \rightarrow y$	$x \leftrightarrow y$	$x y$	$x \dot{\vee} y$
W	W	W	W	W	F	F	F
W	F	F	W	F	F	W	W
F	W	F	W	W	F	W	W
F	F	F	F	W	W	W	F

(1.4) Bemerkung. Der am schwierigsten zu verstehende Junktor ist die Subjunktion. Andere Ausdrucksweisen für die Subjunktion $A \rightarrow B$ sind:

Wenn A , dann B .

A ist hinreichend für B .

A impliziert B .

B ist notwendig für A .

Aus A folgt B .

A nur wenn B .

(1.5) Beispiel. Die Aussage „Eine differenzierbare Funktion ist stetig“ kann man wie folgt anders formulieren:

- Differenzierbarkeit impliziert Stetigkeit.
- Aus Differenzierbarkeit folgt Stetigkeit.
- Differenzierbarkeit ist hinreichend für Stetigkeit.
- Stetigkeit ist notwendig für Differenzierbarkeit.

2. Die Sprache der Aussagenlogik

(2.1) Definition. Sei $\mathcal{V} = \{A_i \mid i \in \mathbb{N}\}$ und $\mathcal{J} = \{\neg, \wedge, \vee, \rightarrow, \leftrightarrow, |\}\}. Das Alphabet der Aussagenlogik ist $\mathcal{A} = \mathcal{V} \cup \mathcal{J} \cup \{(,)\}$. Die Elemente von \mathcal{V} nennen wir aussagenlogische Variable, und die Elemente von \mathcal{J} Junktoren. Die Menge aller Worte mit Buchstaben aus \mathcal{A} (inklusive des leeren Wortes) bezeichnen wir mit \mathcal{A}^* und die Menge aller Worte positiver Länge mit \mathcal{A}^+ .$

Unter den Wörtern aus \mathcal{A}^* wollen wir induktiv eine Teilmenge „sinnvoller“ Ausdrücke definieren, die Sprache \mathcal{L} der Aussagenlogik. Die Elemente von \mathcal{L} nennen wir Formeln der Aussagenlogik. Informell lautet die Definition:

- (1) Jede aussagenlogische Variable ist eine Formel der Aussagenlogik und
- (2) wenn A und B Formeln der Aussagenlogik sind, dann auch $(\neg A)$, $(A \wedge B)$, $(A \vee B)$, $(A \rightarrow B)$, $(A \leftrightarrow B)$ und $(A|B)$.

Formal definieren wir zu jedem Junktor eine Konstruktion, die Worte in \mathcal{A}^* zu einem neuen Wort verknüpft, d.h. eine (1- oder 2-stellige) Funktion auf \mathcal{A}^* , nämlich

$$K_{\neg}: \mathcal{A}^* \longrightarrow \mathcal{A}^* \quad K_{\neg}(A) = (\neg A) \quad \text{und}$$

$$K_{\wedge}: \mathcal{A}^* \times \mathcal{A}^* \longrightarrow \mathcal{A}^* \quad K_{\wedge}(A, B) = (A \wedge B)$$

und K_{\vee} , K_{\rightarrow} , K_{\leftrightarrow} , $K_{|}$ analog zu K_{\wedge} .

Wir gelangen zu folgender Definition.

(2.2) Definition. Sei $\mathcal{A} = \{A_i \mid i \in \mathbb{N}\} \cup \{\neg, \wedge, \vee, \rightarrow, \leftrightarrow, |\} \cup \{(,)\}$ und $K_{\neg}, K_{\wedge}, K_{\vee}, K_{\rightarrow}, K_{\leftrightarrow}, K_{|}$ wie oben. Wir definieren $\mathcal{L} \subseteq \mathcal{A}^*$ induktiv durch

- (1) $\{A_1, A_2, A_3, \dots\} \subseteq \mathcal{L}$
- (2) Wenn $\alpha, \beta \in \mathcal{L}$, dann sind auch $K_{\neg}(\alpha), K_{\wedge}(\alpha, \beta), K_{\vee}(\alpha, \beta), K_{\rightarrow}(\alpha, \beta), K_{\leftrightarrow}(\alpha, \beta)$ und $K_{|}(\alpha, \beta)$ in \mathcal{L} .

Eine Formel, die genau n Variablen A_1, A_2, \dots, A_n enthält, nennt man auch *aussagenlogische Aussageform* in den Variablen A_1, A_2, \dots, A_n .

Aus der Definition von \mathcal{L} mit Hilfe der *induktiven Struktur* (s.u.) $(\mathcal{B}, \mathcal{K}) = (\mathcal{V}, \{K_{\neg}, K_{\wedge}, K_{\vee}, K_{\rightarrow}, K_{\leftrightarrow}, K_{|}\})$ sehen wir, dass jede aussagenlogische Formel entweder ein Element von $\mathcal{B} = \mathcal{V}$ ist, oder durch eine Verknüpfung von Funktionen aus $\mathcal{K} = \{K_{\neg}, K_{\wedge}, K_{\vee}, K_{\rightarrow}, K_{\leftrightarrow}, K_{|}\}$, angewandt auf Elemente von \mathcal{V} , erzeugt wird, z.B.

$$((A_2 \wedge A_3) \rightarrow A_1) = K_{\rightarrow}(K_{\wedge}(A_2, A_3), A_1).$$

Die formale Sprache der Logik über der Variablenmenge \mathcal{V} ist eine sogenannte *kontextfreie Sprache*¹ und rekursiv definiert durch die BNF-Grammatik²:

$$\begin{aligned} \mathcal{L} ::= & \mathcal{V} \cup \{\top, \perp\} \\ | & \neg \mathcal{L} && \text{Negation} \\ | & \mathcal{L} \wedge \mathcal{L} && \text{Konjunktion} \\ | & \mathcal{L} \vee \mathcal{L} && \text{Disjunktion} \\ | & \mathcal{L} \rightarrow \mathcal{L} && \text{Implikation} \\ | & \mathcal{L} \leftrightarrow \mathcal{L} && \text{Äquivalenz} \\ | & \mathcal{L} | \mathcal{L} && \text{Scheffer} \end{aligned}$$

Hierbei steht \top für eine grundsätzlich wahre Aussage (auch „Tautologie“, siehe Definition 4.4) und \perp für eine grundsätzlich falsche Aussage (auch „Kontradiktion“, siehe Definition 4.4).

(2.3) Bemerkung. Die folgenden Ausdrücke sind mangels Klammern keine aussagenlogischen Formeln:

$$A \rightarrow B \rightarrow C \quad A \wedge B \vee C$$

¹strenggenommen sind kontextfreie Sprachen nur für endliche Variablenmengen definiert.

²Backus-Naur-Form

3. Exkurs: Induktive Strukturen

(3.1) Definition. Eine *induktive Definition* einer Menge M besteht aus

- (1) einer Vorschrift, dass die Elemente einer Grundmenge \mathcal{B} definitionsgemäß zu M gehören und
- (2) einer Vorschrift, dass für jede Konstruktion K aus einer Menge \mathcal{K} von Konstruktionen (Funktionen) das Resultat von K (der Funktionswert von K) in M liegt, sofern die Objekte, auf die K angewendet wurde (die Argumente der Funktion K), alle in M liegen.

Die definierte Menge M umfasst dann die und nur die Elemente, die aufgrund dieser Vorschriften in M liegen müssen, d.h. jene, die entweder schon in \mathcal{B} sind oder, ausgehend von Elementen in \mathcal{B} , in endlich vielen Schritten durch Anwendung von Konstruktionen aus \mathcal{K} (auf Elemente von \mathcal{B} und bereits konstruierte Objekte) gewonnen werden können. Das Paar $(\mathcal{B}, \mathcal{K})$ heißt *induktive Struktur* von M .

(3.2) Bemerkung. Eine Menge kann natürlich viele verschiedene induktive Strukturen haben.

Das Gute an einer induktiven Definition ist, dass sie die Möglichkeit von induktiven Beweisen eröffnet, d.h., um eine Eigenschaft für jedes Element von M zu zeigen, genügt es,

- (1) (Induktionsanfang) die Eigenschaft für jedes Element von \mathcal{B} zu zeigen und
- (2) (Induktionsschritt) zu zeigen, dass jede Konstruktion in \mathcal{K} die Eigenschaft erhält (d.h., dass das Ergebnis der Konstruktion die Eigenschaft hat, sofern alle Argumente die Eigenschaft haben).

(3.3) Beispiel. Die übliche „Induktion“ (für die Menge der natürlichen Zahlen \mathbb{N}) ist ein Beispiel für einen solchen auf einer induktiven Struktur (welcher?) der Menge \mathbb{N} beruhenden induktiven Beweis.

Im Fall von \mathcal{L} ist diese Darstellung auch eindeutig, d.h., jede aussagenlogische Formel ist entweder eine Variable (d.h. ein Element von \mathcal{B}) oder das Ergebnis einer eindeutig bestimmten Verknüpfung von Konstruktionen auf eindeutig

bestimmte Variable (und nicht beides zugleich). Wenn solches für eine induktive Struktur $(\mathcal{B}, \mathcal{K})$ zutrifft, dann sagt man, $(\mathcal{B}, \mathcal{K})$ erfüllt eindeutige Lesbarkeit.

Dass die Sprache der Aussagenlogik die eindeutige Lesbarkeit erfüllt, zeigt man mit folgendem Lemma:

Unter einem echten Anfangsabschnitt eines Wortes $\alpha \in \mathcal{A}^*$ verstehen wir ein $\beta \in \mathcal{A}^+$, sodass ein $\gamma \in \mathcal{A}^+$ existiert mit $\beta\gamma = \alpha$. (Weder das leere Wort noch α selbst ist also ein echter Anfangsabschnitt von α .)

(3.4) Lemma. Sei $A \in \mathcal{L}$. Dann ist kein echter Anfangsabschnitt von A selbst ein Element von \mathcal{L} .

Beweis. Man zeigt (durch Induktion nach der Struktur von \mathcal{L}), dass in jeder Formel $A \in \mathcal{L}$ gleich viele linke Klammern wie rechte Klammern vorkommen, und dass in jedem echten Anfangsabschnitt einer Formel aus \mathcal{L} mehr linke als rechte Klammern vorkommen.

(3.5) Satz. Für jede Formel $A \in \mathcal{L}$ gilt genau einer der Fälle

- (1) $A \in \mathcal{V}$
- (2) es gibt eine Formel $B \in \mathcal{L}$, sodass $A = (\neg B)$
- (3) es gibt Formeln $B, C \in \mathcal{L}$ und einen Junktor $* \in \{\wedge, \vee, \rightarrow, \leftrightarrow, | \}$, sodass $A = (B * C)$

Im Fall (2) ist B eindeutig und im Fall (3) sind der Junktor $*$ und die Formeln B, C eindeutig.

(3.6) Korollar. Die Sprache der Aussagenlogik (induktiv definiert wie in Definition 2.2) erfüllt eindeutige Lesbarkeit.

Eindeutige Lesbarkeit einer induktiven Struktur $(\mathcal{B}, \mathcal{K})$ einer Menge M ermöglicht es uns, eine Funktion f auf M induktiv zu definieren, nämlich durch Angabe von Werten von f auf den Elementen von \mathcal{B} und Angabe einer Methode, für jede Konstruktion in \mathcal{K} den Wert von f für das Resultat der Konstruktion aus den Werten von f für die Argumente der Konstruktion zu bestimmen. (Ohne eindeutige Lesbarkeit könnten die verschiedenen Arten, ein Element von M zu konstruieren, zu widersprüchlichen Vorschriften für den Wert von f führen.)

(3.7) Lemma. Sei M eine Menge mit induktiver Struktur $(\mathcal{B}, \mathcal{K})$, welche die eindeutige Lesbarkeit erfüllt, und S eine Menge. Gegeben sei eine Funktion $f_0: \mathcal{B} \rightarrow S$ und für jede n -stellige Funktion $K \in \mathcal{K}$ eine Funktion $\Phi_K: S^n \rightarrow S$. Dann gibt es genau eine Funktion $\bar{f}: M \rightarrow S$ mit den Eigenschaften

- (1) für jedes $\alpha \in \mathcal{B}$ gilt $\bar{f}(\alpha) = f(\alpha)$ und
- (2) für jedes $K \in \mathcal{K}$ und jedes $\alpha = K(\beta_1, \dots, \beta_n)$ gilt

$$\bar{f}(\alpha) = \Phi_K(\bar{f}(\beta_1), \dots, \bar{f}(\beta_n)).$$

4. Aussagenlogik

Wir haben die Sprache der Aussagenlogik definiert, und wollen den Formeln der Aussagenlogik jetzt diejenige inhaltliche Bedeutung zuschreiben (Semantik), die uns bei der Definition der Syntax schon vorgeschwobt ist. Die aussagenlogischen Variablen stehen im Prinzip für beliebige Aussagen, wie „Mir ist fad.“ oder „ $e^{\pi i} = -1$ “. Im Rahmen der Aussagenlogik interessiert uns an diesen Aussagen aber nur, dass sie entweder wahr oder falsch sein können, nicht deren Inhalt. Die Aussagenlogik handelt davon, wie der Wahrheitsgehalt einer logischen Verknüpfung von Aussagen sich aus dem Wahrheitsgehalt der Einzelaussagen ergibt. Z.B. ist die bekannte Bauernregel „Kräht der Hahn auf dem Mist, ändert sich's Wetter oder es bleibt wie's ist.“ allein aufgrund ihrer logischen Struktur „ $(A_1 \rightarrow (A_2 \vee (\neg A_2)))$ “ wahr, egal ob der Hahn kräht oder nicht, und egal wie das Wetter wird.

Bei den aussagenlogischen Variablen beschränken wir uns also darauf, jeder Variable den Wert „wahr“ oder „falsch“ zuzuweisen, und definieren die Bedeutung der Junktoren dadurch, dass wir für jeden Junktor eine Funktion angeben, welche abhängig von den Wahrheitswerten von Einzelaussagen den Wahrheitswert der entsprechenden Verknüpfung der Aussagen angibt.

(4.1) Definition. Eine Belegung (mit Wahrheitswerten) ist eine Funktion $\beta: \mathcal{V}' \rightarrow \{W, F\}$, wobei \mathcal{V}' eine Teilmenge der Menge der aussagenlogischen Variablen \mathcal{V} ist. Eine vollständige Belegung mit Wahrheitswerten ist eine Funktion $\beta: \mathcal{V} \rightarrow \{W, F\}$.

(4.2) Definition. Sei $\beta: \mathcal{V} \rightarrow \{W, F\}$ eine vollständige Belegung mit Wahrheitswerten. Wir definieren induktiv eine Fortsetzung von β auf \mathcal{L} , $\bar{\beta}: \mathcal{L} \rightarrow \{W, F\}$:

- (1) für $A_i \in \mathcal{V}$ sei $\bar{\beta}(A_i) = \beta(A_i)$.
- (2a) für $A = (\neg B)$ sei $\bar{\beta}(A) = \Phi_{\neg}(\bar{\beta}(B))$
- (2b) für $A = (B * C)$ mit $* \in \{\wedge, \vee, \rightarrow, \leftrightarrow, |\}$ sei $\bar{\beta}(A) = \Phi_{*}(\bar{\beta}(B), \bar{\beta}(C))$

$\bar{\beta}(A)$ heißt der Wahrheitswert von A unter β , oft auch geschrieben $\llbracket A \rrbracket \beta$.

(4.3) Lemma. Sei A eine Formel der Aussagenlogik, $\mathcal{V}' \subseteq \mathcal{V}$ die Menge der in A vorkommenden Variablen, und β, γ zwei Belegungen. Wenn für jede Variable $A_i \in \mathcal{V}'$ gilt $\beta(A_i) = \gamma(A_i)$, dann gilt $\bar{\beta}(A) = \bar{\gamma}(A)$.

Dieses (durch Induktion nach der Struktur von \mathcal{L}) leicht zu beweisende Lemma besagt also, dass der Wahrheitswert von A unter der Belegung β nur von deren Werten auf denjenigen (endlich vielen!) Variablen abhängt, die in A vorkommen.

Es definiert also jede (nicht notwendig vollständige) Belegung β , deren Definitionsbereich die in A vorkommenden Variablen umfasst, einen Wahrheitswert für A : wir setzen $\bar{\beta}(A)$ einfach gleich $\bar{\gamma}(A)$ für eine beliebige vollständige Belegung γ , die auf den in A vorkommenden Variablen mit β übereinstimmt.

(4.4) Definition. Eine vollständige Belegung $\beta: \mathcal{V} \rightarrow \{W, F\}$ mit Wahrheitswerten erfüllt eine Formel $A \in \mathcal{L}$ (geschrieben $\beta \models A$), falls $\bar{\beta}(A) = W$. Man sagt auch, β ist ein Modell für A .

Eine Formel $A \in \mathcal{L}$ heißt erfüllbar, wenn eine Belegung $\beta: \mathcal{V} \rightarrow \{W, F\}$, für die $\bar{\beta}(A) = W$ gilt, existiert.

Die Menge der Belegungen β , für die gilt $\bar{\beta}(A) = W$, heißt Erfüllungsmenge von A .

Eine Formel $A \in \mathcal{L}$ heißt eine Tautologie, geschrieben

$$\models A,$$

wenn $\beta \models A$ für jede Belegung $\beta: \mathcal{V} \rightarrow \{W, F\}$ gilt.

Eine Formel $A \in \mathcal{L}$ heißt unerfüllbar oder Kontradiktion, wenn für jede Belegung $\beta: \mathcal{V} \rightarrow \{W, F\}$ gilt $\bar{\beta}(A) = F$.

Aus der Definition von Φ_{\neg} folgt sofort, dass eine Formel A genau dann eine Tautologie ist, wenn $(\neg A)$ unerfüllbar ist. Genauso ist A erfüllbar genau dann, wenn $(\neg A)$ keine Tautologie ist.

Sei A eine Formel, die keine Variable außer A_1, \dots, A_n enthält. Wenn wir β alle Belegungen der Variablen A_1, \dots, A_n durchlaufen lassen und jeweils $\bar{\beta}(A)$ notieren, erhalten wir eine Funktion $\Phi_A: \{W, F\}^n \rightarrow \{W, F\}$, die Wahrheitsfunktion von A , d.h. $\Phi_A(\beta(A_1), \dots, \beta(A_n)) = \bar{\beta}(A)$. Aus der Definition von $\bar{\beta}$ sieht man, dass die Wahrheitsfunktion von A genau jene Funktion ist, die man durch Ersetzen der Konstruktionen K_* in der Darstellung von A durch die entsprechenden Wahrheitsfunktionen Φ_* erhält.

(4.5) Beispiel. Sei $A = (\neg(A_1) \wedge A_2) \rightarrow (\neg(A_3 \wedge A_4))$, dann ist

$$\Phi_A(A_1, \dots, A_4) = \Phi_{\rightarrow}(\Phi_{\wedge}(\Phi_{\neg}(A_1), A_2), (\Phi_{\neg}(\Phi_{\wedge}(A_3, A_4)))).$$

Um Festzustellen, ob eine Formel A , die n verschiedene Variable enthält, erfüllbar ist (bzw. eine Tautologie ist), genügt es nach Lemma 4.3 die Wahrheitswerte von A unter den 2^n verschiedenen Belegungen der in A vorkommenden Variablen (bzw. die Wahrheitsfunktion von A) zu betrachten.

Der exponentielle Aufwand dieses Verfahrens ist nicht zufällig; das Problem, aussagenlogische Formeln auf Erfüllbarkeit zu testen, ist NP-vollständig. (Es gibt aber Verfahren, die für eine umfangreiche Teilmenge von \mathcal{L} schnell zum Ziel führen.)

5. Äquivalenz von aussagenlogischen Formeln

(5.1) Definition. Seien A und B aussagenlogische Formeln. Wir schreiben $A \Rightarrow B$, wenn jede vollständige Belegung β , die A erfüllt, auch B erfüllt.

A und B heißen äquivalent (geschrieben $A \Leftrightarrow B$), wenn für jede vollständige Belegung β gilt $\bar{\beta}(A) = \bar{\beta}(B)$.

(5.2) Beispiel. Die Formeln $A \wedge B$ und $B \wedge A$ sind äquivalent, aber nicht gleich.

Da nach Lemma 4.3 die Wahrheitswerte von A und B bei der Belegung β nur von den Werten $\beta(A_i)$ für jene Variablen A_i , die in A oder B vorkommen, abhängen, gilt $A \Leftrightarrow B$ genau dann, wenn die Wahrheitsfunktionen von A und

B übereinstimmen. (Die beiden Wahrheitsfunktionen definieren wir dabei auf einer gemeinsamen Obermenge der in A und der in B vorkommenden Variablen.) Wenn β eine (nicht notwendig vollständige) Belegung ist, die für alle in A vorkommenden Variablen definiert ist, dann schreiben wir auch $\Phi_A(\beta)$ für $\Phi_A(\beta(A_1), \dots, \beta(A_n))$.

(5.3) Beispiel. Wir zeigen anhand einer Wahrheitstafel, dass $A \rightarrow B \iff \neg(A \wedge \neg B)$.

A	B	$A \rightarrow B$	A	$\neg B$	$A \wedge \neg B$	$\neg(A \wedge \neg B)$
W	W	W	W	F	F	W
W	F	F	W	W	W	F
F	W	W	F	F	F	W
F	F	W	F	W	F	W

(5.4) Bemerkung. Es gibt genau 2^{2^n} paarweise nichtäquivalente logische Aussageformeln in n Variablen A_1, \dots, A_n .

(5.5) Bemerkung. $A \leftrightarrow B$ ist eine aussagenlogische Formel; $A \Leftrightarrow B$ hingegen ist eine (metasprachliche) Aussage über zwei aussagenlogische Formeln. Es besteht folgender Zusammenhang:

(5.6) Lemma. Seien A und B aussagenlogische Formeln. Dann gilt $A \Rightarrow B$ genau dann, wenn $A \rightarrow B$ eine Tautologie ist. Genauso gilt $A \Leftrightarrow B$ genau dann, wenn $A \leftrightarrow B$ eine Tautologie ist.

Beweis. Übung. □

Im informellen Gebrauch lassen wir in aussagenlogischen Formeln oft jene Klammern weg, die sich durch folgende Konvention rekonstruieren lassen:

\neg bindet stärker als alle zweistelligen Junktoren (d.h., \neg „geht vor“) und \wedge und \vee binden stärker als \rightarrow und \leftrightarrow (d.h., \wedge und \vee gehen vor \rightarrow und \leftrightarrow).

Auch lassen wir die äußerste Klammer einer Formel oft weg; siehe jedoch Bemerkung 2.3.

(5.7) Beispiel.

$$\begin{array}{lll} \neg A \vee B & \text{meint} & ((\neg A) \vee B), \\ A \wedge \neg B \rightarrow C & \text{meint} & ((A \wedge (\neg B)) \rightarrow C) \end{array}$$

Alternativ zur Methode der Wahrheitstafeln kann man Äquivalenzen auch algebraisch beweisen, indem man Teilformeln durch semantisch äquivalente Formeln ersetzt.

(5.8) Lemma. Sei A eine aussagenlogische Formel, A_1, \dots, A_n aussagenlogische Variable und B_1, \dots, B_n weitere aussagenlogische Formeln. Wenn wir in A jedes Vorkommen von A_i durch B_i ersetzen (für alle i gleichzeitig), dann ist das Resultat dieser Ersetzung, geschrieben $A(A_1/B_1, \dots, A_n/B_n)$ wieder eine aussagenlogische Formel.

(5.9) Beispiel. Betrachte die Aussageformen $A = A_1 \wedge A_2 \rightarrow A_1 \wedge A_3$ und $B_1 = C_1 \vee C_2$, $B_2 = C_1 \rightarrow C_3$ und $B_3 = C_1 \wedge C_3$. Dann ist die Ersetzung

$$A(A_1/B_1, A_2/B_2, A_3/B_3) = (C_1 \vee C_2) \wedge (C_1 \rightarrow C_3) \rightarrow (C_1 \vee C_2) \wedge (C_1 \wedge C_3)$$

(5.10) Bemerkung. Es folgt eine Tafel von allerlei nützlichen und teils mit historischen Namen versehenen Äquivalenzen, die sich durch Vergleich der Wahrheitstafeln leicht beweisen lassen. Hier stehen A , B , C für beliebige aussagenlogische Formeln.

	$\neg\neg A \iff A$	(Negationsgesetz)
	$(A \wedge B) \wedge C \iff A \wedge (B \wedge C)$	
	$(A \vee B) \vee C \iff A \vee (B \vee C)$	(Assoziativität)
	$A \wedge B \iff B \wedge A$	
	$A \vee B \iff B \vee A$	(Kommutativität)
	$A \wedge (B \vee C) \iff (A \wedge B) \vee (A \wedge C)$	
	$A \vee (B \wedge C) \iff (A \vee B) \wedge (A \vee C)$	(Distributivität)
(5.11)	$\neg(A \vee B) \iff \neg A \wedge \neg B$	
	$\neg(A \wedge B) \iff \neg A \vee \neg B$	(de Morgan ³)

$$\begin{aligned} A \rightarrow B &\iff \neg A \vee B && (\text{Gesetz der Implikation}) \\ A \rightarrow B &\iff \neg B \rightarrow \neg A && (\text{Kontrapositionsgesetz}) \end{aligned}$$

$$\begin{aligned} A \vee \neg A &\iff \top && (\text{Tertium non datur}) \\ A \wedge \neg A &\iff \perp && (\text{Gesetz des Widerspruchs}) \end{aligned}$$

$$(A \rightarrow B) \wedge (B \rightarrow C) \implies (A \rightarrow C) \quad (\text{Transitivität})$$

Sei K eine Kontradiktion und T eine Tautologie, dann gilt

$$\begin{aligned} \perp &\implies A && (\text{ex falso quodlibet}) \\ A &\implies \top && (\text{ex quodlibet verum}) \end{aligned}$$

(5.12) Satz. Das logische Schließen beruht auf den folgenden Einsetzungsregeln.

Einsetzungsregel 1: Wenn $A(A_1, \dots, A_n)$ eine Tautologie ist und B_i beliebige Ausdrücke, dann ist auch $A(A_1/B_1, \dots, A_n/B_n)$ eine Tautologie.

Einsetzungsregel 2: Ersetzt man in einer beliebigen logischen Formel einen Teilausdruck durch einen dazu äquivalenten Ausdruck, so entsteht eine zur gegebenen Formel äquivalente Formel.

(5.13) Beispiel. Man kann mithilfe der obengenannten Regeln Äquivalenzen beweisen wie im folgenden Beispiel. Wir zeigen das Kontrapositionsgesetz

$$A \rightarrow B \iff \neg B \rightarrow \neg A$$

mithilfe der anderen Regeln.

Beweis.

$$\begin{aligned} A \rightarrow B &\iff \neg A \vee B && (\text{Implikationsgesetz}) \\ &\iff B \vee \neg A && (\text{Kommutativität}) \\ &\iff \neg \neg B \vee \neg A && (\text{Negationsgesetz}) \\ &\iff \neg B \rightarrow \neg A && (\text{Implikationsgesetz}) \end{aligned}$$

³Auguste de Morgan (1806–1871)

(5.14) Beispiel. Die Formel $((P \rightarrow Q) \wedge P) \rightarrow Q$ ist eine Tautologie, bekannt als *Modus Ponens*.

Beweis.

$$\begin{aligned}
 ((P \rightarrow Q) \wedge P) \rightarrow Q &\iff \neg((P \rightarrow Q) \wedge P) \vee Q && (\text{Implikationsgesetz}) \\
 &\iff (\neg(P \rightarrow Q) \vee \neg P) \vee Q && (\text{de Morgan}) \\
 &\iff \neg(P \rightarrow Q) \vee (\neg P \vee Q) && (\text{Assoziativitat}) \\
 &\iff \neg(P \rightarrow Q) \vee (P \rightarrow Q) && (\text{Implikationsgesetz}) \\
 &\iff \top && (\text{Tertium non datur})
 \end{aligned}$$

(5.15) Definition. Eine Aussageform $P(A_1, A_2, \dots, A_n)$ heit *aussagenlogische Folgerung* oder *Konklusion* aus den Aussageformen

$$P_1(A_1, A_2, \dots, A_n), \dots, P_m(A_1, A_2, \dots, A_n),$$

den *Pramissen*, wenn fur jede Belegung β der Variablen A_1, A_2, \dots, A_n mit

$$\begin{aligned}
 \bar{\beta}(P_1(A_1, A_2, \dots, A_n)) &= W \\
 \bar{\beta}(P_2(A_1, A_2, \dots, A_n)) &= W \\
 &\vdots \\
 \bar{\beta}(P_m(A_1, A_2, \dots, A_n)) &= W
 \end{aligned}$$

auch gilt $\bar{\beta}(P(A_1, A_2, \dots, A_n)) = W$; mit anderen Worten,

$$P_1(A_1, A_2, \dots, A_n) \wedge \dots \wedge P_m(A_1, A_2, \dots, A_n) \implies P(A_1, A_2, \dots, A_n),$$

bzw. $P_1(A_1, A_2, \dots, A_n) \wedge \dots \wedge P_m(A_1, A_2, \dots, A_n) \rightarrow P(A_1, A_2, \dots, A_n)$ ist eine Tautologie.

Die Pramissenmenge P_1, \dots, P_m heit *konsistent*, wenn $P_1 \wedge \dots \wedge P_m$ keine Kontradiktion ist.

Die Folgerung P heit *trivial*, wenn P eine Tautologie ist. Im letzteren Fall gilt $P_1 \wedge \dots \wedge P_m \implies P$ fur jede beliebige Pramissenmenge $\{P_1, \dots, P_m\}$.

(5.16) Definition. Eine Teilmenge \mathcal{L}' der Sprache der Aussagenlogik \mathcal{L} heit *vollstandig*, wenn jede Formel aus \mathcal{L} zu einer Formel aus \mathcal{L}' aquivalent ist.

(5.17) Definition. Sei $\mathcal{J}' \subseteq \mathcal{J} = \{\neg, \wedge, \vee, \rightarrow, \leftrightarrow, | \}$ eine Menge von Junktoren. Dann bezeichnen wir mit $\mathcal{L}_{\mathcal{J}'}$ die Menge jener Formeln von \mathcal{L} , die keine Junktoren außer jenen in \mathcal{J}' enthalten. \mathcal{J}' heißt Verknüpfungsbasis oder Boole⁴-Basis, wenn die Formelmenge $\mathcal{L}_{\mathcal{J}'}$ vollständig ist.

(5.18) Beispiel. $\mathcal{L}_{\{\neg, \vee\}}$ ist vollständig.

Lösung. Die Junktoren \leftrightarrow und $|$ lassen sich durch andere Junktoren ausdrücken:

$$\begin{aligned} A \leftrightarrow B &\iff (A \rightarrow B) \wedge (B \rightarrow A) \\ A|B &\iff \neg(A \wedge B) \end{aligned}$$

Es genügt also, die restlichen fehlenden Junktoren \wedge, \rightarrow durch \neg und \vee auszudrücken:

$$\begin{aligned} A \wedge B &\iff \neg(\neg A \vee \neg B) \\ A \rightarrow B &\iff \neg A \vee B \end{aligned}$$

Wir überprüfen dies anhand einer Wahrheitstafel:

A	B	$A \wedge B$	$\neg A \vee \neg B$	$\neg(\neg A \vee \neg B)$	$A \rightarrow B$	$\neg A \vee B$
W	W	W	F	W	W	W
W	F	F	W	F	F	F
F	W	F	W	F	W	W
F	F	F	W	F	W	W

Wir können auf jene Junktoren verzichten, die sich durch andere Junktoren ausdrücken lassen, in folgendem Sinn:

(5.19) Satz. Sei $I \subseteq \{\neg, \wedge, \vee, \rightarrow, \leftrightarrow, |\}$.

i) Sei $* \in \{\wedge, \vee, \rightarrow, \leftrightarrow, |\}$. Wenn für jede Formel der Gestalt $P_1 * P_2$ mit $P_1, P_2 \in \mathcal{L}_I$ eine Formel $Q \in \mathcal{L}_I$ existiert, sodass $P_1 * P_2 \Leftrightarrow Q$ und $\mathcal{L}_{I \cup \{*\}}$ vollständig ist, dann ist auch \mathcal{L}_I vollständig. Beispiel: $\mathcal{J} \setminus \{\rightarrow\}$ ist vollständig.

ii) Wenn für jede Formel der Gestalt $\neg P$ mit $P \in \mathcal{L}_I$ eine Formel $Q \in \mathcal{L}_I$ existiert, sodass $\neg P \Leftrightarrow Q$, und $\mathcal{L}_{I \cup \{\neg\}}$ vollständig ist, dann ist auch \mathcal{L}_I vollständig.

⁴George Boole 1815–1864

6. Konjunktive und disjunktive Normalform

Aus der Assoziativität von \vee folgt durch Induktion, dass (für aussagenlogische Formeln B_1, \dots, B_n) alle aussagenlogischen Formeln der Gestalt $B_1 \vee B_2 \vee \dots \vee B_n$ (mit beliebiger Klammerung) äquivalent sind. Analog sind alle Formeln der Gestalt $B_1 \wedge B_2 \wedge \dots \wedge B_n$ (unabhängig von der Klammeranzahl) äquivalent. Wenn es uns nur auf die Wahrheitsfunktion einer Formel ankommt, können wir also die Klammern in Disjunktionen bzw. Konjunktionen von Formeln, weglassen.

Eine *Normalform* einer logischen Formel ist, grob gesprochen, eine nach gewissen Regeln daraus abgeleitete logisch äquivalente Formel, die überdies eindeutig ist, d.h., für zwei verschiedene aussagenlogisch äquivalente Formeln das gleiche Ergebnis liefert.

(6.1) Definition. Ein *Literal* ist entweder eine aussagenlogische Variable A_i (positives Literal) oder ein Ausdruck der Gestalt $\neg A_i$, wobei A_i eine aussagenlogische Variable ist (negatives Literal).

(6.2) Definition. Gegeben seien n verschiedene aussagenlogische Variable (o.B.d.A. die ersten n) A_1, \dots, A_n . Eine n -*Klausel* über A_1, A_2, \dots, A_n ist eine Formel der Gestalt

$$B_1 \vee B_2 \vee \dots \vee B_n,$$

wobei für jedes i entweder $B_i = A_i$ oder $B_i = \neg A_i$ (ein Literal).

Eine Formel ist in n -konjunktiver Normalform (n -KNF), wenn sie (für ein $k \in \mathbb{N}$) folgende Gestalt hat:

$$C_1 \wedge \dots \wedge C_k, \quad \text{jedes } C_i \text{ eine } n\text{-Klausel über } A_1, A_2, \dots, A_n.$$

(6.3) Satz. Sei A eine aussagenlogische Formel, die keine Variable außer A_1, \dots, A_n enthält.

Wenn A keine Tautologie ist, dann ist A äquivalent zu einer Formel in n -KNF.

Beweis. Wir ordnen jeder Belegung β von A_1, \dots, A_n eine Klausel C_β zu, und zwar

$$C_\beta = B_1 \vee \dots \vee B_n \quad \text{mit} \quad B_i = \begin{cases} A_i & \text{wenn } \beta(A_i) = F \\ \neg A_i & \text{wenn } \beta(A_i) = W \end{cases}$$

Dann gilt: β ist die einzige Belegung von A_1, \dots, A_n mit $\bar{\beta}(C_\beta) = F$.

Sei

$$\mathcal{F}_A = \{\beta: \{A_1, \dots, A_n\} \rightarrow \{W, F\} \mid \bar{\beta}(A) = F\}.$$

Da A keine Tautologie ist, ist $\mathcal{F}_A \neq \emptyset$. Sei $\mathcal{F}_A = \{\beta_1, \dots, \beta_k\}$, dann gilt

$$A \Leftrightarrow C_{\beta_1} \wedge \dots \wedge C_{\beta_k}.$$

(Es ist $C_{\beta_1} \wedge \dots \wedge C_{\beta_k}$ genau bei jenen Belegungen falsch, bei denen A falsch ist.) \square

(6.4) Beispiel. Wir bestimmen die n -KNF der Formel $A \wedge (B \vee C)$.

Lösung. Wir stellen die Wahrheitstafel auf:

A	B	C	$A \wedge (B \vee C)$
W	W	W	W
W	W	F	W
W	F	W	W
W	F	F	F
F	W	W	F
F	W	F	F
F	F	W	F
F	F	F	F

Dann suchen wir jene Belegungen, die den Wert F ergeben, bilden die entsprechenden Klauseln und erhalten

$$\begin{aligned} A \wedge (B \vee C) &\Leftrightarrow \\ (\neg A \vee B \vee C) \wedge (A \vee \neg B \vee \neg C) \wedge (A \vee \neg B \vee C) \wedge (A \vee B \vee \neg C) \wedge (A \vee B \vee C) \end{aligned}$$

(6.5) Definition. Gegeben seien n verschiedene aussagenlogische Variable (o.B.d.A. die ersten n) A_1, \dots, A_n . Eine *duale n-Klausel* über A_1, A_2, \dots, A_n ist eine Formel der Gestalt

$$B_1 \wedge B_2 \wedge \dots \wedge B_n,$$

wobei für jedes i entweder $B_i = A_i$ oder $B_i = \neg A_i$.

Eine Formel ist in *n-disjunktiver Normalform (n-DNF)*, wenn sie (für ein $k \in \mathbb{N}$) folgende Gestalt hat:

$$D_1 \vee \dots \vee D_k, \quad \text{jedes } D_i \text{ eine duale } n\text{-Klausel über } A_1, A_2, \dots, A_n.$$

(6.6) Satz. Sei A eine aussagenlogische Formel, die keine Variable außer A_1, \dots, A_n enthält.

Wenn A erfüllbar ist, dann ist A äquivalent zu einer Formel in n -DNF.

Beweis. Wir ordnen jeder Belegung β von A_1, \dots, A_n eine duale Klausel D_β zu, und zwar

$$D_\beta = B_1 \wedge \dots \wedge B_n \quad \text{mit} \quad B_i = \begin{cases} A_i & \text{wenn } \beta(A_i) = W \\ \neg A_i & \text{wenn } \beta(A_i) = F \end{cases}$$

Dann gilt: β ist die einzige Belegung von A_1, \dots, A_n mit $\bar{\beta}(D_\beta) = W$.

Sei \mathcal{W}_A die Menge aller Belegungen von A_1, \dots, A_n mit $\bar{\beta}(A) = W$. Da A erfüllbar ist, ist $\mathcal{W}_A = \{\beta_1, \dots, \beta_k\} \neq \emptyset$.

Es gilt

$$A \Leftrightarrow D_{\beta_1} \vee \dots \vee D_{\beta_k}.$$

□

(6.7) Beispiel. Wir bestimmen die n -DNF der Formel $A \wedge (B \vee C)$.

Lösung. Wir suchen in der Wahrheitstabelle von Beispiel 6.4 diejenigen Belegungen, die den Wert W ergeben, bilden die entsprechenden dualen Klauseln und erhalten

$$A \wedge (B \vee C) \iff (A \wedge B \wedge C) \vee (A \wedge B \wedge \neg C) \vee (A \wedge \neg B \wedge C)$$

Als Anwendung betrachten wir die Frage, wieviele Konklusionen (Definition 5.15) man aus einer gegebenen Prämissenmenge herleiten kann.

(6.8) Satz. Seien P_1, \dots, P_m n -stellige aussagenlogische Aussageformen und sei $B = B_1 \wedge B_2 \wedge \dots \wedge B_k$ die n -konjunktive Normalform des Konjugats $P_1 \wedge \dots \wedge P_m$. Dann besteht die Gesamtheit der nichttrivialen Folgerungen aus den Prämissen P_1, \dots, P_m aus allen möglichen Konjugaten der Terme B_1, \dots, B_k . Inklusive der trivialen Folgerung gibt es also genau 2^k nicht-äquivalente Folgerungen aus den P_i . Ist das Konjugat $P_1 \wedge \dots \wedge P_m$ jedoch eine Tautologie, so gibt es nur die triviale Folgerung.

(6.9) Beispiel. Gesucht sind alle Folgerungen aus den Prämissen

$$\begin{aligned} P_1 &:\iff B \rightarrow (\neg A \rightarrow C) \\ P_2 &:\iff B \vee (\neg A \wedge \neg C) \vee (A \wedge C) \\ P_3 &:\iff (\neg B \wedge \neg C) \rightarrow \neg A \\ P_4 &:\iff A \vee ((B \vee \neg C) \wedge (\neg B \vee C)) \end{aligned}$$

Lösung. Wir bilden die konjunktiven Normalformen der P_i

$$\begin{aligned} P_1 &\iff A \vee \neg B \vee C \\ P_2 &\iff (\neg A \vee B \vee C) \wedge (A \vee B \vee \neg C) \\ P_3 &\iff \neg A \vee B \vee C \\ P_4 &\iff (A \vee \neg B \vee C) \wedge (A \vee B \vee \neg C) \end{aligned}$$

Das Konjugat ist also

$$P_1 \wedge P_2 \wedge P_3 \wedge P_4 \iff (A \vee \neg B \vee C) \wedge (A \vee B \vee \neg C) \wedge (\neg A \vee B \vee C)$$

und neben der trivialen Folgerung gibt es die folgenden Folgerungen:

$$\begin{aligned} C_1 &:\iff (A \vee \neg B \vee C) \\ C_2 &:\iff (A \vee B \vee \neg C) \\ C_3 &:\iff (\neg A \vee B \vee C) \\ C_4 &:\iff (A \vee B \vee \neg C) \wedge (\neg A \vee B \vee C) \\ C_5 &:\iff (A \vee \neg B \vee C) \wedge (\neg A \vee B \vee C) \\ C_6 &:\iff (A \vee \neg B \vee C) \wedge (A \vee B \vee \neg C) \\ C_7 &:\iff (A \vee \neg B \vee C) \wedge (A \vee B \vee \neg C) \wedge (\neg A \vee B \vee C) \end{aligned}$$

(6.10) Definition. Eine *Klausel* ist eine Formel der Gestalt

$$B_1 \vee B_2 \vee \dots \vee B_n, \quad \text{jedes } B_i \text{ ein Literal.}$$

Eine Formel ist in *konjunktiver Normalform (KNF)*, wenn sie (für ein $k \in \mathbb{N}$) folgende Gestalt hat:

$$C_1 \wedge \dots \wedge C_k, \quad \text{jedes } C_i \text{ eine Klausel.}$$

Eine Formel in KNF wird oft durch eine „Klauselmenge“ abgekürzt; man schreibt $\{ \{B_{11}, \dots, B_{1n_1}\}, \dots, \{B_{k1}, \dots, B_{kn_k}\} \}$ für

$$(B_{11} \vee \dots \vee B_{1n_1}) \wedge \dots \wedge (B_{k1} \vee \dots \vee B_{kn_k}).$$

(6.11) Satz. Jede aussagenlogische Formel ist äquivalent zu einer Formel in konjunktiver Normalform.

Beweis. Das muss nur mehr für Tautologien gezeigt werden. Wenn A Tautologie, dann ist A äquivalent zu $A_1 \vee \neg A_1$. \square

(6.12) Definition. Eine *duale Klausel* ist eine Formel der Gestalt

$$B_1 \wedge B_2 \wedge \dots \wedge B_n, \quad \text{jedes } B_i \text{ ein Literal.}$$

Eine Formel ist in *disjunktiver Normalform (DNF)*, wenn sie (für ein $k \in \mathbb{N}$) folgende Gestalt hat:

$$D_1 \vee \dots \vee D_k, \quad \text{jedes } D_i \text{ eine duale Klausel.}$$

In einer DNF der Form $D_1 \vee \dots \vee D_k$ kommen also im Allgemeinen nicht in jedem der D_i dieselben Variablen vor, und eine duale Klausel darf (im Gegensatz zu einer dualen n -Klausel) auch Ausdrücke wie $A_i \wedge \neg A_i$ enthalten.

(6.13) Satz. Jede aussagenlogische Formel A ist äquivalent zu einer Formel in disjunktiver Normalform.

Beweis. Das muss nur mehr für unerfüllbare Formeln gezeigt werden. Wenn A unerfüllbar, dann ist A äquivalent zu $A_1 \wedge \neg A_1$. \square

(6.14) Notation Seien ℓ_i Literale, $i \in I = \{1, \dots, n\}$, dann definieren wir

$$\begin{aligned} \bigwedge_{i \in I} \ell_i &:= \ell_1 \wedge \ell_2 \wedge \dots \wedge \ell_n \\ \bigvee_{i \in I} \ell_i &:= \ell_1 \vee \ell_2 \vee \dots \vee \ell_n \end{aligned}$$

und weiters definieren wir

$$(6.15) \quad \bigwedge_{i \in \emptyset} \ell_i := \top \qquad \bigvee_{i \in \emptyset} \ell_i := \perp.$$

(Begründung in Worten: eine leere Konjunktion stellt keine Bedingungen und ist immer erfüllt, eine leere Disjunktion ist unerfüllbar, weil keine erfüllbare Formel vorhanden ist). Eine pragmatische Begründung für diese Konvention besteht darin, dass damit uneingeschränkt gilt

$$\begin{aligned} \left(\bigwedge_{i \in I} \ell_i \right) \wedge \left(\bigwedge_{i \in J} \ell_i \right) &= \bigwedge_{i \in I \cup J} \ell_i \\ \left(\bigvee_{i \in I} \ell_i \right) \vee \left(\bigvee_{i \in J} \ell_i \right) &= \bigvee_{i \in I \cup J} \ell_i. \end{aligned}$$

Außerdem besitzt mit der Konvention (6.15) jede aussagenlogische Formel eine KNF und eine DNF, auch Tautologie und Kontradiktion, nämlich die leere Klausel bzw. die leere Disjunktion.

Die Herleitung einer n -KNF/DNF mit Hilfe der Wahrheitstafel ist bereits behandelt worden.

Daneben kann eine KNF/DNF auch durch logische Umformungen nach folgendem Rezept konstruiert werden:

- (1) Ersetze $A \rightarrow B$ durch $\neg A \vee B$.
- (2) Verschiebe alle Negationen ins Innere der Klammern (de Morgan).
- (3) Wende die Distributivgesetze an, bis eine KNF/DNF übrigbleibt.

(6.16) Beispiel. Gesucht ist eine KNF der Formel

$$P = (A \wedge B) \vee (C \rightarrow D \wedge \neg A).$$

Wir verwenden die Gesetzentafel aus Bemerkung 5.10.

$$\begin{aligned} P &\iff (A \wedge B) \vee (\neg C \vee (D \wedge \neg A)) && \text{(Implik.)} \\ &\iff ((A \wedge B) \vee \neg C) \vee (D \wedge \neg A) && \text{(Assoz.)} \end{aligned}$$

Hiermit ist eine DNF erreicht.

$$\begin{aligned} &\iff [(A \vee \neg C) \wedge (B \vee \neg C)] \vee (D \wedge \neg A) && \text{(Distrib.)} \\ &\iff [(A \vee \neg C) \vee (D \wedge \neg A)] \wedge [(B \vee \neg C) \vee (D \wedge \neg A)] && \text{(Distrib.)} \\ &\iff (A \vee \neg C \vee D) \wedge (A \vee \neg C \vee \neg A) \\ &\quad \wedge (B \vee \neg C \vee D) \wedge (B \vee \neg C \vee \neg A) && \text{(Distrib.)} \\ &\iff (A \vee \neg C \vee D) \wedge (B \vee \neg C \vee D) \wedge (B \vee \neg C \vee \neg A) && \text{(TND)} \end{aligned}$$

TND („Tertium non datur“), das Gesetz vom ausgeschlossenen Dritten bezieht sich auf die Tautologie $A \vee \neg C \vee \neg A$, die aus der KNF gestrichen werden kann.

(6.17) Bemerkung. Zu beachten ist, dass die n -KNF/DNF bis auf die Reihenfolge eindeutig ist, während es verschiedene einfache KNF/DNF geben kann, z.B. ist

$$(A \wedge C) \vee (B \wedge C) \iff (A \wedge B \wedge C) \vee (A \wedge \neg B \wedge C) \vee (\neg A \wedge B \wedge C)$$

Aus einer vorhandenen *KNF* kann durch Ergänzung der fehlenden Variablen eine n -KNF hergeleitet werden.

7. Prädikatenlogik

Die Definition einer Gruppe diene als Beispiel für die Verwendung der Prädikatenlogik.

(7.1) Definition. Eine Gruppe ist definiert als Tripel (G, \circ, e) wobei G eine Menge ist, \circ eine zweistellige Operation und $e \in G$ (neutrales Element), wobei folgende Eigenschaften erfüllt sein müssen:

- (1) $\forall x \forall y \forall z x \circ (y \circ z) = (x \circ y) \circ z$
- (2) $\forall x x \circ e = x = e \circ x$
- (3) $\forall x \exists y x \circ y = e \wedge y \circ x = e$

Hier sind

x, y, z Variable

e eine Konstante

\circ ein Funktionssymbol

$=$ eine Relation

\forall, \exists Quantoren („für alle“ bzw. „es existiert ein“).

Der Ausdruck $x = y$ ist ein *Prädikat*, d.h., eine logische Aussage, die von Variablen abhängt.

(7.2) Definition. Eine Sprache 1. Stufe ist aus folgendem Alphabet aufgebaut:

- (1) Eine Variablenmenge $\mathcal{V} = \{x, y, z, \dots\}$
- (2) Eine Signatur Σ bestehend aus
 - (a) Konstanten (z.B. $e, 0, 1, 2, \dots$)
 - (b) Funktionssymbolen (z.B. $f, g, +, \cdot, \dots$)
 - (c) Relationssymbolen (z.B. $<, >, |, \dots$).

Dabei ist zu beachten, dass die Funktions- und Relationssymbole fixe Stelligkeiten (=Anzahl der Argumente) besitzen.

Die Menge der Terme wird induktiv definiert:

- (1) Alle Variablen und Konstante sind Terme.
- (2) Ist f ein n -stelliges Funktionssymbol und sind t_1, t_2, \dots, t_n Terme, dann ist auch $f(t_1, t_2, \dots, t_n)$ ein Term.

Die Menge der Formeln wird induktiv definiert:

- (1) \top, \perp sind Formeln.
- (2) Sind s und t Terme, dann ist $s = t$ eine Formel.
- (3) Sind t_1, t_2, \dots, t_n Terme und $r \in \Sigma$ ein n -stelliges Relationssymbol, dann ist $r(t_1, t_2, \dots, t_n)$ eine Formel.
- (4) Sind P und Q Formeln und $x \in \mathcal{V}$ eine Variable, dann sind auch

$$P \wedge Q \quad P \vee Q \quad \neg P \quad P \rightarrow Q \quad \forall x P \quad \exists x P$$

Formeln.

Die Formeln aus (1)–(3) heißen *Primformeln*. Wir bezeichnen die Menge der Formeln über dem Alphabet \mathcal{V} und der Signatur Σ mit $\mathcal{F}_{\mathcal{V}, \Sigma}$.

(7.3) Beispiel. Man könnte Beispiel 7.1 auch wie folgt schreiben ($f(x, y)$ entspricht $x \circ y$ und $g(x) = x^{-1}$):

$$\begin{aligned} \forall x \forall y \forall z f(x, f(y, z)) &= f(f(x, y), z) \\ \forall x (f(e, x) &= x \wedge f(x, e) = x) \\ \forall x (f(x, g(x)) &= e \wedge f(g(x), x) = e) \end{aligned}$$

(7.4) Definition. Sei P eine Formel. Die Menge $\text{FV}(P)$ der *freien Variablen* einer Formel ist induktiv definiert als

- (1) $\text{FV}(P) = \text{Variable, die in } P \text{ vorkommen, wenn } P \text{ eine Primformel ist}$ (Konstante sind keine Variable!).
- (2) $\text{FV}(\neg P) = \text{FV}(P)$.
- (3) $\text{FV}(P \wedge Q) = \text{FV}(P \vee Q) = \text{FV}(P) \cup \text{FV}(Q)$
- (4) $\text{FV}(\forall x P) = \text{FV}(\exists x P) = \text{FV}(P) \setminus \{x\}$. Die Variable x heißt in diesem Fall *gebunden*.

Eine geschlossene Formel oder Aussage ist eine Formel ohne freie Variable.

(7.5) Beispiel. Die folgenden Formeln sind geschlossen:

$$\forall x f(x, x) = x \quad \forall x \forall y x = y \quad \forall x \neg \forall y f(x, y) = f(y, x)$$

Beispiele für nicht geschlossene Formeln (eine Variable kann sowohl gebunden als auch ungebunden vorkommen!):

$$\begin{aligned}\text{FV}((\forall x \neg x = e) \wedge (\forall y y = f(x, z))) &= \{x, z\} \\ \text{FV}(\forall x (\neg x = e \wedge \forall y y = f(x, z))) &= \{z\}\end{aligned}$$

(7.6) Definition. Eine Σ -Struktur $\mathcal{A} = (A, \Sigma^{\mathcal{A}})$ besteht aus einer Grundmenge A und einer Strukturmenge $\Sigma^{\mathcal{A}}$, die

zu jeder Konstanten $c \in \Sigma$ eine Konstante $c^{\mathcal{A}} \in A$

zu jedem n -stetigen Funktionssymbol $f \in \Sigma$ eine Funktion $f : A^n \rightarrow A$

zu jedem n -stetigen Relationssymbol $r \in \Sigma$ eine Relation $r^{\mathcal{A}} \subseteq A^n$ enthält.

Eine Belegung ist eine Abbildung $\omega : V \rightarrow A$ und eine Σ -Struktur zusammen mit einer Belegung $\omega : V \rightarrow A$ heißt Σ -Modell, geschrieben $\mathcal{M} = (A, \Sigma^{\mathcal{A}}, \omega)$. Für einen Term oder eine Formel $P \in \mathcal{F}_{V, \Sigma}$ bezeichnet $P^{\mathcal{M}}$ die „Auswertung“ der Formel im Modell \mathcal{M} (letztere induktiv definiert, Details leicht eruierbar anhand des folgenden Beispiels).

(7.7) Beispiel. Die Gruppe $\mathcal{Z} = (\mathbb{Z}, +)$ ist eine Σ -Struktur für die Sprache erster Stufe über der Struktur $\Sigma = \{e, o\}$, die wir in Beispiel 7.1 für die Definition einer Gruppe verwendet haben, und zwar ist $o^{\mathcal{Z}} = +$ und $e^{\mathcal{Z}} = 0$. Sei $\omega : x \mapsto 1, y \mapsto 2$ und $\mathcal{M} = (\mathcal{Z}, \omega)$ das entsprechende Modell. Dann hat die Formel

$$P = \exists z x = y + z$$

die Auswertung

$$P^{\mathcal{M}} = \exists z 1 = 2 + z$$

(7.8) Definition. Man kann auch in der Prädikatenlogik Variablen substituieren, allerdings dürfen *nur freie Variablen* ersetzt werden. Darüber hinaus gibt es eine weitere Einschränkung: Betrachte die Formel

$$\exists x x = y$$

die in jedem Modell gilt. Sie bleibt auch gültig wenn wir z.B. $z + 1$ für y substituieren:

$$(\exists x x = y)[y/z + 1] = \exists x x = z + 1$$

Wenn wir allerdings $x + 1$ für y substituieren, erhalten wir

$$(\exists x x = y)[y/x + 1] = \exists x x = x + 1$$

die offensichtlich ungültig ist. Eine Substitution heißt *zulässig*, wenn der zu substituierende Term t frei für x in P ist, d.h., wenn bei der Substitution von t für y keine freie Variable von t gebunden wird.

Im obigen Beispiel kann man das Problem umgehen, indem man vor der Substitution eine *Umbenennung* durchführt, z.B., die gebundene Variable x in v umbenennt, d.h., die Substitution $y \mapsto x + 1$ mit dem äquivalenten Ausdruck

$$\exists v v = y$$

vornimmt. Solche Umbenennungen werden oft implizit vorgenommen.

(7.9) Definition. Sei $\mathcal{M} = (\mathcal{A}, \omega)$ ein Modell, $x \in \mathcal{V}$ eine Variable und $a \in A$ ein Element der Grundmenge. Die Substitution $\mathcal{M}[x/a]$ ist das Modell $\mathcal{M}_x^a = (\mathcal{A}, \omega_x^a)$ mit der Belegung

$$\omega_x^a(y) = \begin{cases} a & \text{wenn } y = x \\ \omega(y) & \text{wenn } y \neq x. \end{cases}$$

(7.10) Definition. Sei $\mathcal{M} = (\mathcal{A}, \omega)$ ein Modell für $\mathcal{F}_{\mathcal{V}, \Sigma}$. Wir definieren die Erfüllungsrelation für Formeln induktiv wie folgt:

(1) Für Primformeln gilt

$$\mathcal{M} \models s = t \text{ wenn } s^{\mathcal{M}} = t^{\mathcal{M}}$$

$\mathcal{M} \models r(t_1, t_2, \dots, t_n)$ wenn die Relation $r^{\mathcal{M}}(t_1^{\mathcal{M}}, t_2^{\mathcal{M}}, \dots, t_n^{\mathcal{M}})$ erfüllt ist.

(2) Für die Formeln P und Q gilt wie zu erwarten

$$\mathcal{M} \models P \wedge Q \text{ wenn gilt } \mathcal{M} \models P \text{ und } \mathcal{M} \models Q.$$

$$\mathcal{M} \models \neg P \text{ wenn nicht gilt } \mathcal{M} \models P.$$

$$\mathcal{M} \models \forall x P \text{ wenn } \mathcal{M}_x^a \models P \text{ für alle } a \in A.$$

(7.11) Definition. Zwei prädikatenlogische Formeln P und Q heißen semantisch äquivalent wenn für alle Modelle \mathcal{M} gilt $\mathcal{M} \models P$ genau dann, wenn $\mathcal{M} \models Q$.

Zum Beispiel führt die Umbenennung gebundener Variablen auf semantisch äquivalente Formeln:

$$x \wedge \forall x f(x) \iff x \wedge \forall y f(y)$$

(7.12) Lemma. Seien P und Q Formeln, dann gelten die folgenden Umformungsregeln:

$$\begin{array}{ll} \neg \forall x P \iff \exists x \neg P & \neg \exists x P \iff \forall x \neg P \\ (\forall x P) \wedge (\forall x Q) \iff \forall x (P \wedge Q) & (\exists x P) \vee (\exists x Q) \iff \exists x (P \vee Q) \\ \forall x \forall y P \iff \forall y \forall x P & \exists x \exists y P \iff \exists y \exists x P \end{array}$$

Wenn $x \notin \text{FV}(Q)$, dann gilt außerdem:

$$\begin{array}{ll} (\forall x P) \wedge Q \iff \forall x (P \wedge Q) & (\exists x P) \wedge Q \iff \exists x (P \wedge Q) \\ (\forall x P) \vee Q \iff \forall x (P \vee Q) & (\exists x P) \vee Q \iff \exists x (P \vee Q) \end{array}$$

(7.13) Definition. Eine prädikatenlogische Formel P ist in pränexer Normalform, wenn $P = Q_1 x_1 Q_2 x_2 \cdots Q_k x_k R$, wobei $Q_i \in \{\forall, \exists\}$, $x_i \in \mathcal{V}$ und R eine quantorenfreie Formel ist.

(7.14) **Satz.** Jede Formel besitzt eine äquivalente pränexe Normalform.

(7.15) **Beispiel.** Wir bestimmen eine zur Formel

$$P = (\neg \exists x S(x, y) \vee \forall x R(f(x))) \wedge (\forall y \neg Q(x, g(y)))$$

äquivalente Formel in pränexer Normalform.

$$\begin{aligned} P &\iff (\forall x \neg S(x, y) \vee \forall x R(f(x))) \wedge (\forall y \neg Q(x, g(y))) \\ &\iff (\forall w \neg S(w, y) \vee \forall v R(f(v))) \wedge (\forall z \neg Q(x, g(z))) \\ &\iff (\forall w (\neg S(w, y) \vee \forall v R(f(v)))) \wedge (\forall z \neg Q(x, g(z))) \\ &\iff (\forall w \forall v (\neg S(w, y) \vee R(f(v)))) \wedge (\forall z \neg Q(x, g(z))) \\ &\iff \forall w \forall v ((\neg S(w, y) \vee R(f(v))) \wedge (\forall z \neg Q(x, g(z)))) \\ &\iff \forall w \forall v \forall z ((\neg S(w, y) \vee R(f(v))) \wedge \neg Q(x, g(z))) \end{aligned}$$

KAPITEL C

Erzeugende Funktionen

Inhaltsangabe

- | | |
|----------------------------|------|
| 1. Kombinatorik | C.3 |
| 2. Abzählende Potenzreihen | C.7 |
| 3. Rekursionen | C.16 |
-

1. Kombinatorik

Die *Kombinatorik* ist ein Teilgebiet der Mathematik, das sich mit dem Abzählen verschiedener Strukturen (z.B. Anordnungen, Auswahlen, Unterteilungen, Variationen, etc.) beschäftigt. Wir werden zunächst ein paar grundlegende Abzählungen kennenlernen, bevor wir uns dem Abzählen mittels Potenzreihen zuwenden.

(1.1) Definition. Unter einer *Permutation* von n Elementen versteht man eine Anordnung der Elemente in einer Reihe. Die Anzahl der Permutationen von n Elementen wird mit $n!$ bezeichnet (*Fakultät* oder *Faktorielle*).

(1.2) Bemerkung. Es gilt

$$0! = 1, \\ \forall n \in \mathbb{N}: \quad n! = n \cdot (n-1)! = n \cdot (n-1) \cdot (n-2) \cdots 3 \cdot 2 \cdot 1.$$

(1.3) Beispiel. Bei den klassischen Kartenspielen mit 32 oder 52 Karten gibt es

$$32! \approx 2.6 \cdot 10^{35} \quad \text{bzw.} \quad 52! \approx 8 \cdot 10^{67}$$

Möglichkeiten, wie der Kartenstoß zu Beginn eines Spiels gemischt sein kann.

(1.4) Definition. Wählt man aus n unterscheidbaren Objekten k Objekte aus, wobei die Reihenfolge der ausgewählten Objekte egal ist, dann spricht man von einer *Kombination*. Falls hingegen die Reihenfolge der ausgewählten Objekte berücksichtigt wird, handelt es sich um *Variationen*.

Sowohl Kombinationen als auch Variationen gibt es *ohne Wiederholung* (jedes der n Objekte darf nur einmal ausgewählt werden) und *mit Wiederholung* (jedes Objekt darf mehrfach ausgewählt werden).

(1.5) Bemerkung. Bei Variationen ohne Wiederholung von k Objekten aus n Objekten gibt es n Möglichkeiten, das erste Objekt auszuwählen. Für das zweite Objekt gibt es dann noch $n - 1$ Möglichkeiten, für das dritte Objekt sind es $n - 2$ Möglichkeiten usw. Für das k -te Objekt bleiben letztlich noch $n - k + 1$ Möglichkeiten. Insgesamt ist also die Anzahl der Variationen ohne Wiederholung

$$n \cdot (n-1) \cdots (n-k+1) = \frac{n!}{(n-k)!}.$$

(1.6) Beispiel. In einer Lotterie mit 100 Teilnehmern werden fünf unterschiedliche Preise vergeben. Hierfür gibt es

$$100 \cdot 99 \cdot 98 \cdot 97 \cdot 96 = 9034502400$$

verschiedene Möglichkeiten.

(1.7) Bemerkung. Für Variationen mit Wiederholung von k Objekten aus n Objekten gibt es

$$\underbrace{n \cdot n \cdot n \cdots n}_{k \text{ mal}} = n^k$$

Möglichkeiten, da für jede der k Positionen jedes der n Objekte in Frage kommt.

(1.8) Beispiel. Eine Münze wird 20 mal geworfen. Jeder einzelne Münzwurf hat zwei mögliche Ergebnisse, wir wählen also aus $n = 2$ Objekten (Kopf oder Zahl) $k = 20$ Objekte (die Ergebnisse der einzelnen Münzwürfe) mit Zurücklegen aus. Es gibt daher $2^{20} = 1048576$ verschiedene Möglichkeiten für das Ergebnis der 20 Münzwürfe.

(1.9) Definition. Die Anzahl der Kombinationen ohne Wiederholung (auch *Auswahl*) von k Objekten aus n Objekten heißt *Binomialkoeffizient* und wird mit

$$\binom{n}{k}$$

(ausgesprochen „ n über k “) bezeichnet.

(1.10) Bemerkung. Es gilt

$$\binom{n}{k} = \begin{cases} 0 & \text{für } k > n, \\ \frac{n!}{k!(n-k)!} & \text{sonst.} \end{cases}$$

Insbesondere ist $\binom{n}{k} = \binom{n}{n-k}$ und $\binom{n}{0} = \binom{n}{n} = 1$ für alle $n, k \in \mathbb{N}_0$.

(1.11) Beispiel. Für das Ergebnis einer Lottoziehung „6 aus 45“ gibt es

$$\binom{45}{6} = 8145060$$

Möglichkeiten.

(1.12) Bemerkung. Die Anzahl der Kombinationen mit Wiederholung von k Objekten aus n Objekten ist

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

Dies kann man wie folgt einsehen: Die n Objekte nennen wir O_1, \dots, O_n . Bei der Kombination mit Wiederholung haben wir jedes O_j genau a_j mal gewählt (mit $a_j \geq 0$ für alle $j = 1, \dots, n$ und $a_1 + \dots + a_n = k$). Nun sind

$$\begin{aligned} b_1 &= a_1 + 1, \\ b_2 &= a_1 + a_2 + 2, \\ &\vdots \\ b_{n-1} &= a_1 + \dots + a_{n-1} + n - 1 \end{aligned}$$

$n-1$ unterschiedliche Zahlen aus $\{1, \dots, n+k-1\}$, wofür es $\binom{n+k-1}{n-1}$ Möglichkeiten gibt. Jede mögliche Auswahl von b_1, \dots, b_{n-1} entspricht hierbei genau einer Kombination mit Wiederholung von k Objekten aus O_1, \dots, O_n . Also ist $\binom{n+k-1}{n-1}$ die gesuchte Anzahl an Kombinationen mit Wiederholung.

(1.13) Beispiel. Werden fünf Würfel gleichzeitig geworfen, dann kommt es bei dem Ergebnis nicht darauf an, welcher Würfel welche Zahl anzeigt, sondern es ist lediglich von Interesse, wie oft jede Zahl angezeigt wird. Es handelt sich also um eine Kombination mit Wiederholung von $k = 5$ Objekten (die einzelnen Würfel) aus $n = 6$ Objekten (die Zahlen, die ein Würfel anzeigen kann). Deren Anzahl ist

$$\binom{10}{5} = 252.$$

(1.14) Binomischer Lehrsatz. Für $n \in \mathbb{N}$ gilt

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

Die Binomialkoeffizienten, die im Binomischen Lehrsatz vorkommen, lassen sich mit Hilfe des *Pascalschen Dreiecks* graphisch anordnen. Hierbei enthält

die $(n + 1)$ -te Zeile des Dreiecks jene Binomialkoeffizienten, die bei $(a + b)^n$ auftreten.

$$\begin{array}{ccccc}
 \binom{0}{0} & & & & \\
 \binom{1}{0} & \binom{1}{1} & & & 1 \\
 \binom{2}{0} & \binom{2}{1} & \binom{2}{2} & & 1 \quad 1 \\
 \binom{3}{0} & \binom{3}{1} & \binom{3}{2} & \binom{3}{3} & 1 \quad 2 \quad 1 \\
 \binom{4}{0} & \binom{4}{1} & \binom{4}{2} & \binom{4}{3} & \binom{4}{4} \\
 & & \vdots & & \vdots
 \end{array}$$

(1.15) Beispiel. Möchte man $(a + b)^4$ bestimmen, so betrachtet man die fünfte Zeile des Pascalschen Dreiecks.

$$(a + b)^4 = 1a^4 + 4a^3b + 6a^2b^2 + 4ab^3 + 1b^4.$$

(1.16) Bemerkung.

- Jeder Eintrag, der nicht am Rand seiner Zeile steht, lässt sich als Summe der beiden Einträge, die in der Zeile darüber als unmittelbare Nachbarn stehen, berechnen:

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

Dies kann man auch leicht anhand der Darstellung des Binomialkoeffizienten aus Bemerkung 1.10 nachrechnen.

- Bildet man die Summe über alle Einträge der $(n + 1)$ -ten Zeile, so erhält man mit Hilfe des binomischen Lehrsatzes (mit $a = b = 1$)

$$\binom{n}{0} + \binom{n}{1} + \cdots + \binom{n}{n} = \sum_{k=0}^n \binom{n}{k} = \sum_{k=0}^n \binom{n}{k} 1^{n-k} 1^k = (1 + 1)^n = 2^n.$$

2. Abzählende Potenzreihen

(2.1) Definition. Eine *formale Potenzreihe* ist ein Ausdruck der Form

$$\sum_{k=0}^{\infty} a_k x^k$$

mit Koeffizienten $a_k \in \mathbb{R}$, wobei x als formaler Parameter zu betrachten ist und keinerlei Konvergenzfragen gestellt werden. Ein *Polynom* ist eine formale Potenzreihe mit nur endlich vielen nichtverschwindenden Koeffizienten, d.h., von der Form

$$\sum_{k=0}^n a_k x^k.$$

(2.2) Definition. Sei $(a_k)_{k \in \mathbb{N}}$ eine Folge von Zahlen, dann heißt $A(x) = \sum a_k x^k$ die *erzeugende Potenzreihe* oder etwas ungenau *erzeugende Funktion* dieser Folge. Wenn a_n die Anzahl gewisser kombinatorischer Objekte darstellt, dann nennt man $A(x)$ auch die *abzählende Potenzreihe* dieser Objekte.

(2.3) Beispiel. Die Folgen $a_n = 1$ (konstante Folge) und $b_n = 2^n$ (geometrische Folge) haben die erzeugenden Potenzreihen

$$A(x) = \sum_{n=0}^{\infty} x^n = \frac{1}{1-x} \quad B(x) = \sum_{n=0}^{\infty} 2^n x^n = \frac{1}{1-2x}.$$

(2.4) Beispiel. Das Polynom

$$(1+x)^n = \sum_{k=0}^n \binom{n}{k} x^k$$

ist die erzeugende Potenzreihe der Binomialkoeffizienten.

(2.5) Verallgemeinerte Binomialkoeffizienten. Die Funktion $(\alpha)_k = \alpha \cdot (\alpha - 1) \cdot (\alpha - 2) \cdots (\alpha - k + 1)$ heißt *Pochhammersymbol*. Damit kann man den Binomialkoeffizienten für beliebige Zahlen $\alpha \in \mathbb{R}$ und $k \in \mathbb{N}$ definieren als

$$\binom{\alpha}{k} = \begin{cases} 1 & k = 0 \\ \frac{(\alpha)_k}{k!} = \frac{\alpha \cdot (\alpha - 1) \cdot (\alpha - 2) \cdots (\alpha - k + 1)}{k!} & k \geq 1. \end{cases}$$

Dann gilt die Reihenentwicklung

$$(2.6) \quad (1+x)^\alpha = \sum_{k=0}^{\infty} \binom{\alpha}{k} x^k.$$

(2.7) Beispiel.

$$\begin{aligned} \frac{1}{\sqrt{1-x}} &= (1-x)^{-1/2} = \sum_{k=0}^{\infty} \binom{-\frac{1}{2}}{k} (-1)^k x^k \\ &= \sum_{k=0}^{\infty} \frac{\left(-\frac{1}{2}\right) \left(-\frac{3}{2}\right) \cdots \left(-\frac{1}{2}-k+1\right)}{k!} (-1)^k x^k \\ &= \sum_{k=0}^{\infty} \frac{1 \cdot 3 \cdot 5 \cdots (2k-1)}{k!} \left(\frac{x}{2}\right)^k \\ &= \sum_{k=0}^{\infty} \frac{1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdots (2k-1) \cdot 2k}{k! \cdot 2 \cdot 4 \cdots 2k} \left(\frac{x}{2}\right)^k \\ &= \sum_{k=0}^{\infty} \binom{2k}{k} \left(\frac{x}{4}\right)^k \end{aligned}$$

(2.8) Rechenregeln. Formale Potenzreihen können addiert und multipliziert werden wie Polynome:

$$\begin{aligned} \sum_{k=0}^{\infty} a_k x^k + \sum_{k=0}^{\infty} b_k x^k &= \sum_{k=0}^{\infty} (a_k + b_k) x^k, \\ \lambda \cdot \sum_{k=0}^{\infty} a_k x^k &= \sum_{k=0}^{\infty} (\lambda \cdot a_k) x^k, \\ (2.9) \quad \sum_{k=0}^{\infty} a_k x^k \cdot \sum_{l=0}^{\infty} b_l x^l &= \sum_{n=0}^{\infty} c_n x^n, \end{aligned}$$

wobei $c_0 = a_0 b_0 \quad c_1 = a_0 b_1 + a_1 b_0 \quad c_2 = a_0 b_2 + a_1 b_1 + a_2 b_0 \quad \dots$

und allgemein

$$c_n = \sum_{k=0}^n a_k b_{n-k}.$$

Es gehen also in die Berechnung der Koeffizienten c_n jeweils nur endlich viele Koeffizienten a_k und b_k ein.

(2.10) Differentiation. Formale Potenzreihen können wie Polynome differenziert werden.

$$\frac{d}{dx} \sum_{n=0}^{\infty} a_n x^n = \sum_{n=1}^{\infty} n a_n x^{n-1}$$

(2.11) Beispiel.

$$\sum_{n=1}^{\infty} n x^n = x \cdot \sum_{n=1}^{\infty} n x^{n-1} = x \cdot \frac{d}{dx} \sum_{n=1}^{\infty} x^n = x \frac{d}{dx} \left(\frac{1}{1-x} - 1 \right) = \frac{x}{(1-x)^2}$$

(2.12) Koeffizienten rationaler Funktionen. Es sei eine rationale Funktion gegeben, d.h. eine Funktion der Form

$$f(x) = \frac{p(x)}{q(x)}$$

wobei $p(x)$ und $q(x)$ Polynome sind mit $q(0) \neq 0$ und der Grad von p kleiner ist als der Grad von q . Wir wollen die Taylorkoeffizienten der Reihenentwicklung

$$f(x) = \sum_{k=0}^{\infty} a_k x^k$$

bestimmen. Angenommen $q(x)$ hat die Faktorisierung

$$q(x) = \alpha_0 (1 - \alpha_1 x) (1 - \alpha_2 x) \cdots (1 - \alpha_n x),$$

wobei alle Nullstellen verschieden sind, dann gibt es eine Partialbruchzerlegung

$$(2.13) \quad f(x) = \frac{c_1}{1 - \alpha_1 x} + \frac{c_2}{1 - \alpha_2 x} + \cdots + \frac{c_n}{1 - \alpha_n x}$$

mit gewissen Konstanten c_i , die sich bestimmen lassen durch Multiplizieren der Gleichung (2.13) mit $q(x)$:

$$p(x) = c_1 \frac{q(x)}{1 - \alpha_1 x} + c_2 \frac{q(x)}{1 - \alpha_2 x} + \cdots + c_n \frac{q(x)}{1 - \alpha_n x}.$$

und Einsetzen von $x = 1/\alpha_i$ in diese Gleichung. Die geometrischen Reihenentwicklung ergibt dann

$$\frac{1}{1 - \alpha x} = \sum_{k=0}^{\infty} \alpha^k x^k$$

und wir erhalten die explizite Formel

$$a_k = c_1 \alpha_1^k + c_2 \alpha_2^k + \cdots + c_n \alpha_n^k.$$

(2.14) Beispiel. Berechnung der Koeffizienten a_n der rationalen formalen Potenzreihe

$$f(x) = \sum_{n=0}^{\infty} a_n x^n = \frac{1}{1+x-4x^2-4x^3}.$$

Beweis. Nach Faktorisierung finden wir

$$f(x) = \frac{1}{(1-2x)(1+x)(1+2x)}$$

und daher

$$f(x) = \frac{c_1}{1-2x} + \frac{c_2}{1+x} + \frac{c_3}{1+2x}.$$

Multiplizieren mit dem kgV der Nenner ergibt

$$1 = c_1(1+x)(1+2x) + c_2(1-2x)(1+2x) + c_3(1-2x)(1+x)$$

und durch Einsetzen von $x = 1/2, -1, -1/2$ erhalten wir

$$c_1 = \frac{1}{3} \quad c_2 = -\frac{1}{3} \quad c_3 = 1.$$

Wir finden als Ergebnis

$$a_n = \frac{1}{3} 2^n - \frac{1}{3} (-1)^n + (-2)^n.$$

(2.15) Bemerkung. Wenn es mehrfache Nullstellen gibt, etwa die Potenz $(1-\alpha x)^k$ im Nenner, dann lautet der entsprechende Ansatz

$$\frac{c_1}{1-\alpha x} + \frac{c_2}{(1-\alpha x)^2} + \cdots + \frac{c_k}{(1-\alpha x)^k}.$$

(2.16) Anwendungen. Mithilfe von erzeugenden Potenzreihen kann man unendlich viele Abzählprobleme „auf einen Streich“ lösen, und zwar von folgender Art. Es sei eine Klasse von Objekten \mathcal{A} gegeben zusammen mit einer Funktion $\mathcal{A} \rightarrow \mathbb{N}$ die jedem Objekt α aus \mathcal{A} eine natürliche Zahl $|\alpha|$, seine „Größe“ zuordnet. Gesucht ist für jedes n die Anzahl a_n der Elemente $\alpha \in \mathcal{A}$ für die gilt $|\alpha| = n$. Die erzeugende Funktion kann man dann schreiben als

$$A(x) = \sum_{n=0}^{\infty} a_n x^n = \sum_{\alpha \in \mathcal{A}} x^{|\alpha|}.$$

Die algebraischen Manipulationen von Potenzreihen haben folgende Bedeutung:

(2.17) Bemerkung. Der *disjunkten Vereinigung* von Mengen entspricht die Addition von Potenzreihen. Wenn $C = A \dot{\cup} B$, dann ist

$$C(z) = A(z) + B(z)$$

(2.18) Bemerkung. Dem *kartesischen Produkt* von Mengen entspricht die Multiplikation von Potenzreihen. Wenn $C = A \times B$ mit Größenfunktion $|(\alpha, \beta)| = |\alpha| + |\beta|$, dann ist

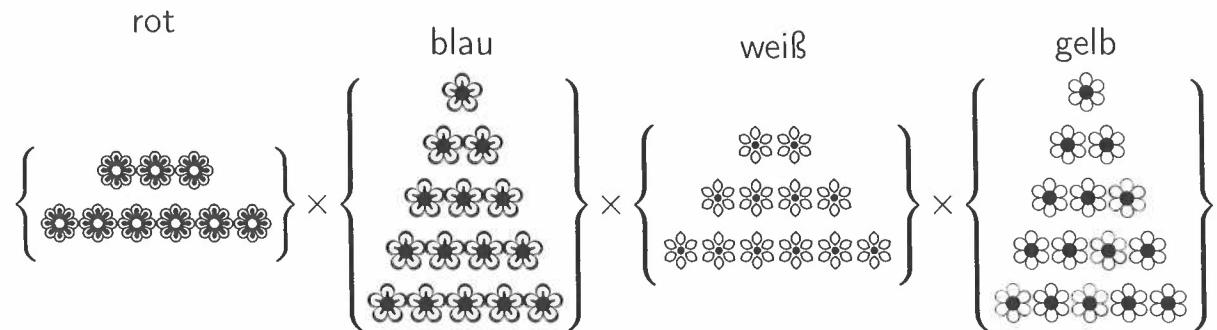
$$C(z) = A(z) \cdot B(z)$$

Beweis. In der Tat ist

$$C(z) = \sum_{\gamma \in C} z^{|\gamma|} = \sum_{(\alpha, \beta) \in A \times B} z^{|\alpha|+|\beta|} = \sum_{\alpha \in A} z^{|\alpha|} \cdot \sum_{\beta \in B} z^{|\beta|}$$

(2.19) Beispiel. Ein Gärtner bietet 6 rote, 5 blaue, 7 weiße und 5 gelbe Blumen zum Verkauf an. Auf wieviele Arten kann man daraus einen zehnblütigen Blumenstrauß bilden, der mindestens eine Blume von jeder Farbe, eine durch drei teilbare Zahl roter und nur eine gerade Anzahl weißer Blumen enthält?

Lösung. Wir schreiben alle Möglichkeiten formal an: und zählen die Elemente der „Länge“ 10 in der Menge



ab. Da am Ende nur die Anzahl der Blumen interessant ist, können wir jede Blume durch ein x ersetzen. Gesucht ist also der Koeffizient von x^{10} des Polynoms

$$\begin{aligned} & (x^3 + x^6)(x + x^2 + x^3 + x^4 + x^5)(x^2 + x^4 + x^6)(x + x^2 + x^3 + x^4 + x^5) \\ &= x^7 + 2x^8 + 4x^9 + 7x^{10} + 11x^{11} + 14x^{12} + 17x^{13} + 19x^{14} \\ & \quad + 19x^{15} + 17x^{16} + 14x^{17} + 11x^{18} + 7x^{19} + 4x^{20} + 2x^{21} + x^{22}. \end{aligned}$$

(2.20) Beispiel.

$$\binom{m+n}{k} = \sum_{i=0}^k \binom{m}{i} \binom{n}{k-i}$$

Lösung. Dies folgt aus (2.9) und Beispiel 2.4.

(2.21) Bemerkung. Sei $\mathcal{C} = S(\mathcal{A})$ (manchmal auch als \mathcal{A}^* bezeichnet) die Menge aller *endlichen Folgen* von Elementen aus \mathcal{A} , d.h., wenn wir „+“ anstatt \cup schreiben,

$$\mathcal{C} = \emptyset + \mathcal{A} + \mathcal{A} \times \mathcal{A} + \mathcal{A} \times \mathcal{A} \times \mathcal{A} + \dots$$

Dann ist $C(z)$ eine geometrische Reihe in $A(z)$, nämlich

$$C(z) = 1 + A(z) + A(z)^2 + \dots = \frac{1}{1 - A(z)}.$$

(2.22) Beispiel. Auf einer CD können nicht beliebige Bitfolgen gespeichert werden, sondern nur solche, bei denen niemals die Folge „11“ vorkommt, z.B. ist 000101001 erlaubt, aber 000110101 verboten.

Wieviele zulässige Bitfolgen der Länge n gibt es?

Lösung. Zunächst bestimmen wir die Anzahl aller zulässigen Bitfolgen, die mit 0 beginnen. Das sind alle möglichen Wörter, die aus den Bausteinen 0 und 01 gebildet werden können (diese Darstellung ist eindeutig), d.h., aus dem Alphabet $\mathcal{A} = \{0, 01\}$ bilden wir die Menge aller Folgen

$$S(\mathcal{A}) = \{0, 00, 01, 000, 001, 010, 0000, 0001, 0010, 0100, 0101, \dots\}.$$

Die Elemente des Alphabets haben verschiedene Längen und die erzeugende Funktion ist $A(z) = z + z^2$, daher ist die erzeugende Funktion von $S(\mathcal{A})$

$$\tilde{C}(z) = \sum_{n=1}^{\infty} A(z)^n = \frac{A(z)}{1 - A(z)} = \frac{z + z^2}{1 - z - z^2} = \frac{1}{1 - z - z^2} - 1,$$

vgl. Beispiel 3.3. Die Menge aller zulässigen Bitfolgen erhalten wir, indem wir die erste 0 streichen. Dadurch wird die Länge aller Bitfolgen um 1 reduziert, d.h. wir müssen die erzeugende Funktion durch z dividieren und erhalten

$$C(z) = \frac{\tilde{C}(z)}{z} = \frac{1 + z}{1 - z - z^2} = 1 + 2z + 3z^2 + 5z^3 + 8z^4 + 13z^5 + 21z^6 + \dots$$

Die gesuchten Zahlen sind also $2, 3, 5, 8, 13, 21, 34, 55, 89, \dots$. Wir werden diese Zahlen in Beispiel 3.3 wiedertreffen, wo eine zweite Lösung gezeigt wird.

(2.23) Beispiel. (Euler¹ 1748, Pólya² 1956)

Auf wieviele Arten kann man 1€ bezahlen?

Wir beginnen mit einer Auflistung aller möglichen Münzkombinationen. Zunächst die formale Summe aller möglichen Kombinationen von 1-Centmünzen (dies entspricht Bemerkung 2.21):

$$\mathcal{K}(\textcircled{1}) = \emptyset + (\textcircled{1}) + (\textcircled{1})(\textcircled{1}) + (\textcircled{1})(\textcircled{1})(\textcircled{1}) + \dots$$

Hierbei steht \emptyset für „keine“ Münze; dasselbe mit den 2-Centmünzen ergibt

$$\mathcal{K}(\textcircled{2}) = \emptyset + (\textcircled{2}) + (\textcircled{2})(\textcircled{2}) + (\textcircled{2})(\textcircled{2})(\textcircled{2}) + \dots$$

Wenn wir nun das formale „Produkt“ bilden, wobei $\emptyset \cdot (\textcircled{k}) = (\textcircled{k})$, erhalten wir die formale Summe aller Kombinationen von 1- und 2-Centmünzen

$$\mathcal{K}(\textcircled{1}) \cdot \mathcal{K}(\textcircled{2}) = \emptyset + (\textcircled{1}) + (\textcircled{2}) + (\textcircled{1})(\textcircled{1}) + (\textcircled{1})(\textcircled{2}) + (\textcircled{2})(\textcircled{2}) + (\textcircled{1})(\textcircled{1})(\textcircled{1}) + \dots$$

Wenn wir also alle Münzen mit Wert $\leq 1\text{€}$ kombinieren wollen, müssen wir das Produkt

$$\mathcal{E} = \mathcal{K}(\textcircled{1}) \mathcal{K}(\textcircled{2}) \mathcal{K}(\textcircled{5}) \mathcal{K}(\textcircled{10}) \mathcal{K}(\textcircled{20}) \mathcal{K}(\textcircled{50}) \mathcal{K}(\textcircled{1\text{€}})$$

betrachten. Wenn wir nun (\textcircled{k}) mit $W(\textcircled{k}) = x^k$ bewerten, erhalten wir Terme wie z.B.

$$W(\textcircled{1})(\textcircled{1})(\textcircled{5}) = x^1 x^1 x^5 = x^7$$

und daher finden wir mit

$$f(x) = 1 + x + x^2 + \dots = \frac{1}{1-x}$$

dass $W(\mathcal{K}(\textcircled{k})) = f(x^k) = \frac{1}{1-x^k}$ und insgesamt

$$W(\mathcal{E}) = f(x) f(x^2) f(x^5) f(x^{10}) f(x^{20}) f(x^{50}) f(x^{100}) = \sum_{n=0}^{\infty} a_n x^n$$

¹Leonhard Euler (1707–1783)

²Gyorgy Pólya (1887–1985)

wobei a_n die Anzahl der Münzkombinationen mit Gesamtwert n ist. Die Lösung für unser Problem ist daher (unter Zuhilfenahme eines Computeralgebrapakets) $a_{100} = 4562$.

(2.24) Bemerkung. Teilsummenbildung entspricht der Multiplikation mit $\frac{1}{1-x}$. Es sei $A(x) = \sum a_n x^n$ und $b_n = \sum_{k=0}^n a_k$, dann ist wegen (2.9)

$$B(x) = \sum b_n x^n = \frac{1}{1-x} A(x).$$

(2.25) Beispiel. (vgl. Beispiel 2.11)

$$\sum_{n=0}^{\infty} (n+1)x^n = \sum_{n=0}^{\infty} \left(\sum_{k=0}^n 1 \right) x^n = \frac{1}{1-x} \sum_{n=0}^{\infty} x^n = \frac{1}{(1-x)^2}$$

(2.26) Beispiel. Berechne

$$s_n = \sum_{k=1}^n k(k-1)$$

durch Bestimmung der abzählenden Potenzreihe $\sum_{n=0}^{\infty} s_n x^n$.

Lösung. Ausgehend von der Formel

$$\frac{d^2}{dx^2} \sum_{n=0}^{\infty} x^n = \sum_{n=2}^{\infty} n(n-1) x^{n-2}$$

sehen wir

$$x^2 \frac{d^2}{dx^2} \frac{1}{1-x} = \sum_{n=2}^{\infty} n(n-1) x^n = \sum_{n=0}^{\infty} n(n-1) x^n.$$

Wir verwenden Bemerkung 2.24 und erhalten

$$\begin{aligned}
 \sum_{n=0}^{\infty} \left(\sum_{k=1}^n k(k-1) \right) x^n &= \frac{x^2}{1-x} \frac{d^2}{dx^2} \frac{1}{1-x} \\
 &= \frac{2x^2}{(1-x)^4} \\
 &= \frac{x^2}{3} \frac{d^3}{dx^3} \frac{1}{1-x} \\
 &= \frac{x^2}{3} \sum_{n=3}^{\infty} n(n-1)(n-2) x^{n-3} \\
 &= \frac{1}{3} \sum_{n=2}^{\infty} (n+1)n(n-1) x^n
 \end{aligned}$$

und daher

$$\sum_{k=1}^n k(k-1) = \frac{(n+1)n(n-1)}{3}.$$

(2.27) Formale Potenzreihen mit mehreren Veränderlichen. Gegeben eine Doppelfolge $(a_{n,k})_{k,n \in \mathbb{N}}$, kann man die formale Potenzreihe

$$f(x, y) = \sum_{n=0}^{\infty} \sum_{k=0}^{\infty} a_{nk} x^n y^k$$

betrachten.

(2.28) Beispiel. Gesucht ist die erzeugende Potenzreihe der Doppelfolge $\binom{n}{k}$.

Lösung.

$$\begin{aligned}
 \sum_{n=0}^{\infty} \sum_{k=0}^n \binom{n}{k} x^n y^k &= \sum_{n=0}^{\infty} x^n (1+y)^n \\
 &= \frac{1}{1-x(1+y)}
 \end{aligned}$$

3. Rekursionen

Erzeugende Funktionen eignen sich hervorragend zur Lösung von Rekursionen, d.h. zur Auffindung von expliziten Formeln für Zahlenfolgen die durch Angabe eines Anfangswerts a_0 und einer Rekursionsvorschrift

$$a_n = f_n(a_{n-1}, a_{n-2}, \dots, a_0) \quad n \geq 1$$

gegeben sind. Das Rezept besteht darin, die Rekursion als

$$A(x) = \sum_{n=0}^{\infty} a_n x^n = a_0 + \sum_{n=1}^{\infty} f_n(a_{n-1}, a_{n-2}, \dots, a_0) x^n$$

anzuschreiben und in eine algebraische Gleichung für $A(x)$ umzuwandeln.

(3.1) Beispiel. Gesucht ist ein Ausdruck für die Zahlen a_n , die durch die Rekursion

$$\begin{aligned} a_0 &= 1 & a_1 &= -1 \\ a_n + 3a_{n-1} + 2a_{n-2} &= 2^n & n \geq 2 \end{aligned}$$

definiert sind.

Lösung. Wir leiten zuerst eine Gleichung für die erzeugende Funktion $A(x) = \sum_{n=0}^{\infty} a_n x^n = 1 - x + \sum_{n=2}^{\infty} a_n x^n$ her. Dazu multiplizieren wir beide Seiten der Rekursionsvorschrift mit x^n und summieren:

$$\begin{aligned} \sum_{n=2}^{\infty} a_n x^n + 3 \sum_{n=2}^{\infty} a_{n-1} x^n + 2 \sum_{n=2}^{\infty} a_{n-2} x^n &= \sum_{n=2}^{\infty} 2^n x^n \\ \sum_{n=2}^{\infty} a_n x^n + 3 \sum_{n=1}^{\infty} a_n x^{n+1} + 2 \sum_{n=0}^{\infty} a_n x^{n+2} &= \sum_{n=2}^{\infty} (2x)^n \\ A(x) + x - 1 + 3x(A(x) - 1) + 2x^2 A(x) &= \frac{1}{1-2x} - 2x - 1 \\ A(x)(1 + 3x + 2x^2) &= \frac{1}{1-2x} \\ A(x) &= \frac{1}{(1-2x)(1+3x+2x^2)} = \frac{1}{1+x-4x^2-4x^3} \end{aligned}$$

Die Koeffizienten dieser Reihe wurden bereits in Beispiel 2.14 bestimmt.

(3.2) Satz. Sei $(a_n)_{n \geq 0}$ Lösung einer linearen Rekursionsgleichung, dann ist die erzeugende Potenzreihe $A(x)$ rational, d.h. es gibt Polynome $p(x)$ und $q(x)$, sodass

$$A(x) = \frac{p(x)}{q(x)}.$$

(3.3) Beispiel. Die Fibonaccifolge³ F_n ist rekursiv definiert durch die Anfangsbedingung

$$F_0 = 1 \quad F_1 = 1$$

und die Vorschrift

$$F_{n+1} = F_n + F_{n-1}.$$

Gesucht ist eine explizite Formel für F_n .

Lösung. Zunächst berechnen wir die erzeugende Potenzreihe:

$$\begin{aligned} F(x) &= \sum_{n=0}^{\infty} F_n x^n = 1 + x + \sum_{n=2}^{\infty} (F_{n-1} + F_{n-2}) x^n \\ &= 1 + (x + x^2) F(x) \end{aligned}$$

und folglich

$$F(x) = \frac{1}{1 - x - x^2}.$$

Durch Partialbruchzerlegung (2.12) finden wir

$$\frac{1}{1 - x - x^2} = \frac{1}{2\sqrt{5}} \left(\frac{\sqrt{5} - 1}{1 - \frac{1-\sqrt{5}}{2}x} + \frac{\sqrt{5} + 1}{1 - \frac{1+\sqrt{5}}{2}x} \right)$$

und daher gilt

$$F_n = \frac{1}{\sqrt{5}} \left(\left(\frac{1 + \sqrt{5}}{2} \right)^{n+1} - \left(\frac{1 - \sqrt{5}}{2} \right)^{n+1} \right).$$

Lösung. Zweite Lösung von Beispiel 2.22.

Sei

a_n die Anzahl der zulässigen Bitfolgen, die auf 0 enden.

³Leonardo von Pisa („Fibonacci“), (1170–1250)

b_n die Anzahl der zulässigen Bitfolgen, die auf 1 enden.

$c_n = a_n + b_n$ die Anzahl aller zulässigen Bitfolgen.

Es ist $a_1 = 1$ und $b_1 = 1$.

Sei eine zulässige Bitfolge der Länge $n + 1$ gegeben.

Wenn das letzte Zeichen 0 ist, kann das vorletzte Zeichen beliebig sein.

Wenn das letzte Zeichen 1 ist, dann muss das vorletzte Zeichen 0 sein.

Es ist daher $a_{n+1} = c_n$, $b_{n+1} = a_n$ und $c_{n+1} = a_{n+1} + b_{n+1} = c_n + a_n = c_n + c_{n-1}$. Das ist die in Beispiel 3.3 behandelte Fibonaccifolge, um einen Index verschoben.

(3.4) Beispiel. Das Catalansche Problem. (Catalan⁴ 1838) Gegeben sei ein Produkt von n Zahlen $x_1 \cdot x_2 \cdots x_n$ und gefragt ist nach der Anzahl der möglichen Klammerungen dieses Produktes, die jeweils einer Berechnung des Produktes durch fortgesetzte Multiplikation von je zwei Faktoren entsprechen. Zum Beispiel gibt es zwei Klammerungen von 3 Faktoren

$$((x_1 x_2) x_3) \quad (x_1 (x_2 x_3))$$

und es gibt 5 Klammerungen von 4 Faktoren.

$$\begin{aligned} & ((x_1 x_2)(x_3 x_4)) \quad (((x_1 x_2)x_3)x_4) \quad ((x_1(x_2 x_3))x_4) \\ & \quad (x_1((x_2 x_3)x_4)) \quad (x_1(x_2(x_3 x_4))) \end{aligned}$$

Lösung. Jeder dieser Klammerungen entspricht ein *planarer* (oder *ebener*) *binärer Wurzelbaum*, wie in Abb. 1 gezeigt. Die Bäume sind *binär*, d.h., jeder innere Knoten hat genau zwei Nachfolger ("Kinder"). Die Endknoten heißen *Blätter*.

Sei C_n die Anzahl der verschiedenen Bäume mit n Blättern. Da die Wurzel zwei Kinder hat, kann man jeden Baum zerlegen in die Wurzel und zwei Teilbäume, die in jeweils einem der Kinder verwurzelt sind. Diese beiden Teilbäume können wieder als binäre Wurzelbäume aufgefasst werden. Man kann also, wenn man alle kleineren Bäume abgezählt hat, die Anzahl der Bäume angeben als

$$C_n = \sum_{k=1}^{n-1} C_k C_{n-k}.$$

⁴Eugene Charles Catalan (1814–1894)

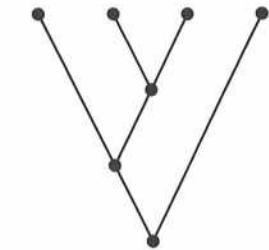
$$(x_1x_2)(x_3x_4)$$



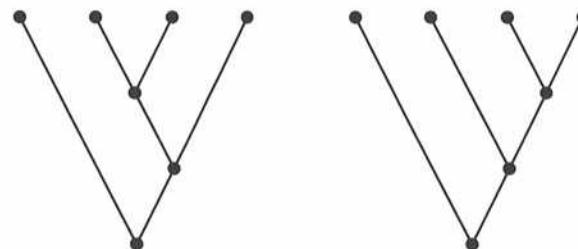
$$((x_1x_2)x_3)x_4$$



$$(x_1(x_2x_3))x_4$$



$$x_1((x_2x_3)x_4)$$



$$x_1(x_2(x_3x_4))$$

ABBILDUNG 1. Klammerungen und Bäume

Um diese Rekursion zu lösen, setzen wir die erzeugende Funktion an

$$C(x) = \sum_{n=1}^{\infty} C_n x^n$$

und unter Verwendung von (2.9) finden wir

$$\begin{aligned} C(x) &= x + \sum_{n=2}^{\infty} C_n x^n \\ &= x + \sum_{n=2}^{\infty} \left(\sum_{k=1}^{n-1} C_k C_{n-k} \right) x^n \\ &= x + C(x)^2. \end{aligned}$$

Die Funktion $C(x)$ erfüllt also die quadratische Gleichung

$$C(x)^2 - C(x) + x = 0$$

und daher gilt

$$C(x) = \frac{1}{2} (1 \pm \sqrt{1 - 4x})$$

Aufgrund der Bedingung $C_0 = 0$ wählen wir den „–“ Zweig der Lösung und es gilt gemäß (2.6)

$$C(x) = \frac{1}{2} (1 - \sqrt{1 - 4x}) = \frac{1}{2} \left(1 - \sum_{n=0}^{\infty} \binom{\frac{1}{2}}{n} (-4)^n x^n \right)$$

d.h.

$$C_n = (-1)^{n+1} \frac{4^n}{2} \binom{\frac{1}{2}}{n} = \frac{1}{n} \binom{2n-2}{n-1}.$$

Diese Zahlen heißen *Catalansche Zahlen* (eigentlich schon gefunden in China⁵ um 1730, L. Euler und C. Goldbach 1751 sowie allgemeiner N. Fuss⁶ 1795).

⁵Ming Antu (1692–1763)

⁶Nicolaus Fuss (1755–1826)

KAPITEL D

Graphen und Bäume

Inhaltsangabe

- | | |
|--|------|
| 1. Ungerichtete und gerichtete Graphen | D.3 |
| 2. Wege und Kreise in Graphen | D.4 |
| 3. Bäume | D.13 |
| 4. Matchings | D.19 |
| 5. Färbungen | D.22 |
| 6. Planare Graphen | D.24 |
-

1. Ungerichtete und gerichtete Graphen

(1.1) Definition. Ein (endlicher ungerichteter) Graph $G = (V, E)$ besteht aus einer endlichen Knotenmenge $V = V(G)$ und einer Menge $E = E(G)$ ungerichteter Kanten.

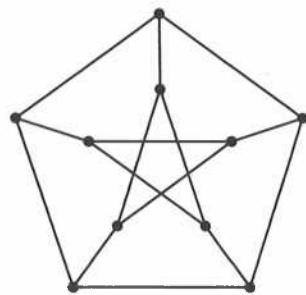
Jede Kante $e \in E$ ist von der Form $e = [x, y]$ mit $x, y \in V$. Ungerichtet bedeutet, dass $[x, y] = [y, x]$.

Im Fall $y = x$ spricht man von einer *Schleife*.

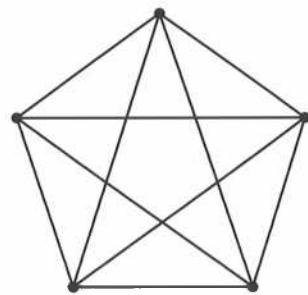
(1.2) Definition. Ein ungerichteter Graph G heißt *vollständig*, falls je zwei Knoten durch eine Kante verbunden sind, also $[x, y] \in E(G)$ für alle $x, y \in V(G)$ mit $x \neq y$.

Der vollständige Graph mit n Knoten wird als K^n bezeichnet.

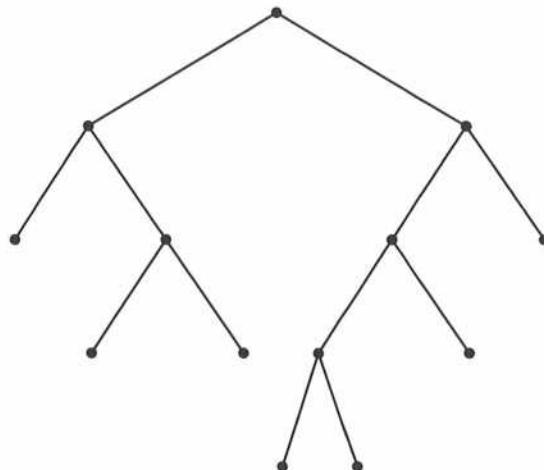
(1.3) Beispiele. Graphen stellt man oft zeichnerisch dar, indem man für jeden Knoten einen Punkt zeichnet und für jede Kante eine Linie zwischen den jeweiligen Knoten.



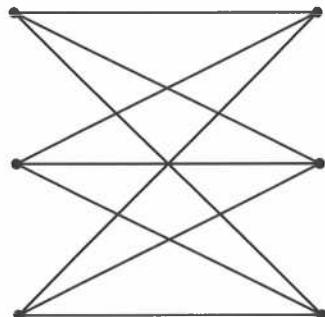
Der Petersen-Graph



Der vollständige Graph K^5

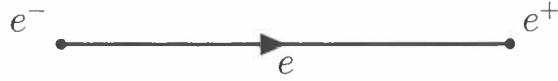


Ein binärer Baum



Der Graph $K_{3,3}$

(1.4) Definition. Analog zu Definition 1.1 definiert man gerichtete Graphen (*Digraphen*). In diesem Fall hat jede Kante $e \in E$ einen Anfangsknoten $e^- \in V$ und einen Endknoten $e^+ \in V$. Zeichnerisch deutet man dies durch einen Pfeil von e^- nach e^+ an.



Einem ungerichteten Graphen entspricht ein Digraph, indem jede ungerichtete Kante $[x, y]$ durch zwei entgegengesetzt orientierte Kanten (von x nach y und von y nach x) ersetzt wird.

Man kann Graphen und Digraphen auch mit *Mehrfachkanten* definieren. In dem Fall können Knoten durch mehr als eine Kante verbunden sein.

Graphen treten in vielen Anwendungen auf. Zum Beispiel lassen sich Netzwerke als Graphen auffassen. Auch bei Optimierungsproblemen wie der Erstellung von Zeitplänen treten Graphen als Modelle auf.

2. Wege und Kreise in Graphen

(2.1) Definition. (a) Ein Weg der Länge $k \geq 0$ in $G = (V, E)$ ist eine Sequenz $W = [x_0, e_1, x_1, \dots, e_k, x_k]$ von Knoten x_0, \dots, x_k und Kanten $e_i = [x_{i-1}, x_i] \in E$ ($i = 1, \dots, k$). Genauer heißt w dann auch Weg von x_0 nach x_k .

Ein Weg W besucht alle Knoten in $V(W) = \{x_0, \dots, x_k\}$ und durchläuft alle Kanten in $E(W) = \{e_1, \dots, e_k\}$.

Ein Weg der Länge 0 besteht also aus einem einzigen Knoten.

(b) Ein *Pfad* in G ist ein Weg ohne Wiederholungen.

(c) Ein *geschlossener Weg* ist ein Weg mit $x_k = x_0$.

(d) Ein geschlossener Weg positiver Länge, der außer $x_k = x_0$ keine Wiederholung (weder Knoten noch Kanten) hat, heißt *Kreis*.

Insbesondere sind Kreise der Länge 1 nur bei Schleifen möglich. Kreise der Länge 2 können nur bei Mehrfachkanten auftreten.

(e) Ein *Hamiltonscher Kreis* in G ist ein Kreis, der alle Knoten besucht.

Ein gerichteter Weg in einem gerichteten Graphen ist eine Sequenz $W = [x_0, e_1, x_1, \dots, e_k, x_k]$ mit $e_i \in E$, $x_{i-1} = e_i^-$ und $x_i = e_i^+$ ($i = 1, \dots, k$).



Analog betrachtet man in einem gerichteten Graphen zumeist die gerichteten Versionen von Pfaden und Kreisen.

(2.2) Definition. Der Grad eines Knotens $x \in V$ (ungerichteter Graph, Mehrfachkanten erlaubt) ist die Anzahl $\deg(x)$ der Kanten, die x enthalten.

In einem gerichteten Graphen unterteilt man die Kanten an einem Knoten x in ausgehende Kanten (x ist Anfangsknoten) und eingehende Kanten (x ist Endknoten) und unterscheidet dementsprechend zwischen Ausgangs- und Eingangsgrad von x .

- (2.3) Definition.**
- (a) Einen ungerichteten Graphen nennt man zusammenhängend, wenn je zwei Knoten x, y durch mindestens einen Weg verbunden sind.
 - (b) Ein gerichteter Graph heißt stark zusammenhängend, wenn für je zwei Knoten x, y mindestens ein gerichteter Weg von x nach y existiert.
 - (c) Ein gerichteter Graph heißt schwach zusammenhängend, wenn sein ungerichteter „Schatten“ (d.h., Weglassen der Orientierungen der Kanten) zusammenhängend ist.

Ein stark zusammenhängender gerichteter Graph ist auch schwach zusammenhängend.

(2.4) Satz. In einem ungerichteten Graphen $G = (V, E)$ ist durch

$$x R y \iff \exists \text{ Weg von } x \text{ nach } y$$

eine Äquivalenzrelation auf V gegeben.

Die Äquivalenzklassen heißen Zusammenhangskomponenten.

(Beweis als Übung!)

(2.5) Travelling Salesman Problem. Ein Handelsreisender muss Städte S_1, \dots, S_n besuchen und zum Schluss in den Ausgangsort zurückkehren. Er

verfügt über eine Liste der Direktflüge und entsprechenden Preise (nicht zwischen allen Städten gibt es Direktflüge; die jeweiligen Preise sind in beiden Richtungen gleich).

Gesucht: die preisgünstigste Rundreise.

Modell: Graph G mit $V = \{S_1, \dots, S_n\}$, Kanten: $[S_i, S_j] \in E \iff \exists$ Direktflug $S_i \leftrightarrow S_j$.

Die Kanten sind gewichtet mit den jeweiligen Flugpreisen.

Das Gewicht (die Kosten) eines Weges ist die Summe der Gewichte entlang der einzelnen Kanten.

Gesucht: eine Rundreise (geschlossener Weg durch alle Knoten) mit minimalen Kosten. \square

Es ist kein „rascher“ Algorithmus zur allgemeinen Lösung dieses Problems bekannt! Das gleiche gilt für das – ähnliche – Problem der Suche nach einem Hamiltonschen Kreis. Das Travelling Salesman Problem kann man natürlich auch für gerichtete, stark zusammenhängende Graphen stellen (Flüge nicht in beide Richtungen möglich, oder verschieden teuer).

- (2.6) Definition.** (a) Ein *Eulerscher Weg* in einem (ungerichteten) Graphen ist ein Weg, der jede Kante genau einmal durchläuft.
- (b) Ein *Eulerscher Kreis* in einem (ungerichteten) Graphen ist ein geschlossener Eulerscher Weg (Anfangsknoten = Endknoten).

Ebenso wie Hamiltonsche Kreise sind Eulersche Wege/Kreise nur in zusammenhängenden Graphen sinnvoll.

Sei G ein (ungerichteter) Graph ohne Schleifen (Mehrfachkanten erlaubt).

Wenn in G ein Eulerscher Weg $[x_0, e_1, x_1, \dots, e_m, x_m]$ (mit $m = |E|$) existiert, dann muss jeder Knoten darin vorkommen, kann aber auch mehrfach vorkommen. Jede Kante $e \in E$ ist muss von der Form $e = e_i$ sein, mit $i \in \{1, \dots, m\}$. Wir sagen dann, dass in Bezug auf den Eulerschen Weg x_{i-1} der Anfangs- und x_i der Endknoten von e ist. (Der Graph ist aber ursprünglich nicht gerichtet!) Eventuell mit Ausnahme von x_0 und x_m muss jeder Knoten gleich oft als Anfangs- und als Endknoten vorkommen. Daher muss $\deg(x)$ für alle $x \neq x_0, x_m$ eine gerade Zahl sein.

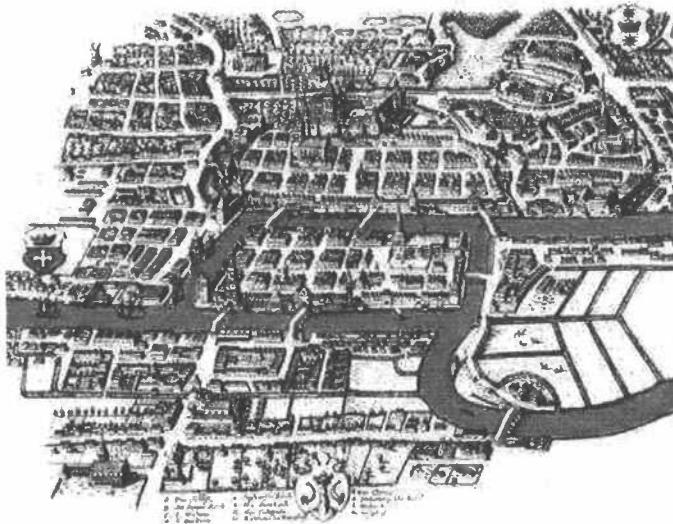


ABBILDUNG 1. Königsberg

Falls es sich um einen Eulerschen Kreis handelt, muss dies auch für $x_0 = x_m$ gelten. (Und umgekehrt: Ist der Eulersche Weg *kein* Kreis, dann sind $\deg(x_0)$ und $\deg(x_m)$ ungerade.)

Falls in G Schleifen existieren, sind diese bei der Bestimmung von $\deg(x)$ (ausnahmsweise) doppelt zu zählen (als einmündend und herausführend).

Die von uns gefundenen Bedingungen sind nicht nur notwendig, sondern auch hinreichend:

(2.7) Satz. [Euler 1736]. Sei $G = (V, E)$ ein (ungerichteter) zusammenhängender Graph.

- (a) In G existiert ein Eulerscher Weg von x nach y genau dann, wenn alle Knoten außer x und y geraden Grad haben (d.h., x und y dürfen – aber müssen nicht – ungeraden Grad haben).
- (b) In G existiert ein Eulerscher Kreis genau dann, wenn alle Knoten geraden Grad haben. □

Man kann (b) wie folgt beweisen; mit einer einfachen Modifikation auch (a).

Wir starten mit einem Knoten x_0 , wählen eine dort ausgehende Kante e_1 usw.

Haben wir bereits einen Weg $[x_0, e_1, x_1, \dots, e_k, x_k]$ ohne Wiederholung von Kanten, dann überprüfen wir, ob von x_k eine noch nicht verwendete Kante ausgeht. Falls so eine Kante existiert, fügen wir sie dem Weg hinzu.

Da nur endlich viele Kanten vorhanden sind, bricht dieses Verfahren bei einem Index k ab. Es muss dann x_k einer der schon vorher besuchten Knoten sein, da sonst noch Kanten zur Verfügung stehen müssten. Es kann sich nicht um einen der Knoten x_1, \dots, x_{k-1} handeln, da dieser sonst ungeraden Grad hätte.

Daher ist $x_k = x_0$, wir haben also einen geschlossenen Weg W_1 aus lauter verschiedenen Kanten gefunden. Enthält dieser alle Kanten, so sind wir fertig. Andernfalls streichen wir die Kanten e_1, \dots, e_k , d.h., wir bilden den Graphen $G_1 = (V_1, E_1)$ mit $E_1 = E \setminus \{e_1, \dots, e_k\}$ und $V_1 = \{x \in V : \exists e \in E_1, e \ni x\}$.

G_1 ist nicht unbedingt zusammenhängend, alle Knoten von G_1 haben aber geraden Grad. Da G zusammenhängend war, muss V_1 einen Knoten von W_1 enthalten. Beginnend mit diesem Knoten verfahren wir in G_1 genauso wie vorher in G und finden einen geschlossenen Weg aus lauter verschiedenen Kanten in G_1 , der mit W_1 (zumindest) einen Knoten, aber keine Kante gemeinsam hat. Durch entsprechendes Koppeln der beiden Wege erhalten wir einen neuen „Kantenkreis“ W_2 , der mehr Kanten als W_1 hat.

Nun fahren wir mit W_2 genauso fort wie vorher mit W_1 , und so weiter, bis keine Kanten mehr übrig sind. Dann durchläuft der Weg W_ℓ jede Kante genau einmal. \square

(Zum tieferen Verständnis: wo bricht dieser Beweis zusammen, wenn nicht alle Knotengrade gerade sind?)

Der obige Beweis enthält einen Algorithmus zum Finden eines Eulerschen Kreises. Allerdings gibt es eine vereinfachte Variante dieses Algorithmus. Hierfür benötigen wir folgende Definition.

(2.8) Definition. Eine Kante e in einem Graphen G heißt *Brücke*, falls durch das Entfernen von e eine Zusammenhangskomponente von G in zwei Teile zerfällt.

(2.9) Algorithmus von Fleury. Wir beginnen mit einem beliebigen Knoten x_0 und fügen nach folgenden Vorschriften Kanten hinzu.

Schon gegeben: Ein Weg $[x_0, e_1, x_1, \dots, e_{k-1}, x_{k-1}]$ ohne Wiederholung von Kanten.

Sei $E_{k-1} = E \setminus \{e_1, \dots, e_{k-1}\}$ und $V_{k-1} = \{x \in V : \exists e \in E_{k-1} \text{ mit } e \ni x\}$. Wir betrachten $G_{k-1} = (V_{k-1}, E_{k-1})$.

Wähle eine Kante $e_k \in E_{k-1}$, die

- (a) von x_{k-1} ausgeht, und
- (b) nur dann eine Brücke von G_{k-1} ist, wenn es keine andere Möglichkeit gibt.

Füge die Kante e_k und ihren zweiten Endpunkt zu dem bisherigen Weg hinzu und fahre mit k anstelle von $k - 1$ fort.

Dieser Algorithmus endet mit dem Erreichen eines Eulerschen Kreises. (Beweis nicht trivial!)

(2.10) Will man einen Graphen in einem Programm einlesen, bzw. darstellen, so gibt es verschiedene Möglichkeiten.

Bei einem ungerichteten Graphen ohne Mehrfachkanten kann man z.B. eine Liste der Knoten angeben, wobei jedem Knoten sein Grad und eine Liste seiner Nachbarn folgt. Zum Beispiel für den Graphen aus Beispiel 2.13 weiter unten:

Knoten	Grad	Nachbarn
1	3	2, 3, 4
2	2	1, 3
3	3	1, 2, 4
4	2	1, 3

Eine weitere Möglichkeit ist die Adjazenzmatrix (diese ist mathematisch besser, braucht aber meistens mehr Speicherplatz).

(2.11) Definition. Die Adjazenzmatrix $A = A(G)$ des (gerichteten oder ungerichteten) Graphen $G = (V, E)$ mit $V = \{x_1, \dots, x_n\}$ (\equiv geeignete Numerierung der Knoten) ist die Matrix

$$A = (a_{i,j})_{i,j=1}^n,$$

wobei $a_{i,j}$ die Anzahl der Kanten von x_i nach x_j ist.

Für einen Graphen oder Digraphen ohne Mehrfachkanten ist insbesondere stets $a_{i,j} \in \{0, 1\}$. Falls es keine Schleifen gibt, sind zudem alle Einträge $a_{i,i}$ auf der Diagonalen 0.

Für ungerichtete Graphen ist A symmetrisch, d.h. $a_{j,i} = a_{i,j}$. Insbesondere ist

$$\deg(x_i) = \sum_{j=1}^n a_{i,j}.$$

Bei Graphen mit gewichteten Kanten (vgl. Traveling Salesman) verwendet man oft eine gewichtete Adjazenzmatrix, bei der $a_{i,j}$ durch das Gewicht der Kante $[x_i, x_j]$ ersetzt wird.

Wir betrachten die k -te Matrixpotenz

$$A^k = \underbrace{A \cdot A \cdots A}_{k \text{ mal}} =: (a_{i,j}^{(k)})_{i,j=1}^n.$$

Für $k = 0$ setzt man dabei $A^0 = I$, die Einheitsmatrix.

(2.12) Satz. Die Anzahl der Wege der Länge k von x_i nach x_j ist $a_{i,j}^{(k)}$.

Beweis durch Induktion nach k . Die Aussage ist für $k = 1$ (auch für $k = 0$) richtig.

Angenommen, die Aussage stimmt für k , und zwar für alle Knoten x_i, x_j .

Jeden Weg der Länge $k + 1$ von x_i nach x_j kann man in eine Anfangskante von x_i zu einem Knoten $x_{i'}$ und den darauffolgenden Weg der Länge k von $x_{i'}$ nach x_j zerlegen.

Zu jeder Anfangskante von x_i nach $x_{i'}$ gibt es nach Annahme $a_{i',j}^{(k)}$ Wege der Länge k von $x_{i'}$ nach x_j . Daher ist die Anzahl der Wege der Länge $k + 1$ von x_i über $x_{i'}$ (1. Schritt) nach x_j gleich

$$a_{i,i'} a_{i',j}^{(k)}.$$

Variieren der $x_{i'}$ ergibt alle verschiedenen Möglichkeiten, also ist die Anzahl der Wege der Länge $k + 1$ von x_i nach x_j

$$\sum_{i'=1}^n a_{i,i'} a_{i',j}^{(k)}.$$

Nach Definition der Matrizenmultiplikation ist das $a_{i,j}^{(k+1)}$. □

Zur Berechnung von $a_{i,j}^{(k)}$ kann man erzeugende Funktionen verwenden. Sei

$$w_{i,j}(x) = \sum_{k=0}^{\infty} a_{i,j}^{(k)} x^k.$$

Da $a_{i,j}^{(k)} \leq m^k$ (mit $m = |E|$, Anzahl der Kanten), konvergiert diese Reihe zumindestens für $|x| < 1/m$. Setzen wir

$$W(x) = (w_{i,j}(x))_{i,j=1}^n$$

so ist in Matrzenschreibweise

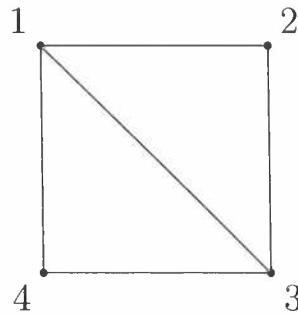
$$\begin{aligned} W(x) &= \sum_{k=0}^{\infty} x^k \cdot A^k = I + x \cdot A + x^2 \cdot A^2 + \dots \\ &= I + x \cdot A(I + x \cdot A + \dots) = I + x \cdot A W(x), \end{aligned}$$

also $(I - x \cdot A)W(x) = I$, d.h.

$$W(x) = (I - x \cdot A)^{-1}$$

(inverse Matrix). Damit kann man $w_{i,j}(x)$ ausrechnen und danach in eine Potenzreihe (Taylorreihe) entwickeln. Die Koeffizienten sind die gesuchten Weganzahlen.

(2.13) Beispiel. Sei $G = (V, E)$ (ungerichtet) wie folgt.



$V = \{1, 2, 3, 4\}$ und $E = \{[1, 2], [1, 3], [1, 4], [2, 3], [4, 3]\}$.

$$A = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} \quad I - x \cdot A = \begin{pmatrix} 1 & -x & -x & -x \\ -x & 1 & -x & 0 \\ -x & -x & 1 & -x \\ -x & 0 & -x & 1 \end{pmatrix}$$

Invertieren (z.B. mit MAXIMA) ergibt

$$w_{1,1}(x) = \frac{1 - 2x^2}{1 - 5x^2 - 4x^3}.$$

Man findet

$$1 - 5x^2 - 4x^3 = (1 + x)(1 - x - 4x^2) = (1 + x)\left(1 - \frac{1-\sqrt{17}}{2}x\right)\left(1 - \frac{1+\sqrt{17}}{2}x\right).$$

Partialbruchzerlegung: Ansatz

$$w_{1,1}(x) = \frac{A}{1+x} + \frac{B}{1 - \frac{1-\sqrt{17}}{2}x} + \frac{C}{1 - \frac{1+\sqrt{17}}{2}x}$$

liefert

$$1 - 2x^2 = A(1 - x - 4x^2) + B(1 + x)\left(1 - \frac{1+\sqrt{17}}{2}x\right) + C(1 + x)\left(1 - \frac{1-\sqrt{17}}{2}x\right).$$

Einsetzen von $x = -1$, $x = \frac{2}{1-\sqrt{17}}$ und $x = \frac{2}{1+\sqrt{17}}$ ergibt

$$A = \frac{1}{2}, \quad B = -\frac{1 - \sqrt{17}}{4\sqrt{17}}, \quad C = \frac{1 + \sqrt{17}}{4\sqrt{17}}.$$

Wir verwenden:

$$\frac{1}{1 - \lambda x} = \sum_{k=0}^{\infty} \lambda^k x^k.$$

Daher, mit

$$\lambda_1 = -1, \quad \lambda_2 = \frac{1-\sqrt{17}}{2}, \quad \lambda_3 = \frac{1+\sqrt{17}}{2},$$

$$w_{1,1}(x) = \sum_{n=0}^{\infty} (A \lambda_1^n + B \lambda_2^n + C \lambda_3^n) x^n.$$

Wir erhalten

$$a_{1,1}^{(k)} = \frac{1}{2}(-1)^k - \frac{1}{2\sqrt{17}} \left(\frac{1 - \sqrt{17}}{2}\right)^{k+1} + \frac{1}{2\sqrt{17}} \left(\frac{1 + \sqrt{17}}{2}\right)^{k+1}$$

als Anzahl der geschlossenen Wege, die beim Knoten 1 beginnen. Das ist für jedes k eine ganze Zahl! \square

3. Bäume

(3.1) Definition. Ein *Baum* ist ein zusammenhängender Graph ohne Kreise.

Ein Graph ohne Kreise (nicht unbedingt zusammenhängend) heißt *Wald*. So gesehen ist ein Baum ein zusammenhängender Wald. Insbesondere sind alle Zusammenhangskomponenten eines Waldes Bäume. Für Bäume schreibt man oft T („tree“) statt G .

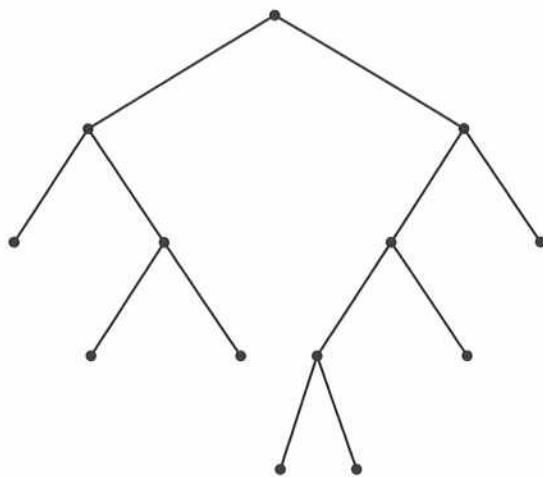


ABBILDUNG 2. Ein Baum T .

(3.2) Satz. Jede der folgenden Bedingungen charakterisiert einen Graphen $T = (V, E)$ mit n Knoten als Baum.

- (1) T ist zusammenhängend und hat $n - 1$ Kanten.
- (2) T ist zusammenhängend und jede Kante ist eine Brücke.
- (3) Für je zwei Knoten x, y gibt es genau einen Pfad von x nach y .

Vier Beweisschritte (\rightarrow Übungen!), z.B. in der Reihenfolge:

- (i) Baum \implies (1)
- (ii) (1) \implies (2)
- (iii) (2) \implies (3)
- (iv) (3) \implies Baum.

□

(3.3) Darstellung von Bäumen. Die Adjazenzmatrix A eines Baumes $T = (V, E)$ mit n Knoten hat n Zeilen und Spalten, also n^2 Elemente, aber nur $2(n - 1)$ Einsen (warum?). Daher braucht A unnötig viel Speicherplatz.

Aus diesem Grund – und auch, um in einem Baum den Pfad zwischen zwei Knoten effizient suchen zu können – ist es sinnvoll, weitere Informationen gemeinsam mit jedem Knoten anzugeben.

Wir wählen eine Wurzel $x_0 \in V$. Diese bewirkt eine Orientierung der (ursprünglich ungerichteten) Kanten, und zwar „weg von der Wurzel“. Für jeden Knoten $x \neq x_0$ geben wir seinen *Vorgänger* $v(x)$ (den zu x_0 näheren Nachbarn) und seine *Höhe* $h(x)$ (die Länge des Pfades von x_0 nach x) an. (Man veranschauliche sich dies anhand des Baumes in Abbildung 2.) Zusätzlich kann es noch sinnvoll sein, für jeden Knoten die Liste seiner Nachfolger anzugeben. Knoten ohne Nachfolger heißen *Blätter*.

Ist $k \leq h(x)$, dann schreiben $v^k(x)$ für den k -ten Vorgänger von x . Insbesondere ist also $v^{h(x)}(x) = x_0$.

Um den Pfad von x nach y in T zu finden, müssen wir den ersten „gemeinsamen Vorfahren“ bestimmen. Wenn $h(x) = h(y) + r$, wobei o.B.d.A. $r \geq 0$, dann suchen wir den kleinsten Index j sodass

$$v^{j+r}(x) = v^j(y).$$

Die Knoten des Pfades sind dann

$$[x, v(x), \dots, v^{j+r}(x), v^{j-1}(y), \dots, v(y), y].$$

Dies ergibt einen Algorithmus zum Auffinden des Pfades zwischen zwei Knoten eines Baumes.

(3.4) Definition. Sei G ein Graph. Ein (*auf*)spannender Baum oder Spannbaum von G ist ein Baum T mit $V(T) = V(G)$ und $E(T) \subset E(G)$.

(3.5) Algorithmus (Spannbaum) Sei $G = (V, E)$.

Wir wählen eine Wurzel $x_0 \in V$.

In jedem Durchgang des Algorithmus haben wir eine gereihte Liste der abzuarbeitenden Knoten und einen Baum T mit Wurzel x_0

- 1.) Am Anfang besteht die Liste aus x_0 , und der Baum T besteht nur aus x_0 , und $h(x_0) = 0$.
- 2.) In jedem Durchgang nehmen wir den ersten Knoten x der Liste.
- 3.) Wenn alle Nachbarn von x schon zu T gehören, streicht man x aus der Liste, der nächste Knoten der Liste ist nun der erste.
- 4.) Andernfalls nehme einen Nachbarn y von x in G , der noch nicht zu T gehört, und

- füge y zu $V(T)$ hinzu,
- füge $[x, y]$ zu $E(T)$ hinzu,
- setze $v(y) = x$ und $h(y) = h(x) + 1$, und
- füge y am Ende der Liste hinzu.

Der Algorithmus endet, wenn die Liste leer ist. Falls G zusammenhängend ist, ist T dann ein Spannbaum (warum?).

Der so gefundene spannende Baum heißt BFS-Baum (*breadth first search*, d.h., „zuerst in die Breite suchen“).

Variante: DFS-Baum (*depth first search*, d.h., „zuerst in die Tiefe suchen“). Der einzige Unterschied ist, dass man beim letzten Punkt von 4.) den Knoten y am *Anfang* der Liste hinzufügt (also als nächsten bearbeitet).

(3.6) Beispiel. BFS- und DFS-Baum im Petersen-Graph.

(3.7) Bemerkung. Ein BFS-Baum hat den Vorteil, dass er kürzeste Pfade (kürzeste Länge im ursprünglichen Graphen) von der Wurzel x_0 zu jedem beliebigen anderen Knoten enthält. In anderen Zusammenhängen ist hingegen ein DFS-Baum praktischer.

(3.8) Algorithmus (alle Spannbäume). Ist man an *allen* Spannbäumen eines Graphen interessiert, kann man wie folgt vorgehen („cut and fuse“):

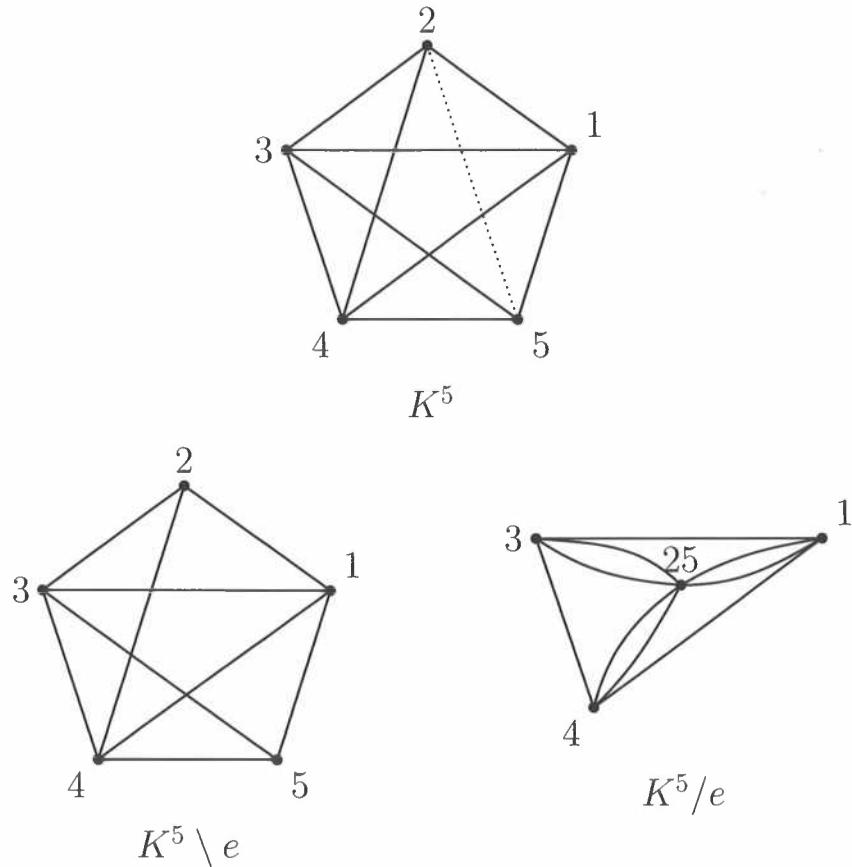
Wähle eine beliebige Kante $e \in E$. Dann kann man die Menge der Spannbäume in zwei Klassen zerlegen: Diejenigen, die e enthalten und diejenigen, die e nicht enthalten.

Andererseits kann man die folgenden Operationen am Graphen ausführen:

Auslöschen: $G \setminus e = (V, E \setminus \{e\})$

Kontraktion: G/e ist der Graph, den man erhält, wenn man die Endknoten der Kante e miteinander identifiziert (dabei können auch Mehrfachkanten entstehen!).

Beispiel:



Sei $\mathcal{T}(G)$ die Menge der Spannbäume von G . Dann gibt es offensichtliche Bijektionen

$$\begin{aligned} \{T \in \mathcal{T}(G) : e \notin T\} &\longleftrightarrow \mathcal{T}(G \setminus e), \\ \{T \in \mathcal{T}(G) : e \in T\} &\longleftrightarrow \mathcal{T}(G/e). \end{aligned}$$

Man kann also alle Spannbäume von G ermitteln, indem man rekursiv Kanten löscht bzw. kontrahiert und die Spannbäume in den reduzierten Graphen bestimmt.

Ist man nur an der Anzahl der Spannbäume interessiert, dann hilft Kirchhoffs¹ Matrix-Tree Theorem weiter.

¹Gustav Kirchhoff (1824–1887)

(3.9) Definition. Die Laplace-Matrix eines Graphen G ist die Matrix $L(G)$ mit den Einträgen

$$[L(G)]_{ij} = \begin{cases} -a_{ij} & \text{wenn } i \neq j \\ \deg x_i & \text{wenn } i = j \end{cases}$$

wobei $A = (a_{ij})$ die Adjazenzmatrix aus Definition 2.11 bezeichnet, d.h., a_{ij} ist die Anzahl der Kanten zwischen den Knoten v_i und v_j .

(3.10) Satz. (Matrix-Tree Theorem) Sei G ein Graph ohne Schleifen mit Laplace-Matrix $L(G)$. Weiters sei $v_i \in V(G)$ ein beliebiger Knoten und $\tilde{L}(G)$ die Matrix, die man erhält, wenn man die i -te Zeile und i -te Spalte aus $L(G)$ streicht. Dann ist

$$|\mathcal{T}(G)| = \det \tilde{L}(G).$$

(3.11) Beispiel. Wir überprüfen den Algorithmus (3.8) anhand des Graphen K^5 auf Seite D.16:

$$L(K^5 \setminus e) = \begin{pmatrix} 3 & -1 & -1 & 0 & -1 \\ -1 & 4 & -1 & -1 & -1 \\ -1 & -1 & 4 & -1 & -1 \\ 0 & -1 & -1 & 3 & -1 \\ -1 & -1 & -1 & -1 & 4 \end{pmatrix} \quad L(K^5/e) = \begin{pmatrix} 6 & -2 & -2 & -2 \\ -2 & 4 & -1 & -1 \\ -2 & -1 & 4 & -1 \\ -2 & -1 & -1 & 4 \end{pmatrix}$$

Und nach Streichen der jeweils ersten Zeile und Spalte erhalten wir

$$|\mathcal{T}(K^5 \setminus e)| = \begin{vmatrix} 4 & -1 & -1 & -1 \\ -1 & 4 & -1 & -1 \\ -1 & -1 & 3 & -1 \\ -1 & -1 & -1 & 4 \end{vmatrix} = 75 \quad |\mathcal{T}(K^5/e)| = \begin{vmatrix} 4 & -1 & -1 \\ -1 & 4 & -1 \\ -1 & -1 & 4 \end{vmatrix} = 50$$

In der Tat ist

$$L(K^5) = \begin{pmatrix} 4 & -1 & -1 & -1 & -1 \\ -1 & 4 & -1 & -1 & -1 \\ -1 & -1 & 4 & -1 & -1 \\ -1 & -1 & -1 & 4 & -1 \\ -1 & -1 & -1 & -1 & 4 \end{pmatrix}$$

und wir erhalten durch Streichen der letzten Zeile und Spalte

$$|\mathcal{T}(K^5)| = \begin{vmatrix} 4 & -1 & -1 & -1 \\ -1 & 4 & -1 & -1 \\ -1 & -1 & 4 & -1 \\ -1 & -1 & -1 & 4 \end{vmatrix} = 125.$$

Da jeder Baum mit n Knoten ein Spannbaum des vollständigen Graphen K^n ist, kann man u.a. mit dem Matrix-Tree Theorem den folgenden Satz beweisen:

(3.12) Satz. (Cayley) *Es gibt n^{n-2} verschiedene Bäume mit n Knoten.*

Im folgenden wollen wir uns mit Spannbäumen von gewichteten Graphen beschäftigen.

(3.13) Algorithmus von Kruskal. Sei $G = (V, E)$ ein zusammenhängender Graph mit n Knoten und m gewichteten Kanten (Gewicht $w(e)$, $e \in E$).

Gesucht: ein Spannbaum T , sodass

$$\sum_{e \in E(T)} w(e) \rightarrow \text{minimal.}$$

Vorgangsweise: Nummeriere die Kanten aufsteigend nach ihrem Gewicht, also $E = \{e_1, \dots, e_m\}$ mit

$$w(e_1) \leq w(e_2) \leq \dots \leq w(e_m).$$

Zu Beginn sei T der Graph mit Knotenmenge V und leerer Kantenmenge. Im Folgenden fügen wir sukzessive Kanten zu T hinzu, bis ein Baum entsteht.

Wir sehen uns die Kanten e_1, \dots, e_m in dieser Reihenfolge an. Sei e_k die aktuelle Kante.

Falls das Hinzufügen von e_k zu $E(T)$ einen Kreis erzeugen würde, übergehen wir e_k (d.h., wir fügen die Kante nicht hinzu), ansonsten fügen wir e_k zu $E(T)$ hinzu.

Damit ist e_k abgearbeitet und wir gehen zu e_{k+1} über.

Wir fahren fort, bis $E(T)$ genau $n-1$ Kanten enthält. Dann ist T ein Spannbaum von G .

(3.14) Bemerkung. Der Algorithmus von Kruskal findet einen Spannbaum mit minimalem Gesamtgewicht. Falls alle Kanten von G unterschiedliche Gewichte haben, ist dieser Spannbaum sogar eindeutig.

4. Matchings

(4.1) Definition. Ein *Matching* (oder eine *Paarung*) in einem Graphen $G = (V, E)$ ist eine Menge $M \subseteq E$ von Kanten, sodass jeder Knoten von G in höchstens einer Kante von M liegt.

Ist M ein Matching und $[x, y] \in M$, dann sagt man, dass x mit y gematcht ist (und umgekehrt ist y mit x gematcht).

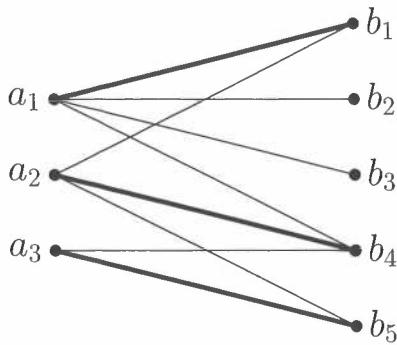
Ein Matching in G heißt *perfekt*, falls jeder Knoten von G gematcht ist. Dies ist offensichtlich genau dann der Fall, wenn $|M| = |V|/2$.

(4.2) Beispiel. Eine Firma erhält Aufträge a_1, a_2, a_3 . Zur Bearbeitung dieser Aufträge stehen die Mitarbeiter b_1, b_2, b_3, b_4, b_5 zur Verfügung. Den Fähigkeiten der Mitarbeiter entsprechend weiß man, dass b_1 die Aufträge a_1 oder a_2 übernehmen könnte, b_2 und b_3 jeweils nur a_1 , b_4 könnte jeden der drei Aufträge übernehmen und b_5 könnte a_2 oder a_3 übernehmen.

Jeder Auftrag soll von einem (und nur einem) Mitarbeiter bearbeitet werden und kein Mitarbeiter soll mehrere Aufträge übernehmen.

Dieses Problem lässt sich in Form eines Matchings in einem Graphen $G = (V, E)$ modellieren. Dazu setzt man $V = \{a_1, a_2, a_3, b_1, b_2, b_3, b_4, b_5\}$ und $E = \{[a_i, b_j] : b_j \text{ kann } a_i \text{ bearbeiten}\}$.

Gesucht ist nun ein Matching, in welchem jeder der Knoten a_1, a_2, a_3 gematcht ist. Dies ist zum Beispiel bei $M = \{[a_1, b_1], [a_2, b_4], [a_3, b_5]\}$ der Fall (die fett gezeichneten Kanten in der folgenden Abbildung).



Der Graph aus Beispiel 4.2 hat eine spezielle Struktur: Schreibt man $A = \{a_1, a_2, a_3\}$ und $B = \{b_1, b_2, b_3, b_4, b_5\}$, dann verläuft jede Kante zwischen einem Knoten in A und einem Knoten in B . Solch einen Graphen G nennt man *bipartit*. Die Knotenmengen A und B heißen *Seiten* von G . Bipartite Graphen sind vor allem in Hinsicht auf Matchings leichter zu untersuchen als allgemeine Graphen. In vielen Anwendungen ist zudem ein modellierender Graph automatisch bipartit.

(4.3) Satz. (Hall) Ein bipartiter Graph G mit Seiten A, B hat genau dann ein Matching, in welchem jeder Knoten $a \in A$ gematcht ist, wenn die folgende Bedingung gilt.

$$(4.4) \quad \forall S \subseteq A: |N(S)| \geq |S|,$$

wobei $N(S) = \{b \in B : \exists a \in S \text{ mit } [a, b] \in E\}$ die Nachbarschaft von S bezeichnet.

Die Bedingung (4.4) im Satz von Hall ist offensichtlich notwendig für das gesuchte Matching, denn falls die Ungleichung für eine Menge $S \subseteq A$ nicht gilt, dann wird in jedem Matching mindestens ein Knoten aus S (und somit mindestens ein Knoten aus A) ungematcht bleiben. Dass (4.4) auch hinreichend für das gesuchte Matching ist, ist nicht ganz so leicht zu sehen.

Wir betrachten nun eine Möglichkeit, in einem bipartiten Graphen ein größtmögliches Matching algorithmisch zu finden. Hierfür sind die folgenden Begriffe hilfreich.

(4.5) Definition. Angenommen, M ist ein Matching in einem bipartiten Graphen G . Ein Pfad $P = [x_0, e_1, x_1, \dots, e_k, x_k]$ heißt *alternierend* (bezüglich M), falls er in einem ungematchten Knoten beginnt und abwechselnd Kanten

aus $E(G) \setminus M$ und aus M durchläuft (formal: x_0 ist ungematcht und $e_i \in M$ genau dann, wenn i gerade ist).

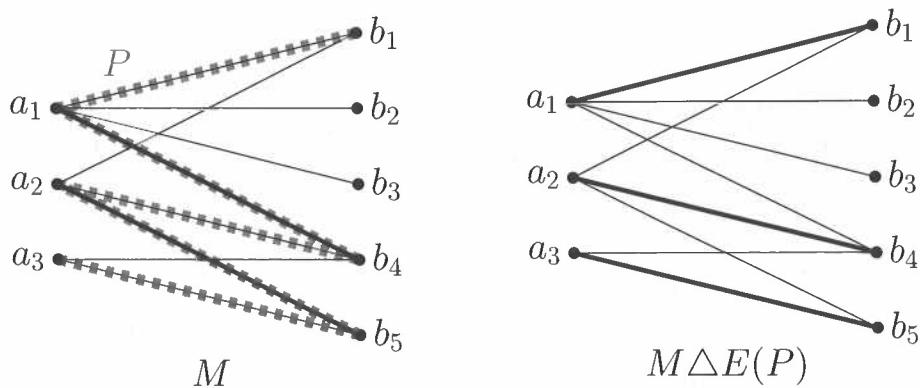
Ist $P = [x_0, e_1, x_1, \dots, e_k, x_k]$ ein alternierender Pfad und zusätzlich sein letzter Knoten x_k ebenfalls ungematcht, dann nennt man P verbessernd (oder augmentierend).

(4.6) Bemerkung. Sei M ein Matching in einem bipartiten Graphen G .

(a) Ist P ein verbessernder Pfad, dann ist die *symmetrische Differenz*

$$M \Delta E(P) = (M \cup E(P)) \setminus (M \cap E(P))$$

ein Matching mit einer Kante mehr als M (vgl. Abbildung unten).



(b) Ist M nicht größtmöglich, dann gibt es einen verbessernden Pfad P .

Zum Beispiel ist für jedes Matching M' , das größer als M ist, mindestens eine Komponente des Graphen $(V(G), M \Delta M')$ ein verbessernder Pfad (bezüglich M).

(4.7) Algorithmus (Matching in bipartiten Graphen). Gegeben: Bipartiter Graph G mit Seiten A und B .

Gesucht: Größtmögliche Matching in G .

Vorgehensweise: Wir beginnen mit $M = \emptyset$ und vergrößern M rekursiv durch verbessernde Pfade, bis M größtmöglich ist.

Hierfür verwenden wir eine Variante von Algorithmus 3.5 (BFS) mit den folgenden zwei Unterschieden:

- (1) Die gereihte Liste der abzuarbeitenden Knoten besteht am Anfang nicht nur aus einem Knoten, sondern aus allen ungematchten Knoten in A (in beliebiger Reihenfolge).

(2) Liegt der aktuell erste Knoten x der Liste in B , dann

- beenden wir den Algorithmus, falls x ungematcht ist;
- ansonsten fügen wir den Knoten y , mit dem x gematcht ist, an das Ende der Liste hinzu, setzen $v(y) = x$ und entfernen x aus der Liste.

Solange M noch nicht größtmöglich ist, findet dieser Algorithmus einen ungemachten Knoten $x \in B$. Zusammen mit allen seinen Vorgängern (und den Kanten zwischen ihnen) bildet x einen verbessernden Pfad P .

Wir ersetzen M durch $M \Delta E(P)$ und wiederholen die obige BFS-Variante.

Der Algorithmus endet, sobald die BFS-Variante keinen ungemachten Knoten in B findet. Dann ist M größtmöglich.

(4.8) Bemerkung. Für nicht bipartite Graphen gibt es ebenfalls Algorithmen zum Auffinden größtmöglicher Matchings. Diese sind allerdings weitaus komplizierter als Algorithmus 4.7.

(4.9) Bemerkung. In Anwendungen können oft zusätzliche Bedingungen an das Matching gestellt werden. Zum Beispiel können die Kanten des Graphen gewichtet sein, um Kosten oder Zeitdauer zu repräsentieren. Dann ist zumeist ein Matching M gesucht, in welchem nicht nur möglichst viele Knoten gematcht sind, sondern zum Beispiel das Gesamtgewicht der Kanten in M so klein wie möglich ist (Kostenminimierung) oder das maximale Gewicht aller Kanten in M minimal ist (Minimierung der Arbeitsdauer bei parallelen Abläufen).

5. Färbungen

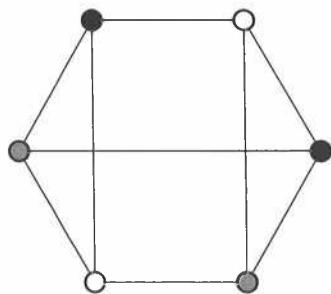
(5.1) Definition. Eine Färbung eines Graphen $G = (V, E)$ ist eine Funktion $c: V \rightarrow F$ mit der Eigenschaft, dass $[x, y] \in E \implies c(x) \neq c(y)$. Hierbei nennen wir die Elemente von F die Farben.

(5.2) Beispiel. An einer Universität soll ein Stundenplan für alle Lehrveranstaltungen eines Semesters erstellt werden. Dabei dürfen einige Lehrveranstaltungen nicht parallel stattfinden, weil sie (a) gleiche Lehrende haben oder (b) für die gleichen Studierenden vorgesehen sind.

Man modelliert das Problem durch einen Graphen mit Knotenmenge $V = \{\text{Lehrveranstaltungen}\}$ und der folgenden Kantenmenge: Zwei Knoten (Lehrveranstaltungen) sind durch eine Kante verbunden, wenn sie nicht parallel stattfinden dürfen.

Interpretiert man Farben als Zeiträume für die Lehrveranstaltungen, dann sucht man eine Färbung mit möglichst wenigen Farben, um den Zeitplan so kompakt wie möglich zu halten.

In einem simplen Beispiel mit nur sechs Lehrveranstaltungen kann eine Färbung des Hilfsgraphen wie folgt aussehen.



(5.3) Beispiel. Auch die bekannten Sudoku-Rätsel lassen sich als Färbungsproblem interpretieren. Dafür betrachtet man einen Graphen mit 81 Knoten (für jedes Feld des Rätsels ein Knoten) und einer Kante zwischen zwei Knoten, wann immer die entsprechenden Felder in der gleichen Zeile, Spalte oder im gleichen (3×3) -Teilquadrat liegen.

Es ist eine Färbung dieses Graphen mit neun Farben (Zahlen von 1 bis 9) gesucht, wobei für einige Knoten bereits die Farben vorgegeben sind. Bei einem lösbareren Sudoku sollte diese Färbung existieren und zudem eindeutig sein.

(5.4) Definition. Die *chromatische Zahl* $\chi(G)$ eines Graphen G ist die kleinste Zahl k , für die es eine Färbung von G mit k Farben gibt.

Es ist kein „schneller“ Algorithmus bekannt, der für einen allgemeinen Graphen die chromatische Zahl berechnet. Einfacher wird es, wenn man mehr als $\chi(G)$ Farben erlaubt.

(5.5) Algorithmus (Greedy-Algorithmus für Färbungen)

Gegeben: Graph $G = (V, E)$.

Gesucht: Färbung von G mit „nicht zu vielen“ Farben.

Vorgehensweise: Zuerst ordne die Knoten als v_1, \dots, v_n . Rekursiv für jedes $i = 1, \dots, n$ wählen wir als Farbe $c(v_i)$ die kleinste natürliche Zahl, die noch nicht als Farbe unter den bereits gefärbten Nachbarn von v_i auftritt. Formal:

$$c(v_i) := \min\{k \in \mathbb{N} : \forall 1 \leq j < i \text{ mit } [v_i, v_j] \in E \text{ gilt } c(v_j) \neq k\}.$$

Ist m die größte Zahl, die im Algorithmus als Farbe verwendet wird, dann ist $c: V \rightarrow \{1, \dots, m\}$ eine Färbung von G mit m Farben.

(5.6) Bemerkung. Der Greedy-Algorithmus erzeugt eine Färbung mit maximal $\Delta(G) + 1$ Farben, wobei

$$\Delta(G) := \max_{v \in V} (\deg(v))$$

den *Maximalgrad* von G bezeichnet.

Die tatsächlich verwendete Anzahl an Farben hängt dabei von der gewählten Reihenfolge der Knoten ab und kann sehr viel größer als $\chi(G)$ sein.

(5.7) Beispiel. Man färbe den Graphen aus Beispiel 5.2 mit dem Greedy-Algorithmus für verschiedene Reihenfolgen der Knoten.

(5.8) Bemerkung. In manchen Anwendungen sind Varianten von Färbungen relevant:

- **Kantenfärbungen:** Färben der Kanten anstelle der Knoten. Dabei erhalten Kanten mit gemeinsamen Knoten unterschiedliche Farben.
(Eine Kantenfärbung entspricht also einer Partition der Kantenmenge in mehrere Matchings.)
- **Listenfärbungen:** Jeder Knoten (oder jede Kante, falls die Kanten gefärbt werden sollen) besitzt eine Liste von erlaubten Farben.

6. Planare Graphen

Wir betrachten das folgende klassische Beispiel für Färbungsprobleme.

(6.1) Beispiel. Man will in einer „politischen“ Landkarte jedes Land so färben, dass benachbarte Länder stets verschiedene Farben haben. Wieviele verschiedene Farben braucht man dafür? (F. Guthrie, 1852).

Man modelliert die Landkarte durch einen Graphen, dessen Knoten die Länder sind und in dem zwei Knoten (Länder) durch eine Kante verbunden werden, wenn sie benachbart sind. Das Problem entspricht dann der Frage nach der chromatischen Zahl eines Graphen, der auf diese Art entsteht.

Nicht jeder Graph kommt auf diese Art zustande:

(6.2) Definition. Ein Graph heißt *planar*, wenn man ihn in der Ebene so zeichnen kann, dass sich keine Kanten überkreuzen.

(6.3) Beispiel. Der Petersen-Graph und die Graphen K^5 und $K_{3,3}$ aus Beispiel 1.3 können nicht ohne Überkreuzungen in der Ebene gezeichnet werden und sind daher nicht planar. Hingegen ist der Graph aus Beispiel 5.2 planar. (Übung: Zeichnen Sie ihn in die Ebene, ohne Kanten zu überkreuzen.)

(6.4) Beispiel. Zeichnen Sie den Graphen, der den Ländern der Europäischen Union entspricht. Färben Sie ihn mit möglichst wenigen Farben.

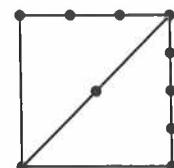
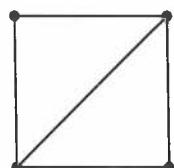
Beispiel 6.1 entspricht also der Frage, wie viele Farben zum Färben eines planaren Graphen notwendig sind. Dieses Problem wurde erst viele Jahre später gelöst.

(6.5) Vierfarbensatz [K. Appel und W. Haken, 1976] Jeder planare Graph kann mit 4 oder weniger Farben gefärbt werden.

Der Beweis des Vierfarbensatzes gelang durch Reduktion auf 1936 Spezialfälle, die dann mit dem Computer überprüft wurden.

(6.6) Definition. Eine *Unterteilung* eines Graphen G erhält man, indem man Kanten von G durch Pfade beliebiger positiver Länge ersetzt.

(6.7) Beispiel.



Ein Graph G

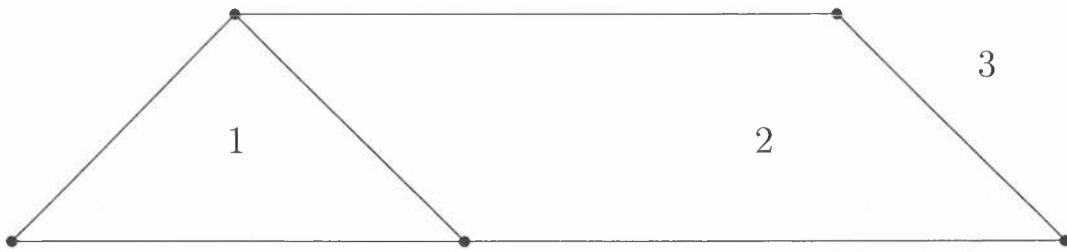
Eine Unterteilung von G

(6.8) Satz. (Kuratowski) Ein Graph ist planar genau dann, wenn er keine Unterteilung von K^5 oder $K_{3,3}$ enthält.

Zeichnet man einen planaren Graphen (ohne Überkreuzungen der Kanten) in die Ebene, kann man drei Parameter definieren:

- $n = |V|$ Anzahl der Knoten
- $m = |E|$ Anzahl der Kanten
- $f =$ Anzahl der Flächen, in die die Ebene unterteilt wird.

(6.9) Beispiel.



Ein Graph mit $n = 5$, $m = 6$, $f = 3$.

(6.10) Satz. [Euler] In einem zusammenhängenden planaren Graphen G (Schleifen und Mehrfachkanten erlaubt) gilt

$$n - m + f = 2.$$

Beweis. Induktion nach f .

(i) $f = 1$

Der Graph enthält keine Kreise, weil er ansonsten die Ebene in mindestens zwei Flächen unterteilen würde. Also ist er ein Baum und wegen Satz 3.2 gilt $m = n - 1$.

$$n - (n - 1) + 1 = 2.$$

(ii) $f - 1 \rightarrow f$ (für $f \geq 2$)

Der Graph G enthält mindestens einen Kreis, weil ansonsten $f = 1$ wäre. Man wähle eine beliebige Kante aus einem Kreis und entferne sie aus dem Graphen. Der erhaltene Graph G' ist immer noch zusammenhängend und planar. Da durch das Entfernen der Kante zwei Flächen zu einer verschmelzen, hat G' die Parameter

n Knoten, $m - 1$ Kanten, $f - 1$ Flächen.

Nun ist die Induktionsannahme anwendbar und es muss gelten

$$n - m + f = n - (m - 1) + (f - 1) = 2.$$

□

(6.11) Korollar. Ein planarer Graph (ohne Schleifen und Mehrfachkanten) mit $n \geq 3$ Knoten hat höchstens $3n - 6$ Kanten.

Beweis. Jeder Graph mit drei Knoten hat höchstens drei Kanten, die Aussage ist also wahr für $n = 3$.

Sei nun $G = (V, E)$ ein planarer Graph mit $n \geq 4$ Knoten und f Flächen. Wir dürfen annehmen, dass G zusammenhängend ist. (Ansonsten können wir Kanten hinzufügen bis der Graph zusammenhängend wird, ohne die Anzahl der Knoten und Flächen zu ändern.) Betrachte die Menge

$$P = \{(e, F) : F \text{ Fläche}, e \in E \text{ liegt auf dem Rand von } F\}.$$

Dann gilt

- jede Kante begrenzt maximal 2 Flächen $\implies |P| \leq 2m$.
- jede Fläche wird von mindestens 3 Kanten begrenzt $\implies |P| \geq 3f$.

und insgesamt $3f \leq |P| \leq 2m$. Einsetzen in die Eulersche Formel liefert

$$6 = 3n - 3m + 3f \leq 3n - 3m + 2m = 3n - m$$

und daher $m \leq 3n - 6$. □

