Diskrete Mathematik 1

Ein Skriptum zur Vorlesung
im Sommersemester 2010

Arne Dür

3.Auflage

© 2010 ARNE DÜR INNSBRUCK, ÖSTERREICH

Vorwort

Die *Inhalte* der Lehrveranstaltung „Diskrete Mathematik 1“ sind

- wohlfundierte Induktion
- Graphen und Bäume
- Grundlagen des Abzählens
- elementare Zahlentheorie.

Die *Ziele* dieser Lehrveranstaltung sind

- der Aufbau eines Grundwissens über formale Techniken sowie
- die Vermittlung eines methodischen Vorgehens bei der Lösung von Problemen.

In der Software-Entwicklung angewandt können diese Techniken in der Schwester-Lehrveranstaltung „Algorithmen und Datenstrukturen“ werden.

Das vorliegende Skriptum soll den Hörerinnen und Hörern der Vorlesung das Mitschreiben erleichtern und Zeit zum *Mitdenken* schaffen. Das Skriptum enthält alle Definitionen und Sätze der Vorlesung, aber fast keine vollständigen Beispiele. In der Vorlesung werden die Definitionen motiviert, die wesentlichen Ergebnisse ausführlich erläutert und Rechenverfahren in konkreten Beispielen vorgeführt. Daher ist dieses Skriptum als Grundlage, aber nicht als Ersatz für eine Vorlesungsmitschrift zu verstehen.

Bedanken möchte ich mich bei Herrn Kristian Kuhnert für die kritische Durchsicht des Skriptums.

Inhaltsverzeichnis

Vorwort ii

Kapitel 1. Grundlagen 1
1. Formales Beweisen 1
2. Wachstum von Funktionen und asymptotische Notation 4
3. Äquivalenzrelationen 5
4. Partielle Ordnungen 8
5. Die wohlfundierte Induktion 11
6. Das Wortmonoid 12
7. Strukturelle Induktion 16

Kapitel 2. Graphentheorie 18
1. Gerichtete Graphen 18
2. Ungerichtete Graphen 28

Kapitel 3. Zähltheorie 36
1. Aufzählen und Numerieren von Objekten 36
2. Abzählbarkeit von Mengen 42
3. Lösen von Rekursionsformeln 49

Kapitel 4. Zahlentheorie 53
1. Rechnen mit ganzen Zahlen 53
2. Der euklidische Algorithmus 55
3. Primzahlen 60
4. Restklassen 62

Anhang 66
Das griechische Alphabet 66
Der ASCII-Code 67

Literaturverzeichnis 68

KAPITEL 1

Grundlagen

1. Formales Beweisen

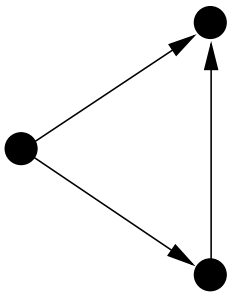
1.1. Wozu exakte Definitionen ?

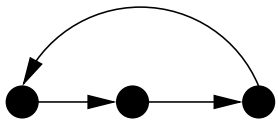
Alle Begriffe der diskreten Mathematik werden aus den Begriffen „Menge“ und „Abbildung“ abgeleitet, z.B.

Numerierung, Ordnung, Graph, Wort .

Dem Nachteil des Aufwandes für die exakte Definition stehen folgende Vorteile gegenüber:

- Reduktion auf das Wesentliche
- Gleichheit mit ja oder nein entscheidbar
- Programmierung naheliegend





baba

b
a
b
a

1.2. Wozu Beweise ?

- Mit einem ausformulierten Beweis kann man sich selbst oder Kollegen überzeugen, dass richtig überlegt wurde. Oft stellt sich erst im Beweis an Hand der Argumentationskette heraus, dass gewisse Voraussetzungen fehlen oder überflüssig sind.
- Durch das Studium von Beweisen trainieren Sie das logische Denken und werden befähigt, eigene Ideen korrekt zu formulieren und durch einen Beweis zu verifizieren.
- Beweise führen oft zu programmierbaren Verfahren, weil die einzelnen Schritte im Beweis so klein sind, dass sie leicht in eine Programmiersprache übertragen werden können.

1.3. Deduktive Beweise.

Ein *deduktiver Beweis* besteht aus einer Folge von Aussagen, die von einer *Hypothese* zu einer *Konklusion* führen. Jeder Beweisschritt muss sich nach einer akzeptierten logischen Regel aus den gegebenen Fakten oder aus vorangegangenen Aussagen ergeben. Der Aussage, dass die Folge der Beweisschritte von einer Hypothese H zu einer Konklusion K führt, entspricht der Satz:

$$\text{Wenn } H\text{, dann } K.$$

„Wenn-dann“-Sätze können auch in anderer Form auftreten.

- Wenn H gilt, folgt daraus K.
- H nur dann, wenn K.
- K, wenn H.
- H impliziert K.

Die Aussage

$$H \text{ impliziert } K$$

und ihre *Kontraposition*

$$(\text{nicht } K) \text{ impliziert } (\text{nicht } H)$$

sind äquivalent, dh. aus dem einen Satz folgt der andere und umgekehrt. Statt zu zeigen, dass die Aussage H die Aussage K impliziert, ist es manchmal leichter zu beweisen, dass die Negation von K die Negation von H impliziert.
Gelegentlich finden wir Aussagen der Form

$$F \text{ genau dann, wenn } G.$$

Andere Varianten dieses Satzes sind etwa:

- F dann und nur dann, wenn G.
- F ist äquivalent zu G.

„Genau dann, wenn“-Aussagen können bewiesen werden, indem man *zwei* Behauptungen zeigt:

- „F, wenn G“, d.h. G impliziert F.
- „F nur dann, wenn G“, d.h. F impliziert G.

1.4. Beweisformen.

Im Folgenden gehen wir auf drei häufig auftretende Beweisprinzipien ein.

1.4.1. *Beweise von Mengeninklusionen.* Seien A und B Mengen. Um die Teilmengeneigenschaft (Inklusion)

$$A \subseteq B$$

zu zeigen, genügt es nach der Definition, die folgende „Wenn-dann“-Aussage zu beweisen:

$$\text{Wenn } x \in A\text{, dann } x \in B.$$

Die *Gleichheit von Mengen* A und B kann bewiesen werden, indem man *zwei* Behauptungen zeigt:

- Wenn $x \in A$, dann $x \in B$.
- Wenn $x \in B$, dann $x \in A$.

1.4.2. *Widerspruchsbeweise (indirekte Beweise).*
Um zu zeigen, dass aus einer Hypothese H die Konklusion K folgt, benutzen *Widerspruchsbeweise* das folgende Schema:

$$\frac{\dfrac{\text{Hypothese} \quad \text{Negation der Konklusion}}{\textit{Widerspruch}}}{\text{Konklusion}}$$

1.4.3. *Widerlegung durch ein Gegenbeispiel.*
Wenn Sätze *allgemeine* Aussagen behandeln, genügt es, die Aussage für bestimmte Werte zu widerlegen, um den Satz zu widerlegen. In dieser Situation haben wir dann ein *Gegenbeispiel* gefunden. Gegenbeispiele können auch verwendet werden, um allgemein gefasste Aussagen so weit einzuschränken, dass sie dann als Satz gezeigt werden können.

1.5. Induktive Beweise.
Aus der Lehrveranstaltung „Einführung in die Mathematik 1“ ist das *Prinzip der vollständigen Induktion* bekannt:

Sei m eine fest gewählte natürliche Zahl, z.B. $m = 0$ oder $m = 1$. Eine Aussage $A(n)$ soll für alle natürlichen Zahlen $n \geq m$ gezeigt werden. In diesem Fall gehen wir wie folgt vor:

- INDUKTIONSBASIS: Wir zeigen, dass A für den Basiswert m gilt.
- INDUKTIONSSCHRITT: Wir zeigen, dass für alle $n \geq m$ aus $A(n)$ auch $A(n+1)$ folgt.

Nach Satz 16 von [**5**] gilt dann $A(n)$ für alle $n \geq m$.

In einigen Beispielen sind folgende *Erweiterungen* nützlich:

- Es gibt mehrere Basiswerte
$$A(m), A(m+1), \ldots, A(l),$$
und wir setzen im Beweis des Induktionsschritts $n \geq l$ voraus. Da man $A(m)$, $A(m+1)$, …, $A(l)$ eigens zeigt und damit $A(n)$ impliziert $A(n+1)$ für $n = m, m+1, \ldots, l-1$ bewiesen ist, folgt diese Erweiterung aus der Urform durch Umgruppieren.
- Um $A(n+1)$ zu beweisen, können als Hypothesen alle Aussagen
$$A(m), A(m+1), \ldots, A(n)$$
verwendet werden. Diese Erweiterung folgt aus der Urform durch Wechsel von den Aussagen $A(n)$ zu den Aussagen
$$B(n) := A(m) \wedge A(m+1) \wedge \ldots \wedge A(n),$$
weil $B(m) = A(m)$ ist und $B(n)$ impliziert $B(n+1)$ äquivalent zu
$$A(m) \wedge A(m+1) \wedge \ldots \wedge A(n) \quad \text{impliziert} \quad A(n+1)$$
ist.

Diese Erweiterungen des Prinzips der vollständigen Induktion stellen Erweiterungen in der Anwendbarkeit des Prinzips dar, fügen aber der Beweisstärke des Prinzips nichts hinzu. Eine echte Erweiterung der Beweiskraft bringt die wohlfundierte Induktion im Abschnitt 5.

2. Wachstum von Funktionen und asymptotische Notation

Um die Größe von Datenmengen oder die Laufzeit von Algorithmen in Abhängigkeit von der Größe der Eingabe asymptotisch abzuschätzen, vergleicht man ihr Wachstum mit jenem bekannter Funktionen. Diese Funktionen haben als Definitionsbereiche eine Menge natürlicher Zahlen der Form

$$\{\ell, \ell+1, \ell+2, \ldots\}$$

mit $\ell \in \mathbb{N}$. Als Wertebereiche verwendet wir die reellen Intervalle

$$[0,\infty) := \{x \in \mathbb{R} \mid x \geq 0\} \quad \text{bzw.} \quad (0,\infty) := \{x \in \mathbb{R} \mid x > 0\}.$$

Definition 1.1 : *(asymptotische Notation)*
Sei $g : \{\ell, \ell+1, \ell+2, \ldots\} \to [0,\infty)$ mit $\ell \in \mathbb{N}$.

(1) *(Groß-O)*
Die Menge $\mathrm{O}(g)$ umfasst alle Funktionen

$$f : \{k, k+1, k+2, \ldots\} \to [0,\infty) \quad \text{mit} \quad k \in \mathbb{N}\,,$$

für die eine positive reelle Zahl c und eine natürliche Zahl m mit $m \geq k$ und $m \geq \ell$ existieren, sodass für alle natürlichen Zahlen n mit $n \geq m$

$$f(n) \leq c \cdot g(n)$$

gilt. In Kurzform ist $f \in \mathrm{O}(g)$, wenn für hinreichend große Argumente der Funktionswert von f durch ein konstantes Vielfaches des Funktionswerts von g *nach oben beschränkt* ist.

(2) *(Groß-Omega)*
Die Menge $\Omega(g)$ umfasst alle Funktionen

$$f : \{k, k+1, k+2, \ldots\} \to [0,\infty) \quad \text{mit} \quad k \in \mathbb{N}\,,$$

für die eine positive reelle Zahl c und eine natürliche Zahl m mit $m \geq k$ und $m \geq \ell$ existieren, sodass für alle natürlichen Zahlen n mit $n \geq m$

$$f(n) \geq c \cdot g(n)$$

gilt. In Kurzform ist $f \in \Omega(g)$, wenn für hinreichend große Argumente der Funktionswert von f durch ein konstantes Vielfaches des Funktionswerts von g *nach unten beschränkt* ist.

(3) *(Groß-Theta)*
Schließlich ist

$$\Theta(g) := \mathrm{O}(g) \cap \Omega(g)\,.$$

Satz 1.1 : (Limes-Kriterium für Groß-O und Groß-Omega)
Seien $f : \{k, k+1, \ldots\} \to [0, \infty)$ *und* $g : \{\ell, \ell+1, \ldots\} \to (0, \infty)$. *Dann gilt*

$$f \in \mathrm{O}(g) \;\Leftrightarrow\; \limsup_{n\to\infty} \frac{f(n)}{g(n)} < \infty$$

und

$$f \in \Omega(g) \;\Leftrightarrow\; \liminf_{n\to\infty} \frac{f(n)}{g(n)} > 0\,.$$

Beweis: Wenn $f(n) \le c \cdot g(n)$ für hinreichend große n gilt, dann ist $\limsup_{n\to\infty} \frac{f(n)}{g(n)} \le c$. Wenn umgekehrt $s := \limsup_{n\to\infty} \frac{f(n)}{g(n)} < \infty$ ist, dann gilt $\frac{f(n)}{g(n)} \le s+1$ für hinreichend große n. Details findet man in [**4**].

Beispiel 1.1 : Sei $p(n)$ eine Polynomfunktion in n vom Grad d mit Leitkoeffizient $c > 0$. Dann ist

$$\lim_{n\to\infty} \frac{p(n)}{n^d} = \lim_{n\to\infty} \frac{c \cdot n^d + c_1 \cdot n^{d-1} + \ldots + c_d}{n^d} = \lim_{n\to\infty} (c + c_1 \cdot \frac{1}{n} + \ldots + c_d \frac{1}{n^d}) = c$$

und somit $p(n) \in \Theta(n^d)$.

Definition 1.2 : *(Klein-o)*
Seien $f : \{k, k+1, \ldots\} \to [0, \infty)$ und $g : \{\ell, \ell+1, \ldots\} \to (0, \infty)$. Dann ist $f \in \mathrm{o}(g)$, wenn

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0\,,$$

d.h. asymptotisch ist f vernachlässigbar gegenüber g.

Beispiel 1.2 : In der Hierarchie bekannter Funktionen

$$1 < \log\log n < \log n < n < n\log n < n^2 < n^3 < 2^n < 3^n < n! < n^n$$

wird $f < g$ für $f \in \mathrm{o}(g)$ geschrieben.

3. Äquivalenzrelationen

Äquivalenzrelationen werden verwendet, um ähnliche Objekte in Klassen zusammenzufassen und damit Datenmengen zu reduzieren.

Definition 1.3 : *(Relation)*
Sei M eine Menge. Eine Teilmenge R von $M \times M$ heißt eine *Relation auf* M. Eine Relation R auf M heißt

- *reflexiv*, wenn für alle $x \in M \quad (x,x) \in R$ gilt,
- *irreflexiv*, wenn für kein $x \in M \quad (x,x) \in R$ gilt,

- *symmetrisch*, wenn für alle $x,y \in M$ aus $(x,y) \in R$ auch $(y,x) \in R$ folgt,
- *antisymmetrisch*, wenn für alle $x,y \in M$ aus $(x,y) \in R$ und $(y,x) \in R$ folgt, dass $x = y$ ist,
- *transitiv*, wenn für alle $x,y,z \in M$ aus $(x,y) \in R$ und $(y,z) \in R$ folgt, dass auch $(x,z) \in R$ ist.

Definition 1.4 : *(Äquivalenzrelation)*
Eine *Äquivalenzrelation* auf M ist eine reflexive, symmetrische und transitive Relation auf M.

Definition 1.5 : *(Äquivalenzklasse)*
Sei $\sim$ eine Äquivalenzrelation auf M. Elemente $x,y \in M$ heißen *äquivalent*, wenn $(x,y) \in \sim$ ist. Man schreibt dann kurz $x \sim y$. Für $x \in M$ heißt die Menge

$$\{y \in M \mid x \sim y\}$$

die *Äquivalenzklasse von x*. Die Elemente einer Äquivalenzklasse K heissen die *Repräsentanten* von K. Ein *Repräsentantensystem* von $\sim$ ist eine Teilmenge von M, die aus jeder Äquivalenzklasse genau ein Element enthält.

Beispiel 1.3 : Tupel $(x_1,x_2,\ldots,x_n)$ und $(y_1,y_2,\ldots,y_n)$ reeller Zahlen seien äquivalent, wenn sie durch Umordnen der Komponenten ineinander übergeführt werden können, d.h. wenn es eine Permutation $s \in S_n$ gibt mit

$$y_i = x_{s(i)} \qquad \text{für } i = 1,2,\ldots,n.$$

Dann ist die Menge aller monotonen Tupel

$$\{(x_1,x_2,\ldots,x_n) \in \mathbb{R}^n \mid x_1 \leq x_2 \leq \ldots \leq x_n\}$$

ein Repräsentantensystem.

Satz 1.2 : (Äquivalenz entspricht Gleichheit der Äquivalenzklassen)
Sei R eine Äquivalenzrelation auf M. Dann sind Elemente von M genau dann äquivalent, wenn ihre Äquivalenzklassen gleich sind.

Beweis: Seien $x,z \in M$. Wenn die Äquivalenzklassen von x und z gleich sind, dann ist nach Definition der Äquivalenzklasse $x \sim z$.
Sei umgekehrt $x \sim z$. Da R symmetrisch ist, ist auch $z \sim x$. Wenn y ein Element der Äquivalenzklasse von x ist, dann ist $x \sim y$. Da R transitiv ist, folgt daraus $z \sim y$, also ist y auch ein Element der Äquivalenzklasse von z. Daher ist die Äquivalenzklasse von x eine Teilmenge der Äquivalenzklasse von z. Analog zeigt man die andere Inklusion.

Satz 1.3 : *(Abbildungen liefern Äquivalenzrelationen)*
Sei $f : M \to N$ eine beliebige Abbildung. Dann wird durch

$$x \sim z \;:\Leftrightarrow\; f(x) = f(z)$$

eine Äquivalenzrelation definiert. Die Äquivalenzklassen sind die Urbildmengen

$$f^{-1}(y) = \{x \in M \mid f(x) = y\}$$

mit $y \in f(M)$.

*Anschaulich kann man sich diese Konstruktion von Äquivalenzklassen als das „*Zusammenfassen aller Objekte mit dem gleichen Merkmal*“ vorstellen.*

Beweis: Offensichtlich ist $\sim$ reflexiv, symmetrisch und transitiv. Für ein Element x aus M ist die Äquivalenzklasse von x die Menge aller Elemente aus M mit dem gleichen Bild, d.h. $f^{-1}(f(x))$.

Definition 1.6 : *(Partition)*
Sei M eine Menge. Eine *Partition* von M ist eine Menge von paarweise disjunkten nichtleeren Teilmengen von M, deren Vereinigung ganz M ist. Diese Teilmengen nennt man dann die *Blöcke* der Partition.

Satz 1.4 : (Äquivalenzrelationen und Partitionen entsprechen sich)
Sei M eine Menge.

(1) *Sei P eine Partition von M. Für Elemente x und y von M sei*

$$x \sim y \text{ genau dann, wenn } x \text{ und } y \text{ im gleichen Block von } P \text{ liegen.}$$

Dann ist $\sim$ eine Äquivalenzrelation auf M.

(2) *Sei R eine Äquivalenzrelation auf M. Dann ist die Menge P aller Äquivalenzklassen bezüglich R eine Partition von M.*

(3) *Die Abbildungen*

$$P \mapsto \sim \text{ aus (1)} \quad \text{und} \quad R \mapsto P \text{ aus (2)}$$

sind zueinander invers.

Beweis: (1) Man wendet Satz 1.3 an auf die Abbildung $f : M \to P$, die einem Element x von M jenen Block B von P zuordnet, in dem x liegt.
(2) Da R reflexiv ist, liegt jedes Element $x \in M$ in der Äquivalenzklasse von x, also ist die M die Vereinigung aller Äquivalenzklassen.
Wenn die Äquivalenzklassen von $x \in M$ und $z \in M$ nicht disjunkt sind, dann gibt es ein $y \in M$ mit

$$x \sim y \text{ und } z \sim y.$$

Da R symmetrisch ist, gilt auch $y \sim z$. Da R transitiv ist, folgt daraus $x \sim z$. Nach Satz 1.2 sind dann die Äquivalenzklassen von x und z gleich.

(3) Wenn man von einer Partition P ausgeht, die Äquivalenzrelation „im gleichen Block“ nimmt und dazu die Partition in Äquivalenzklassen bildet, dann erhält man die Partition P zurück. Wenn man umgekehrt von einer Äquivalenzrelation R ausgeht, die Partition in Äquivalenzklassen nimmt und dazu die Relation „im gleichen Block“ bildet, dann bekommt man nach Satz 1.2 die Äquivalenzrelation R zurück.

4. Partielle Ordnungen

Ordnungen erlauben, Datenmengen hierarchisch zu strukturieren.

Definition 1.7 : *(partielle Ordnung)*
Sei M eine Menge. Eine *(partielle) Ordnung* R auf M ist eine reflexive, antisymmetrische und transitive Relation auf M. In diesem Fall schreiben wir statt $(x,y) \in R$ kürzer

$$x \leq y \text{ (Sprechweise: „}x\text{ ist kleiner oder gleich }y\text{“).}$$

Wir schreiben

$$x < y \text{ (Sprechweise: „}x\text{ ist kleiner als }y\text{“),}$$

wenn $x \leq y$ und $x \neq y$ ist, und nennen x einen *Vorgänger* von y und y einen *Nachfolger* von x. Statt $x \leq y$ oder $x < y$ schreiben wir auch $y \geq x$ bzw. $y > x$. Eine Ordnung $\leq$ auf M heisst *total* (linear), wenn für je zwei verschiedene Elemente $x,y \in M$ entweder $x < y$ oder $y < x$ gilt. Wenn R eine partielle oder totale Ordnung auf einer Menge M ist und N eine Teilmenge von M ist, dann ist

$$R \cap (N \times N)$$

eine partielle bzw. totale Ordnung auf N.

Beispiel 1.4 : Die *natürliche Ordnung* auf $\mathbb{Z}$, definiert durch

$$x \leq y \text{ genau dann, wenn } y - x \in \mathbb{N} \text{ ist,}$$

ist eine totale Ordnung.

Beispiel 1.5 : Eine Zahl $m \in \mathbb{N}$ *teilt* eine Zahl $n \in \mathbb{N}$, wenn es eine Zahl $p \in \mathbb{N}$ gibt, sodass

$$n = m \cdot p.$$

Die *Teilbarkeitsordnung* auf $\mathbb{N}$ ist eine partielle, aber keine totale Ordnung auf $\mathbb{N}$.

Beispiel 1.6 : Seien $M_1, M_2, \ldots, M_k$ partiell geordnete Mengen. Für Tupel $x = (x_1, x_2, \ldots, x_k)$ und $y = (y_1, y_2, \ldots, y_k)$ in $M_1 \times M_2 \times \ldots \times M_k$ sei

$$x \leq y \quad \text{genau dann, wenn} \quad x_i \leq y_i \quad \text{für alle } i = 1, \ldots, k\,.$$

Die so definierte partielle Ordnung heißt die *komponentenweise Ordnung* auf $M_1 \times M_2 \times \ldots \times M_k$.

Definition 1.8 : *(Potenzmenge)*
Sei M eine Menge. Die Menge aller Teilmengen von M

$$\mathcal{P}(M) := \{T \mid T \subseteq M\}$$

heißt die *Potenzmenge* von M . Für $k \in \mathbb{N}$ bezeichne

$$\mathcal{P}_k(M) := \{T \mid T \subseteq M \text{ und } \#(T) = k\}$$

die Menge aller k-elementigen Teilmengen von M. Dann ist die Teilmengenrelation oder *Inklusion*

$$S \subseteq T$$

eine partielle Ordnung auf $\mathcal{P}(M)$.

Definition 1.9 : *(Verfeinerung und Vergröberung von Partitionen)*
Sei M eine Menge. Für Partitionen P und Q von M sei

$$P \leq Q$$

genau dann, wenn jeder Block von P Teilmenge eines Blocks von Q ist. In diesem Fall ist für jeden Block $T \in Q$ die Menge

$$\{S \in P \mid S \subseteq T\}$$

eine Partition von T. Wenn $P < Q$ ist, dann heißt P *feiner* als Q und Q *gröber* als P.

Satz 1.5 : (Vorgängerrelation)
Sei M eine Menge.

(1) *Wenn $\leq$ eine partielle Ordnung auf M ist, dann ist die Vorgängerrelation $<$ irreflexiv und transitiv.*
(2) *Wenn R eine irreflexive und transitive Relation auf M ist, dann wird durch*
$$x \leq y \;:\Leftrightarrow\; xRy \vee x = y$$
eine partielle Ordnung auf M definiert.
(3) *Die Abbildungen*
$$\leq \mapsto < \text{ aus (1)} \quad \text{und} \quad R \mapsto \leq \text{ aus (2)}$$
sind zueinander invers.

Eine partielle Ordnung kann daher auch über eine irreflexive und transitive Relation definiert werden.

Beweis: (1) Nach Definition gilt $x < y$ genau dann, wenn $x \leq y$ und $x \neq y$. Somit ist $<$ irreflexiv. Um die Transitivität von $<$ zu zeigen, seien $x, y, z \in M$ mit $x < y$ und $y < z$. Aus der Transitivität von $\leq$ folgt $x \leq z$. Wegen der Antisymmetrie von $\leq$ kann x nicht gleich z sein. Daher ist $x < z$.
(2) Nach Definition ist $\leq$ reflexiv. Um die Antisymmetrie von $\leq$ zu zeigen, seien $x, y \in M$ mit $x \leq y$ und $y \leq x$. Wenn $x \neq y$, dann wäre xRy und yRx,

wegen der Transitivität von R somit xRx, was der Irreflexivität von R widerspricht. Daher muss $x = y$ sein. Um die Transitivität von $\leq$ zu zeigen, seien $x, y, z \in M$ mit $x \leq y$ und $y \leq z$. Wenn $x = y$ oder $y = z$ ist, dann folgt $x \leq z$. Andernfalls ist xRy und yRz, wegen der Transitivität von R somit xRz und $x \leq z$.
(3) Wenn man von einer partiellen Ordnung $\leq$ ausgeht, dann bekommt man durch $x < y \vee x = y$ die partielle Ordnung $x \leq y$ zurück. Wenn man von einer irreflexiven und transitiven Relation R ausgeht, dann bekommt man durch $x \leq y \wedge x \neq y$ die Relation R zurück.

Beispiel 1.7 : Die Relation $f \in \mathrm{o}(g)$ ist irreflexiv und transitiv und kann daher suggestiv als $f < g$ wie in Beispiel 1.2 geschrieben werden.

Definition 1.10 : *(kleinste, größte, minimale, maximale Elemente)*
Sei $\leq$ eine partielle Ordnung auf einer Menge M. Ein Element $x \in M$ heisst

- *kleinstes Element* von M , falls x kleiner als alle anderen Elemente von M ist, d.h. für alle $y \in M$ mit $y \neq x$ ist $x < y$,
- *größtes Element* von M , falls x größer als alle anderen Elemente von M ist, d.h. für alle $y \in M$ mit $y \neq x$ ist $y < x$,
- *minimales Element* von M, falls kein anderes Element von M kleiner als x ist, d.h. für alle $y \in M$ mit $y \neq x$ ist nicht $y < x$,
- *maximales Element* von M , falls kein anderes Element von M größer als x ist, d.h. für alle $y \in M$ mit $y \neq x$ ist nicht $x < y$.

Beispiel 1.8 : Sei $\leq$ eine totale Ordnung auf M und sei $x \in M$. Dann ist x kleinstes Element von M genau dann, wenn x minimales Element von M ist. Analog ist x größtes Element von M genau dann, wenn x maximales Element von M ist.

Satz 1.6 : (über kleinste, größte, minimale, maximale Elemente)
Sei $\leq$ eine partielle Ordnung auf einer Menge M .

(1) *Wenn ein kleinstes Element von M existiert, dann ist es eindeutig bestimmt und das einzige minimale Element von M .*
(2) *Wenn ein größtes Element von M existiert, dann ist es eindeutig bestimmt und das einzige maximale Element von M .*
(3) *Wenn M endlich ist, dann gibt es zu jedem Element $x \in M$ ein minimales Element $w \in M$ mit $w \leq x$ und ein maximales Element $z \in M$ mit $x \leq z$.*
(4) *Wenn M endlich ist und nur ein minimales Element besitzt, dann ist dieses Element das kleinste Element von M.*
(5) *Wenn M endlich ist und nur ein maximales Element besitzt, dann ist dieses Element das größte Element von M.*

Beweis: (1) Wenn sowohl x als auch w kleinste Elemente von M sind, dann ist $w \leq x \leq w$ und somit $w = x$. Wenn x das kleinste Element von M ist und y ein beliebiges Element von M mit $y \leq x$ ist, dann ist $y \leq x \leq y$ und somit $y = x$.
(2) beweist man analog.
(3) Wir zeigen nur die Existenz eines minimalen Elements: Wenn x selbst minimal ist, kann man $w = x$ wählen. Andernfalls gibt es ein $x_1 \in M$ mit $x_1 < x$. Wenn x_1 nicht minimal ist, dann gibt es ein $x_2 \in M$ mit $x_2 < x_1$, usw. Da

$$x > x_1 > x_2 > \dots$$

verschiedene Elemente von M sind, erreicht man nach endlich vielen Schritten ein minimales Element x_n mit $x_n < x$.
(4) und (5) folgen aus (3).

5. Die wohlfundierte Induktion

Definition 1.11 : *(wohlfundierte Menge)*
Sei $\leq$ eine partielle Ordnung auf einer Menge M. Eine Folge $(x_0, x_1, x_2, \dots)$ von Elementen in M heißt eine *unendliche absteigende Kette*, falls

$$x_0 > x_1 > x_2 > \dots .$$

Man nennt $\leq$ *wohlfundiert*, wenn es keine unendlichen absteigenden Ketten in M gibt.

Beispiel 1.9 : Sei $f : M \to \mathbb{N}$ eine beliebige Abbildung. Dann wird durch

$$x < y \quad :\Leftrightarrow \quad f(x) < f(y)$$

eine wohlfundierte Ordnung auf M definiert.

Satz 1.7 : (Existenz minimaler Elemente)
Sei $\leq$ eine partielle Ordnung auf einer Menge M. Dann ist $\leq$ wohlfundiert genau dann, wenn jede nichtleere Teilmenge von M ein minimales Element besitzt.

Beweis: Sei $\leq$ wohlfundiert und sei N eine nichtleere Teilmenge von M. Dann gibt es ein Element x_0 in N. Wenn x_0 minimal in N ist, ist man fertig. Andernfalls gibt es ein Element $x_1 \in N$ mit $x_1 < x_0$. Wenn x_1 nicht minimal ist, dann gibt es ein $x_2 \in N$ mit $x_2 < x_1$, usw. Wegen

$$x_0 > x_1 > x_2 > \dots$$

erreicht man nach endlich vielen Schritten ein minimales Element x_n.

Um die umgekehrte Richtung zu beweisen, nehmen wir an, M sei nicht wohlfundiert. Dann gibt es eine unendliche absteigende Kette

$$x_0 > x_1 > x_2 > \dots,$$

und die nichtleere Teilmenge $N = \{x_0, x_1, x_2, \dots\}$ hat kein minimales Element.

Satz 1.8 : (Grundlage der wohlfundierten Induktion)
Sei $\leq$ eine wohlfundierte Ordnung auf einer Menge M, und sei W eine nichtleere Teilmenge von M mit folgenden zwei Eigenschaften:

- *W enthält alle minimalen Elemente von M.*
- *Wenn für ein nicht-minimales Element x in M alle Vorgänger in W liegen, dann liegt auch x in W.*

Dann ist $W = M$.

Beweis: Wir führen einen Widerspruchsbeweis und nehmen an, dass $W \neq M$ ist. Dann ist die Menge

$$N := \{x \in M \mid x \notin W\}$$

nichtleer und hat somit ein minimales Element y. Nach der ersten Eigenschaft von W ist y kein minimales Element von M. Da alle Vorgänger von y in W liegen, folgt nach der zweiten Eigenschaft von W der Widerspruch $y \in W$.

Folgerung : (wohlfundierte Induktion)
Sei $\leq$ eine wohlfundierte Ordnung auf einer Menge M. Eine Aussage $A(x)$ soll für alle Elemente x in M gezeigt werden. In diesem Fall gehen wir wie folgt vor:

- INDUKTIONSBASIS*: Wir zeigen, dass $A(m)$ wahr ist für alle minimalen Elemente m von M.*
- INDUKTIONSSCHRITT*: Sei x ein nicht-minimales Element von M, und sei $A(y)$ wahr für alle Vorgänger y von x. Wir zeigen, dass auch $A(x)$ wahr ist.*

Nach Satz 1.8 ist die Menge W aller Elemente x, für die $A(x)$ wahr ist, ganz M.

6. Das Wortmonoid

Definition 1.12 : *(Alphabet)*
Sei Σ eine beliebige Menge, die im Folgenden ein *Alphabet* und deren Elemente *Zeichen* genannt werden. Wir verwenden üblicherweise lateinische Buchstaben vom Anfang des Alphabets, um beliebige Elemente des Alphabets zu notieren.

Beispiel 1.10 :

- $\mathbb{B} = \{0,1\}$ ist das *binäre* Alphabet
- $\{a,b,\ldots,z\}$ ist das Alphabet aller (lateinischen) Kleinbuchstaben
- $\{\ ,!,\texttt{"},\#,\$,\%,\&,\ldots,\tilde{}\}$ ist das Alphabet der druckbaren ASCII-Zeichen (siehe Anhang).

Satz 1.9 : *(Wortmonoid)*
Sei Σ ein Alphabet. Dann heißt ein Tupel

$$(w_0,\ldots,w_{n-1})\ ,$$

wobei $n \in \mathbb{N}$ beliebig ist und $w_0,\ldots,w_{n-1} \in \Sigma$ sind, ein Wort *(eine Zeichenkette, ein String) der Länge n über dem Alphabet Σ . Üblicherweise schreiben wir kleine griechische Buchstaben oder Buchstaben vom Ende des Alphabets, um Wörter zu notieren. Bezeichne*

$$\Sigma^*$$

die Menge aller Wörter über dem Alphabet Σ . Für Wörter

$$v = (v_0,\ldots,v_{m-1}) \in \Sigma^* \qquad \textit{und} \qquad w = (w_0,\ldots,w_{n-1}) \in \Sigma^*$$

ist die Verkettung *oder* Konkatenation *definiert als das Wort*

$$vw := (v_0,\ldots,v_{m-1},w_0,\ldots,w_{n-1}) \in \Sigma^*\ .$$

Dann gilt für Wörter $u,v,w \in \Sigma^*$ das Assoziativgesetz

$$(uv)w = u(vw)\,,$$

und das leere Wort $\varepsilon = ()$ *ist das neutrale Element:*

$$w\varepsilon = \varepsilon w = w\,.$$

Die Menge Σ^* mit der Verkettung wird das Wortmonoid *über dem* Alphabet Σ *genannt. Für die* Längenfunktion

$$\ell : \Sigma^* \to \mathbb{N}\ ,\ (w_0,\ldots,w_{n-1}) \mapsto n\ ,$$

gilt

$$\ell(vw) = \ell(v) + \ell(w) \quad \textit{und} \quad \ell(\varepsilon) = 0\ .$$

Üblicherweise lässt man in den Wörtern die Klammern und Beistriche weg, weil keine Verwechslungsgefahr besteht. Insbesondere werden Wörter der Länge 1 wie Zeichen des Alphabets geschrieben.

Beweis: Wörter sind Tupel und damit Funktionen. Die angeführten Wörter sind gleich, weil sie als Funktionen gleich sind.

Beispiel 1.11 : 01101 ist ein Wort über dem Alphabet $\{0,1\}$ der Länge 5. Für $x = 01101$, $y = 110$ und $z = 10101$ sind

$$\begin{aligned} xy &= 01101110 \\ yx &= 11001101 \\ (xy)z &= (01101110)10101 = 0110111010101 \\ x(yz) &= 01101(11010101) = 0110111010101\,. \end{aligned}$$

Satz 1.10 : (lexikographische Ordnung)
Sei $\leq$ eine totale Ordnung auf Σ . Für Wörter $v, w \in \Sigma^*$ sei

$$v <_{lex} w\,,$$

falls ein $k \in \mathbb{N}$ mit $k \leq \ell(v)$ und $k \leq \ell(w)$ existiert, sodass

(1) $v_i = w_i$ *für* $i = 0, \ldots, k-1$ *und*

(2) $(\ell(v) = k$ *und* $\ell(w) > k)$ *oder*
$(\ell(v) > k$ *und* $\ell(w) > k$ *und* $v_k < w_k)$

ist. Dann ist $\leq_{lex}$ eine totale Ordnung auf Σ^* und heißt lexikographische Ordnung.

Beweis: Offensichtlich ist $<_{\text{lex}}$ irreflexiv. Um die Transitivität zu zeigen, seien $u, v, w \in \Sigma^*$ mit

$$u < v \qquad \text{und} \qquad v < w\,.$$

Dann gibt es ein $k \in \mathbb{N}$ mit $k \leq \ell(u)$ und $k \leq \ell(v)$ und

(1) $u_i = v_i$ für $i = 0, \ldots, k-1$ und

(2) $(\ell(u) = k$ und $\ell(v) > k)$ oder
$(\ell(u) > k$ und $\ell(v) > k$ und $u_k < v_k)$

sowie ein $l \in \mathbb{N}$ mit $l \leq \ell(v)$ und $l \leq \ell(w)$ und

(1) $v_i = w_i$ für $i = 0, \ldots, l-1$ und

(2) $(\ell(v) = l$ und $\ell(w) > l)$ oder
$(\ell(v) > l$ und $\ell(w) > l$ und $v_l < w_l)$.

Dann gilt für $m := \min(k, l)$ auch $m \leq \ell(u)$ und $m \leq \ell(w)$ und

(a) $u_i = w_i$ für $i = 0, \ldots, m-1$ und

(b) $(\ell(u) = m$ und $\ell(w) > m)$ oder
$(\ell(u) > m$ und $\ell(w) > m$ und $u_m < w_m)$,

sodass $u <_{\text{lex}} w$ folgt. Um zu zeigen, dass $\leq_{\text{lex}}$ eine totale Ordnung ist, seien $v, w \in \Sigma^*$ mit $v \neq w$. Dann existiert ein $k \in \mathbb{N}$ mit $k \leq \ell(v)$ und $k \leq \ell(w)$, sodass

(a) $v_i = w_i$ für $i = 0, \ldots, k-1$ und

(b) $(\ell(v) = k$ und $\ell(w) > k)$ oder $(\ell(v) > k$ und $\ell(w) = k)$ oder
$(\ell(v) > k$ und $\ell(w) > k$ und $v_k \neq w_k)$

ist. Da $\leq$ eine totale Ordnung auf Σ ist, gilt somit entweder $v <_{\text{lex}} w$ oder $w <_{\text{lex}} v$.

Beispiel 1.12 : In den meisten Programmiersprachen sind die Zeichen nach dem ASCII-Code (siehe Anhang) total geordnet und die Zeichenketten nach der lexikographischen Ordnung, z.B. mit der Funktion `strcmp` von C.

Lemma 1.1 : (graduiert-lexikographische Ordnung auf Wörtern)
Sei $\leq$ eine totale Ordnung auf Σ. Für Wörter $v, w \in \Sigma^*$ sei

$$v <_{gradlex} w \ ,$$

falls entweder $\ell(v) < \ell(w)$ oder $(\ell(v) = \ell(w)$ und $v <_{lex} w)$ ist. Dann ist $\leq_{gradlex}$ eine totale Ordnung auf Σ^* und heißt graduiert-lexikographische Ordnung.

Beweis: Man prüft die Irreflexivität, Transitivität und Totalität nach.

Definition 1.13 : *(formale Sprache)*
Sei Σ eine beliebige Menge. Eine Teilmenge von Σ^* heißt eine *formale Sprache* über dem *Alphabet* Σ .

Beispiel 1.13 : Die formale Sprache der Palindrome über dem Alphabet $\Sigma = \{a, b\}$ ist

$$\{w_0 w_1 \ldots w_{n-1} \mid w_0 w_1 \ldots w_{n-1} = w_{n-1} w_{n-2} \ldots w_0\} = \\ \{\varepsilon, a, b, aa, bb, aaa, aba, bab, bbb, aaaa, abba, baab, bbbb, \ldots\} \ .$$

Beispiel 1.14 : Für Alphabete mit mindestens zwei Buchstaben ist die lexikographische Ordnung nicht wohlfundiert, weil

$$b >_{\text{lex}} ab >_{\text{lex}} aab >_{\text{lex}} aaab >_{\text{lex}} \ldots$$

eine unendliche absteigende Kette ist. Hingegen ist die graduiert-lexikographische Ordnung auf einem endlichen Alphabet wohlfundiert.

Satz 1.11 : (lex auf $\mathbb{N}^k$ wohlfundiert)
Sei $\mathbb{N}$ mit der natürlichen Ordnung versehen. Dann ist die lexikographische Ordnung auf der Menge $\mathbb{N}^k$ wohlfundiert.

Beweis: Wir führen eine Induktion über k. Die Menge $\mathbb{N}$ ist wohlfundiert. Sei

$$x_1 >_{\text{lex}} x_2 >_{\text{lex}} \ldots$$

eine unendliche absteigende Kette in $\mathbb{N}^{k+1}$. Die Menge der ersten Komponenten der x_i hat ein kleinstes Element $m = (x_n)_1$. Dann ist

$$x_n >_{\text{lex}} x_{n+1} >_{\text{lex}} \cdots$$

eine unendliche absteigende Kette in $\mathbb{N}^{k+1}$ mit konstanter erster Komponente m, was der Wohlfundiertheit von $\mathbb{N}^k$ widerspricht.

Beispiel 1.15 : Die lexikographische Ordnung auf der Menge $\mathbb{N}^2$ ist wohlfundiert, obwohl z.B. das Paar $(1,0)$ unendlich viele Vorgänger $(0,n)$ hat. Daher bricht die Rekursion

$$f(n,m) := \begin{cases} m+1 & \text{falls } n = 0 \\ f(n-1,1) & \text{falls } n > 0 \text{ und } m = 0 \\ f(n-1, f(n,m-1)) & \text{sonst} \end{cases}$$

nach endlich vielen Schritten ab. Die Funktion

$$A(n) := f(n,n)$$

heisst *Ackermannfunktion* und wächst sehr schnell.

7. Strukturelle Induktion

In der theoretischen Informatik ist man weniger an Induktion über natürliche Zahlen interessiert, sondern mehr an Induktion über Strukturen wie etwa Aussagen, arithmetische Ausdrücke oder Bäume.

Definition 1.14 : *(induktive Definition einer Menge M)*

- BASIS: Man gibt ein oder mehr Elemente von M an.
- SCHRITT: Man spezifiziert, wie man neue Elemente von M aus den vorliegenden Elementen von M bekommt.

Die Menge M besteht dann aus genau jenen Elementen, die man durch Induktionsbasis und ein- oder mehrmalige Anwendung des Induktionsschritts erhält.

Beispiel 1.16 : (Syntax der Aussagenlogik)
Seien $E_1, E_2, E_3, \ldots$ Aussagen, die entweder wahr oder falsch sind. Diese Aussagen werden *atomare Aussagen* genannt. Die Menge aller Aussagen ist dann als formale Sprache mit Hilfe der *logischen Symbole*

- $\neg$
- $\wedge$
- $\vee$

und der *Trennzeichen*

- (

-)

induktiv definiert:

(1) Die atomaren Aussagen $E_1, E_2, \ldots$ sind Aussagen.
(2) Ist A eine Aussage, so ist auch $\neg A$ eine Aussage. Die Aussage $\neg A$ heißt die *Negation* von A.
(3) Sind A und B Aussagen, so sind auch $(A \vee B)$ und $(A \wedge B)$ Aussagen. Die Aussage $(A \vee B)$ heißt die *Disjunktion* von A und B, die Aussage $(A \wedge B)$ die *Konjunktion* von A und B.

Satz 1.12 : (Prinzip der strukturellen Induktion)
Die Aussage $A(x)$ soll für alle Strukturen $x \in M$, die induktiv definiert sind, gezeigt werden. In diesem Fall gehen wir wie folgt vor:

- INDUKTIONSBASIS*: Wir zeigen, dass $A(x)$ für die Basisstruktur(en) x gilt.*
- INDUKTIONSSCHRITT*: Wir wählen eine Struktur y, die rekursiv aus Strukturen $y_1, y_2, \ldots, y_k$ gebildet wird. Unsere Induktionshypothese ist, dass $A(y_1), A(y_2), \ldots, A(y_k)$ wahr sind. Mit Hilfe der Induktionshypothese zeigen wir nun $A(y)$.*

Beweis: Wir definieren eine Abbildung $f : M \to \mathbb{N}$, welche die maximale Anzahl der Schritte zählt:

- BASIS: Für alle Basiselemente $x \in M$ sei $f(x) = 0$.
- SCHRITT: Wenn y aus Strukturen $y_1, y_2, \ldots, y_k$ gebildet wird, sei

$$f(y) := \max\{f(y_1), f(y_2), \ldots, f(y_k)\} + 1 .$$

Dann wird durch $x < y :\Leftrightarrow f(x) < f(y)$ eine wohlfundierte Ordnung auf M definiert, und wir können die strukturelle Induktion durch eine wohlfundierte Induktion beweisen. Dazu nehmen wir an, dass Basis und Schritt der strukturellen Induktion erfüllt sind.

Nach Konstruktion sind die minimalen Elemente genau die Basiselemente von M, sodass auch die Basis der wohlfundierten Induktion erfüllt ist. Für den Schritt der wohlfundierten Induktion sei y nicht minimal mit $A(z)$ wahr für alle $z < y$. Dann wird y aus Strukturen $y_1, y_2, \ldots, y_k$ gebildet. Nach Definition von f gilt $y_i < y$ für alle i. Somit sind alle Aussagen $A(y_i)$ wahr. Nach dem Schritt der strukturellen Induktion ist auch $A(z)$ wahr. Aus der wohlfundierten Induktion folgt schließlich die Gültigkeit der Aussagen $A(x)$ für alle $x \in M$.

Beispiel 1.17 : Für die zusammengesetzte Aussage

$$A = ((E_1 \wedge E_2) \vee \neg E_2)$$

ergeben sich die Funktionswerte

$$f(E_1) = 0 \,,\; f(E_2) = 0 \,,\; f(E_1 \wedge E_2) = 1 \,,\; f(\neg E_2) = 1 \quad \text{und} \quad f(A) = 2 \,.$$

KAPITEL 2

Graphentheorie

1. Gerichtete Graphen

Definition 2.1 : *(gerichteter Multigraph)*
Ein *gerichteter Multigraph* G ist gegeben durch

- eine *Eckenmenge* (oder *Knotenmenge*) E
- eine *Kantenmenge* K
- zwei Abbildungen

$$q : K \to E \quad \text{und} \quad z : K \to E\,,$$

  die jeder Kante k ihre *Anfangsecke* $q(k)$ bzw. ihre *Endecke* $z(k)$ zuordnen (q für Quelle, z für Ziel). Man nennt dann k eine Kante von $q(k)$ nach $z(k)$.

Eine Ecke c heißt *unmittelbarer Vorgänger* der Ecke d , wenn es eine Kante von c nach d gibt. Man nennt d dann *unmittelbarer Nachfolger* von c. *Schleifen* sind Kanten mit der gleichen Anfangs- wie Endecke. Kanten mit den gleichen Anfangsecken und den gleichen Endecken heißen *parallel*. Für eine Ecke e heißt die Zahl der Kanten mit Endecke e der *Eingangsgrad* von e und die Zahl der Kanten mit Anfangsecke e der *Ausgangsgrad* von e. Wenn zusätzlich Abbildungen

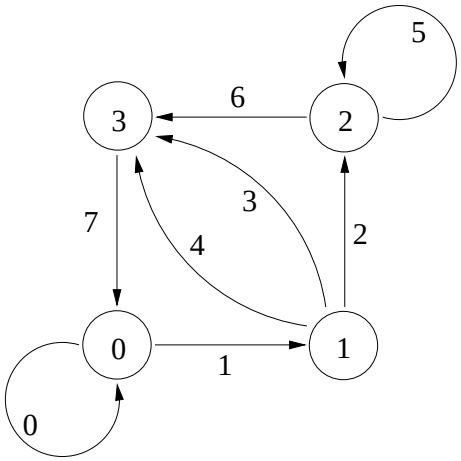
$$a : E \to M \quad \text{oder} \quad b : K \to N$$

gegeben werden, dann heißt der Multigraph *ecken-* bzw. *kantenbeschriftet*, im Spezialfall $M = \mathbb{R}$ oder $N = \mathbb{R}$ *ecken-* bzw. *kantenbewertet*.

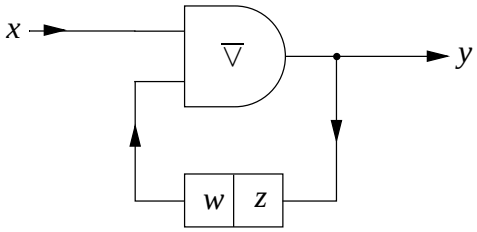
Beispiel 2.1 : Sei ein gerichteter Multigraph gegeben durch die Eckenmenge $E = \{0,1,2,3\}$, die Kantenmenge $K = \{0,1,2,\ldots,7\}$ und die Abbildungen q und z laut folgender Tabelle:

k	$q(k)$	$z(k)$
0	0	0
1	0	1
2	1	2
3	1	3
4	1	3
5	2	2
6	2	3
7	3	0

Visualisiert kann dieser Graph durch folgendes Bild werden:



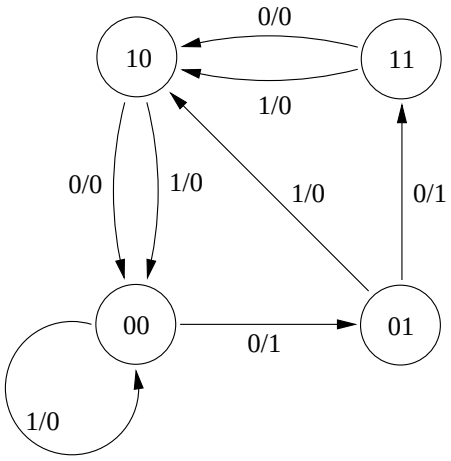
Beispiel 2.2 : Im synchronen Schaltwerk mit Eingang x, Ausgang y, einem Nor-Gatter und einem Schieberegister der Länge 2



lauten die Gleichungen für die binären Signalfolgen

$$\begin{aligned} y(t) &= x(t)\,\overline{\vee}\,w(t) \\ w(t+1) &= z(t) \\ z(t+1) &= y(t)\,, \end{aligned}$$

wobei $t \in \mathbb{N}$ eine diskrete Zeit ist. Eine anschauliche Beschreibung des Schaltverhaltens liefert das Zustandsdiagramm:



Das Zustandsdiagramm ist ein gerichteter Multigraph mit der Eckenmenge $E = \mathbb{B}^2$ und einer Kantenmenge $K \subset \mathbb{B}^6$, wobei $e = (e_0, e_1)$ den Inhalt des Schieberegisters beschreibt und

$$k = (k_0, k_1, k_2, k_3, k_4, k_5)$$

den momentanen Zustand (k_0, k_1), die Eingabe k_2, die Ausgabe k_3 und den Folgezustand (k_4, k_5) angibt. Dann sind Quelle und Ziel

$$q(k) = (k_0, k_1) \quad \text{bzw.} \quad z(k) = (k_4, k_5)\,,$$

und

$$b(k) = (k_2, k_3)$$

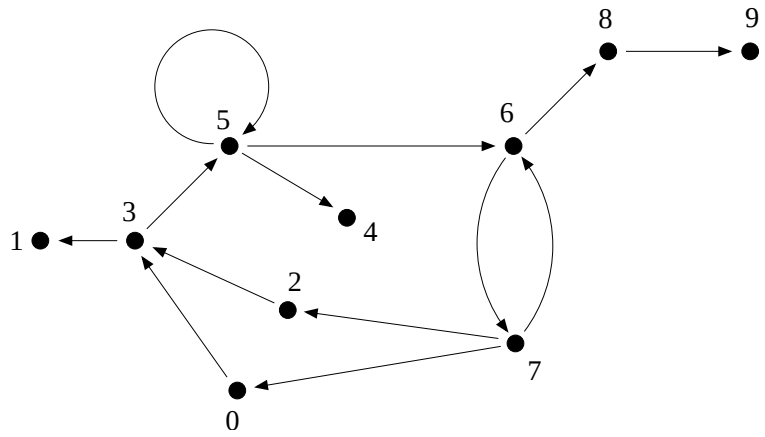
ist die Beschriftung der Kanten durch Eingabe und Ausgabe.

Definition 2.2 : *(gerichteter Graph)*
Ein *gerichteter Graph* (oder *Digraph* für directed graph) ist ein gerichteter Multigraph ohne parallele Kanten. Dann gibt es zu jedem Eckenpaar (c, d) höchstens eine Kante $k \in K$ mit $q(k) = c$ und $z(k) = d$, und statt der abstrakten Kante k wird häufig das Eckenpaar (c, d) genommen.

Beispiel 2.3 : Sei R eine Relation auf einer Menge M . Dann ist der *gerichtete Graph der Relation* gegeben durch

- die Eckenmenge M
- die Kantenmenge R
- die Abbildungen $q((x,y)) = x$ und $z((x,y)) = y$.

Offenbar ist jeder gerichtete Graph der Graph einer Relation.



Definition 2.3 : *(Teilmultigraph,Teilgraph)*
Sei $G = (E, K, q, z)$ ein gerichteter Multigraph. Ein gerichteter Multigraph $G' = (E', K', q', z')$ heißt *Teilmultigraph* von G , wenn $E' \subseteq E$, $K' \subseteq K$ und

$$q'(k) = q(k) \quad \text{und} \quad z'(k) = z(k) \quad \text{für alle } k \in K'\,.$$

Ein *Teilgraph* ist ein *Teilmultigraph*, der selbst Graph ist.

Definition 2.4 : *(Wege, starker Zusammenhang, Zyklen, Wurzelbäume, Blätter)* Sei (E,K,q,z) ein gerichteter Multigraph, und seien c,d Ecken. Ein Tupel

$$(k_0,k_1,\ldots,k_{\ell-1}) \in K^\ell$$

heißt ein *Weg* von c nach d der Länge ℓ, wenn es Ecken $e_0,e_1,\ldots,e_\ell$ gibt mit $e_0 = c,\ e_\ell = d$, und

$$q(k_i) = e_i \quad \text{sowie} \quad z(k_i) = e_{i+1} \quad \text{für } i = 0,1,\ldots,\ell-1\ .$$

Die Ecken $e_0,e_1,\ldots,e_\ell$ sind dann eindeutig bestimmt, und man nennt e_0 die *Anfangsecke*, e_ℓ die *Endecke* sowie $e_1,e_2,\ldots,e_{\ell-1}$ die *Zwischenecken* des Weges. Für jede Ecke $e \in E$ wird das leere Tupel $() \in K^0$ der *leere Weg* mit *Anfangsecke* e und *Endecke* e genannt.

Der gerichtete Multigraph heißt *stark zusammenhängend*, wenn es von jeder Ecke zu jeder anderen Ecke einen Weg gibt. Ein Weg heißt *einfach*, wenn er nichtleer ist und die Ecken

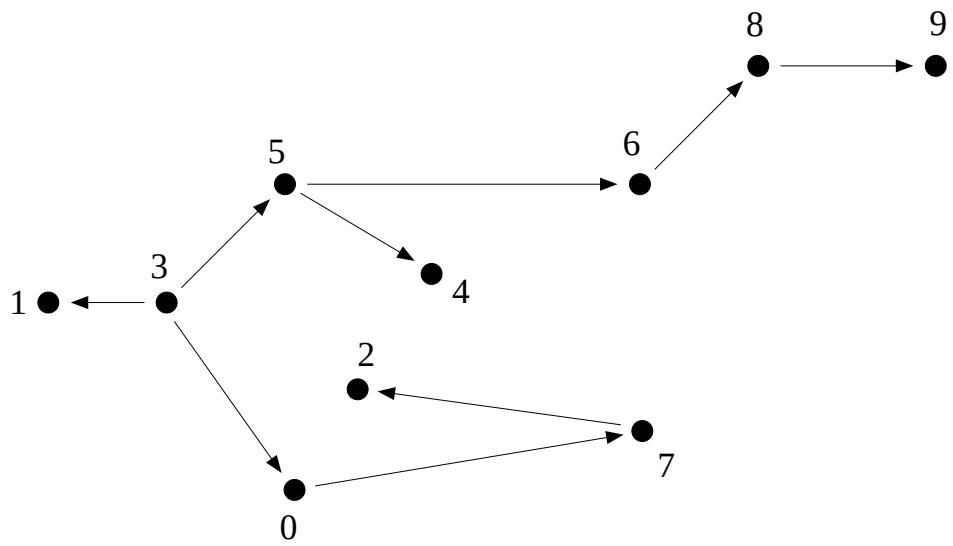
$$e_0,e_1,\ldots,e_\ell$$

paarweise verschieden sind bis auf die mögliche Ausnahme $e_0 = e_\ell$. Wenn $(k_0,k_1,\ldots,k_{\ell-1})$ ein Weg von c nach d ist und $q = (h_0,h_1,\ldots,h_{m-1})$ ein Weg von d nach e ist, dann ist die *Verkettung*

$$(k_0,k_1,\ldots,k_{\ell-1},h_0,h_1,\ldots,h_{m-1})$$

ein Weg von c nach e. Ein Weg $(k_0,k_1,\ldots,k_{\ell-1})$ heißt *geschlossen*, wenn Anfangs- und Endecke gleich sind. Ein nichtleerer geschlossener Weg mit paarweise verschiedenen Kanten wird ein *Zykel* genannt. Der gerichteter Multigraph heißt *zyklenfrei* oder *azyklisch*, wenn es keine Zyklen gibt.

Ein *Wurzelbaum* ist ein gerichteter Graph, in dem es eine Ecke gibt, von der zu jeder Ecke genau ein Weg führt. Diese Ecke ist dann eindeutig bestimmt und heißt die *Wurzel* des Baumes. Insbesondere ist jeder Wurzelbaum zyklenfrei. Ecken eines Wurzelbaums vom Ausgangsgrad 0 nennt man *Blätter*.



Lemma 2.1 : (nichtleere Wege enthalten einfache Wege)
Sei G ein gerichteter Multigraph.

(1) *Wenn es einen nichtleeren Weg p von der Ecke c zur Ecke d gibt, dann kann man aus p durch Weglassen von Kanten einen einfachen Weg von c nach d erhalten.*
(2) *Jeder einfache geschlossene Weg ist ein Zykel.*

Beweis: (1) Seien $e_0, e_1, \ldots, e_\ell$ die Ecken des Weges $(k_0, k_1, \ldots, k_{\ell-1})$ von c nach d. Wenn es Indizes i, j mit $i < j$ und $e_i = e_j$ gibt, dann führt auch der verkürzte Weg

$$(k_0, k_1, \ldots, k_{i-1}, k_j, \ldots, k_{\ell-1})$$

von c nach d. Wenn $i > 0$ oder $j < \ell$ ist, dann ist der verkürzte Weg nichtleer. Nach endlich vielen Verkürzungen erhält man einen einfachen Weg von e nach d.
(2) Sei p ein einfacher Weg von e nach e. Wenn zwei Kanten gleich sind, dann wären auch ihre Anfangsecken gleich, was der Einfachheit von p widerspricht.

Satz 2.1 : (Adjazenzmatrix)
Sei (E, K, q, z) ein gerichteter Multigraph mit endlicher Ecken- und Kantenmenge. Wir numerieren die Ecken als $e_0, e_1, \ldots, e_{n-1}$. Dann heißt die Matrix $A \in \mathbb{Z}^{n \times n}$,

$$A_{ij} := \#(\{k \in K \mid q(k) = e_i \text{ und } z(k) = e_j\}) \quad \text{für } i, j = 0, 1, \ldots, n-1,$$

die Adjazenzmatrix *(oder* Nachbarschaftsmatrix*) des Multigraphen. Für $i, j = 0, 1, \ldots, n-1$ und $\ell \in \mathbb{N}$ gibt*

$$(A^\ell)_{ij}$$

die Anzahl der Wege von e_i nach e_j der Länge ℓ an.

Beweis: Für $\ell = 0$ ist

$$A^\ell = I_n$$

und es gibt nur den leeren Weg. Für $\ell = 1$ erhalten wir die Definition. Für $\ell > 1$ ist

$$(A^\ell)_{ij} = \sum_{r=0}^{n-1} (A^{\ell-1})_{ir} \cdot A_{rj}.$$

Nach Induktionannahme zählt $(A^{\ell-1})_{ir}$ alle Wege von e_i nach e_r der Länge $\ell - 1$, und A_{rj} zählt alle Kanten von e_r nach e_j. Somit zählt die Summe alle Wege von e_i nach e_j der Länge ℓ.

Satz 2.2 : (transitive Hülle)
Sei R eine Relation auf einer Menge M und sei G der gerichtete Graph von R. Dann ist

$$T := \{(x,y) \in M^2 \mid \textit{es gibt einen einfachen Weg von } x \textit{ nach } y\}$$

die kleinste transitive Relation, die R enthält, und heisst die transitive Hülle *von R.*

Beweis: Offensichtlich ist die Relation T transitiv und enthält die Relation R. Sei S eine transitive Relation mit $R \subseteq S$. Wenn $(x,y) \in T$ ist, dann gibt es einen einfachen Weg in G von x nach y mit Zwischenecken $z_1, z_2, \ldots, z_{\ell-1}$, somit gilt

$$(x,z_1) \in R,\ (z_1,z_2) \in R, \ldots,\ (z_{\ell-1},y) \in R,$$

daher auch

$$(x,z_1) \in S,\ (z_1,z_2) \in S, \ldots,\ (z_{\ell-1},y) \in S,$$

und wegen der Transitivität von S auch $(x,y) \in S$.

Satz 2.3 : (Algorithmus von Warshall)
Sei R eine Relation auf einer Menge M mit n Elementen und sei A die Adjazenzmatrix von R. Der folgende Algorithmus mit $\mathrm{O}(n^3)$ Bitoperationen überschreibt A mit der Adjazenzmatrix der transitiven Hülle von R.

Für r von 0 bis $n-1$ wiederhole:
  Setze $N = A$.
  Für i von 0 bis $n-1$ wiederhole:
    Für j von 0 bis $n-1$ wiederhole:
      Falls $A_{ij} = 0$ und $A_{ir} = 1$ und $A_{rj} = 1$, setze $N_{ij} = 1$.
  Setze $A = N$.

Beweis: Für $r \in \{0,1,\ldots,n\}$ sei P_r die Menge aller einfachen Wege im Graphen von R, die nur Zwischenecken aus der Menge $\{x_0, x_1, \ldots, x_{r-1}\}$ haben. Dann ist

- P_0 die Menge aller Kanten von G, und
- P_n ist die Menge aller einfachen Wege in G.

Sei nun $r < n$. Für einen Weg p in P_{r+1} gibt es zwei Fälle:

- x_r ist keine Zwischenecke von p. Dann ist p in P_r.
- x_r ist eine Zwischenecke von p. Dann kann der Weg p von e nach d als Verkettung eines Weges u von e nach x_r und eines Weges v von x_r nach d geschrieben werden, die beide in P_r liegen.

Für $r = 0, 1, \ldots, n$ sei

$$R_r := \{(x,y) \in M^2 \mid \text{es gibt einen Weg in } P_r \text{ von } x \text{ nach } y\}\,.$$

Dann ist $R_0 = R$ und R_n ist die transitive Hülle von R. Der Algorithmus von Warshall berechnet für $r = 0, 1, \ldots, n-1$ aus der Adjazenzmatrix von R_r die Adjazenzmatrix von R_{r+1}.

Beispiel 2.4: Für die Relation $R = \{(0,2),(1,0),(2,1)\}$ ist die transitive Hülle $T = \{(0,0),(0,1),(0,2),(1,0),(1,1),(1,2),(2,0),(2,1),(2,2)\}$.

Definition 2.5: *(Länge von Wegen, Abstand von Ecken)*
Sei G ein gerichteter Multigraph mit einer nichtnegativen Kantenbewertung b. Die Länge eines Weges $(k_0, k_1, \ldots, k_{\ell-1})$ bezüglich b ist die Summe der Bewertungen seiner Kanten k_i (Somit ergibt sich bei Einheitsbewertung der Kanten die Länge ℓ). Der Abstand von Ecken e und d ist die kleinste Länge eines Weges von e nach d, falls ein solcher existiert, und ∞ sonst.

Satz 2.4: (Algorithmus von Floyd)
Sei G ein gerichteter Multigraph mit einer endlichen Eckenmenge E, einer endlicher Kantenmenge K und einer nichtnegativen Kantenbewertung b. Wir numerieren die Ecken als

$$e_0, e_1, \ldots, e_{n-1}\,.$$

Sei B die $n \times n$-Matrix mit den Einträgen

$$B_{ij} := \begin{cases} 0 & \textit{falls } i = j \\ \min\{b(k) \mid k \textit{ Kante von } e_i \textit{ nach } e_j\} & \textit{falls } i \neq j \textit{ und eine Kante} \\ & \textit{von } e_i \textit{ nach } e_j \textit{ existiert} \\ \infty & \textit{sonst.} \end{cases}$$

Der folgende Algorithmus mit $\mathrm{O}(n^3)$ Rechenoperationen überschreibt die Matrix B mit der Matrix der Eckenabstände.

Für r von 0 bis $n-1$ wiederhole:
  Setze $N = B$.
  Für i von 0 bis $n-1$ wiederhole:
    Für j von 0 bis $n-1$ wiederhole:
      Falls $B_{ir} + B_{rj} < B_{ij}$, setze $N_{ij} = B_{ir} + B_{rj}$.
  Setze $B = N$.

Beweis: Der Abstand der Ecke e zu sich ist 0, der Abstand zu einer anderen Ecke d ist gleich der kleinsten Länge eines einfachen Weges von e nach d. Wir verwenden dieselbe Idee wie im Beweis des Algorithmus von Warshall, um die Wege im Graphen zu analysieren.

Für $r \in \{0,1,\ldots,n\}$ sei P_r die Menge aller einfachen Wege in G, die nur Zwischenecken aus der Menge $\{x_0,x_1,\ldots,x_{r-1}\}$ haben. Dann ist

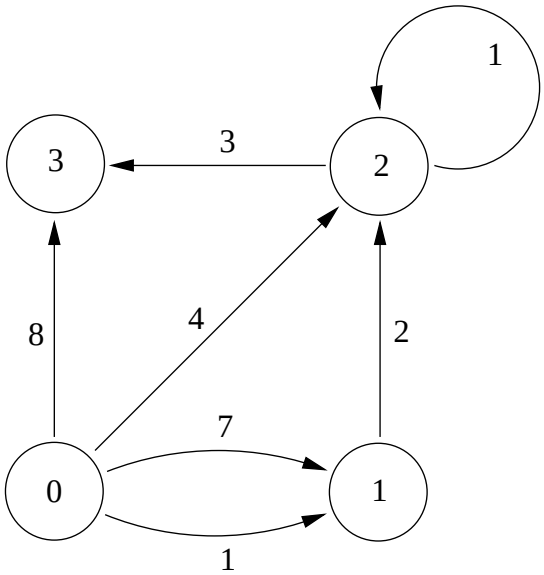
- P_0 die Menge aller Kanten von G, und
- P_n ist die Menge aller einfachen Wege in G.

Sei nun $r < n$. Für einen kürzesten Weg p von e nach d in P_{r+1} gibt es zwei Fälle:

- x_r ist keine Zwischenecke von p. Dann ist p ein kürzester Weg von e nach d in P_r.
- x_r ist eine Zwischenecke von p. Dann kann der Weg p von e nach d als Verkettung eines kürzesten Weges u von e nach x_r und eines kürzesten Weges v von x_r nach d geschrieben werden, die beide in P_r liegen.

Der Algorithmus von Floyd berechnet für $r = 0,1,\ldots,n-1$ aus den kleinsten Längen von Wegen in P_r die kleinsten Längen von Wegen in P_{r+1}.

Beispiel 2.5 : Für den gerichteten Graphen



erhält man die Abstandsmatrix

$$\begin{pmatrix} 0 & 1 & 3 & 6 \\ \infty & 0 & 2 & 5 \\ \infty & \infty & 0 & 3 \\ \infty & \infty & \infty & 0 \end{pmatrix}.$$

Definition 2.6 : (*erreichbare Ecken*)
Sei G ein gerichteter Multigraph und sei S eine Teilmenge der Eckenmenge. Eine Ecke d von G heisst *von S erreichbar*, wenn es einen Weg in G gibt mit der Endecke d und der Anfangsecke in S.

Satz 2.5 : (Nachfolgersuche)
Sei G ein gerichteter Multigraph mit endlicher Eckenmenge E und endlicher Kantenmenge K. Der folgende Algorithmus markiert alle von einer Startmenge S erreichbaren Ecken mit $\mathrm{O}(\#(E)\cdot\#(K))$ Operationen.

Markiere die Ecken in S.
Solange S nichtleer ist, wiederhole:
Wähle eine Ecke e in S und entferne sie aus S.
Bestimme alle unmarkierten unmittelbaren Nachfolger von e,
markiere sie und gebe sie zu S dazu.

Beweis: Da jede Ecke von G höchstens einmal aus S entfernt wird, terminiert der Algorithmus. Eine Ecke d ist genau dann von der Ecke e erreichbar, wenn d gleich e ist oder d von den unmittelbaren Nachfolgern von e erreicht werden kann. Bei jeder Iteration bleibt in Zeile 2 die Eigenschaft von Ecken,

markiert oder von S erreichbar zu sein,

gleich. Am Anfang bedeutet dies, von der Startmenge erreichbar zu sein, und am Ende, markiert zu sein.

Beispiel 2.6 : Im gerichteten Graphen aus Beispiel 2.3 sind alle Ecken von der Startecke 0 aus erreichbar.

Satz 2.6 : (azyklische gerichtete Graphen legen Hierarchie auf Ecken fest)
Sei G ein gerichteter Multigraph. Für Ecken e und d sei

$$d < e\,,$$

falls es einen einfachen Weg von d nach e gibt. Dann ist $\leq$ eine partielle Ordnung auf der Eckenmenge E genau dann, wenn G keine Zykeln enthält.

Beweis: Die Relation $<$ ist offenbar transitiv. Wenn G einen Zykel von e nach e enthält, dann ist $e < e$ und $<$ nicht irreflexiv. Wenn umgekehrt $<$ nicht irreflexiv ist, dann enthält G einen einfachen geschlossenen Weg und nach Lemma 2.1 auch einen Zykel.

Definition 2.7 : *(unmittelbarer Vorgänger bzw. Nachfolger)*
Sei $\leq$ eine partielle Ordnung auf einer Menge M, und seien x und y Elemente von M mit $x < y$. Dann heißt x ein *unmittelbarer Vorgänger* von y bzw. y ein *unmittelbarer Nachfolger* von x, in Zeichen

$$x \prec y,$$

wenn es kein $z \in M$ mit $x < z < y$ gibt.

Beispiel 2.7 : In der Potenzmenge von M ist eine Menge S unmittelbarer Vorgänger einer Menge T genau dann, wenn

$$T = S \cup \{x\}$$

für ein $x \in M \setminus S$ ist.

Beispiel 2.8 : Eine Partition P einer Menge M ist unmittelbarer Vorgänger einer Partition Q von M genau dann, wenn Q aus P durch Vereinigen zweier Blöcke von P entsteht.

Der folgende Satz zeigt, dass eine partielle Ordnung auf einer endlichen Menge durch den Graphen der unmittelbaren Vorgängerrelation visualisiert werden kann.

Satz 2.7 : *Sei $\leq$ eine partielle Ordnung auf der endlichen Menge M und sei G der Graph der unmittelbaren Vorgängerrelation $\prec$ auf M. Dann gilt für $x, y \in M$ genau dann $x \leq y$, wenn in G ein Weg von x nach y existiert, d.h. $\leq$ ist die transitive Hülle von $\prec$.*

Beweis: Wenn in G ein Weg von x nach y einer Länge ℓ existiert, dann ist entweder $\ell = 0$ und $x = y$, oder $\ell > 0$ und

$$x \prec z_1 \prec z_2 \prec \ldots z_{\ell-1} \prec y,$$

sodass insgesamt $x \leq y$ folgt. Zur Beweis der umgekehrten Implikation seien $x, y \in M$ mit $x \leq y$. Da für $x = y$ der leere Weg von x nach y führt, können wir $x < y$ annehmen. Wir zeigen nun durch Induktion nach der Zahl der Elemente des Intervalls

$$[x, y] := \{z \in M \mid x \leq z \leq y\},$$

dass es Elemente $z_1, \ldots, z_n \in M$ gibt mit

$$x \prec z_1 \prec \ldots \prec z_n \prec y.$$

Dann ist $((x, z_1), (z_1, z_2), \ldots, (z_n, y))$ ein Weg in G von x nach y.
Für $\#([x, y]) = 2$ ist offenbar $x \prec y$.

Für $\#([x,y]) > 2$ gibt es ein $z \in M \setminus \{x,y\}$ mit

$$x < z < y\,.$$

Da die Intervalle $[x,z]$ und $[z,y]$ weniger Elemente als $[x,y]$ enthalten, existieren nach Induktionsannahme Elemente $v_1,\ldots,v_k,w_1,\ldots,w_\ell \in M$ mit

$$x \prec v_1 \prec \ldots \prec v_k \prec z \prec w_1 \prec \ldots \prec w_\ell \prec y\,.$$

2. Ungerichtete Graphen

Definition 2.8 : *(ungerichteter Multigraph)*
Ein *ungerichteter Multigraph* ist gegeben durch

(1) eine *Eckenmenge* (oder *Knotenmenge*) E
(2) eine *Kantenmenge* K
(3) eine Abbildung

$$\begin{aligned} r : K &\to \{\{c,d\} \mid c,d \in E\}\,, \\ k &\mapsto r(k)\,, \end{aligned}$$

die jeder Kante k eine Menge $r(k)$ mit einer oder zwei *Endecken* zuordnet (r für Rand). Man nennt dann k eine Kante zwischen diesen Ecken.

Eine Ecke c heißt *Nachbar* der Ecke d , wenn es eine Kante zwischen c und d gibt. Man nennt eine Kante mit nur einer Endecke eine *Schleife*. Kanten mit den gleichen Endecken heißen *parallel*. Für eine Ecke e heißt die Zahl der Kanten mit Endecke e der *Grad* von e . Wenn zusätzlich Abbildungen

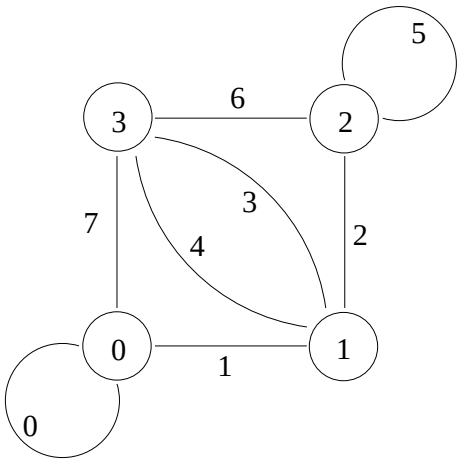
$$a : E \to M \quad \text{oder} \quad b : K \to N$$

gegeben werden, dann heißt der Multigraph *ecken-* bzw. *kantenbeschriftet*, im Spezialfall $M = \mathbb{R}$ oder $N = \mathbb{R}$ *ecken-* bzw. *kantenbewertet*.

Beispiel 2.9 : Sei ein ungerichteter Multigraph gegeben durch die Eckenmenge $E = \{0,1,2,3\}$, die Kantenmenge $K = \{0,1,2,\ldots,7\}$ und die Abbildung r laut folgender Tabelle:

k	$r(k)$
0	$\{0\}$
1	$\{0,1\}$
2	$\{1,2\}$
3	$\{1,3\}$
4	$\{1,3\}$
5	$\{2\}$
6	$\{2,3\}$
7	$\{0,3\}$

Visualisiert kann dieser Graph durch folgendes Bild werden:

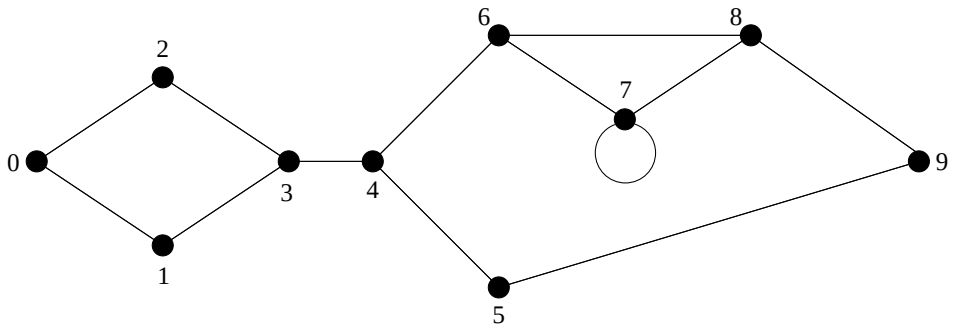


Definition 2.9 : *(ungerichteter Graph)*
Ein *ungerichteter Graph* ist ein ungerichteter Multigraph ohne parallele Kanten. Dann gibt es zu jeder Eckenmenge $\{c,d\}$ höchstens eine Kante $k \in K$ mit $r(k) = \{c,d\}$.

Beispiel 2.10 : Eine symmetrische Relation S auf einer Menge M kann durch den ungerichteten Graphen mit

- der Eckenmenge M
- der Kantenmenge $\{\{x,y\} \mid (x,y) \in S\}$
- der Abbildung $r(\{x,y\}) = \{x,y\}$

visualisiert werden. Offenbar ist jeder ungerichtete Graph der Graph einer symmetrischen Relation.



Definition 2.10 : *(Teilmultigraph, Teilgraph)*
Sei $G = (E,K,r)$ ein ungerichteter Multigraph. Ein ungerichteter Multigraph $G' = (E',K',r')$ heißt *Teilmultigraph* von G, wenn $E' \subseteq E$, $K' \subseteq K$ und

$$r'(k) = r(k) \quad \text{für alle } k \in K' .$$

Ein *Teilgraph* ist ein *Teilmultigraph*, der selbst Graph ist.

Definition 2.11 : *(Wege, Zusammenhang, Zyklen, Wälder, Bäume, Blätter)* Sei (E,K,r) ein ungerichteter Multigraph, und seien c,d Ecken. Ein Tupel

$$(k_0,k_1,\ldots,k_{\ell-1}) \in K^\ell$$

heißt ein *Weg* von c nach d der Länge ℓ, wenn es Ecken $e_0,e_1,\ldots,e_\ell$ gibt mit $e_0 = c,\ e_\ell = d$, und

$$r(k_i) = \{e_i,e_{i+1}\} \quad \text{für } i = 0,1,\ldots,\ell-1\ .$$

Die Ecken $e_0,e_1,\ldots,e_\ell$ sind dann eindeutig bestimmt, und man nennt e_0 die *Anfangsecke*, e_ℓ die *Endecke* sowie $e_1,e_2,\ldots,e_{\ell-1}$ die *Zwischenecken* des Weges. Für jede Ecke $e \in E$ wird das leere Tupel $() \in K^0$ der *leere Weg* mit *Anfangsecke* e und *Endecke* e genannt.

Der ungerichtete Multigraph heißt *zusammenhängend*, wenn es von jeder Ecke zu jeder anderen Ecke einen Weg gibt. Ein Weg heißt *einfach*, wenn er nichtleer ist und die Ecken

$$e_0,e_1,\ldots,e_\ell$$

paarweise verschieden sind bis auf die mögliche Ausnahme $e_0 = e_\ell$. Für jeden Weg $(k_0,k_1,\ldots,k_{\ell-2},k_{\ell-1})$ von c nach d ist der *reziproke* Weg

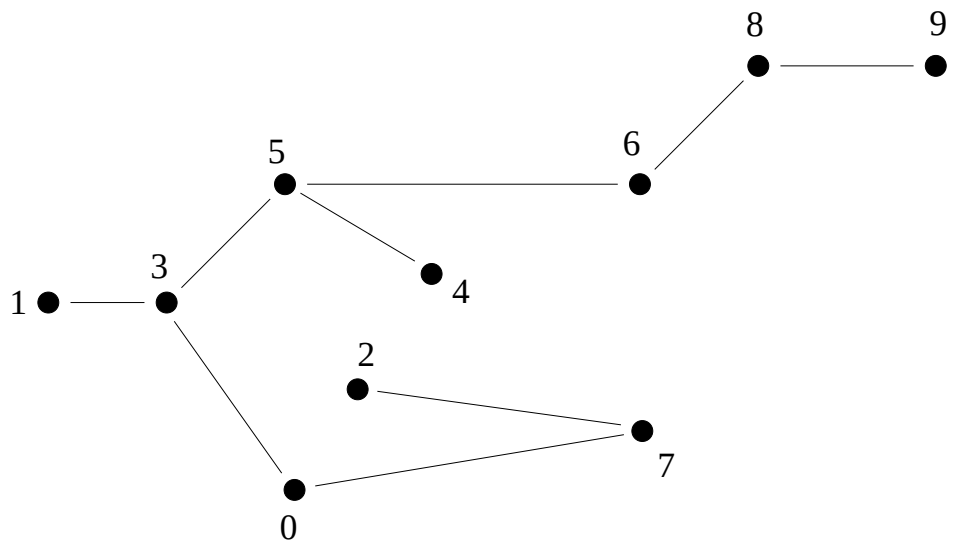
$$(k_{\ell-1},k_{\ell-2},\ldots,k_1,k_0)$$

ein Weg von d nach c. Wenn $(k_0,k_1,\ldots,k_{\ell-1})$ ein Weg von c nach d ist und $(h_0,h_1,\ldots,h_{m-1})$ ein Weg von d nach e ist, dann ist die *Verkettung*

$$(k_0,k_1,\ldots,k_{\ell-1},h_0,h_1,\ldots,h_{m-1})$$

ein Weg von c nach e. Ein Weg $(k_0,k_1,\ldots,k_{\ell-1}) \in K^\ell$ heißt *geschlossen*, wenn Anfangs- und Endecke gleich sind. Ein nichtleerer geschlossener Weg mit paarweise verschiedenen Kanten wird ein *Zykel* genannt. Der ungerichtete Multigraph heißt *zyklenfrei* oder *azyklisch*, wenn es keine Zyklen gibt.

Ein *Wald* ist ein zyklenfreier ungerichteter Multigraph, ein *Baum* ist ein zusammenhängender Wald. Ecken eines Waldes vom Grad ≤ 1 nennt man *Blätter*.



Lemma 2.2 : (nichtleere Wege enthalten einfache Wege)
Sei G ein ungerichteter Multigraph.

(1) *Wenn es einen nichtleeren Weg p von der Ecke c zur Ecke d gibt, dann kann aus p durch Weglassen von Kanten ein einfacher Weg von c nach d gewonnen werden.*
(2) *Jeder einfache geschlossene Weg der Länge mindestens* 3 *ist ein Zykel.*
(3) *Aus jedem Zykel kann durch Weglassen von Kanten ein einfacher Zykel erhalten werden.*

Beweis: (1) wie für Lemma 2.1, (3) folgt aus (1).
(2) Sei $(k_0, k_1, \ldots, k_{\ell-1})$ ein einfacher Weg von e nach e mit zwei gleichen Kanten. Dann stimmen auch ihre Eckenmengen überein. Da der Weg einfach ist, enthält er nur die Ecke e doppelt. Daher müssen diese Kanten erste und letzte Kante des Weges sein und aufeinanderfolgen. Somit ist $\ell = 2$ und $k_0 = k_1$. Insbesondere folgt die Behauptung (2) des Lemmas.

Beispiel 2.11 : In einem ungerichteten Multigraphen kann es Zyklen der Länge 2 geben, in einem ungerichteten Graphen nicht.

Lemma 2.3 : (Eindeutigkeit von einfachen Wegen in Bäumen)
Sei G ein Baum. Dann existiert zu verschiedenen Ecken c und d genau ein einfacher Weg von c nach d .

Beweis: Seien p und q zwei einfache Wege von c nach d . Wir können annehmen, dass die Mengen der Zwischenecken disjunkt sind, ansonsten zerlegen wir die Wege in entsprechende Teilwege. Dann ist die Verkettung eines Weges mit dem reziproken Weg des anderen Weges ein einfacher geschlossener Weg. Nach Lemma 2.2 besteht dieser Weg aus zwei gleichen Kanten. Daher sind die Wege p und q gleich.

Definition 2.12 : *(schwacher Zusammenhang, Orientierung)*

(1) Wenn G ein gerichteter Multigraph ist, dann erhält man durch
$$r(k) := \{q(k), z(k)\} \quad \text{für } k \in K$$
einen ungerichteten Multigraphen, indem man die Richtung der Kanten vergisst. Man nennt G *schwach zusammenhängend*, wenn sein ungerichteter Multigraph zusammenhängend ist.
(2) Umgekehrt liegen zwei Verfahren nahe, aus einem ungerichteten Multigraphen G einen gerichteten Multigraphen zu konstruieren:
   – Man dupliziert jede Kante von G und wählt für Original und Kopie jeweils eine andere Richtung, d.h. man interpretiert die Kanten als Doppelpfeile.

– Man wählt für jede Kante k von G eine Richtung aus, d.h. man setzt für $r(k) = \{c, d\}$

entweder $q(k) := c$ und $z(k) := d$ oder umgekehrt.

Der gerichtete Multigraph heißt dann eine *Orientierung* von G.

Satz 2.8 : (Wurzelbaum als Baum mit ausgezeichneter Ecke)

(1) *Für jeden Wurzelbaum W mit Wurzel w ist der zugehörige ungerichtete Multigraph B ein Baum mit Ecke w.*
(2) *Zu jedem nichtleeren Baum B und jeder Ecke e von B gibt es genau einen Wurzelbaum mit Wurzel e, der eine Orientierung von B ist.*
(3) *Die Zuordnungen*

$$W \mapsto (B, w)$$

von (1) und

$$(B, e) \mapsto W$$

von (2) sind zueinander invers. Daher kann man einen Wurzelbaum auch als einen Baum mit einer ausgezeichneten Ecke auffassen.

Beweis: (1) Offensichtlich ist B zusammenhängend. Wenn B einen Zykel enthält, dann gäbe es in W zwei verschiedene Kanten mit gleichem Endpunkt und somit zwei verschiedene Wege von der Wurzel zu dieser Ecke.
(2) Da B zyklenfrei ist, gibt es zu jeder Ecke d ungleich e genau einen einfachen Weg von e nach d. Die dadurch festgelegte Orientierung W ist ein Wurzelbaum mit der Wurzel e.
(3) Der ungerichtete Multigraph einer Orientierung ist der ursprüngliche Multigraph. Daher erhält man aus dem Wurzelbaum durch Vergessen der Richtungen den alten Baum zurück. Wenn im neuen Wurzelbaum eine Kante anders orientiert wäre als im alten Wurzelbaum, dann gäbe es im Baum zwei verschiedene einfache Wege von der Wurzel zu einem der Endknoten.

Definition 2.13 : *(Zusammenhangskomponenten)*
Sei G ein ungerichteter Multigraph. Zwei Ecken c und d seien äquivalent, wenn es einen Weg von c nach d gibt. Dann heißen die Äquivalenzklassen die *Zusammenhangskomponenten* von G. Offensichtlich kann man in G Schleifen und parallele Kanten entfernen, ohne die Partition in Zusammenhangskomponenten zu verändern.

Satz 2.9 : (Anzahlbedingung für Bäume)
Sei G ein zusammenhängender ungerichteter Multigraph mit mindestens einer Ecke. Dann ist G genau dann ein Baum, wenn er eine Ecke mehr als Kanten hat.

Beweis: Sei $G = (E, K, r)$ ein Baum. Wir zeigen

$$\#(E) = \#(K) + 1$$

durch Induktion nach $\#(K)$. Wenn $\#(K) = 0$ ist, dann gibt es nur eine Ecke. Sei nun $\#(K) > 0$. Herausnehmen einer beliebigen Kante liefert einen Wald mit zwei Zusammenhangskomponenten E_1 und E_2, die wieder Bäume sind. Nach Induktionsannahme ist

$$\#(K) = 1 + (\#(E_1) - 1) + (\#(E_2) - 1) = \#(E) - 1\,.$$

Sei umgekehrt G kein Baum. Dann gibt es einen einfachen Zykel

$$(k_0, k_1, \ldots, k_{\ell-1})\,.$$

Sei Z die Menge der Ecken des Zykels. Für jede Ecke e in $E \setminus Z$ wählen wir einen Weg kürzester Länge zu einer Ecke aus Z und bezeichnen die erste Kante mit $k(e)$. Dann ist die Abbildung

$$E \setminus Z \to K \setminus \{k_0, k_1, \ldots, k_{\ell-1}\}\,,\; e \mapsto k(e)\,,$$

injektiv. Es folgt

$$\#(E) - \ell \leq \#(K) - \ell$$

und somit

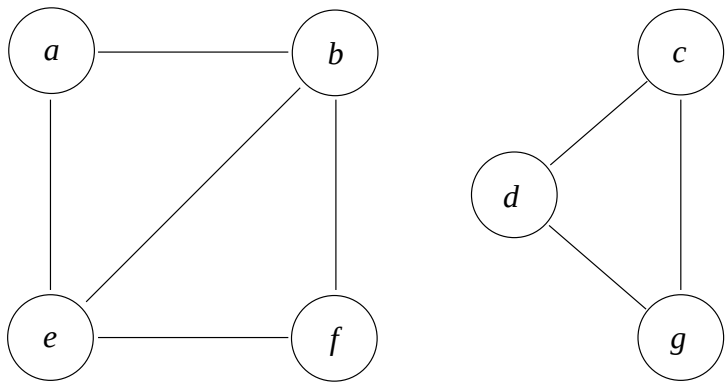
$$\#(E) \leq \#(K)\,.$$

Definition 2.14 : *(spannender Wald)*
Sei G ein ungerichteter Multigraph. Ein Teilgraph G' von G heißt ein *spannender Wald* von G, wenn

(a) G' ein Wald ist und
(b) die Partitionen in Zusammenhangskomponenten von G bzw. G' übereinstimmen.

Notwendigerweise ist dann $E' = E$.

Beispiel 2.12 : Für den Graphen



gibt es

$$(10-2)\cdot 3 = 24$$

spannende Wälder.

Satz 2.10 : (Algorithmus von Kruskal)
Sei $G = (E, K, r)$ ein ungerichteter Multigraph mit Kantenbewertung b. Gesucht werden die Partition von E in Zusammenhangskomponenten sowie die Kantenmenge W eines spannenden Waldes von G mit minimaler Bewertung

$$\sum_{k\in W} b(k)\,.$$

Als Vorbereitung werden im Multigraphen alle Schleifen entfernt, parallele Kanten bis auf jene mit kleinster Bewertung gestrichen, und die verbleibenden Kanten sortiert, sodass

$$b(k_0) \le b(k_1) \le \ldots \le b(k_{m-1})$$

gilt. Der eigentliche Algorithmus operiert dann mit $\mathrm{O}(\#(E)\cdot\#(K))$ *Operationen wie folgt.*

Setze $W = \emptyset$ *und* $P = \{\{e\} \mid e \in E\}$.
Für i von 0 *bis* $m-1$ *wiederhole:*
Falls die Ecken e und d von k_i in verschiedenen Blöcken von P liegen,
vereinige die beiden Blöcke von P und nimm k_i in die Menge W auf.

Beweis: Sei G_i der Teilgraph von G mit Eckenmenge E und Kantenmenge

$$\{k_0, k_1, , \ldots, k_i\}\,.$$

Der Algorithmus startet mit der Partition in einzelne Ecken und vereinigt anschließend Blöcke mit Verbindungskante. Nach Schritt i ist P die Partition in Zusammenhangskomponenten von G_i . Die Menge W ist zunächst leer und wird im Schritt i um eine etwaige Verbindungskante erweitert, deren Ecken dann im Vereinigungsblock liegen. Für jeden Block B ist der Teilgraph mit Eckenmenge B und den entsprechenden Kanten aus W ein Baum, weil er zusammenhängt und die Anzahlbedingung

$$\#(E_1) + \#(E_2) = (\#(K_1)+1) + (\#(K_2)+1) = (\#(K_1)+\#(K_2)+1)+1$$

erfüllt. Somit ist nach Schritt i der Teilgraph mit Eckenmenge E und Kantenmenge W ein spannender Wald von G_i .

Zu zeigen bleibt noch, dass die Greedy-Strategie bei der Wahl der Kanten einen spannenden Wald mit minimaler Bewertung liefert. Sei dazu M die Kantenmenge eines spannenden Waldes mit minimaler Bewertung. Wenn $M = W$ ist, haben wir die Behauptung gezeigt. Wenn $M \neq W$ ist, dann existiert eine Kante k_i in W, die nicht in M liegt. Seien e_1, e_2 die Ecken von k_i

und E_1, E_2 die zugehörigen Blöcke im Algorithmus. Da es einen Weg p von e_1 nach e_2 aus Kanten in M gibt, existiert eine Kante k_j im Weg p, die eine Endecke in E_1 und die andere außerhalb von E_1 hat. Somit ist

$$j > i \quad \text{und} \quad b(k_j) \geq b(k_i)\,.$$

Der neue Teilgraph mit Eckenmenge E und Kantenmenge

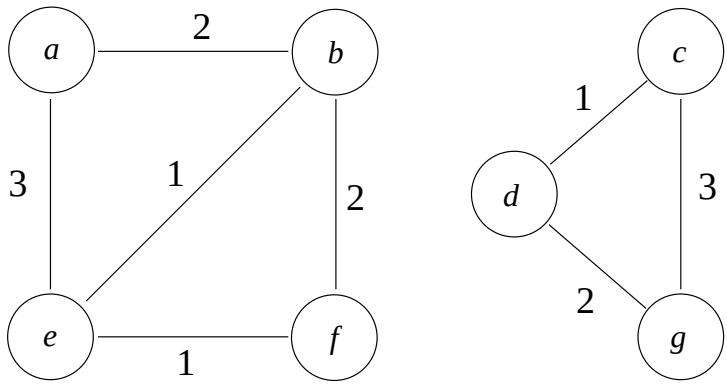
$$N := (M \setminus \{k_j\}) \cup \{k_i\}$$

ist ein spannender Wald, weil jeder Weg über k_j auch über k_i und die restlichen Kanten von p geführt werden kann und umgekehrt. Wegen

$$\sum_{k \in N} b(k) = \sum_{k \in M} b(k) - b(k_j) + b(k_i) \leq \sum_{k \in M} b(k)$$

hat der neue spannende Wald ebenfalls eine minimale Bewertung und zusätzlich eine kleinere Indexsumme. Somit erhält man durch endlich viele Austäusche den Kruskalwald.

Beispiel 2.13 : Für den bewerteten Graphen



startet der Algorithmus von Kruskal mit

$$W = \emptyset \quad \text{und} \quad P = \{\{a\},\{b\},\{c\},\{d\},\{e\},\{f\},\{g\}\}$$

und endet mit

$$W = \{\{a,b\},\{b,e\},\{c,d\},\{d,g\},\{e,f\}\} \quad \text{und} \quad P = \{\{a,b,e,f\},\{c,d,g\}\}\,.$$

KAPITEL 3

Zähltheorie

1. Aufzählen und Numerieren von Objekten

Definition 3.1 : *(Aufzählung, Numerierung)*
Eine Menge M heißt *endlich* , wenn es eine natürliche Zahl m und eine bijektive Abbildung

$$\alpha : \{0,1,\ldots,m-1\} \to M$$

gibt. In diesem Fall ist m eindeutig bestimmt und man nennt

$$\#(M) := m$$

die Anzahl der Elemente von M. Die Abbildung α ist im Allgemeinen nicht eindeutig und heißt eine *Aufzählung* von M . Eine bijektive Abbildung

$$v : M \to \{0,1,\ldots,m-1\}$$

wird eine *Numerierung* von M genannt. Offenbar ist die Umkehrabbildung einer Aufzählung von M eine Numerierung von M, und die Umkehrabbildung einer Numerierung von M ist eine Aufzählung von M. Wenn M nicht endlich ist, dann heißt M *unendlich* und man schreibt

$$\#(M) = \infty \, .$$

Satz 3.1 : (elementare Zählregeln)

(1) Gleichheitsregel: *Sind M und N endliche Mengen und ist $f : M \to N$ eine bijektive Abbildung, so gilt*

$$\#(M) = \#(N) \, .$$

(2) Summenregel: *Sind $A_1, A_2, \ldots, A_k$* paarweise disjunkte *endliche Mengen, so gilt für die Vereinigung*

$$\#(A_1 \cup A_2 \cup \ldots \cup A_k) = \sum_{i=1}^{k} \#(A_i) \, .$$

(3) Differenzregel: *Für endliche Mengen A und B gilt*

$$\#(A \setminus B) = \#(A) - \#(A \cap B) \, .$$

(4) Siebformel: *Für endliche Mengen $A_1, A_2, \ldots, A_k$ gilt*

$$\#(A_1 \cup \ldots \cup A_k) = \sum_{\substack{I \subseteq \{1,2,\ldots,k\} \\ I \neq \emptyset}} (-1)^{\#(I)-1} \#(\bigcap_{i \in I} A_i) \, .$$

Insbesondere ist für endliche Mengen A und B

$$\#(A \cup B) = \#(A) + \#(B) - \#(A \cap B)\ .$$

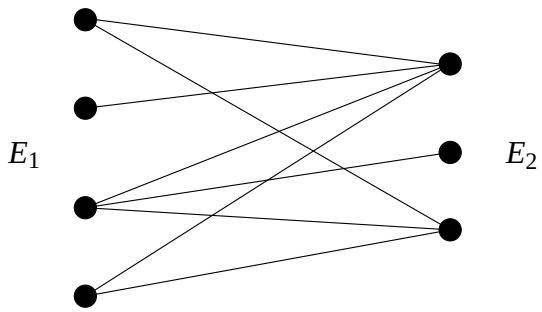
(5) Produktregel: *Sind $M_1, M_2, \ldots, M_k$ endliche Mengen, so gilt für das kartesische Produkt*

$$\#(M_1 \times M_2 \times \ldots \times M_k) = \prod_{i=1}^{k} \#(M_i)\ .$$

Insbesondere ist für eine endliche Menge M

$$\#(M^k) = \#(M)^k\ .$$

(6) Regel des zweifachen Abzählens:
Ein ungerichteter Graph heißt bipartit, *wenn es eine Partition der Eckenmenge in zwei Blöcke E_1 und E_2 gibt, sodass jede Kante eine Endecke in E_1 und eine Endecke in E_2 hat.*



Für einen endlichen bipartiten Graphen ist dann

$$\sum_{e_1 \in E_1} \mathrm{Grad}(e_1) = \sum_{e_2 \in E_2} \mathrm{Grad}(e_2)\,.$$

Beweis: (1) Da M endlich ist, gibt es ein $m \in \mathbb{N}$ und eine bijektive Abbildung $\alpha : \{0, 1, \ldots, m-1\} \to M$. Dann ist auch die zusammengesetzte Abbildung

$$f \circ \alpha : \{0, 1, \ldots, m-1\} \to N\ ,\ i \mapsto f(\alpha(i)),$$

bijektiv.
(2) Seien $\alpha_1 : \{0, 1, \ldots, m_1 - 1\} \to M_1, \ldots, \alpha_k : \{0, 1, \ldots, m_k - 1\} \to M_k$ bijektiv. Dann ist auch die zusammengesetzte Abbildung

$$\alpha : \{0, 1, \ldots, m_1 + \ldots + m_k - 1\} \to M_1 \cup \ldots \cup M_k\ ,$$

$$i \mapsto \begin{cases} \alpha_1(i) & \text{falls } i \in \{0, 1, \ldots, m_1 - 1\} \\ \alpha_2(i - m_1) & \text{falls } i \in \{m_1, \ldots, m_1 + m_2 - 1\} \\ \vdots & \vdots \\ \alpha_k(i - m_1 - \ldots - m_{k-1}) & \text{falls } i \in \{m_1 + \ldots + m_{k-1}, \ldots, m_1 + \ldots + m_k - 1\}, \end{cases}$$

bijektiv.
(3) Aus der disjunkten Vereinigung

$$A = (A \setminus B) \cup (A \cap B)$$

folgt nach (2)

$$\#(A \setminus B) = \#(A) - \#(A \cap B) \ .$$

(4) Wir führen eine Induktion über k. Aus der disjunkten Vereinigung

$$A_1 \cup A_2 = A_1 \cup (A_2 \setminus A_1)$$

folgt

$$\#(A_1 \cup A_2) = \#(A_1) + \#(A_2 \setminus A_1) = \#(A_1) + \#(A_2) - \#(A_1 \cap A_2) \ .$$

Für $k > 2$ ist nach Induktionsannahme

$$\#(\bigcup_{i=1}^{k} A_i) = \#((\bigcup_{i=1}^{k-1} A_i) \cup A_k) = \#(\bigcup_{i=1}^{k-1} A_i) + \#(A_k) - \#(\bigcup_{i=1}^{k-1} (A_i \cap A_k)) =$$
$$\sum_{\substack{I \subseteq \{1,\ldots,k-1\} \\ I \neq \emptyset}} (-1)^{\#(I)-1} \#(\bigcap_{i \in I} A_i) + \#(A_k) - \sum_{\substack{I \subseteq \{1,\ldots,k-1\} \\ I \neq \emptyset}} (-1)^{\#(I)-1} \#(\bigcap_{i \in I} A_i \cap A_k) =$$
$$\sum_{\substack{J \subseteq \{1,\ldots,k\} \\ J \neq \emptyset}} (-1)^{\#(J)-1} \#(\bigcap_{i \in J} A_i) \ ,$$

weil entweder $J = I$ oder $J = \{k\}$ oder $J = I \cup \{k\}$ gewählt werden kann.
(5) Seien $\nu_1 : M_1 \to \{0, 1, \ldots, m_1 - 1\}, \ldots, \nu_k : M_k \to \{0, 1, \ldots, m_k - 1\}$ bijektiv. Dann ist auch die Abbildung

$$\begin{array}{rcl} \nu : M_1 \times \ldots \times M_k & \to & \{0, 1, \ldots, m_1 \cdots m_k - 1\} \\ (x_1, \ldots, x_k) & \mapsto & \nu_1(x_1) \cdot m_2 \cdots m_k + \nu_2(x_2) \cdot m_3 \cdots m_k + \ldots + \nu_k(x_k) \end{array}$$

bijektiv, weil man die Einzelnummern

$$i_1 := \nu_1(x_1) \ , \ \ldots \ , \ i_k := \nu_k(x_k)$$

aus der Gesamtnummer

$$I := i_1 \cdot m_2 \cdots m_k + i_2 \cdot m_3 \cdots m_k + \ldots + i_{k-1} \cdot m_k + i_k$$

durch *ganzzahlige Division mit Rest* zurückerhält:

$$\begin{array}{rcl} i_k & = & I \% m_k \\ i_{k-1} & = & (I/m_k) \% m_{k-1} \\ & \vdots & \\ i_2 & = & (I/(m_3 \cdots m_k)) \% m_2 \\ i_1 & = & I/(m_2 \cdots m_k) \end{array}$$

(6) Beide Summen geben die Zahl der Kanten an, einmal über die Endecken in E_1 gezählt, das andere Mal über die Endecken in E_2.

Beispiel 3.1 : In C-Programmen werden die Elemente mehrdimensionaler Felder hintereinander im Speicher abgelegt, wobei die Reihenfolge so geregelt ist, dass

„hintere Indizes schneller laufen als vordere“.

Zum Beispiel liegen für

```
int M[2][3] = {{3,5,-2},{1,0,2}};
```

die Feldelemente wie folgt im Speicher:

M[0][0]	M[0][1]	M[0][2]	M[1][0]	M[1][1]	M[1][2]
3	5	-2	1	0	2

M

Wird ein mehrdimensionales Feld an eine Funktion übergeben, die variable Dimensionen zuläßt, dann muß der Programmierer in der Funktion die relativen Adressen der Feldelemente selbst berechnen, z.B. im folgenden Codefragment:

```
...
double f(double *z, int m1, int m2, int m3)
{
  ...
}
...
int main( void)
{
   double x, y , A[2][3][4], B[3][4][2];
   ...
   x = f(&A[0][0][0],2,3,4);
   y = f(&B[0][0][0],3,4,2);
   ...
}
```

Nach Satz 3.1 kann in der Funktion `f` das Feldelement „`z[i][j][k]`“ als

```
*(z+i*m2*m3+j*m3+k)
```

angesprochen werden. Die Indizes `i`, `j`, `k` des Feldelements an der Adresse `z+l` können aus den Formeln

```
k = l%m3
j = (l/m3)%m2
i = l/(m2*m3)
```

berechnet werden.

Satz 3.2 : (Schubfachprinzip, Taubenschlagprinzip)
Seien M und N endliche Mengen und sei $f : M \to N$ eine Abbildung. Wenn $\#(M) > \#(N)$ ist, dann gibt es mindestens ein Element $y \in N$ mit mehr als einem Urbild.

Beweis: Wenn jedes Element von N höchstens ein Urbild hätte, dann wäre f injektiv, die eingeschränkte Abbildung $M \to f(M)$ bijektiv, und somit $\#(M) \leq \#(N)$.

Satz 3.3 : (Zahl der injektiven Abbildungen)
Seien K und M endliche Mengen mit k bzw. n Elementen. Dann gibt es genau

$$(n)_k := \begin{cases} n(n-1)(n-2)\cdots(n-k+1) & \text{falls } k \geq 1 \\ 1 & \text{falls } k = 0 \end{cases}$$

verschiedene injektive Abbildungen von K nach M. Man nennt die Zahl $(n)_k$ die fallende Faktorielle *von n und k.*

Beweis: Wir zeigen die Formel durch Induktion über k. Am Induktionsanfang ist $k = 0$, somit K leer und die einzige injektive Abbildung die leere Abbildung. Für den Induktionsschluss schreiben wir

$$K = \{x_0, x_1, \ldots, x_k\}$$

und überlegen uns, wie viele injektive Abbildungen $f : K \to M$ es geben kann. Für x_0 gibt es n Möglichkeiten, ein Bild $f(x_0) \in M$ zu wählen. Dieses Element

$$y_0 := f(x_0)$$

darf dann aber nicht mehr als Bild eines anderen Elements in K gewählt werden, sodass für die Wahl der Bilder von $x_1, \ldots, x_k$ nur die Elemente in $M \setminus \{y_0\}$ in Frage kommen. Nach Induktionsannahme gibt es dafür $(n-1)_k$ Möglichkeiten. Die Gesamtzahl der Möglichkeiten ist somit

$$n \cdot (n-1)_k = (n)_{k+1}\,.$$

Beispiel 3.2 : Sei M eine endliche Menge von Objekten. In der Literatur werden injektive Abbildungen

$$\{0, 1, \ldots, k-1\} \to M\,,\; i \mapsto x_i\,,$$

als k-Tupel

$$(x_0, x_1, \ldots, x_{k-1})$$

von *unterschiedlichen* Elementen von M beschrieben und *Permutationen* von jeweils k Objekten aus M genannt.

Satz 3.4 : (Zahl der bijektiven Abbildungen)
Seien K und M endliche Mengen mit jeweils n Elementen. Dann gibt es genau

$$n! := \begin{cases} n(n-1)(n-2)\cdots 3\cdot 2\cdot 1 & \textit{falls } n \geq 1 \\ 1 & \textit{falls } n = 0 \end{cases}$$

verschiedene bijektive Abbildungen von K nach M. Man nennt die Zahl $n!$ die Faktorielle *oder* Fakultät *von n.*

Die Faktorielle wächst sehr schnell und kann für große n mit der Stirlingschen Formel

$$n! \approx \sqrt{2\pi}\cdot e^{(n+\frac{1}{2})\log n - n} = \Theta(\sqrt{n}\cdot\left(\frac{n}{e}\right)^n)$$

approximiert werden (ohne Beweis).

Beweis: Wegen $\#(K) = \#(M) = n$ ist jede injektive Abbildung von K nach M bijektiv. Damit folgen die Behauptungen aus Satz 3.3 mit $(n)_n = n!$.

Beispiel 3.3 : Für n Objekte gibt es $n!$ verschiedene Möglichkeiten, eine Reihenfolge festzulegen, d.h. die Objekte zu numerieren.

Satz 3.5 : (Zahl von Teilmengen)
Sei M eine endliche Menge mit n Elementen. Dann gilt

$$\#(\mathcal{P}(M)) = 2^n \ .$$

Beweis: Wir fixieren eine Aufzählung $\alpha : \{0,1,\ldots,n-1\} \to M$. Dann ist die folgende Abbildung bijektiv, insbesondere können Teilmengen als Bitmuster programmiert werden.

$$F : \mathcal{P}(M) \to \{0,1\}^n \ , \ T \mapsto (t_0,\ldots,t_{n-1}) \ , \ t_i := \begin{cases} 1 & \text{falls } \alpha(i) \in T \\ 0 & \text{sonst.} \end{cases}$$

Satz 3.6 : (Zahl von Teilmengen mit vorgegebener Anzahl von Elementen)
Sei M eine endliche Menge mit n Elementen und sei k eine natürliche Zahl. Dann gilt

$$\#(\mathcal{P}_k(M)) = \binom{n}{k} \ .$$

Dabei ist der Binomialkoeffizient *„n über k“ definiert als*

$$\binom{n}{k} := \frac{n\cdot(n-1)\cdots(n-k+1)}{k\cdot(k-1)\cdots 1} = \begin{cases} \dfrac{n!}{k!(n-k)!} & \textit{falls } k \leq n \\ 0 & \textit{sonst.} \end{cases}$$

Beweis: Eine Aufzählung $\alpha : \{0,1,\ldots,k-1\} \to T$ einer k-elementigen Teilmenge T von M erhält man durch Wählen

- eines beliebigen Elements $\alpha(0) \in M$,
- eines beliebigen Elements $\alpha(1) \in M \setminus \{\alpha(0)\}$,
- eines beliebigen Elements $\alpha(2) \in M \setminus \{\alpha(0), \alpha(1)\}$, usw.

Da es bei der Teilmenge T nicht auf die Reihenfolge der gewählten Elemente ankommt, ergibt sich die gesuchte Anzahl als

$$n \cdot (n-1) \cdots (n-k+1)/k! \, .$$

Beispiel 3.4 : Sei M eine endliche Menge von Objekten. In der Literatur werden k-elementige Teilmengen von M *Kombinationen* von jeweils k Objekten aus M genannt. Im Gegensatz zu den Permutationen von Objekten aus M spielt bei den Kombinationen die Reihenfolge der ausgewählten Objekte keine Rolle.

Beispiel 3.5 : Sei M eine endliche Menge mit n Elementen und sei k eine natürliche Zahl $\leq n$. Dann ist die Abbildung

$$\mathcal{P}_k(M) \to \mathcal{P}_{n-k}(M) \, , \; T \mapsto M \setminus T \, ,$$

bijektiv und somit

$$\binom{n}{k} = \binom{n}{n-k} \, .$$

2. Abzählbarkeit von Mengen

Definition 3.2 : *(abzählbare Unendlichkeit)*
Eine Menge M heisst *abzählbar unendlich*, wenn eine bijektive Abbildung

$$\alpha : \mathbb{N} \to M \, , \; i \mapsto x_i \, ,$$

existiert. Man schreibt dann

$$M = \{x_0, x_1, x_2, \ldots\} \, ,$$

nennt α eine *Aufzählung* von M und α^{-1} eine *Numerierung* von M.

Beispiel 3.6 : Die Menge $\mathbb{N}$ der natürlichen Zahlen ist abzählbar unendlich, weil die identische Abbildung bijektiv ist. Auch die Menge $\mathbb{Z}$ der ganzen Zahlen ist abzählbar unendlich, weil die Abbildung

$$\mathbb{N} \to \mathbb{Z} \, , \; i \mapsto \begin{cases} i/2 & \text{falls } i \text{ gerade} \\ -(i+1)/2 & \text{falls } i \text{ ungerade} \end{cases}$$

bijektiv ist.

Satz 3.7 : *Die Menge* $\mathbb{N} \times \mathbb{N}$ *ist abzählbar unendlich.*

Beweis: Wir schreiben die Paare (m,n) zweidimensional

$$\begin{array}{ccccc} (0,0) & (0,1) & (0,2) & (0,3) & \dots \\ (1,0) & (1,1) & (1,2) & (1,3) & \dots \\ (2,0) & (2,1) & (2,2) & (2,3) & \dots \\ (3,0) & (3,1) & (3,2) & (3,3) & \dots \\ \vdots & & & & \end{array}$$

auf und numerieren diagonal

$$\begin{array}{ccc} (0,0) & \mapsto & 0 \\ (0,1) & \mapsto & 1 \\ (1,0) & \mapsto & 2 \\ (0,2) & \mapsto & 3 \\ (1,1) & \mapsto & 4 \\ (2,0) & \mapsto & 5 \\ (0,3) & \mapsto & 6 \\ & \vdots & \end{array}$$

Dabei bekommt das Paar (m,n) die Nummer

$$\left(\sum_{i=0}^{m+n-1} (i+1) \right) + m\,.$$

Somit ist die Abbildung

$$\mathbb{N} \times \mathbb{N} \to \mathbb{N}\,,\ (m,n) \mapsto \frac{(m+n)(m+n+1)}{2} + m\,,$$

bijektiv.

Definition 3.3 : *(graduiert-lexikographische Ordnung auf Zahlentupeln)*
Für $x,y \in \mathbb{N}^k$ sei

$$x <_{\text{gradlex}} y\,,$$

falls entweder

$$\sum_{i=1}^{k} x_i < \sum_{i=1}^{k} y_i$$

oder

$$\sum_{i=1}^{k} x_i = \sum_{i=1}^{k} y_i \quad \text{und} \quad x <_{\text{lex}} y$$

ist. Dann ist $\leq_{\text{gradlex}}$ eine totale Ordnung auf $\mathbb{N}^k$ und heisst die *graduiert-lexikographische Ordnung*.

Satz 3.8 : *Die Menge $\mathbb{N}^k$ ist abzählbar unendlich.*

Beweis: Jedes $x \in \mathbb{N}^k$ hat nur endlich viele Vorgänger. Daher liefert die Numerierung der Elemente in der graduiert-lexikographischen Ordnung eine bijektive Abbildung $\mathbb{N}^k \to \mathbb{N}$.

Definition 3.4 : Eine Menge heisst *abzählbar*, wenn sie endlich oder abzählbar unendlich ist.

Satz 3.9 : (Abzählbarkeitseigenschaften)

(1) *Jede Teilmenge einer abzählbaren Menge ist abzählbar.*
(2) *Das Bild einer abzählbaren Menge ist abzählbar.*
(3) *Die Vereinigung einer Folge von abzählbaren Mengen ist abzählbar.*
(4) *Das kartesische Produkt endlich vieler abzählbarer Mengen ist abzählbar.*

Beweis: (1) Sei $M = \{x_0, x_1, x_2, \ldots\}$ abzählbar unendlich und sei N eine unendliche Teilmenge von M. Wähle y_0 als Element x_{n_0} von N mit dem kleinsten Index n_0, dann y_1 als Element x_{n_1} von $N \setminus \{y_0\}$ mit dem kleinsten Index n_1 , weiters y_2 als Element x_{n_2} von $(N \setminus \{y_0\}) \setminus \{y_1\}$ mit dem kleinsten Index n_2, usw. Da jedes Element von N einen Index hat, ist

$$N = \{y_0, y_1, y_2, \ldots\}$$

abzählbar.
(2) Sei $M = \{x_0, x_1, x_2, \ldots\}$ und sei $f : M \to N$ mit $f(M)$ unendlich. Dann ist

$$f(M) = \{\, f(x_n) \mid n = 0, 1, 2, \ldots\}\,.$$

Wähle $y_0 = f(x_0)$, dann $y_1 = f(x_{n_1})$, wobei $x_{n_1} \in M \setminus f^{-1}(y_0)$ den kleinsten Index n_1 hat, weiters $y_2 = f(x_{n_2})$, wobei $x_{n_2} \in (M \setminus f^{-1}(y_0)) \setminus f^{-1}(y_1)$ den kleinsten Index n_2 hat, usw. Dann ist $f(M) = \{y_0, y_1, y_2, \ldots\}$.
(3) Seien

$$\begin{aligned} M_0 &= \{x_{00}, x_{01}, x_{02}, \ldots\} \\ M_1 &= \{x_{10}, x_{11}, x_{12}, \ldots\} \\ M_2 &= \{x_{20}, x_{21}, x_{22}, \ldots\} \\ &\vdots \end{aligned}$$

Dann ist die Abbildung

$$f : \mathbb{N}^2 \to \bigcup_{n=0}^{\infty} M_n\,,\ (n,m) \mapsto x_{nm}\,,$$

surjektiv. Nach Satz 3.7 und (2) ist die Vereinigung abzählbar.
(4) Wir können annehmen, dass die Mengen $M_0, \ldots, M_{k-1}$ abzählbar unendlich sind, ansonsten ergänzen wir Elemente. Aus Numerierungen $\nu_0, \ldots, \nu_{k-1}$ von $M_0, \ldots, M_{k-1}$ bekommt man die bijektive Abbildung

$$M_0 \times \cdots \times M_{k-1} \to \mathbb{N}^k, (x_0, \ldots, x_{k-1}) \mapsto (\nu_0(x_0), \ldots, \nu_{k-1}(x_{k-1})).$$

Nach Satz 3.8 gibt es eine bijektive Abbildung von $\mathbb{N}^k$ nach $\mathbb{N}$. Hintereinanderausführen liefert eine bijektive Abbildung von $M_0 \times \cdots \times M_{k-1}$ nach $\mathbb{N}$.

Beispiel 3.7 : Sei Σ ein endliches Alphabet. Dann ist das Wortmonoid

$$\Sigma^* = \bigcup_{n=0}^{\infty} \Sigma^n$$

abzählbar.

Satz 3.10 : (Diagonalisierung)
Sei Σ ein Alphabet mit mindestens zwei Buchstaben a und b , und seien $s_0, s_1, s_2, \ldots$ eine Folge von Folgen in Σ:

$$\begin{aligned} s_0 &= s_{00}s_{01}s_{02}\ldots \\ s_1 &= s_{10}s_{11}s_{12}\ldots \\ s_2 &= s_{20}s_{21}s_{22}\ldots \\ &\vdots \end{aligned}$$

Dann ist die Folge

$$d_n = \begin{cases} b & \text{falls } s_{nn} = a \\ a & \text{falls } s_{nn} \neq a \end{cases}$$

eine neue Folge.

Beweis: Wenn d keine neue Folge ist, dann gibt es einen Index n mit $d = s_n$, woraus $d_n = s_{nn}$ im Widerspruch zur Konstruktion von d folgt.

Beispiel 3.8 : Die Menge $\mathbb{B}^{\mathbb{N}}$ aller binären Folgen ist nicht abzählbar.

Definition 3.5 : *(Mächtigkeit, Kardinalität)*
Zwei Mengen M und N heissen *gleichmächtig*, wenn es eine bijektive Abbildung $f : M \to N$ gibt. Offensichtlich ist Gleichmächtigkeit eine Äquivalenzrelation, und man nennt die Äquivalenzklasse

$$|M| := \{N \mid N \text{ gleichmächtig wie } M\}$$

die *Mächtigkeit* oder *Kardinalität* der Menge M. Für eine endliche Menge M ist die Menge

$$\{0,1,2,\ldots,\#(M)-1\}$$

ein Repräsentant von $|M|$. Daher werden die Kardinalitäten endlicher Mengen oft mit den natürlichen Zahlen identifiziert.

Satz 3.11 : (Ordnung von Kardinalitäten)
Für Mengen M und N sei

$$|M| \leq |N|,$$

wenn es eine injektive Abbildung $f : M \to N$ gibt. Dann ist $\leq$ eine partielle Ordnung auf den Kardinalitäten.

Beweis: Die Relation $\leq$ ist wohldefiniert, weil für andere Repräsentanten M' von $|M|$ bzw. N' von $|N|$ es bijektive Abbildungen $g : M \to M'$ und $h : N \to N'$ gibt und dann die Abbildung

$$hfg^{-1} : M' \to N'$$

injektiv ist. Klarerweise ist $\leq$ reflexiv und transitiv. Die Antisymmetrie ergibt sich aus dem folgenden Satz.

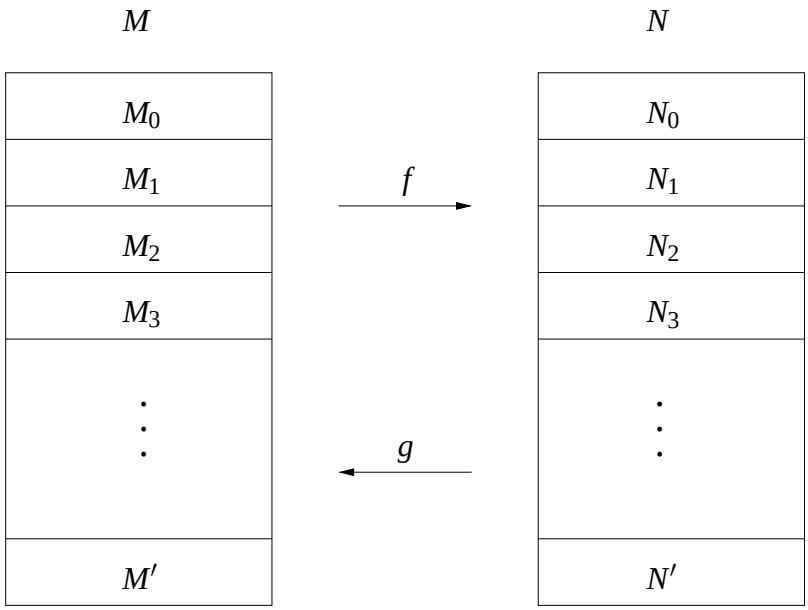
Satz 3.12 : (Satz von Bernstein)
Seien $f : M \to N$ und $g : N \to M$ injektive Abbildungen. Dann existiert eine bijektive Abbildung $h : M \to N$.

Beweis: Wir definieren induktiv Teilmengen $M_0, M_1, M_2, \ldots$ von M sowie Teilmengen $N_0, N_1, N_2, \ldots$ von N durch

$$M_0 := M \setminus g(N) \quad \text{und} \quad N_0 := f(M_0)$$

und, für $n > 0$,

$$M_n := g(N_{n-1}) \quad \text{und} \quad N_n := f(M_n).$$



Wir zeigen durch wohlfundierte Induktion, dass die Mengen $M_0, M_1, \ldots$ paarweise disjunkt sind: Für $n > 0$ ist

$$M_n \subseteq g(N)\,,$$

daher sind M_0 und M_n disjunkt. Für $m, n \in \mathbb{N}$ mit $0 < m < n$ folgt mit der Injektivität von f und g

$$M_m \cap M_n = gf(M_{m-1}) \cap gf(M_{n-1}) = gf(M_{m-1} \cap M_{n-1}) = \emptyset\,.$$

Da f injektiv ist, sind auch die Mengen $N_0, N_1, \ldots$ paarweise disjunkt, und die eingeschränkten Abbildungen

$$f_n : M_n \to N_n\,,\, x \mapsto f(x)\,,$$

sind bijektiv. Seien

$$M' := M \setminus \bigcup_{n=0}^{\infty} M_n \quad \text{und} \quad N' := N \setminus \bigcup_{n=0}^{\infty} N_n\,.$$

Wenn für $y \in N$ ein $n \geq 0$ existiert mit $g(y) \in M_n$, dann ist $n > 0$ und $y \in N_{n-1}$. Somit erhalten wir eine injektive Abbildung

$$g' : N' \to M'\,,\, y \mapsto g(y)\,.$$

Für $x \in M'$ gibt es wegen $M' \subseteq g(N)$ ein $y \in N$ mit $g(y) = x$. Wenn ein $n \geq 0$ existiert mit $y \in N_n$, dann wäre

$$x \in g(N_n) = M_{n+1}$$

im Widerspruch zu $x \in M'$. Somit ist $y \in N'$ und g' bijektiv.

Nach Konstruktion sind die Mengen $M_0, M_1, \ldots . M'$ paarweise disjunkt und

$$\left(\bigcup_{n=0}^{\infty} M_n\right) \cup M' = M\,.$$

Ebenso sind die Mengen $N_0, N_1, \ldots, N'$ paarweise disjunkt und

$$\left(\bigcup_{n=0}^{\infty} N_n\right) \cup N' = N\,.$$

Daher können die bijektiven Abbildungen $f_0, f_1, \ldots, (g')^{-1}$ zu einer bijektiven Abbildung $h : M \to N$ zusammengesetzt werden:

$$h(x) := \begin{cases} f_n(x) & \text{falls } x \in M_n \text{ für ein } n \geq 0 \\ (g')^{-1}(x) & \text{sonst.} \end{cases}$$

Beispiel 3.9 : Seien $M = N = \mathbb{N}$,

$$f : M \to N\,,\, x \mapsto 2x\,, \quad \text{und} \quad g : N \to M\,,\, y \mapsto 2y+1\,.$$

Dann sind

$$\begin{aligned} M_0 &= \{2z \mid z \in \mathbb{N}\} \quad , \quad N_0 = \{4z \mid z \in \mathbb{N}\} \\ M_1 &= \{8z+1 \mid z \in \mathbb{N}\} \quad , \quad N_1 = \{16z+2 \mid z \in \mathbb{N}\} \\ M_2 &= \{32z+5 \mid z \in \mathbb{N}\} \quad , \quad N_2 = \{64z+10 \mid z \in \mathbb{N}\} \\ &\vdots \end{aligned}$$

Satz 3.13 : (Hierarchie der Kardinalitäten)
Für endliche Mengen M und N gilt

$$|M| \leq |N| \;\; \textit{genau dann, wenn} \;\; \#(M) \leq \#(N)\,.$$

Die kleinste Kardinalität einer unendlichen Menge ist $|\mathbb{N}|$, und es gilt

$$|\mathbb{N}| < |\mathbb{B}^{\mathbb{N}}|\,.$$

Beweis: Für endliche Mengen M und N existiert genau dann eine injektive Abbildung $f : M \to N$, wenn M höchstens so viele Elemente wie N hat. Wenn M unendlich ist, dann kann man ein Element x_0 aus M wählen, ein Element x_1 aus $M \setminus \{x_0\}$, ein Element x_2 aus $(M \setminus \{x_0\}) \setminus \{x_1\}$, usw. Somit erhält man eine injektive Abbildung

$$f : \mathbb{N} \to M\,,\, n \mapsto x_n\,,$$

woraus

$$|\mathbb{N}| \leq |M|$$

folgt. Insbesondere ist $|\mathbb{N}| \leq |\mathbb{B}^{\mathbb{N}}|$. Nach Beispiel 2 ist $\mathbb{B}^{\mathbb{N}}$ nicht abzählbar und somit $|\mathbb{N}| < |\mathbb{B}^{\mathbb{N}}|$.

3. Lösen von Rekursionsformeln

Die Laufzeit oder der Speicherplatzbedarf eines Algorithmus hängt üblicherweise von der Eingabegrösse n der Instanz ab. Im Allgemeinen ist es schwierig, für diese Funktion $f(n)$ eine explizite Formel zu finden. In vielen Fällen, besonders bei induktiv definierten Strukturen, ist es leichter, eine Rekursionsformel aufzustellen, die dann gelöst werden soll.

Definition 3.6 : *(erzeugende Funktion)*
Für eine Folge $f : \mathbb{N} \to \mathbb{R}$ heißt die Potenzreihe

$$F(x) := \sum_{n=0}^{\infty} f(n)x^n$$

die *erzeugende Funktion* von f. Die Methode der erzeugenden Funktionen versucht, aus den Rekursionsformeln für $f(n)$ Gleichungen für $F(x)$ herzuleiten und diese mit algebraischen oder analytischen Mitteln zu lösen.

Beispiel 3.10 : Die *Fibonacci-Zahlen* sind rekursiv definiert durch

$$f(n) = \begin{cases} 0 & \text{falls } n = 0 \\ 1 & \text{falls } n = 1 \\ f(n-1) + f(n-2) & \text{falls } n \geq 2\,. \end{cases}$$

Für die erzeugende Funktion $F(x)$ folgt aus obiger Rekursion die Gleichung

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

und durch Partialbruchzerlegung

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

Einsetzen der geometrischen Reihe

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

liefert

$$F(x) = \frac{1}{\sqrt{5}} \left[\sum_{n=0}^{\infty} \left(\frac{1+\sqrt{5}}{2} \right)^n x^n - \sum_{n=0}^{\infty} \left(\frac{1-\sqrt{5}}{2} \right)^n x^n \right].$$

Koeffizientenvergleich ergibt die explizite Formel

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

Beispiel 3.11 : Die Menge der *binären Bäume* über der Menge M wird als formale Sprache mit Hilfe der Klammern „(“ und „)“ induktiv definiert:

(1) Die leere Zeichenkette ε ist ein binärer Baum.
(2) Wenn $x \in M$ und L, R binäre Bäume sind, dann ist (LxR) ein binärer Baum mit Knoten x.

Wir nennen binäre Bäume strukturell gleich, wenn sie durch Umbezeichnen der Elemente von M ineinander übergehen, z.B.

$$((a)b(c)) \text{ und } ((c)a(b)) \,,$$

nicht aber

$$((a)b(c)) \text{ und } (a(b(c))) \,.$$

Offensichtlich ist strukturelle Gleichheit eine Äquivalenzrelation. Nach Konstruktion gilt für die Zahl $f(n)$ der Äquivalenzklassen binärer Bäume mit n Knoten die Rekursionsformel

$$f(n) = \begin{cases} 1 & \text{falls } n = 0 \\ \sum_{k=0}^{n-1} f(k) \cdot f(n-1-k) & \text{falls } n > 0 \,. \end{cases}$$

Für die erzeugende Funktion $F(x)$ folgt

$$F(x) = \sum_{n=0}^{\infty} f(n)x^n = 1 + \sum_{n=1}^{\infty} \left(\sum_{k=0}^{n-1} f(k) \cdot f(n-1-k) \right) x^n = 1 + x \cdot F(x) \cdot F(x)$$

und

$$F(x)^2 - \frac{1}{x} \cdot F(x) + \frac{1}{x} = 0 \,.$$

Lösen der quadratischen Gleichung gibt

$$F(x) = (1 \pm \sqrt{1-4x})/2x \,.$$

Mit der Binomialreihe

$$\sqrt{1-4x} = (1-4x)^{1/2} = \sum_{n=0}^{\infty} \binom{1/2}{n} (-4x)^n = 1 - \sum_{n=1}^{\infty} \frac{2}{n} \binom{2n-2}{n-1} x^n$$

erhält man

$$F(x) = (1 - \sqrt{1-4x})/2x = \sum_{n=0}^{\infty} \frac{1}{n+1} \binom{2n}{n} x^n \,.$$

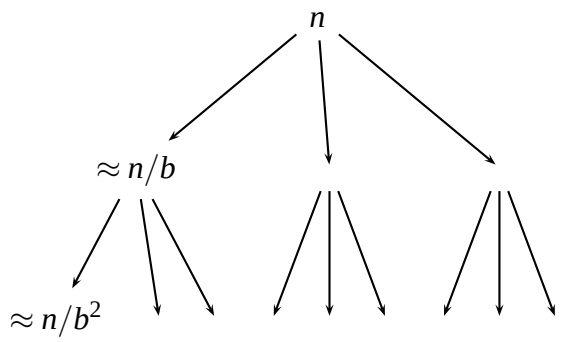
Koeffizientenvergleich liefert

$$f(n) = \frac{1}{n+1} \binom{2n}{n} \,.$$

Beispiel 3.12 : *(Divide-and-Conquer-Algorithmen)*
Ein Algorithmus für ein bestimmtes Problem löst Instanzen bis zur Größe m direkt. Eine Instanz der Größe $n > m$ hingegen zerlegt der Algorithmus in a Teilinstanzen mit Größen

$$\lfloor n/b \rfloor \quad \text{oder} \quad \lceil n/b \rceil \,,$$

löst diese rekursiv und setzt die Teillösungen zu einer Gesamtlösung zusammen. Dabei sind a und b Konstante mit $a \geq 1$ und $b > 1$, und $\lfloor\ \rfloor$ und $\lceil\ \rceil$ bezeichnen die Rundung nach unten bzw. oben.



Die Zeit zum Aufteilen der Instanz und zum Zusammenfügen der Lösungen betrage $f(n)$, die Gesamtzeit sei $t(n)$. Eine natürliche Annahme ist, dass

$$t(n+1) \geq t(n)$$

für alle n gilt. Dann folgt

$$a \cdot t(\lfloor n/b \rfloor) + f(n) \leq t(n) \leq a \cdot t(\lceil n/b \rceil) + f(n)\,.$$

Mit den rekursiv definierten Funktionen

$$t^-(n) := \begin{cases} a \cdot t^-(\lfloor n/b \rfloor) + f(n) & \text{falls } n > m \\ t(n) & \text{falls } n \leq m \end{cases}$$

und

$$t^+(n) := \begin{cases} a \cdot t^+(\lceil n/b \rceil) + f(n) & \text{falls } n > m \\ t(n) & \text{falls } n \leq m \end{cases}$$

ist

$$t^-(n) \leq t(n) \leq t^+(n)$$

für alle n, sodass für die asymptotische Analyse der Laufzeit die Funktionen $t^{\pm}(n)$ an Stelle von $t(n)$ verwendet werden können.

Im Spezialfall $n = m \cdot b^k$ teilt der Algorithmus k-mal, bevor er auf Basisinstanzen der Größe m trifft. Die Gesamtzahl der Basisinstanzen ist

$$a^k = (b^r)^k = (b^k)^r = m^{-r} \cdot n^r\,,$$

wobei

$$r := \log_b a\,.$$

Da jede Basisinstanz in konstanter Zeit gelöst werden kann, liegt die Zeit für die Lösung aller Basisinstanzen in

$$\Theta(n^r)\,,$$

was auch allgemein für $t^{\pm}(n)$ gilt [**1**]. Im folgenden Satz wird diese Zeit mit der Zeit $f(n)$ zum Aufteilen und Zusammenfügen verglichen und entsprechend das asymptotische Wachstum von $t(n)$ angegeben.

Satz 3.14 : (Master-Theorem)

(1) *Wenn* $f \in \mathrm{O}(n^s)$ *für eine reelle Zahl s mit* $s < r$ *, dann ist*

$$t(n) \in \Theta(n^r)\,.$$

(2) *Wenn* $f \in \Theta(n^r)$ *, dann ist*

$$t(n) \in \Theta(n^r \cdot \log n)\,.$$

(3) *Wenn eine reelle Zahl c mit* $c < 1$ *und eine natürliche Zahl k existieren, sodass*

$$a \cdot f(\lceil n/b \rceil) \leq c f(n)$$

für alle n mit $n \geq k$ *, dann ist* $f \in \Omega(n^s)$ *für eine reelle Zahl s mit* $s > r$ *und*

$$t(n) \in \Theta(f)\,.$$

Beweis: [**1**].

Beispiel 3.13 : Für

$$f(n) = n^r \log n$$

liegt das Wachstum von f zwischen Fall (2) und (3), somit ist das Master-Theorem nicht anwendbar.

Beispiel 3.14 : Der Merge-Sort-Algorithmus zerlegt ein n-Tupel von Zahlen in zwei Tupel mit ungefähr $n/2$ Zahlen, sortiert getrennt und mischt die sortierten Tupel zusammen. Wegen $a = b = 2$ und $f \in \Theta(n)$ gibt das Master-Theorem die Laufzeitabschätzung

$$t(n) \in \Theta(n \cdot \log n)\,.$$

KAPITEL 4

Zahlentheorie

1. Rechnen mit ganzen Zahlen

Neben den üblichen Zifferndarstellungen zur Basis 10 (Mensch) oder Basis 2 (Hardware) werden in der Software Basen verwendet, bei denen die Ziffern im nativen Ganzzahlformat abgespeichert werden können. Zum Beispiel verwendet die Langzahl-Bibliothek

GNU Multiple Precision Arithmetic Library (`http://gmplib.org/`)

die Basen $2^{32} \approx 4 \cdot 10^9$ oder $2^{64} \approx 2 \cdot 10^{19}$, damit die Ziffern in die Datentypen 32- bzw. 64-bit unsigned integer von C passen.

Definition 4.1 : *(Ziffernoperationen)*
Sei b eine natürliche Zahl ≥ 2 . Für Zahlen $u, v \in \{0, 1, \ldots, b-1\}$ und $w \in \{0, 1\}$ heißt

$$C(u,v,w) := \begin{cases} 0 & \text{falls } u+v+w < b \\ 1 & \text{sonst} \end{cases}$$

der *Übertrag* (carry) modulo b und

$$S(u,v,w) := \begin{cases} u+v+w & \text{falls } u+v+w < b \\ u+v+w-b & \text{sonst} \end{cases}$$

die *Summe* modulo b. Offenbar gilt

$$S(u,v,w) \in \{0, 1, \ldots, b-1\}$$

und

$$u+v+w = C(u,v,w) \cdot b + S(u,v,w).$$

Satz 4.1 : (Addition natürlicher Zahlen in Zifferndarstellung)
Es seien x und y natürliche Zahlen mit Ziffern

$$x_k x_{k-1} \cdots x_0 \quad \textit{bzw.} \quad y_\ell y_{\ell-1} \cdots y_0$$

zur Basis $b \geq 2$. Ohne Einschränkung der Allgemeinheit nehmen wir $k \geq \ell$ an und setzen

$$y_{\ell+1} := 0\,, y_{\ell+2} := 0\,, \ldots, y_k := 0\,.$$

Dann können die Ziffern der Summe $x+y$ durch den folgenden Algorithmus mit $\mathrm{O}(k)$ Ziffernoperationen berechnet werden:

Setze $c_0 = 0$.
Für i von 0 *bis k wiederhole:*
Setze $z_i = S(x_i, y_i, c_i)$.
Setze $c_{i+1} = C(x_i, y_i, c_i)$.
Falls $c_{k+1} \neq 0$ *, setze* $z_{k+1} = c_{k+1}$.

Beweis: Wir führen eine Induktion nach k. Um Fallunterscheidungen zu vermeiden, setzen wir im letzten Schritt des Algorithmus $z_{k+1} = 0$, falls $c_{k+1} = 0$. Für $k = 0$ ergibt sich

$$z_1 b + z_0 = C(x_0, y_0, 0)b + S(x_0, y_0, 0) = x_0 + y_0 + 0 = x + y\,.$$

Für den Induktionsschluss seien

$$x = \sum_{i \leq k+1} x_i b^i \quad , \quad y = \sum_{i \leq k+1} y_i b^i$$

und z_0, z_1,...,z_{k+2} die Ziffern aus dem Algorithmus. Nach Induktionshypothese gilt

$$\sum_{i \leq k} x_i b^i + \sum_{i \leq k} y_i b^i = \sum_{i \leq k} z_i b^i + c_{k+1} b^{k+1}\,.$$

Damit folgt

$$\sum_{i \leq k+1} x_i b^i + \sum_{i \leq k+1} y_i b^i = \sum_{i \leq k} z_i b^i + (x_{k+1} + y_{k+1} + c_{k+1}) b^{k+1} =$$
$$\sum_{i \leq k} z_i b^i + (S(x_{k+1}, y_{k+1}, c_{k+1}) + C(x_{k+1}, y_{k+1}, c_{k+1}) b)\, b^{k+1} =$$
$$\sum_{i \leq k} z_i b^i + z_{k+1} b^{k+1} + z_{k+2} b^{k+2} = \sum_{i \leq k+2} z_i b^i\,.$$

Für die Subtraktion kann analog ein Algorithmus mit $\mathrm{O}(\max(k, \ell))$ Ziffernoperationen angegeben werden, für die Multiplikation oder die Division mit Rest Algorithmen mit $\mathrm{O}(k \cdot \ell)$ Ziffernoperationen [**3**]. Hier wird noch ein Algorithmus für das Potenzieren diskutiert.

Satz 4.2 : (schnelles Potenzieren)
Sei x eine ganze Zahl oder allgemeiner ein Element eines Ringes, und sei e eine positive ganze Zahl mit Binärziffern

$$e_t e_{t-1} \cdots e_0\,.$$

Dann kann die Potenz $y := x^e$ *durch t-faches Quadrieren und Multiplizieren berechnet werden:*

Setze $y = x$.
Für i von $t-1$ *hinab bis* 0 *wiederhole:*
Setze $y = y^2$.
Falls $e_i = 1$, *setze* $y = y * x$.

Beweis: Für $t = 0$ ist $e = 1$ und der Algorithmus liefert $x^1 = x$. Für $t > 0$ schreiben wir

$$e = \sum_{i=0}^{t} e_i 2^i = d \cdot 2 + e_0 \quad \text{mit} \quad d = \sum_{i=1}^{t} e_i 2^{i-1}.$$

Nach Induktionshypothese hat y vor dem letzten Schleifendurchlauf den Wert x^d. Daher ist das Resultat

$$(x^d)^2 \cdot x^{e_0} = x^e .$$

2. Der euklidische Algorithmus

Seien a, b, c ganze Zahlen mit $b \neq 0$ und $c \neq 0$. Dann ist

$$\frac{a}{b} = \frac{a \cdot c}{b \cdot c} \in \mathbb{Q}$$

eine rationale Zahl [**5**]. Der Übergang von der Darstellung durch das Zahlenpaar $(a \cdot c, b \cdot c)$ zu der durch das Zahlenpaar (a, b) heißt

durch c kürzen.

Rechnet man mit rationalen Zahlen, dann ist es empfehlenswert, alle auftretenden Brüche sofort durch möglichst große Zahlen zu kürzen, um Zähler und Nenner klein zu halten. In diesem Abschnitt wird ein Verfahren zum „optimalen Kürzen“ vorgestellt.

Definition 4.2 : *(Teiler, Vielfaches)*
Sei a eine ganze Zahl ungleich null. Eine ganze Zahl d heißt ein *Teiler* von a, wenn es eine Zahl $c \in \mathbb{Z}$ gibt mit

$$a = c \cdot d .$$

Äquivalente Sprechweisen sind „d teilt a“ oder „a ist *Vielfaches* von d“. Die Teiler $\pm 1, \pm a$ nennt man *triviale Teiler*.

Definition 4.3 : *(größter gemeinsamer Teiler, kleinstes gemeinsames Vielfaches)*
Seien a und b ganze Zahlen ungleich 0.

- Der *größte gemeinsame Teiler* von a und b ist die größte ganze Zahl, die sowohl a als auch b teilt, und wird mit

$$\mathrm{ggT}(a,b) \quad (\gcd(a,b), \text{ greatest common divisor})$$

  bezeichnet.
- Das *kleinste gemeinsame Vielfache* von a und b ist die kleinste positive ganze Zahl, die sowohl Vielfaches von a als auch Vielfaches von b ist, und wird mit

$$\mathrm{kgV}(a,b) \quad (\mathrm{lcm}(a,b), \text{ least common multiple})$$

  bezeichnet.

Lemma 4.1 : (Reduktion des größten gemeinsamen Teilers)
Seien $a,b,c \in \mathbb{Z}$ mit $a \neq 0$, $b \neq 0$ und $a \neq c \cdot b$. Dann gilt

$$\mathrm{ggT}(a,b) = \mathrm{ggT}(|a|,|b|) \quad \textit{und} \quad \mathrm{ggT}(a,b) = \mathrm{ggT}(a - c \cdot b, b)\,.$$

Beweis: Wenn eine ganze Zahl d die Zahl a teilt, dann auch die Zahl $-a$. Somit stimmen die gemeinsamen Teiler von a und b mit den gemeinsamen Teilern von $|a|$ und $|b|$ überein, und die größten gemeinsamen Teiler sind gleich.

Wenn eine ganze Zahl d die Zahlen a und b teilt, dann teilt sie auch die Zahl $a - c \cdot b$. Somit stimmen die gemeinsamen Teiler von a und b mit den gemeinsamen Teilern von $a - c \cdot b$ und b überein, und die größten gemeinsamen Teiler sind gleich.

Satz 4.3 : (euklidischer Algorithmus für ganze Zahlen)
Der größte gemeinsame Teiler zweier ganzer Zahlen ungleich null kann wie folgt berechnet werden:

Ersetze die Zahlen durch ihre Beträge.
Solange die Zahlen verschieden sind, wiederhole:
Ersetze die größere der Zahlen durch die Differenz der größeren und der kleineren.
Die resultierende Zahl ist dann der größte gemeinsame Teiler.

Ersetzt man mehrfaches Abziehen derselben Zahl durch eine Division mit Rest, dann bekommt dieses Verfahren die folgende Form:

Ersetze die Zahlen durch ihre Beträge.
Solange keine der zwei Zahlen ein Teiler der anderen ist, wiederhole:
Ersetze die größere der Zahlen durch ihren Rest
nach Division durch die kleinere.
Der resultierende Teiler ist dann der größte gemeinsame Teiler.

Beweis: Nach dem Lemma bleiben bei jedem Schritt die größten gemeinsamen Teiler der beiden Zahlen gleich. Da in jedem Schleifendurchlauf die Zahlen positiv bleiben, aber ihr Maximum um mindestens 1 sinkt, bricht der Algorithmus nach endlich vielen Schritten ab.

Im euklidischen Algorithmus wird die folgende Strategie zur Lösung von Problemen verwendet: Wenn man ein gegebenes Problem nicht sofort lösen kann, reduziert man das Problem auf ein einfacheres mit derselben Lösungsmenge. Das wiederholt man solange, bis man ein Problem bekommt, dessen Lösungen man kennt. Diese Lösungen sind dann auch die Lösungen des ursprünglichen Problems.

Satz 4.4 : (erweiterter euklidischer Algorithmus)
Seien a und b ganze Zahlen ungleich null. Dann gibt es ganze Zahlen u und v mit

$$u \cdot a + v \cdot b = \mathrm{ggT}(a,b)$$

*(*Darstellung des größten gemeinsamen Teilers als Linearkombination*).*
Diese Zahlen u und v können durch folgenden Algorithmus berechnet werden:

Setze $A = (|a|,1,0)$ *und* $B = (|b|,0,1)$.
Solange B_1 *die Zahl* A_1 *nicht teilt, wiederhole:*
Berechne den ganzzahligen Quotienten q von A_1 *und* B_1.
Setze $C = B$.
Setze $B = A - qC$.
Setze $A = C$.
Setze $u = \mathrm{vz}(a) \cdot B_2$ *und* $v = \mathrm{vz}(b) \cdot B_3$.

Dabei sind $A = (A_1,A_2,A_3)$, $B = (B_1,B_2,B_3)$, $C = (C_1,C_2,C_3)$ *Tripel ganzer Zahlen und* $A - q \cdot C = (A_1 - q \cdot C_1, A_2 - q \cdot C_2, A_3 - q \cdot C_3)$.

Beweis: Wenn zwei Zahlentripel A und B die Eigenschaft

$$T_1 = |a| \cdot T_2 + |b| \cdot T_3$$

haben, dann auch alle Tripel $A - q \cdot B$ mit $q \in \mathbb{Z}$. Die ersten zwei Tripel im Algorithmus haben diese Eigenschaft, daher auch alle anderen auftretenden Tripel. In der ersten Komponente der Tripel wird der euklidische Algorithmus durchgeführt, für das letzte Tripel B gilt daher

$$\mathrm{ggT}(a,b) = |a| \cdot B_2 + |b| \cdot B_3 = (vz(a) \cdot B_2) \cdot a + (vz(b) \cdot B_3) \cdot b\,.$$

Satz 4.5 : (binärer erweiterter euklidischer Algorithmus)
Seien a und b positive ganze Zahlen mit höchstens n Binärziffern. Dann können

$$g := \mathrm{ggT}(a,b) \quad \textit{und} \quad u,v \in \mathbb{Z} \;\; \textit{mit} \;\; ua + vb = g$$

durch Additionen, Subtraktionen und Shifts mit $\mathrm{O}(n^2)$ Bitoperationen berechnet werden:

1. Setze $e = 0$.
2. Solange a und b gerade sind, wiederhole:
  Setze $a = a/2$, $b = b/2$ und $e = e + 1$.
3. Setze $A = (a,1,0)$ und $B = (b,0,1)$.
4. Solange A_1 gerade ist, wiederhole:
  Setze $A_1 = A_1/2$.
  Falls A_2 und A_3 gerade sind,
    setze $A_2 = A_2/2$ und $A_3 = A_3/2$,
  ansonsten falls $A_2 > 0$ ist,
    setze $A_2 = (A_2 - b)/2$ und $A_3 = (A_3 + a)/2$,
  ansonsten
    setze $A_2 = (A_2 + b)/2$ und $A_3 = (A_3 - a)/2$.
5. Solange B_1 gerade ist, wiederhole:
  Setze $B_1 = B_1/2$.
  Wenn B_2 und B_3 gerade sind,
    setze $B_2 = B_2/2$ und $B_3 = B_3/2$,
  ansonsten falls $B_2 > 0$ ist,
    setze $B_2 = (B_2 - b)/2$ und $B_3 = (B_3 + a)/2$,
  ansonsten
    setze $B_2 = (B_2 + b)/2$ und $B_3 = (B_3 - a)/2$.
6. Falls $A_1 > B_1$, setze $A = A - B$ und gehe zu Schritt 4.
7. Falls $A_1 < B_1$, setze $B = B - A$ und gehe zu Schritt 5.
8. Setze $g = B_1 \cdot 2^e$, $u = B_2$ und $v = B_3$.

Beweis: Bei den Reduktionen

$$\mathrm{ggT}(a,b) = \begin{cases} 2\,\mathrm{ggT}(a/2,b/2) & \text{falls } a \text{ und } b \text{ gerade sind} \\ \mathrm{ggT}(a/2,b) & \text{falls } a \text{ gerade und } b \text{ ungerade ist} \\ \mathrm{ggT}(a,b/2) & \text{falls } a \text{ ungerade und } b \text{ gerade ist} \\ \mathrm{ggT}((a-b)/2,b) & \text{falls } a \text{ und } b \text{ ungerade und } a > b \text{ ist} \\ \mathrm{ggT}(a,(b-a)/2) & \text{falls } a \text{ und } b \text{ ungerade und } a < b \text{ ist} \end{cases}$$

sinkt die Summe der Bitlängen um mindestens 1, sodass höchstens $2n-2$ Reduktionen ausgeführt werden. Bei den Operationen für A_2, A_3, B_2, B_3 wird die Bitlänge $n+1$ (plus Vorzeichenbit) nicht überschritten. Wenn c die Binärdarstellung

$$c_k c_{k-1} \cdots c_1 c_0$$

hat, dann haben $2c$ und $c/2$ die Binärdarstellungen

$$c_k c_{k-1} \cdots c_1 c_0 0 \quad \text{bzw.} \quad c_k c_{k-1} \cdots c_1\,,$$

sodass Multiplikationen mit 2 oder Divisionen durch 2 als Shifts ausgeführt werden können. Somit haben alle arithmetische Operationen lineare Komplexität, was quadratische Komplexität für den Algorithmus ergibt.

In den ersten zwei Schritten wird auf den Fall a oder b ungerade reduziert. Mit den neuen Werten a', b' und

$$g' := \mathrm{ggT}(a', b')$$

folgt dann in Schritt 8

$$g = \mathrm{ggT}(a,b) = \mathrm{ggT}(a' \cdot 2^e, b' \cdot 2^e) = g' \cdot 2^e\,.$$

Zur Analyse der Schritte 3-7 betrachten wir die Eigenschaft

$$T_1 = a' \cdot T_2 + b' \cdot T_3 \qquad (*)$$

eines Tripels ganzer Zahlen $T = (T_1, T_2, T_3)$. Die Startwerte für A und B in Schritt 3 haben diese Eigenschaft. Wenn A und B die Eigenschaft $(*)$ besitzen, dann auch die Tripel

$$A - B \quad \text{bzw.} \quad B - A$$

in Schritt 6 bzw. 7. Wenn A_1, A_2 und A_3 gerade sind, dann hat auch das Tripel

$$(A_1/2, A_2/2, A_3/2)$$

in Schritt 4 die Eigenschaft $(*)$. Wenn A_1 gerade ist, A_2 oder A_3 ungerade sind und $A_2 > 0$ ist, dann ist $A_3 \leq 0$, sowohl $A_2 - b'$ als auch $A_3 + a'$ gerade und das Tripel

$$(A_1/2, (A_2 - b')/2, (A_3 + a')/2)$$

in Schritt 4 hat wieder die Eigenschaft $(*)$. Analoges gilt im Fall $A_2 \leq 0$, wo $A_3 > 0$ ist. Schließlich gilt für $A_1 = B_1$ in Schritt 8

$$g' = B_1 = a' \cdot B_2 + b' \cdot B_3$$

und

$$g = B_2 \cdot a + B_3 \cdot b\,.$$

Satz 4.6 : (Berechnung des kleinsten gemeinsamen Vielfaches)
Seien a und b ganze Zahlen ungleich null. Dann gilt

$$\mathrm{kgV}(a,b) = \frac{|a|}{\mathrm{ggT}(a,b)} \cdot |b| = \frac{|b|}{\mathrm{ggT}(a,b)} \cdot |a|\,.$$

Beweis: Offensichtlich ist

$$m := \frac{|a|}{\mathrm{ggT}(a,b)} \cdot |b| = \frac{|b|}{\mathrm{ggT}(a,b)} \cdot |a|$$

ein Vielfaches sowohl von a als auch von b. Wenn eine positive ganze Zahl z gemeinsames Vielfaches von a und b ist, dann gibt es ganze Zahlen c,d mit

$$z = c \cdot a \quad \text{und} \quad z = d \cdot b\,.$$

Nach Satz 4.4 existieren Zahlen u,v mit

$$u \cdot a + v \cdot b = \mathrm{ggT}(a,b)\,.$$

Somit ist

$$z = \frac{u \cdot a + v \cdot b}{\mathrm{ggT}(a,b)} \cdot z = \frac{u \cdot a}{\mathrm{ggT}(a,b)} \cdot z + \frac{v \cdot b}{\mathrm{ggT}(a,b)} \cdot z = \frac{u \cdot a \cdot d \cdot b}{\mathrm{ggT}(a,b)} + \frac{v \cdot b \cdot c \cdot a}{\mathrm{ggT}(a,b)} = \frac{a \cdot b}{\mathrm{ggT}(a,b)} \cdot (u \cdot d + v \cdot c) = m \cdot vz(a \cdot b) \cdot (u \cdot d + v \cdot c)$$

ein Vielfaches von m .

3. Primzahlen

Definition 4.4 : *(Primzahl)*
Eine natürliche Zahl p heißt *Primzahl*, wenn $p \notin \{0,1\}$ und p nur die trivialen Teiler besitzt.

Lemma 4.2 : (Primzahl teilt Faktor)
Sei p eine Primzahl und seien $a,b \in \mathbb{Z}$. Wenn p das Produkt $a \cdot b$ teilt, dann teilt p auch einen der Faktoren a und b.

Beweis: Sei $c \in \mathbb{Z}$ mit

$$c \cdot p = a \cdot b\,.$$

Wenn p die Zahl a nicht teilt, dann ist $\mathrm{ggT}(a,p) = 1$. Daher gibt es ganze Zahlen u und v mit

$$1 = u \cdot a + v \cdot p\,.$$

Es folgt

$$b = b \cdot u \cdot a + b \cdot v \cdot p = u \cdot c \cdot p + b \cdot v \cdot p = (u \cdot c + b \cdot v) \cdot p\,,$$

somit ist p ein Teiler von b.

Satz 4.7 : (Primfaktorzerlegung)
Jede ganze Zahl größer als 1 *kann als Produkt von Primzahlen geschrieben werden. Diese Primzahlen heißen* Primfaktoren *der Zahl und sind bis auf die Reihenfolge eindeutig bestimmt.*

Beweis: Es sei a eine ganze Zahl größer als 1. Wir zeigen die Existenz der Primfaktorzerlegung durch Induktion nach a . Wenn $a = 2$, dann ist a eine Primzahl. Wenn $a > 2$, dann ist a entweder eine Primzahl oder es gibt ganze Zahlen b, c mit

$$a = b \cdot c \quad \text{und} \quad 1 < b < a \quad \text{und} \quad 1 < c < a\,.$$

Nach Induktionsannahme sind die Zahlen b und c Produkte von Primzahlen, somit auch die Zahl a .

Wir beweisen noch die Eindeutigkeit der Primfaktorzerlegung. Seien

$$a = p_1 p_2 \cdots p_k = q_1 q_2 \cdots q_\ell$$

zwei Zerlegungen von a in Primfaktoren. Wir zeigen durch wohlfundierte Induktion, dass die Primfaktoren der zwei Zerlegungen bis auf die Reihenfolge gleich sind. Da p_1 das Produkt $q_1 q_2 \cdots q_\ell$ teilt, gibt es nach Lemma 4.2 eine Zahl $j \in \{1, \ldots, \ell\}$ mit $p_1 = q_j$. Somit ist

$$p_2 \cdots p_k = \prod_{\substack{1 \le i \le \ell \\ i \ne j}} q_i\,,$$

und die Behauptung folgt aus der Induktionsannahme.

Während der größte gemeinsame Teiler mit dem binären euklidischen Algorithmus schnell berechnet werden kann, ist die Berechnung der Primfaktorzerlegung im Allgemeinen aufwendig. Daher sollten Rechenverfahren, in denen Zahlen in Primfaktoren zerlegt werden müssen, nach Möglichkeit vermieden werden.

Satz 4.8 : *Es gibt unendlich viele Primzahlen.*

Beweis: Wir nehmen an, die Zahlen $p_1, p_2, \ldots, p_n$ wären sämtliche Primzahlen. Sei

$$q := \prod_{i=1}^{n} p_i\,.$$

Dann ist $q+1$ größer als jede Primzahl und somit keine Primzahl. Nach Satz 4.7 gibt es eine Primzahl p, die $q+1$ teilt. Da p auch q teilt, würde p auch 1 teilen, was im Widerspruch zur Primheit von p steht.

Satz 4.9 : (Berechnung von ggT und kgV aus der Primfaktorzerlegung)
Seien a und b positive ganze Zahlen mit Primfaktorzerlegungen

$$a = \prod_{i=1}^{n} p_i^{e_i} \quad \textit{und} \quad b = \prod_{i=1}^{n} p_i^{f_i},$$

wobei $p_1, \ldots, p_n$ *paarweise verschiedene Primzahlen und* $e_1, \ldots, e_n, f_1, \ldots, f_n$ *natürliche Zahlen sind. Dann gilt*

$$\mathrm{ggT}(a,b) = \prod_{i=1}^{n} p_i^{\min(e_i, f_i)} \quad \textit{und} \quad \mathrm{kgV}(b) = \prod_{i=1}^{n} p_i^{\max(e_i, f_i)},$$

wobei $\min(e_i, f_i)$ *bzw.* $\max(e_i, f_i)$ *die kleinere bzw. die größere der zwei Zahlen* e_i *und* f_i *bezeichnet.*

Beweis: Die Zahl

$$d := \prod_{i=1}^{n} p_i^{\min(e_i, f_i)}.$$

teilt offenbar die Zahlen a und b. Da nach Satz 4.7 die Primfaktorzerlegungen von a und b eindeutig sind, kann ihr größter gemeinsamer Teiler keine anderen Primfaktoren als $p_1, \ldots, p_n$ enthalten. Somit darf p_i in $\mathrm{ggT}(a,b)$ nur $\min(e_i, f_i)$-mal auftreten. Daher ist $d = \mathrm{ggT}(a,b)$. Die Behauptung für $\mathrm{kgV}(a,b)$ folgt aus Satz 4.6.

4. Restklassen

Definition 4.5 : *(Kongruenz, Restklassen)*
Sei n eine positive ganze Zahl. Zwei ganze Zahlen a, b heißen *kongruent modulo n*, in Zeichen

$$a \equiv b \bmod n,$$

wenn ihre Reste nach Division durch n

$$a \bmod n \quad \text{und} \quad b \bmod n$$

gleich sind. Kongruenz modulo n ist eine Äquivalenzrelation. Die Äquivalenzklasse einer ganzen Zahl a ist

$$\overline{a} := \{a + z \cdot n \mid z \in \mathbb{Z}\}$$

und wird die *Restklasse* von a modulo n genannt. Die Menge aller Restklassen modulo n wird mit

$$\mathbb{Z}/n \qquad \text{(Sprechweise: } \mathbb{Z} \text{ modulo } n\text{)}$$

bezeichnet. Übliche Repräsentantensysteme sind die *kleinsten nicht-negativen Reste*

$$\{0,1,2,\ldots,n-1\}$$

oder die *absolut-kleinsten Reste*

$$\begin{cases} \{-n/2+1,\ldots,-1,0,1,\ldots,n/2\} & \text{falls } n \text{ gerade} \\ \{-(n-1)/2,\ldots,-1,0,1,\ldots,(n-1)/2\} & \text{falls } n \text{ ungerade.} \end{cases}$$

Satz 4.10 : (Restklassenarithmetik)
Die Abbildungen

$$+ : \mathbb{Z}/n \times \mathbb{Z}/n \to \mathbb{Z}/n\,,\ (\overline{a},\overline{b}) \mapsto \overline{a}+\overline{b} := \overline{a+b}\,,$$

und

$$\cdot : \mathbb{Z}/n \times \mathbb{Z}/n \to \mathbb{Z}/n\,,\ (\overline{a},\overline{b}) \mapsto \overline{a}\cdot\overline{b} := \overline{a\cdot b}\,,$$

sind wohldefiniert. Mit diesen Rechenoperationen ist $\mathbb{Z}/n$ ein kommutativer Ring. Das Nullelement von $\mathbb{Z}/n$ ist $\overline{0}$, das Einselement ist $\overline{1}$.

Beweis: Wir zeigen zuerst, dass die Operationen wohldefiniert sind. Seien $a,c,b,d \in \mathbb{Z}$ mit

$$\overline{a}=\overline{c} \quad \text{und} \quad \overline{b}=\overline{d}\,.$$

Dann sind $a-c$ und $b-d$ Vielfache von n. Wegen

$$(a+b)-(c+d) = (a-c)+(b-d)$$

ist auch $(a+b)-(c+d)$ ein Vielfaches von n. Somit gilt

$$\overline{a+b} = \overline{c+d}\,,$$

was die Wohldefiniertheit von $+$ beweist. Wegen

$$ab-cd = a(b-d)+(a-c)d$$

ist

$$\overline{a\cdot b} = \overline{c\cdot d}\,,$$

was die Wohldefiniertheit von $\cdot$ beweist. Da $+$ und $\cdot$ über die Addition und Multiplikation ganzer Zahlen definiert sind, erfüllen sie die Rechenregeln eines kommutativen Rings.

Beispiel 4.1 : In der Programmiersprache C wird im Datentyp `unsigned int` im Restklassenring $\mathbb{Z}/n$ mit $n=2^{32}$ bzw. $n=2^{64}$ gerechnet. Als Summe von $2^{32}-1$ und 1 ergibt sich dann 0.

Satz 4.11 : (Invertierbarbeit von Restklassen)
Sei n eine positive ganze Zahl und a eine ganze Zahl ungleich null.

(1) *Die Restklasse von a modulo n ist genau dann invertierbar, wenn*

$$\mathrm{ggT}(a,n) = 1\,.$$

In diesem Fall können mit dem erweiterten euklidischen Algorithmus ganze Zahlen u,v mit $u \cdot a + v \cdot n = 1$ berechnet werden, und dann ist

$$\overline{a}^{-1} = \overline{u}\,.$$

Bezeichne

$$(\mathbb{Z}/n)^{\times}$$

die Gruppe der invertierbaren Restklassen modulo n.

(2) *Der Ring $\mathbb{Z}/n$ ist genau dann ein Körper, wenn n eine Primzahl ist.*

Beweis: (1) Wenn $\mathrm{ggT}(a,n) = 1$ und $u \cdot a + v \cdot n = 1$ ist, dann ist

$$\overline{1} = \overline{u} \cdot \overline{a} + \overline{v} \cdot \overline{n} = \overline{u} \cdot \overline{a} + \overline{v} \cdot \overline{0} = \overline{u} \cdot \overline{a}\,.$$

Wenn umgekehrt $\overline{a}$ invertierbar ist, dann gibt es eine ganze Zahl b mit

$$\overline{a} \cdot \overline{b} = \overline{1} \quad \text{und} \quad \overline{ab-1} = \overline{0}\,.$$

Somit ist n ein Teiler von $ab-1$. Da $\mathrm{ggT}(a,n)$ sowohl a als auch $1 - a \cdot b$ teilt, ist $\mathrm{ggT}(a,n) = 1$.
(2) Sei n eine Primzahl und a eine ganze Zahl. Dann ist entweder a ein Vielfaches von n oder $\mathrm{ggT}(a,n) = 1$. Somit folgt die Behauptung aus (1).

Beispiel 4.2 : Die Zahl 6 ist nicht invertierbar modulo 26, das Inverse von 5 modulo 26 ist 21. Der Ring $\mathbb{Z}/2$ ist ein Körper mit zwei Elementen, der Ring $\mathbb{Z}/256$ ist kein Körper.

Satz 4.12 : (der kleine Satz von Fermat)
Sei p eine Primzahl und sei a eine ganze Zahl, die nicht von p geteilt wird. Dann gilt

$$a^{p-1} \equiv 1 \mod p\,.$$

Beweis: Die Restklassen

$$\overline{1a}, \overline{2a}, \ldots, \overline{(p-1)a}$$

sind alle von $\overline{0}$ und untereinander verschieden und damit eine Permutation von

$$\overline{1}, \overline{2}, \ldots, \overline{p-1}\,.$$

Somit ist

$$\overline{1 \cdot 2 \cdots (p-1)} \cdot \overline{1} = \overline{1a} \cdot \overline{2a} \cdots \overline{(p-1)a} = \overline{1 \cdot 2 \cdots (p-1)} \cdot \overline{a^{p-1}}\,.$$

Kürzen liefert das Ergebnis.

Satz 4.13 : (chinesischer Restsatz)
Seien p und q positive ganze Zahlen mit $\mathrm{ggT}(p,q) = 1$ *, und seien a und b beliebige ganze Zahlen. Dann hat das Kongruenzensystem*

$$\begin{aligned} x &\equiv a \mod p \\ x &\equiv b \mod q \end{aligned}$$

die eindeutige Lösung

$$x \equiv vqa + upb \mod pq\,,$$

wobei die ganzen Zahlen u und v mit

$$up + vq = 1$$

durch den erweiterten euklidischen Algorithmus berechnet werden können.

Beweis: Nach Konstruktion gilt

$$x \equiv vqa \equiv 1a \equiv a \mod p \qquad \text{und} \qquad x \equiv upb \equiv 1b \equiv b \mod q\,.$$

Beispiel 4.3 : Die eindeutige Lösung des Kongruenzensystems

$$\begin{aligned} x &\equiv 1 \mod 5 \\ x &\equiv 2 \mod 7 \end{aligned}$$

ist

$$x \equiv 16 \mod 35\,.$$

Anwendung findet die modulare Arithmetik zum Beispiel bei Kryptosystemen, wo die Verschlüsselungsfunktion leicht zu berechnen aber schwer umzukehren sein soll (siehe Lehrveranstaltung „Wahrscheinlichkeitsrechnung und Informationstheorie“).

Anhang

Das griechische Alphabet

A	α	alpha	N	ν	nü
B	β	beta	Ξ	ξ	xi
Γ	γ	gamma	O	o	omikron
Δ	δ	delta	Π	π, ϖ	pi
E	ε	epsilon	P	ρ	rho
Z	ζ	zeta	Σ	σ, ς	sigma
H	η	eta	T	τ	tau
Θ	θ, ϑ	theta	Υ	υ	ypsilon
I	ι	iota	Φ	ϕ, φ	phi
K	κ	kappa	X	χ	chi
Λ	λ	lambda	Ψ	ψ	psi
M	μ	mü	Ω	ω	omega

Der ASCII-Code

Nr.	Zeichen	Nr.	Zeichen	Nr.	Zeichen	Nr.	Zeichen
0	NUL	32		64	@	96	‘
1	SOH	33	!	65	A	97	a
2	STX	34	"	66	B	98	b
3	ETX	35	#	67	C	99	c
4	EOT	36	$	68	D	100	d
5	ENQ	37	%	69	E	101	e
6	ACK	38	&	70	F	102	f
7	BEL	39	’	71	G	103	g
8	BS	40	(	72	H	104	h
9	HT	41	)	73	I	105	i
10	LF	42	*	74	J	106	j
11	VT	43	+	75	K	107	k
12	FF	44	,	76	L	108	l
13	CR	45	-	77	M	109	m
14	SO	46	.	78	N	110	n
15	SI	47	/	79	O	111	o
16	DLE	48	0	80	P	112	p
17	DC1	49	1	81	Q	113	q
18	DC2	50	2	82	R	114	r
19	DC3	51	3	83	S	115	s
20	DC4	52	4	84	T	116	t
21	NAK	53	5	85	U	117	u
22	SYN	54	6	86	V	118	v
23	ETB	55	7	87	W	119	w
24	CAN	56	8	88	X	120	x
25	EM	57	9	89	Y	121	y
26	SUB	58	:	90	Z	122	z
27	ESC	59	;	91	[	123	{
28	FS	60	<	92	\	124	|
29	GS	61	=	93	]	125	}
30	RS	62	>	94	^	126	~
31	US	63	?	95	_	127	DEL

Steuerzeichen: ACK (Acknowledge), BEL (Bell), BS (Backspace), CAN (Cancel), CR (Carriage Return), DC1 (Device Control 1), DC2 (Device Control 2), DC3 (Device Control 3), DC4 (Device Control 4), DEL (Delete), DLE (Data Link Escape), EM (End of Medium), ENQ (Enquiry), EOT (End of Transmission), ESC (Escape), ETB (End of Transmission Block), ETX (End of Text), FS (File Separator), FF (Form Feed), GS (Group Separator), HT (Horizontal Tabulation), LF (LineFeed), NAK (Negative Acknowledge), NUL (Null), RS (Record Separator), SI (Shift In), SO (Shift Out), SOH (Start of heading), STX (Start of Text), SUB (Substitute), SYN (Synchronous Idle), US (Unit Separator), VT (Vertical Tabulation)

Literaturverzeichnis

[1] T.H.Cormen, C.E.Leiserson, R.L.Rivest, C.Stein, *Introduction to Algorithms*, MIT Press, Cambridge 2001.

[2] J.L.Hein, *Discrete Structures, Logic, and Computability*, Jones and Bartlett Publishers, London 2002.

[3] A.J.Menezes, P.C.van Oorschot, S.A.Vanstone, *Handbook of Applied Cryptography*, CRC Press, Boca Raton 1997. `http://www.cacr.math.uwaterloo.ca/hac/`.

[4] M.Oberguggenberger, A. Ostermann, *Einführung in die Mathematik 2*, Institut für Mathematik, Innsbruck 2008.

[5] F.Pauer, *Einführung in die Mathematik 1*, Institut für Mathematik, Innsbruck 2008.