Vorlesungsmitschriften

Grundvorlesungen

Universität Bonn *
Semester 1

Fabian

19. April 2022

Online aufrufbar auf
https://github.com/git-fabus/lecutres/blob/main/notes.pdf.
Veränderungsvorschläge und Verbesserungen bitte an git-fabus@uni-bonn.de.

*

Vorwort

Dies sind meine persönlichen Vorlesungsmitschriften. Das Dokument ist in 3 große Teilbereiche unterteilt, dazu zählen:

- Algorithmische Mathematik 1
- Analysis 1
- Lineare Algebra 1

Das Dokument wird stetig verbessert und es werden nach und nach immer mehr Aufgaben, Lösungen etc. erscheinen.

Inhaltsverzeichnis

Teil I

Analysis 1

Grundlagen

1. Aussagenlogik

1.1 Definition (Aussage). Eine *Aussage* ist eine Behauptung, die eindeutig feststeht, so ist diese entweder *wahr* oder *falsch*

1.2 Definition. Falls A und B Aussagen gilt: Die Wahrheitstafeln dazu sehen wie folgt

Symbol	Gesprochen
$\neg A$	"nicht A"
$A \vee B$	"A oder B"
$A \wedge B$	"A und B"
$A \implies B$	"aus A folgt B", "A impliziert B"
$A \iff B$	"A ist äquivalent zu B", "A genau dann wenn B"

aus:

A	B	$\neg A$	$A \vee B$	$A \wedge B$	$A \implies B$	$A \iff B$
w	w	f	w	w	w	w
w	f	f	w	f	f	f
f	w	w	w	f	w	f
f	f	w	f	f	w	w

1.1. Gesetze der Aussagenlogik

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

Wir können diese Gesetze durch Wahrheitstafeln beweisen. Das wichtigste Gesetzt ist die Kontraposition

1.3 Satz (Kontraposition). Falls A und B Aussagen gilt:

$$(A \implies B) \iff (\neg B \implies \neg A)$$

1.4 Bemerkung. Der Satz 1.3 wird oft benutzt um etwas zu beweisen.

Noch ein kurzer Beweis zu

$$(A \implies B) \iff (\neg A \vee B)$$

Beweis. Durch eine Wahrheitstafel. □

A	B	$A \implies B$	$\neg A \vee B$
w	w	w	w
w	f	f	f
f	w	w	w
f	f	w	w

2. Prädikatenlogik

Aussagen, die von freien Variablen abhängen.

2.5 Beispiel. Zwei Ausssagenformen

- Es gibt eine reele Zahl x mit $x^2 + 1 = 0$
- Für alle natürlichen Zahlen x gilt: $x^2 - 1 = (x - 1)(x + 1)$

2.6 Notation (Quantoren). Wir verwenden die Kurzformen

- $\exists$ für "es gibt ein"
- $\forall$ für "für alle"

Regeln:

$$\begin{aligned} \neg(\forall_x : A(x)) &\iff (\exists_x : \neg A(x)) \\ \neg(\exists_x : A(x) &\iff (\forall_x : \neg A(x)) \\ \neg(\forall_x \forall_y : A(x,y)) &\iff (\forall_x \exists_y : \neg A(x,y)) \end{aligned}$$

2.7 Warnung. Es gilt nicht:

$$\forall_{x \in M} : A(x) \implies \exists_{x \in M} : A(x)$$

Da M z.B eine leere Menge sein kann.

3. Mengenlehre

3.8 **Definition** (Menge nach Cantor). Eine Menge ist die Zusammenfassung bestimmter wohlunterschiedener Objekte, welche die Elemente der Menge genannt werden, unserer Anschauung oder unseres Denkens zu einem Ganzen.

3.9 **Beispiel.** Die typischen Zahlenmengen kennt man bereits:

- Menge der natürlichen Zahlen $\mathbb{N}$
- Menge der rationalen Zahlen $\mathbb{Q}$
- Menge der reelen Zahlen $\mathbb{R}$

3.10 **Notation.** Wir bezeichnen:

Symbole	Gesprochen
$x \in M$	x Element der Menge M
$M \subset N$	M Teilmenge von N
$M \subseteq N$	M strikte Teilmenge von N
$\emptyset = \{\}$	leere Menge
$M \cap N$	Schnittmenge von M und N
$M \cup N$	Vereinigung von M und N
$M \setminus N$	M ohne N
N^c	Komplement der Menge
$M \cap N = \{\}$	$\Longrightarrow$ M und N sind disjunkt

3.11 **Bemerkung** (Grenzen der naiven Mengenlehre). Nicht erlaubt ist die Konstruktion der Menge aller Mengen, die sich selbst als Element enthalten.

4. Vollständige Induktion

Das Beweisprinzip der vollständigen Induktion ist:
$\forall_{n\in\mathbb{N}} : A(n)$ falls

1. Induktionsanfang (IA, IV) : $A(1)$ gilt

2. Induktionsschritt (IS) : $\forall_{n\in\mathbb{N}}$: falls $A(n)$ gilt, dann auch $A(n+1)$

4.12 **Satz.** $\forall_{n\in\mathbb{N}} : 1 + 2 + 3 + \ldots + n = \frac{n(n+1)}{2}$

Beweis. per Induktion.

$\Diamond$ <u>IA $n = 1$</u>

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

◊ IS $n \to n+1$ Angenommen $A(n)$ ist wahr, dann muss

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

gelten.

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

nach Induktionsannahme.Dies ist nichts anderes als:

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

□

Um nicht immer $1 + 2 + \ldots + n$ zu schreiben, verwenden wir ab sofort das Summenzeichen.

4.13 Notation. Für ganze Zahlen m, n und $a_k \in \mathbb{R}$ ist

$$\sum_{k=m}^{n} a_k = a_m + a_{m+1} + \ldots + a_n$$

$$\sum_{k=m}^{m-1} a_k = 0$$

4.14 Beispiel. Der Satz 4.12 lässt sich wie folgt schreiben:

$$\sum_{k=1}^{n} k = \frac{n(n+1)}{2}$$

4.15 Satz. Für alle $n \in \mathbb{N}$ gilt:

$$\sum_{k=1}^{n} (2k-1) = n^2$$

Beweis. per Induktion:

◊ IA $n = 1$

$$1 = 1^2$$

◊ IS $n \to n+1$ $A(n)$ gilt, dann muss auch

$$\sum_{k=1}^{n+1} (2k-1) = (n+1)^2$$

gelten.

$$\sum_{k=1}^{n+1} (2k-1) = n^2 + (2(n+1)-1) = n^2 + 2n + 1$$

Damit gilt die Behauptung

□

4.16 Satz (Geometrische Summenformel). Für alle $x \in \mathbb{R} \setminus \{1\}$ und für alle $n \in \mathbb{N}$ gilt:

$$\sum_{k=1}^{n} x^k = \frac{1 - x^{n+1}}{1 - x}$$

Hierbei gilt $x^0 = 1$ auch für $x = 0$.

Beweis. per Induktion.

◊ IA $n = 1$

$$1 + x = \frac{1 - x^2}{1 - x}$$

◊ IS $n \to n + 1$ Angenommen $A(n)$ gilt.

$$\begin{aligned} \sum_{k=0}^{n} x^k + x^{n+1} &= \frac{1 - x^{n+1}}{1 - x} + x^{n+1} \\ &= \frac{1 - x^{n+2}}{1 - x} \end{aligned}$$

□

4.17 Bemerkung. Statt den Anfang bei $n = 1$ kann der Induktionsbeweis auch bei jeder Zahl $n_0 \in \mathbb{Z}$ starten und dann die Aussagen für alle $n \in \mathbb{Z}, n \geq n_0$ liefern.

Analog zu dem Summenzeichen gibt es auch ein Produktzeichen:

4.18 Notation (Produktzeichen). Für $m, n \in \mathbb{N}, a_k \in \mathbb{R}, m \leq n$ gilt:

$$\prod_{k=m}^{n} a_k = a_m \cdot a_{m+1} \cdot \ldots * a_n$$

$$\prod_{k=m}^{m-1} a_k = 1$$

Rekursiv $\forall_{k \geq m}$:

$$\prod_{k=m}^{n} a_k = a_n \cdot \prod_{k=m}^{n-1} a_k$$

4.19 Definition (Fakultät). Die Fakultät ist definiert als:

$$n! = \prod_{k=1}^{n} k = 1 \cdot 2 \cdot \ldots \cdot n$$

5. Binominal-Koeffizienten

Grundlegend handelt es sich hierbei um eine "Funktion" um Grundaufgaben der Kombinatorik zu lösen. Er gibt an, auf wie viele verschiedene Arten k Objekte aus einer Menge von n verschiedenen Objekte ausgewählt werden können.

5.20 Notation. Wir sagen für $\binom{n}{k}$ "n über k" oder "n aus k"

5.21 Definition. Wir schreiben:

$$\binom{n}{k} = \frac{n + (n+1) \cdot \ldots \cdot n(k+1)}{k!} = \prod_{j=1}^{k} \frac{(n+1-j)}{j}$$

falls $k > n$ dann gilt: $\binom{n}{k} = 0$, ebenfalls gilt: $\binom{n}{0} = \binom{n}{n} = 1$ und $\binom{n}{n-1} = n$
Sollte $k \leq n$ sein gilt:

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} = \binom{n}{n-k}$$

5.22 Lemma. $\forall_{n \in \mathbb{N}_0, \forall_{k \in \mathbb{Z}}}$ gilt:

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

Beweis. Für $k \leq 0$ klar.
Sei $k \geq 0$ dann gilt:

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

□

Pascalsche Dreieck

$n=0$						1						
$n=1$					1		1					
$n=2$				1		2		1				
$n=3$			1		3		3		1			
$n=4$		1		4		6		4		1		
$n=5$	1		5		10		10		5		1	
$n=6$	1		6		15		20		15		6	1

5.23 Satz (Binomischer Lehrsatz). Für alle $x, y \in \mathbb{R}$ und für alle $n \in \mathbb{N}_0$ gilt:

$$(x+y)^n = \sum_{k=0}^{n} \binom{n}{k} \cdot x^{n-k} \cdot y^k$$

Beweis. durch Induktion:

◊ IA $n = 1$:

$$(x+y)^1 = x + y$$

◊ IS $n \to +1$:

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

□

5.24 Korollar. Es folgt direkt:

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

5.25 Satz. Die Anzahl der Anordnungen von n verschiedenen Elementen ist $n!$.

Beweis. per Indutkion:

◊ IA $n + 1$ a_1, also eine Anordnung

◊ IS $n \to n + 1$ Sei $N(n)$ die Anzahl der Anordnungen von n Elementen.
Annahme: $N(n) = n!$
Gegeben sei $n + 1$ Elemente $\implies$ $n + 1$ Möglichkeiten für den ersten Platz. Jede dieser Möglichkeiten erlaubt $N(n) = n!$ Möglichkeiten zu Anfang. Daraus folgt es gibt insgesamt $n!(n + 1) = (n + 1)!$ Optionen.

□

5.26 Satz. Die Anzahl der k-elementigen Teilmengen einer n-elementigen Menge ist $\binom{n}{k}$.

Beweis. per Indutkion nach n.

◊ IA $n = 0$

$$\binom{0}{0} = 1$$

◊ IS $n \to n + 1$ Sei $M = \{a_1, \ldots, a_{n+1}\}$ Für $0 \leq k \leq n + 1$ gilt:
k-elementigen Teilmengen zerfallen in 2 Klassen:

- Teilmengen mit a_{n+1}
- Teilmengen ohne a_{n+1}

Für ersten gilt: Mengen $\{a_{n+1}\} \cup M_k$ mit beliebiger k-elementiger Teilmenge von $\{a_1, \ldots, a_n\} \implies \exists \binom{n}{k-1}$ Teilmengen.
Für zweitens gilt: beliebige k-elementige Teilmengen von $\{a_1, \ldots, a_n\} \implies \binom{n}{k}$ Teilmengen.
Also insgesamt:

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

Teilmengen. Damit ist $A(n+1)$ bewiesen.

□

5.27 Definition. Für eine Menge M ist die Potenzmenge $\mathfrak{P}(M)$ die Menge aller Teilmengen von M.

5.28 Beispiel. Für $M = \{a_1, a_2\}$ ist die Potenzmenge von M $\mathfrak{P}(M) = \{\emptyset, \{a_1\}, \{a_2\}, \{a_1, a_2\}\}$.

5.29 Satz. Die Potenzmenge $\mathfrak{P}(M)$ einer n-elementigen Menge M hat 2^n Elemente

Beweis. Es gibt $\binom{n}{k}$ Teilmengen mit k-Elementen. Also insgesamt:

$$\sum_{k=0}^{n} \binom{n}{k} = 2^n$$

Teilmengen. □

Körper der reelen Zahlen

$\mathbb{R}$ ist Körper, angeordnet, archimedisch, vollständig.

6. Körperaxiome

6.1 Definition ($\mathbb{R}$). $\mathbb{R}$ ist eine Menge, auf der zwei Verknüpfungen definiert sind:

$$+: \mathbb{R} \times \mathbb{R} \to \mathbb{R}$$
$$\cdot: \mathbb{R} \times \mathbb{R} \to \mathbb{R}$$

Dabei wird jedem Paar (x, y) mit $x, y \in \mathbb{R}$ ein Element $x + y \in \mathbb{R}$ (bzw $x \cdot y \in \mathbb{R}$ zu geordnet.

6.2 Bemerkung. Die zwei Verknüpfungen sind die zwei Grundrechenarten "plus" und "mal". Jedoch müssen wir die typischen Rechenregeln, die wir bereits kennen erst anhand der Körperaxiome zeigen.

6.3 Definition (Körper). $\mathbb{K}$ ist ein Körper falls folgende Axiome gelten: Es sei $a, b, c \in \mathbb{K}$. Für die Addition:

A1 $\forall_{a,b,c} : a + (b + c) = (a + b) + c$

A2 $\forall_{a,b} : a + b = b + a$

A3 $\exists_{0 \in \mathbb{K}} : a + 0 = a$

A4 $\forall_a \exists_{-a \in \mathbb{K}} : a + (-a) = 0$

Analog weiter für die Multiplikation:

M1 $\forall_{a,b,c} : (a \cdot b) \cdot c = a \cdot (c \cdot b)$

M2 $\forall_{a,b} : a \cdot b = b \cdot a$

M3 $\exists_{1 \in \mathbb{K}} \forall_a : 1 \neq 0 \wedge 1 \cdot a = a$

M4 $\forall_{a \neq 0} \exists_{a^{-1} \in \mathbb{K}} : a \cdot a^{-1} = 1$

Außerdem gilt das Distributivgesetz:

D $\forall_{a,b,c} : (a + b) \cdot c = a \cdot c + b \cdot c$

6.4 Notation. Vereinfachend schreiben wir auch:

- $a \cdot b = ab$
- $a^{-1} = \frac{1}{a}$
- $a \cdot b^{-1} = \frac{a}{b}$

Rechenregeln für Körper

6.5 Bemerkung. Die Eigenschaften A1-A4 besagen außerdem, dass $(\mathbb{R}, +)$ eine Gruppe ist.

a) Eindeutigkeit der 0

Beweis. $0^* = 0^* + 0 = 0 + 0^* = 0$ □

b) Das Negativ $-x$ ist eindeutig.

Beweis. Seien x'', x' für $(x' + x = 0) \wedge (x + x'' = 0)$ gilt.
$\implies x' = x' + 0 = x' + x(-x) = x'' + x + (-x) = x''$ □

c) $-0 = 0$

Beweis. $0 = 0 + (-0) = (-0) + 0 = -0$ □

d) $\forall_x : -(-x) = x$

Beweis. $0 = x+(-x) = (-x)+x$ und Eindeutigkeit des Negativen folgt: $x = -(-x)$ □

e) $\forall_{a,b,c \in \mathbb{R}} : a + b = c \iff a = c - b$

f) $\forall_{a,b \in \mathbb{R}} : -(a + b) = (-a) + (-b)$

6.6 Bemerkung. Die Eigenschaften M1-M4 besagen, dass $\mathbb{R} \setminus \{0\}, \cdot)$ auch eine Gruppe ist. Demnach gelten die Eigenschaften a-f eigentlich fast Analog für die Multiplikation.

Wegen der Distributivität folgt außerdem:

g) $\forall_{a,b,c \in \mathbb{R}} : (a + b) \cdot c = c \cdot a + c \cdot b$

h) $\forall_{a \in \mathbb{R}} : x \cdot 0 = 0$

Beweis. $x \cdot 0 = x \cdot 0 + x \cdot 0 = x \cdot (0 + 0) = x \cdot 0$ □

i) $\forall_{a,b \in \mathbb{R}} : a \cdot b = 0 \iff (x = 0) \vee (y = 0)$

Beweis. Hinrichtung folgt aus h.
Annahme: $xy = 0$ und $x \neq 0$. Zu zeigen: $y = 0$
Mit $y = 0 \cdot x^{-1}$ und h folgt: $y = 0$ □

6.7 Bemerkung. Jede Menge K mit Verknüpfungen + und $\cdot$ für die Axiome gelten heißt Körper.

6.8 Beispiel. Dies sind Körper:

- $\mathbb{R}, \mathbb{Q}, \mathbb{C}$
- $\mathbb{F}_p = \{0, 1, \ldots, p\}$ mit Addition / Multiplikation $\mod p$
- $\mathbb{F}_2 = \{0, 1\}$

j) $\forall_x : -x = (-1)x$

Beweis. $x + (-1)x = 1x + (-1)x = (1 + (-1))x = 0x = 0$ □

k) $(-x) \cdot (-y) = xy$

Beweis. $(-x) \cdot (-y) = (-x)(-1)y = ((-1) \cdot (-x))y = (-(-x) \cdot y) = xy$ □

Im Allgemein gilt:

- Assoziativ:
  $\forall_{n \in \mathbb{N}}, \forall_{x_1,\ldots,x_n \in \mathbb{R}} : x_1 + \ldots + x_n = x_1 + (x_2 + (x_3 \ldots)x_{n-1} + x_n$
- Kommutativ:
  $\forall_{n \in \mathbb{N}}, \forall_{x_1,\ldots,x_n \in \mathbb{R}} : x_1 + \ldots + x_n = x_n + x_{n-1} + \ldots + x_1$

6.9 Satz (Doppelsummen).

$$\forall_{m,n \in \mathbb{N}}, \forall_{a_{ij} \in \mathbb{R}} : \sum_{i=1}^{m} \left(\sum_{j=1}^{n} a_{i,j} \right) = \sum_{j=1}^{n} \left(\sum_{i=1}^{m} a_{i,j} \right)$$

Beweis.

$$\begin{aligned} \sum_{i=1}^{m} \sum_{j=1}^{n} a_{i,j} &= \left(\sum_{j=1}^{n} a_{1,j} + \ldots + \sum_{j=1}^{n} a_{n,j} \right) \\ &= \left(\sum_{i=1}^{m} a_{i,1} + \ldots + \sum_{i=1}^{m} a_{i,m} \right) \\ &= \sum_{j=1}^{n} \sum_{i=1}^{m} a_{i,j} \end{aligned}$$

□

Es gilt ebenfalls das Distributivgesetz für Summen.

6.10 Satz (Distributiv).

$$\left(\sum_{i=1}^{m} x_i \right) \cdot \left(\sum_{j=1}^{n} y_i \right) = \sum_{i=1}^{m} \sum_{j=1}^{n} x_i y_j$$

6.11 Definition (Potenzen). $\forall_{n\in\mathbb{N}_0}, \forall_{x\in\mathbb{R}} : x^n \in \mathbb{R}$
Weiter definieren wir:
$x^0 \coloneqq 1$ und $x^{n+1} \coloneqq x \cdot x^n, \forall_{n\in\mathbb{N}}$.
Außerdem gilt für $x \neq 0 : x^{-n} \coloneqq (x^{-1})^n$

Daraus folgt:

l) $\forall_{m,n\in\mathbb{Z}}, \forall_{x,y\in\mathbb{R}} : x^m x^n = x^{m+n}$, $(x^m)^n = x^{mn}$ und $x^n y^n = (xy)^n$

6.1. Anordnungsaxiome

Trichotomie $\forall_{x\in\mathbb{R}}$ gilt genau eine der folgende Aussagen: $x > 0, x = 0, x < 0$

Abgeschlossenheit $\forall_{x,y\in\mathbb{R}} : x > 0$ und $y > 0 \implies x + y > 0$ und $xy > 0$

6.12 Notation.
- $x > y \iff x - y > 0$
- $x < y \iff x - y < 0$
- $x \geq y \iff x > y$ oder $y = x$
- $x \leq y \iff x < y$ oder $x = y$

Daraus ergeben sich weitere Konsequenzen:

m) $\forall_{x,y} : x < y$ oder $x = y$ oder $x > y$

n) Transitivität: $x < y$ und $y < z \implies x < z$

Beweis. $x < y$ und $y < z \implies y - x > 0$ und $z - y > 0 \implies z - x > (z - y) + (y - x) > 0$ □

o) Translation-Invarianz: $x < y \implies \forall_z : x + z < y + z$

Beweis. $(y + z) - (x - z) = y - x > 0$ □

p) Spiegelung: $x < y \iff (-x) > (-y)$

Beweis. $(-x) - (-y) = y - x > 0$ □

q) $x < y$ und $u < v \implies x + u < y + v$

r) $x < y$ und $t > 0 \implies tx < ty$

s) $x < y$ und $t < 0 \implies tx < ty$

t) $0 \leq x < y$ und $0 \leq x < y$ und $0 \leq u < v \implies xu < yv$

u) $\forall_{x\neq 0} : x > 0 \iff x^{-1} > 0$

v) $0 < x < y \implies x^{-1} > y^{-1}$

w) $\forall_{x\neq 0} : x^2 > 0$ insbesondere gilt: $1 > 0$

Beweis. Falls $x > 0$ gilt: Abgeschlossenheit der Multiplikation
Sonst: $x^2 = (-x) \cdot (-x) > 0$ ebenfalls gilt die Abgeschlossenheit der Multiplikation □

6.2. Das Archimedische Axiom

$$\forall_{x\in\mathbb{R}}\exists_{n\in\mathbb{N}} : n > x$$

Daraus folgt:

x) $\forall_{\varepsilon>0}\exists_{n\in\mathbb{N}} : \frac{1}{n} < \varepsilon$

y) $\forall_{x\in\mathbb{R}}\exists!_{n\in\mathbb{N}} : n \leq x < n+1$

6.13 Satz (Bernulische Ungleichung).

$$\forall_{x\in\mathbb{R},x\geq -1}\forall_{n\in\mathbb{N}} : (1+x)^n \geq 1+nx$$

Beweis. durch Induktion.

◊ IA $n = 1$

$$1 + x \geq 1 + x$$

◊ $n \to n+1$ da $A(n)$ gilt:
Es bleibt zu Zeigen: $(1+x)^{n+1} \geq 1 + (n+1)x$.
Nach der Induktionsannahme gilt:

$$(1+x)(1+nx) \geq 1 + (n+1)x$$
$$1 + nx + x + nx^2 \geq 1 + nx + x$$

□

6.14 Korollar. • $\forall_{a\in\mathbb{R},a>1}\forall_{K\in\mathbb{R}}\exists_{n\in\mathbb{N}} : a^n > K$

• $\forall_{a\in\mathbb{R},0<a<1}\forall_{\varepsilon>0}\exists_{n\in\ N} : a^n < \varepsilon$

Beweis. • Bernulische Ungleichung mit $x = a \cdot 1 > 0$

• Analog zu (i) nur mit $\frac{1}{a}$ statt a und $K = \frac{1}{\varepsilon}$

□

6.15 Definition (Absolut-Betrag).

$$x \in \mathbb{R} : |x| \coloneqq \begin{cases} x & , \text{falls } x \geq 0 \\ -x & , \text{falls } x < 0 \end{cases}$$

6.16 Definition (Maximum).

$$\max(x, y) \coloneqq \begin{cases} x & , x \geq y \\ y & , \text{sonst} \end{cases}$$

6.17 Satz. Für den Betrag $|\bullet|$ gilt:

• $\forall_x : |x| \geq 0$

- $\forall_{x,y} : |xy| = |x| \cdot |y|$
- $\forall_{x,y} : |x + y| \leq |x| + |y|$

Beweis. Wir betrachten nur Aussage 3, da die anderen Aussagen trivial sind:
Weil $x \leq |x|$ gilt:

$$\begin{aligned} & x + y \leq |x| + |y| \\ \implies & -x \leq |x| \\ & -(x + y) = (-x) + (-y) \leq (x) + (y) \\ \implies & |x + y| \leq |x| + |y| \end{aligned}$$

□

Folgen

7. Konvergenz von Folgen

Eine Folge reeler Zahlen ist eine Abbildung von $\mathbb{N} \to \mathbb{R}$. Hierbei wird eine natürliche Zahle auf eine reele Zahl geschickt.

$$a_{n\in\mathbb{N}} = (a_1, a_2, a_3, \ldots)$$
$$a_{k\in\mathbb{Z}} = (a_k, a_{k+1}, \ldots)$$

7.1 Beispiel. Typische Folgen:

- $\frac{1}{n} = (1, \frac{1}{2}, \frac{1}{3}, \ldots)$
- $(-1)^n = (1, -1, \ldots)$
- $2^n = (1, 2, 4, 8, 16, \ldots)$
- Fibonnacci mit $f_0 \coloneqq 0, f_1 \coloneqq 1$ und $f_n \coloneqq f_{n-1} + f_{n-2}$

7.2 Definition (Konvergenz). Eine Folge $(a_n)_{n\in\mathbb{N}}$ reeler Zahlen heißt konvergent gegen $a \in \mathbb{R}$ falls

$$\forall_{\varepsilon>0}\exists_{N\in\mathbb{N}} : |a_n - a| < \varepsilon$$

gilt. Äquivalent hierzu:

$$\forall_{\varepsilon>0}\exists_N\forall_{n\geq N} : |a_n - a| < \frac{\varepsilon}{k}$$
$$\forall_{\varepsilon>0}\exists_N\forall_{n>N} : |a_n - a| \leq \varepsilon$$

Integrale

8. Treppenfunktionen

8.1 Definition (Treppenfunktion). $\phi\colon [a,b] \to \mathbb{R}$ heißt *Treppenfunktion*, falls $n \in \mathbb{N}$ gibt mit $x_0 < x_1 < \ldots < x_n = b$ für die gilt:

$$\phi(x_{i-1}m, x_i) \text{ ist konstant.}$$

Mit anderen Worten:

$$\phi(x) = \sum_{i=1}^{n} \alpha_i 1_{(x_{i-1},x_i)} + \sum_{i=1}^{n} \beta_i 1_{\{x_i\}}$$

8.2 Notation.

$$T_{a,b} \coloneqq \text{ Menge aller Treppenfunktionen auf} [a,b]$$

8.3 Lemma. $T_{a,b}$ ist ein reeler Vektorraum:

- $0 \in T_{a,b}$
- $\phi, \psi \in T_{a,b} \implies \phi + \psi \in T_{a,b}$
- $\phi \in T_{a,b}, \lambda \in \mathbb{R} \implies \lambda \cdot \phi \in T_{a,b}$

Beweis. Wir betrachten nur die mittlere Aussage, da die anderen beiden Trivial sind.

Gegeben sind ϕ und ψ mit Stückstellen:

$$\{x_i\}_{i=1,\ldots,n}\{y_j\}_{j=1,\ldots,m}$$

Konstruiere Stückstellen $\{z_k\}_{k=1,\ldots,l}$ mit

$$\{z_k\} \subset \{x_i\} \cup \{y_j\}$$

Daraus folgt: ϕ und ψ sind konstant auf jeden Intervall (z_{k-1}, z_k)
ebenso $\phi + \psi \implies \phi + \psi \in T_{a,b}$ □

8.4 Definition (Integral für Treppenfumktionen). Sei

$$\phi \in T_{a,b}, \phi = \phi(x) = \sum_{i=1}^{n} \alpha_i 1_{(x_{i-1},x_i)} + \sum_{i=1}^{n} \beta_i 1_{\{x_i\}}$$

definiert als:

$$\int_a^b \phi(x)dx = \sum_{i=1}^{n} \alpha_i(x_i - x_{i-1})$$

Interpolation:
für $\alpha \geq 0$ ist dies die Fläche unter des Graphen.

8.5 Notation.

$$\int_a^b \phi(x)dx = \int_a^b \phi dx$$

8.6 Lemma. Das Integral

$$\int_a^b f dx$$

ist wohldefiniert für $f \in T_{a,b}$

Beweis. Falls

$$\begin{aligned}\phi &= \sum_{i=1}^{n} \alpha_i 1_{(x_{i-1},x_i)} + \sum_{i=1}^{m} \beta_j 1_{\{x_i\}} \\ &= \sum_{j=1}^{n} \gamma 1_{(y_{j-1},y_j)} + \sum_{j=1}^{m} \delta 1_{\{y_j\}}\end{aligned}$$

muss gelten:

$$\sum_{i=1}^{n} \alpha_i(x_i - x_{i-1}) = \sum_{j=1}^{m} \gamma_j(y_j - y_{j-1}$$

Wähle □

Teil II

Analysis 2

Integration

0.1 **Wiederholung.** Zu erst Wiederholen wir einige Begriffe aus der Analysis 1, welche wir später gebrauchen werden.

0.2 **Definition** (Treppenfunktion). Wir nennen $\varphi \in T_{a,b}$ eine Treppenfunktion. Dabei gilt:

$$\varphi = \sum_{i=1}^{n} \alpha_i \cdot 1_{(x_{i-1},x_i)} + \sum_{i=1}^{n} \beta_i \cdot 1_{\{x_i\}}$$

0.3 **Definition** (Integral). Wir nennen :

$$\int_a^b \varphi(x)dx \coloneqq \sum_{i=1}^{n} \alpha_i(x_i - x_{i-1})$$

unser Integral.

Kommt später noch mehr.

1. Partialbruchzerlegung

1.4 **Lemma** (Fundamentalsatz der Algebra). Eigentlich ein starker Satz in der Algebra, wir verwenden ihn jedoch nur als Hilfssatz.

- Sei g ein Polynom in $\mathbb{C}$ von Grad n. Dann $\exists_{w_1,\ldots,w_k,a\in\mathbb{C}}$ und $v_1,\ldots,v_k \in \mathbb{N}$ sowie $\sum_{l>1}^{k} v_l = n, (w_l \neq w_j)$ mit:

$$g(z) = a\prod_{l=1}^{k}(z - w_l)^{v_k}$$

  Wobei $(z - w_l)$ die Nullstelle und v_k die Vielfachheit ist.

- Falls Koeffizienten von g reel, dann ist w_l und auch $\overline{w_l}$ eine Nullstelle.

$\implies$ $\frac{1}{a}g$ ist Produkt quadratischer Polynome:

$$(z - w_l)(z - \overline{w_l}) = z^2 - 2Re(w_l)z + |w_l|^2$$

1.5 Lemma (Abspaltung von Haptteilen). Sei $r = \frac{f}{g}$ eine rationale Funktion, wobei f, g Polynome in $\mathbb{C}$, mit n-fachen Pol in $w \in \mathbb{C}$.
Ohne Beschränkung der Allgemeinheit $f(w) \neq 0$ und

$$r(z) = \frac{f(z)}{(z-w)^n \cdot h(z)} \tag{1.1}$$

mit $h(w) \neq 0$.
rationale Funktion ohne Pol in w:

$$H(z) = \frac{a_n}{(z-w)^n} + \frac{a_{n-1}}{(z-w)^{n-1}} + \ldots + \frac{a_1}{z-w}$$

mit $a_n \neq 0$. Dabei sind $a_n, \ldots, a_1$ sowie r_0 eindeutig.

Beweis. Aus (1.1) folgt:

$$\frac{f(z)}{h(z)} - \frac{f(w)}{h(w)} = \frac{f(z)h(w) - h(z)f(w)}{h(z)h(w)} = \frac{(z-w)p(z}{h(z)}$$

da $z \mapsto f(z)h(w) - h(z)f(w)$ Polynom mit Nullstellen in w ist, $\exists_{\text{Polynom } p}$. Nun können wir (1.1) verwenden und es folgt:

$$\begin{aligned} r(z) = \frac{f(z)}{(z-a)^n \cdot h} &= \frac{1}{(z-a)^n}\frac{f(w)}{h(w)} + \frac{1}{(z-w)^{n-1}}\frac{p(z)}{h(z)} \\ &= \frac{a_n}{(z-w)^n} + \frac{p(z)}{(z-w)^{n-1}h(z)} \end{aligned}$$

Nun führen wir einen Induktionsbeweis nach n.

- IA $n = 0$:
$$r = r_0$$
- IS $n-1 \to n$:
$$\implies r(z) = \frac{a_n}{(z-w)^n} + \left(\frac{a_{n-1}}{(z-w)^{n+1}} + \ldots + \frac{a_1}{z-w}\right)$$
Dies folgt direkt nach Induktionsannahme.

□

1.6 Satz (Satz der Partialbruchzerlegung). Sei $r = \frac{f}{g}$ rational mit Polynom f, g und $g(z)) = \prod_{l=1}^{k}(z-w_l)^{v_l}$ mit paarweise verschiedenen $w_l \in \mathbb{C}$. Dann gilt:

$$r = \sum_{l=1}^{k} H_l + q$$

mit Hauptteilen H_l in w_l und Polynom q.

Beweis. Iterative Abspaltung von Hauptteilen liefert:

$$\begin{aligned} H = H_1 + r_1 &= H_1 + H_2 + r_2 \\ &= \dots \\ &= \sum_{l=1}^{k} H_l + r_k \end{aligned}$$

mit r_k rational. Wir wissen:

$$\{\text{Pole von } r_{l+1}\} \subset \{\text{Pole von } r_l\} \setminus \{w_{l+1}\}$$

Dies impliziert:

- r_k hat keinen Pol
- r_k ist Polynom.

□

1.7 Korollar. $\int r dx = \int q dx + \sum_{l=1}^{k} \sum_{j=1}^{v_l} \frac{a_{l,j}}{(x-w_l)^j} dx$

Dabei ist:

- $\int q dx$ klar.
- $\int_s^t \frac{1}{(x-w)^j} dx$ mit $j > 1$ gleich: $-\frac{1}{j-1} \cdot \frac{1}{(x-w)^{j-1}}\Big|_s^t$
- Nicht trivial: $\int_s^t \frac{a_1}{x-w} dx$.
  Hierbei gilt mit w taucht auch $\overline{w}$ auf (mit Zähler $\overline{a_1}$). Das heißt zu berechnen ist:

$$\begin{aligned} \int_s^t \left[\frac{a}{x-w} + \frac{\overline{a}}{x-\overline{w}}\right] dx &= \int_s^t \frac{2Re(a)x - 2Re(a\overline{w})}{x^2 - 2Re(w)x + |w|^2} dx \\ &= \int_s^t \frac{Bx + c}{x^2 + 2bx + c} dx \text{ mit } b^2 < c \\ &= \frac{B}{2} \int_s^t \frac{2x + 2b}{x^2 + 2bx + c} dx + (c - Bb) \int_s^t \frac{1}{x^2 + 2bx + c} dx \end{aligned}$$

  Wir können nun getrennt beide Integrale ausrechnen:

$$= \frac{B}{2} \log(x^2 + 2bx + c)\Big|_s^t + \dots$$

  und das etwas schwierigere Integral:

$$\int_s^t \frac{1}{x^2 + 2b + c} dx = \int_{\frac{s+b}{\sqrt[2]{c-b^2}}}^{\frac{t+b}{\sqrt[2]{c-b^2}}} \frac{1}{y^2 + 1} dy = \frac{1}{\sqrt[2]{c - b^2}} \arctan(y)\Big|_{\frac{s+b}{\sqrt[2]{c-b^2}}}^{\frac{t+b}{\sqrt[2]{c-b^2}}}$$

Als Übung kann das Integral vollständig ausgeschrieben werden. Dies sei dem Leser überlassen.

2. Uneigentliche Riemann Integrale

Wir wollen die Definition von den uns bekannten Integralen erweitern, sodass $a \vee b \notin \mathbb{R}$ oder f nicht definiert in a oder b ist.

2.8 Definition. Wir definieren:

- Sei $I = [a, b)$ mit $a \in \mathbb{R}, b \in \mathbb{R} \cup \{+\infty\}$ und $f\colon I \to \mathbb{R}$ mit $f[a, \beta] \to \mathbb{R}$ integrierbar für alle $\beta < b$. Falls
$$\lim_{\beta \to b} \int_a^b f(x)dx$$
existiert in $\mathbb{R}$ so heißt f *uneigentliche Riemannintegrierbar auf* I und man schreibt:
$$\int_a^b f(x)dx = \lim_{\beta \to b} \int_a^\beta f(x)dx$$
- Analog für $I = (a, b]$ mit $a \in \mathbb{R} \cup \{-\infty\}, b \in \mathbb{R}$
- Für $I = (a, b)$ mit $a \in [-\infty, \infty), b \in (-\infty, \infty]$ schreibt man:
$$\int_a^b f(x)dx = \lim_{\beta \to b} \int_c^\beta f(x)dx + \lim_{\alpha \to a} \int_\alpha^c f(x)dx$$
falls für ein $c \in (a, b)$ diese Limiten existieren.

2.9 Beispiel. Wir betrachten die uns aus der Analysis 1 schon bekannte Riemannsche Zeta-Funktion als eine Abwandlung eines Integrals:

a) • <u>Für $s > 1$</u>:
$$\begin{aligned}\int_1^\infty \frac{1}{x^s}dx &= \lim_{\beta \to \infty} \int_1^\beta \frac{1}{x^s}dx \\ &= \lim_{\beta \to \infty} \left[\frac{1}{1-s}x^{1-s}\right]_1^\beta \\ &= \lim_{\beta \to \infty} \frac{1}{1-s}(-1 + \beta^{1-s}) = \frac{1}{s-1}\end{aligned}$$

• <u>Für $s \leq 1$</u>:
$$\int_1^\beta \frac{1}{x^s}dx \underset{\beta \to \infty}{\to} \infty$$

b) • <u>Für $s < 1$</u>:
$$\int_\alpha^1 \frac{1}{x^s}dx = \lim_{\alpha \to 0} \frac{1}{x^s} = \frac{1}{1-s}$$

• <u>Für $s \geq 1$</u>:
$$\int_\alpha^1 \frac{1}{x^s}dx \to \infty$$

c) Sei $c > 0$:

$$\int_a^\infty e^{-cx}dx = \lim_{\beta\to\infty}\left[\frac{1}{-c}e^{-cx}\right]_0^\beta = \frac{1}{c}$$

d) Gamma-Funktion (Euler 1729):

$$\Gamma(x) = \int_0^\infty t^{x-1}e^{-t}dt, x > 0$$

Dieses Integral existiert $\forall_{x>0}$, denn:

- in $(0,1]$ ist $0 \le t^{x-1}e^{-t} \le t^{x-1}$
- in $[1,\infty)$ ist $0 \le t^{x-1}e^{-t} \le c \cdot e^{-\frac{t}{2}}$ mit $c \coloneqq c(x)$

und aus den vorherigen Beispielen wissen wir, dass beide Integrale in $\mathbb{R}$ existieren.

2.10 Satz (Eigenschaften der Γ-Funktion). Es gilt:

- $\Gamma(x-1) = x\Gamma(x)\forall_{x>0}$
- $\Gamma(1) = 1$
- $\Gamma(n) = (n-1)!\forall_{n\in\mathbb{N}}$

Beweis. Wir zeigen die erste Eigenschaft mittels Partieller Integration:

$$\int_\alpha^\beta t^x e^{-t}dt = \left[-t^c e^{-t}\right]_\alpha^\beta + \int_\alpha^\beta xt^{x-1}e^{-t}dt$$

Daraus folgt die Aussage direkt. Nach Beispiel 3 folgt auch die Eigenschaft, dass $\Gamma(1) = 1$ ist. Die letzte Eigenschaft zeige man mit Induktion nach n mit 2 als Anfang und 1 als Schritt. □

2.11 Satz (Integralvergleichs Kriterium). Sei $f[1,\infty) \to \mathbb{R}_+$ monoton fallend. Dann gilt:

$$\int_1^\infty f(x)dx \iff \sum_{n=1}^\infty f(n) < \infty$$

Beweis. Definiere: $\varphi, \psi\colon [1,\infty) \to \mathbb{R}$ als Treppenfunktion.
$\psi(x) = f(n)$ und $\varphi(x) = f(n+1)$ für $x \in [n, n+1)$. Da die Funktion monoton fallend ist folgt:

$$\varphi \le f \le \psi$$

und $\forall_{N\in\mathbb{N}}$:

$$\sum_{n=2}^N f(n) = \int_1^n \varphi(x)dx \le \int_1^N f(x)dx \le \int_1^N \psi(x)dx = \sum_{n=1}^{N-1} f(n)$$

Demnach falls $\int_1^\infty f(x)dx$ oder $\sum_{k=1}^\infty f(n)$ konvergent ist das jeweilige andere ebenfalls konvergent. □

2.12 Beispiel.

$$\sum_{n=1}^\infty \frac{1}{n^s} < \infty \iff s > 1 \iff \int_1^\infty \frac{1}{x^s}dx < \infty$$

3. Vertauschen von Limes-Bildung mit Integration und Differentiation

3.13 Definition. Sei K eine Menge und $f, f_n \colon K \to \mathbb{C}$. Dann konvergiert f_n gleichmäßig gegen f genau dann wenn:

- $\forall_{\varepsilon>0}\exists_N\forall_{n\geq N}\forall_{x\in K} : |f_n(x) - f(x)| < \varepsilon$
- $\|f_n - f\|_K \to 0$ für $n \to \infty$ mit Supremumsnorm $\|g\|_K \coloneqq sup\{|g(x)| : x \in K\}$

3.14 Bemerkung. Sollte die Norm eindeutig sein lassen wir das $_K$ weg.

3.15 Korollar. $f_n \to f$ gleichmäßig auf $K \implies f_n \to f$ punktweise auf K. Die Umkehrung gilt nicht!

3.16 Satz. $f, f_n \colon K \to \mathbb{C}, K \subset \mathbb{C}$. Falls f_n stetig, $f_n \to f$ gleichmäßig konvergent $\implies f$ stetig.

Beweis. Sei $x \in K, \varepsilon > 0$.

$$\implies \exists_N\forall_{n\geq N} : |f_N(z) - f(z)| < \varepsilon\forall_{z\in K}$$

Stetigkeit von $f_N : \exists_{\delta>0}\forall_y$ mit $|x - y| < \delta$:

$$|f_N(x) - f_N(y)| < \varepsilon$$

Daraus folgt:

$$\begin{aligned} |f(x) - f(y)| &\leq |f(x) - f_N(x)| + |f_N(x) - f_N(y)| + |f_N(y) - f(y)| \\ &\leq 3\varepsilon \end{aligned}$$

□

3.17 Bemerkung. Gilt nicht, wenn man gleichmäßig durch punktweise ersetzt.

3.18 Satz. Für $K = [a, b] \subset \mathbb{R}$ und $f_n \colon K \to \mathbb{R}$ gilt: Falls f_n integrierbar und $f_n \to f$ gleichmäßig konvergent dann ist f integrierbar und:

$$\int_a^b f_n dx \to \int_a^b f dx$$

Beweis. Wir bemerken:

- $\forall_{\varepsilon>0} : \exists_n : |f_n - f| \leq \varepsilon$ auf K
- $\exists_{\varphi,\psi\in T_{a,b}}$ mit $\varphi \leq f_n \leq \psi$ und $\int_a^b(\psi - \varphi) \leq \varepsilon$
  Wir definieren:

$$\begin{aligned} \tilde{\psi} &\coloneqq \psi + \varepsilon \\ \tilde{\varphi} &\coloneqq \varphi - \varepsilon \\ &\implies \tilde{\varphi} \leq f \leq \tilde{\psi} \\ &\implies \int_a^b (\tilde{\psi} - \tilde{\varphi})dx \leq \varepsilon + 2\varepsilon(b - a) = \tilde{\varepsilon} \end{aligned}$$

Damit ist f integrierbar. Außerdem gilt:

$$\left|\int_a^b f dx - \int_a^b f_n dx\right| \leq \int_a^b |f - f_n| dx$$
$$\leq \varepsilon \cdot (b-a)$$

□

3.19 Bemerkung. Wir betrachten Satz 3.18:

- Diese Folgerung gilt nicht bei punktweiser Konvergenz $f_n \to f$
- Das gleiche gilt für uneigentlichen Integralen allerdings auf unbeschränkten Intervallen.
- Es gilt insbesonders: f_n stetig $\implies$ f_n integrierbar.

3.20 Satz. Seien $f_n \colon [a,b] \to \mathbb{R}$ stetig differenzierbar $f_n \to f$ punktweise konvergent und f_n' konvergieren gleichmäßig (gegen ein g) Dann gilt: f stetig differenzierbar und $f' = \lim_{n\to\infty} f_n'$

Beweis. Nach Satz 3.16 : Aus $f_n' \to g$ gleichmäßig und Stetigkeit auf f_n' folgt: g ist stetig.
Nach dem Hauptsatz der Differential und Integralrechnung folgt: $f_n(x) = f_n(a) + \int_a^x f_n'(y)dy$.
Nach Satz 3.18 folgt weiter: $f(x) = f(a) + \int_a^x g(y)dy$. Nach erneuten Verwendung des HDI gilt: $f' = g$ □

3.21 Beispiel. Wir betrachten: $f_n(x) = \frac{1}{n}\sin(nx)$ auf $\mathbb{R}$. Die Funktionenfolge $f_n \to 0$ ist gleichmäßig konvergent auf $\mathbb{R}$, aber $f_n'(x) = \cos(nx)$ konvergiert nicht (insbesonders nicht gegen $f' = 0$.

3.22 Satz (Weierstraß Kriterium). Seien $f_n \colon K \to \mathbb{C}$ mit $\sum_{n=1}^\infty \|f_n\|_K < \infty$ dann konvergiert $\sum_{n=1}^\infty f_n$ absolut und gleichmäßig auf K gegen eine Funktion $F \colon K \to \mathbb{C}$

Beweis. Es gilt:

- $\forall_{x\in K} : F_n(x) \coloneqq \sum_{l=1}^n f_l(x)$ konvergiert absolut gegen $\sum_{l=1}^\infty f_l(x)$ nach dm Majoranten-Kriterium, da:
$$\sum_{l=1}^n |f_l(x)| \leq \sum_{l=1}^n \|f_l\|_K \leq \sum_{l=1}^\infty \|f_l\|_K$$
- $\forall_{\varepsilon>0} : \exists_N : \sum_{l=1}^\infty \|f_l\|_K < \varepsilon$.

$$\implies \forall_{x\in K}, \forall_{n\geq N} : |F(x) - F_n(x)| \geq \sum_{l=nN+1}^\infty |f_l(x)| \leq \sum_{l=N+1}^\infty \|f_l\|_K < \varepsilon$$
$$\implies F_n - F \text{ gleichmäßig auf K}$$

□

3.23 Beispiel. Sei $f_n(x) = \frac{\cos(nx)}{n^2}$ dann gilt: $\sum_{n=1}^\infty f_n$ konvergiert absolut und gleichmäßig auf $\mathbb{R}$ denn $\sum_{n=1}^\infty \|f_n\| = \sum_{n=1}^\infty \frac{1}{n^2} < \infty$

3.1. Potenzreihen

3.24 **Definition** (Potenzreihen). Sei $a, c_n \in \mathbb{C}(\forall_{n\in\mathbb{N}})$. Die Reihe $f(z) = \sum_{n=1}^{\infty} c_n(z-a)^n$ heißt *Potenzreihe zum Entwicklungspunkt* $a \in \mathbb{C}$ mit Koeffizienten c_n.
Dabei heißt $R \coloneqq sup\{|z-a| : \sum_{n=0}^{\infty} c_n(z-a)^n \text{ konvergent}\} \in [0,\infty]$ *Konvergenzradius* der Potenzreihe.

3.25 **Satz.** Es gilt:

- Die Potenzreihe $f(z) = \sum_{n=1}^{\infty} c_n(z-a)^n$ konvergiert für jedes $r \in (0,R)$ absolut und gleichmäßig im abgeschlossenen Kreis $K_r(a) \coloneqq \{z \in \mathbb{C} : |z-a| \leq r\}$
- Die Potenzreihe konvergiert im offenen Kreis $B_R(a) \coloneqq \{z \in \mathbb{C} : |z-a| < R\}$ und definiert dort eine stetige Funktion $f\colon B_R(a) \to \mathbb{C}$
- die Hadmardsche Formel $R = (\limsup_{n\to\infty} |c_n|^{\frac{1}{n}})^{-1}$ (mit $0^{-1} \coloneqq \infty$ und $\infty^{-1} \coloneqq_0$)

Beweis. Wir zeigen:

- Zu gegebenen $r < R$ wähle $z_1 \in \mathbb{C}$ mit $r < |z_1 - a|$ und $f_n(z) \coloneqq \sum_{l=0}^{n} c_l(z-a)^l$ konvergiert für $n \to \infty$.

$$\implies \exists_{M\in\mathbb{R}} : |c_l(z_1-a)^l| \leq M$$
$$\implies \forall_{z\in K_r(a)} : |c_l(z-a)^l \leq |c_l(z_1-a)^l| \cdot |\frac{za}{z_1 a}|^l \leq M \cdot \upsilon$$

  wobei $0 < \upsilon < 1$.
  Vergleich mit geometrischer Reihe impliziert, dass $\sum_{l=1}^{\infty} c_l(z-a)^l$ absolut konvergiert für jedes $z \in K_r(a)$ sowie (nach dem Weierstraß-Kriterium 3.22) auch gleichmäßig in $K_r(a)$ ist.

- $\forall_{z\in B_R(a)} : \exists_{r\in(0,R)} : z \in K_r(a) \implies f_n \to f$ gleichmäßig in $K_r(a)$. Daraus folgt mit Satz 3.16: Limes f ist stetig in $K_r(a)$ und f ist stetig in $B_R(a)$

- Setze $R_* \coloneqq (\limsup_{n\to\infty} |c_n|^{\frac{1}{n}})^{-1}$. Dann gilt mit dem Wurzelkriterium:

$$|z-a| < R_* \implies \limsup_{n\to\infty} |c_n(z-a)^n|^{\frac{1}{n}} < 1$$

  Dies impliziert die Konvergenz von $\sum_{n=1}^{\infty} c_n(z-a)^n$ und demnach ist $R = R_*$.

□

3.26 **Bemerkung.** In jeden Punkt des Komplements des abgeschlossenen Kreises $K_R(a)$ divergiert die Reihe. AU der Kreislinie $\delta K_R(a) = \{z \in \mathbb{C} : |z-a| = R\}$ können Punkte liegen in denen die Reihe konvergiert oder divergiert.

3.27 **Beispiel.** Für:

- $c_n = n^n \implies R = 0$

- ◦ $f(z) = \sum_{n=0}^{\infty} z^n \implies R = 1.$
  Für $|z| < 1 : f(z) = \frac{1}{1-z}$
  ◦ $f(z) = \frac{1}{2} \sum_{n=0}^{\infty} \left(\frac{1+z}{2}\right)^n \implies R = 2.$
  Für $|z+1| < 2 : f(z) = \frac{1}{2} \left(\frac{1}{1-\frac{1+z}{2}}\right) = \frac{1}{1-z}$

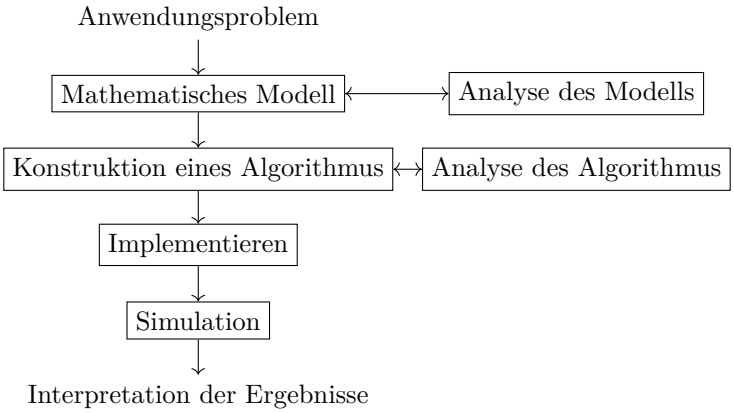
Teil III

Algorithmische Mathematik 1

Einführung

Algorithmische Mathematik -Was ist das?

Gegenstand der Alma ist die Konstruktion und Analyse effizienter Algorithmen zur Lösung mathematischer Problemstellungen mit Hilfe des Computers.
Damit liegt sie im Bereich der Angewandten Mathematik. Konkrete Problemstellungen ergeben sich oft aus technischen Problemen, Naturwissenschaften, Medizin etc.
Man kann hierbei annehmen, dass verschiedene Problemstellungen aus der Anwendungen oft zu ähnlichen oder gleichen mathematischen Modellen zurückgeführt werden können.
Diese Probleme können wie folgt aussehen:



0.1 Beispiel. Ein typisches Problem ist das lösen eines linearen Gleichungssystems.

Im Rahmen dieser Vorlesung konzentrieren wird uns auf die Teilbereiche

a) Numerik

b) Diskrete Mathematik

c) Statistik

der Angewandten Mathematik

Zahlendarstellung am Computer

1. Zahlensystem

Die Darstellung von Zahlen basiert auf sogenannten *Zahlensystemen*. Diese Zahlensysteme unterscheiden sich in der Wahl des zugrundeliegendes Alphabets. Unsere Zahl entspricht dann einem Wort, bestehend aus Elementen des Alphabets.

1.1 Definition (Alphabet). Es sei $\mathbb{N} = \{1, 2, 3 \ldots\}$ und $b \in \mathbb{N}$.
Wir bezeichnen mit $\sum_b$ das Alphabet des *b-adischen Zahlensystems*

1.2 Beispiel. Verschiedene Zahlensysteme

a) Dezimalsystem: $\sum_{10} = \{0, 1, \ldots, 9\}$
wie $(384)_{10}$

b) Dual bzw. Binärsystem $\sum_2 = \{0, 1\}$
wie $(42)_{10} = (101010)_2$

c) Oktalsystem $\sum_8 = \{0, \ldots, 7\}$
wie $(42)_{10} = (52)_8$

d) Hexadezimalsystem $\sum_{16} = \{0, \ldots, 9, A, B, \ldots, F\}$
wie $(42)_{10} = (2A)_{16}$

Um die Zahlendarstellung sinnvoll zu nutzen Ergibt sich folgendes:

1.3 Satz. Seien $b, n \in \mathbb{N}, b > 1$.
Dann ist jede ganze, nicht-negative Zahl z mit $0 \leq z \leq b^n - 1$ *eindeutig* als Wort der Länge n über $\sum_b$ darstellbar durch:

$$z = \sum_{i=0}^{n-1} z_i b^i$$

mit $z_i \in \sum_b$ für alle $i = 0, 1, \ldots, n-1$.
Wir schreiben vereinfachend:

$$z = (z_{n-1}, \ldots, z_0)_b$$

Beweis. • Induktion

◊ *Induktionsanfang*: $\underline{z < b}$ hat die eindeutige Darstellung $z_0 = z$ und $z_i = 0$ sonst.

◊ *Induktionsschritt*: $\underline{z-1 \to z \geq b}$

Betrachte

$$z = \left\lfloor \frac{z}{b} \right\rfloor \cdot b + (z \mod b)$$

Da $\hat{z} < z$ besitzt die eindeutige Darstellung

$$\hat{z} = (z_{\hat{n-1}}, \ldots, \hat{z_0})_b$$

Bemerke $z_{\hat{n-1}} = 0$, da

$$\left(z_{\hat{n-1}} b^{n-1}\right) b \leq z \leq b^n - 1$$

Also ist z_b eine n-stellige Darstellung von z in b-adischen Zahlensystem

- Eindeutigkeit wird durch Widerspruch gezeigt.
  Angenommen: Es gibt zwei verschiedene Darstellungen

$$z = \left(z_{n-1}^{(2)}, \ldots, z_0^{(2)}\right)_b = \left(z_{n-1}^{(1)}, \ldots, z_0^{(1)}\right)_b$$

Sei $m \in \mathbb{N}$ der größte Index mit $z_m^{(1)} \neq z_m^{(2)}$,
Ohne Beschränkung der Allgemeinheit kann gesagt werden $z_m^{(1)>z_m^{(2)}}$.
Dann müssten die Stellen $0, 1, \ldots, m-1$ den niedrigeren Wert von $z_m^{(2)}$ kompensieren.
Die größte mit diesen Stellen darstellbare Zahl ist aber:

$$\sum_{i=0}^{m-1} (b-1) b^i = (b-1) \sum_{i=0}^{m-1} b^i = b^m - 1.$$

Da b^m der kleinstmögliche Wert der fehlende Stelle m ist, kann diese aber nicht kompensiert werden. Dies ist ein Widerspruch.

□

Durch den Beweis ergibt sich sofort ein Algorithmus zur Umwandlung einer Zahl in ein anderes Zahlensystem:

1.4 Beispiel. Umwandlung von $(1364)_{10}$ in das Oktalsystem.

- $1364 = 170 \cdot 8 + 4$
- $170 = 21 \cdot 8 + 2$
- …
- $0 \cdot 8 + 2$

$\Longrightarrow (2524)_8$

Daten: Dezimalzahl $z \in \mathbb{N}_0$, Basis $b \in \mathbb{N}$
Ergebnis: b-adische Darstellung $(z_{n-1}, \ldots, z_0)_b$
Initialisiere $i = 0$
solange $z > 0$ **tue**
  $z_i = z \mod b$
  $z = \lfloor \frac{z}{b} \rfloor$
  $i = i + 1$
Ende

Algorithmus 1: Bestimmung der b-adischen Darstellung

Beobachtung Wir sehen, dass das Honor-Schmea nur eine Schleife, Additionen und Multiplikationen benötigt. Diese Operationen können wir am Computer durchführen. Im Gegensatz dazu steht die Potenz b^i in modernen Programmiersprachen zwar zur Verfügung, wird aber im Hintergrund oft auf Multiplikationen zurückgeführt. Man überprüft leicht, dass das Honor-Schema weniger Multiplikationen benötigt und somit schnellt ist.

2. Vorzeichen-Betrag-Darstellung

Um auch Zahlen mit Vorzeichen am Computer darstellen zu können, betrachten wir im Folgenden die Vorzeichen-Betrag-Darstellung für Binärzahlen. Das Binäralphabet besteht nur aus 0 und 1, welche wir auch als *Bits* bezeichnen. Bei einer Wortlänge von n Bits wir das erste Bit als Vorzeichen verwendet, die restlichen $n-1$-Bits für den Betrag der Zahl. Da die 0 die Darstellung $+0$ und $-O$ besitzt, können wir insgesamt $2^n - 1$ Zahlen darstellen.

2.5 Beispiel. Für $n = 3$

Bitmuster	Dezimaldarstellung
000	$+0$
001	$+1$
010	$+2$
011	$+3$
100	-0
101	-1
110	-2
111	-3

Aber: Diese Darstellung am Computer ist unpraktisch, da die vier Grundrechenarten auf Hardwareebene typischerweise mit Hilfe von Addition und Zusatz-Logik umgesetzt werden.

Lösung: Komplementdarstellung

3. Komplementdarstellung

3.6 **Definition** ((b-1)-Komplement). Sei $z = (z_{n-1} \dots z_1 z_0)_b$ eine n-stellige b-adische Zahl. Das *(b-1)-Komplement* $K_{b-1}(z)$ ist definiert als:

$$K_{b-1} = (b-1-z_{n-1}, \dots, b-1-z_0)_b$$

Geben wir hierzu direkt ein paar Beispiele an

3.7 **Beispiel.** Komplemente

- $K_9((325)_{10}) = (674)_{10}$ (9er-Komplement im 10-er System)
- $K_1((10110)_2) = (01001)_2$ (1er-Komplement im 2er System)

3.8 **Definition** (b-Komplement). Das b-Komplement einer b-adischen Zahl $z \neq 0$ ist definiert als

$$K_b(z) = K_{b-1}(z) + (1)_b$$

3.9 **Beispiel.** • $K_{10}((325)_{10}) = (674)_{10} + (1)_{10} = (675)_{10}$

3.10 **Lemma.** Für jede n-stellige b-adische Zahl z gilt:

- i) $z + K_{b-1}(z) = (b-1, \dots, b-1)_b = b^n - 1$
- ii) $K_{b-1}\left(K_{b-1}(z)\right) = z$

Ist außerdem $z \neq 0$ so gilt:

- iii) $z + K_b(z) = b^n$
- iv) $K_b(K_b(z)) = z$

Beweis. Hilfssatz

(i) Durch nachrechnen:

$$\begin{aligned} z + K_{b-1}(z) &= (z_{n-1} \dots z_0)_b + (b-1-z_{n-1}, \dots, b-1-z_0)_b \\ &= \sum_{i=0}^{n-1} z_i b^i + \sum_{i=0}^{n-1} (b-1-z_i) b^i \\ &= \sum_{i=0}^{n-1} (b-1) b^i = (b-1, \dots, b-1)_b \\ &= (b-1) \sum_{i=0}^{n-1} b^i \\ &= (b-1) \left(\frac{b^n - 1}{b-1} \right) \\ &= b^n - 1 \end{aligned}$$

- (ii) per Definition
- (iii) Nachrechnen:

$$z + K_b(z) = z + K_{b-1} + 1 = b^n - 1 + 1 = b^n$$

- Definiere $\hat{z} = K_b(z) = K_{b-1}(z) + (1)_b > 0$ und rechne

$$z + K_b(z) = b^n = \hat{z} + K_b \hat{z} + K_b(K_b(z)) \implies \text{Behauptung.}$$

□

3.11 Bemerkung. Modifikation

- Die 3. Aussage gilt auch für $z = 0$, falls man dann bei der Addition von 1 die Anzahl der Stellen erweitert.
- die 4. Aussage gilt für $z = 0$, falls überall im Beweis modulo b^n gerechnet wird.

Außerdem impliziert die 3. Aussage des Lemmas, dass

$$K_b(z) = b^n - z. \tag{3.1}$$

Dies können wir geschickt zum Darstellen der b^n verschiedenen ganzen Zahlen z mit

$$-\left\lfloor \frac{b^n}{2} \right\rfloor \leq z \leq \left\lceil \frac{b^n}{2} \right\rceil - 1$$

nutzen. Diesen Bereich nennt man darstellbaren Bereich.

3.12 Definition (b-Komplement-Darstellung). Die b-Komplement-Darstellung $(z)_{K_b} = (z_{n-1} \dots z_0)_b$ einer Zahl $z \in \mathbb{Z}$ im darstellbaren Bereich ist definiert als:

$$(z)_{K_b} = \begin{cases} (z)_b & \text{falls } z \geq 0 \\ (K_b(|z|))_b & \text{falls } z < 0 \end{cases}$$

3.13 Beispiel. Der darstellbare Bereich.

- Sei $b = 10$, $n = 2$.
  Dann impliziert (3.1), dass

$$K_{10}(50) = 10^2 - 50 = 50$$

$$K_{10}(49) = 100 - 49 = 51.$$

  Der darstellbare Bereich ist nun

$$-50 \leq z \leq 49$$

  und hat konkrete Darstellungen:

Darstellung	Zahl
0	+0
1	+1
...	...
49	+49
50	−50
...	...
99	−1

- Sei $b = 2$, $n = 3$ Der darstellbare Bereich ist $-4 \leq z \leq 3$.

Bitmuster	Dezimaldarstellung
000	0
001	1
...	...
100	−4
...	...
111	−1

Wir betrachten nun Addition und Subtraktion zweier Zahlen in b-Komplement-Darstellung. Hierzu bezeichne $(x)_{K_b} \oplus (y)_{K_b}$ die ziffernweise Addition der Darstellungen von x und y mit Übertrag ("schriftlich rechnen"), wobei ein eventueller Überlauf auf die $(n+1)$-te Stelle vernachlässigt wird (wir rechen also immer mit modulo b^n.

3.14 Satz (Addition in b-Komplement-Darstelllung). Seien x und y zwei n-stellige, b-adische Zahlen und x,y und $x + y$ im darstellbaren Bereich. Dann gilt:

$$(x + y)_{K_b} = (x)_{K_b} \oplus (y)_{K_b}$$

Beweis. Wir betrachten dazu mehrere Fälle.

- Fall $x, y \geq 0$:

$$\begin{aligned}(x)_{K_b} \oplus (y)_{K_b} &\overset{\text{Def.}}{=} ((x)_b + (y)_b) \mod b^n \\ &\overset{\text{Def.}}{=} (x+y) \mod b^n \\ &\overset{\text{Def.}}{=} (x+y)_{K_b}\end{aligned}$$

  Der letzte Schritt ist möglich, da $(x+y)_{K_b}$ im darstellbaren Bereich sind.

- Fall $x, y < 0$

$$\begin{aligned}(x)_{K_b} \oplus (y)_{K_b} &\overset{Def.}{=} ((K_b(|x|))_b + (K_b(|y|))_b) \mod b^n \\ &= ((K_b(|x|)) + (K_b(|x|))) \mod b^n \\ &= (b^n - |x| + b^n - |y|) \mod b^n \\ &= (x+y) \mod b^n \\ &= (x+y)_{K_b}\end{aligned}$$

- Fall $x \geq 0, y < 0$:

$$\begin{aligned}(x)_{K_b} \oplus (y)_{K_b} &\overset{Def.}{=} ((x)_b + (K_b(|y|))_b) \mod b^n \\ &= (x + K_b(|y|)) \mod b^n \\ &= (x + b^n - |y|) \mod b^n \\ &= (x+y) \mod b^n \\ &= (x+y)_{K_b}\end{aligned}$$

- Fall $x < 0, y \geq 0$: Analog

□

3.15 Satz (Subtraktion in b-Komplement-Darstellung). Seien x und y n-stellige b-adische Zahlen und x,y und x-y im darstellbaren Bereich. Dann gilt:

$$(x-y)_{K_b} = (x)_{K_b} \oplus (K_b(y))_{K_b}$$

Beweis. • Fall $y = 0$ nichts zu zeigen.

- $y \neq 0$: (3.1) impliziert:

$$-y = K_b(y) - b^n$$

  mit modulo b^n-rechnen folgt, dass

$$(-y)_{K_b} = (K_b(y))_{K_b}$$

ist und somit:

$$\begin{aligned}(x-y)_{K_b} &= (x+(-y))_{K_b} \\ &= (x)_{K_b} \oplus (-y)_{K_b} \\ &= (x)_{K_b} \oplus (K_b(y))_{K_b}\end{aligned}$$

□

3.16 Beispiel. Für $b = 10$ und $n = 2$ ist der darstellbare Bereich $-50 \leq z \leq 49$

- $(20+7)_{K_{10}} = (20)_{K_{10}} \oplus (7)_{K_{10}} = (27)_{K_{10}} = 27$
- $28 - 5 = 28 + (-5) = (28)_{K_{10}} \oplus (95)_{K_{10}} = (23)_{K_{10}} = 23$
- $-18 - 20 = (-18) + (-20) = \ldots = -38$

Die darstellbaren Zahlen kann man sich beim K_b-Komplement als Zahlenrad vorstellen.

Achtung Ein eventueller Überlauf bzw. Unterlauf wird im Allgemeinen nicht aufgefangen.

3.17 Beispiel. In n-stelliger Binärarithmetik ist die größte darstellbare Zahl $x_{max} = (011\ldots1)_{K_2}$ gleich $2^{n-1}-1$. Hingegen ist $x_{max}+1 = (100\ldots0)_{K_2}$ und wir als -2^{n-1} interpretiert.

4. Fest-Komma-Darstellung

4.18 Definition. Bei der Festkommadarstellung einer n-stelligen Zahl werden k Vorkomma und n-k Nachkommastellen definiert:

$$z = \pm(z_{k-1}\ldots z_0.z_{-1}\ldots z_{k-n})_b = \pm \sum_{i=k-n}^{k-1} z_i b^i$$

4.19 Beispiel. Im Zehner System wie gehabt.
Im Binärsystem ergibt sich folgendes: $(101.01)_2 = 2^2 + 2^0 + 2^{-2} = 5.25$

Achtung Im Gegensatz zur Darstellung ganzer Zahlen können bereits bei der Konvertierung von Dezimalzahlen in das b-adische Zahlensystem Rundungsfehler auftreten.

4.20 Beispiel. $(0.8)_{10} = (0.110\overline{1100})_2$

Das größte Problem der Festkommadarstellung ist alllerdings, dass der darstellbare Bereich stark eingeschränkt ist und schlecht aufgelöst ist, da der Abstand zwischen zweier Zahlen immer gleich ist.

4.21 Beispiel. Die kleinste darstellbare Zahl in Fixkommadarstellung ist

$$z_1 = (0 \dots 0.0 \dots 01)_b$$

Die zweitkleinste Zahl ist $z_2 = 2z_1$. Wir wollen $x = \frac{z_1+z_2}{2}$ in Fixkommadarstellung darstellen, müssen wir entweder zu z_1 abrunden oder zu z_2 aufrunden.

4.22 Der relative Fehler dieses Runden ist:

$$\frac{|x - z_1|}{|x|} = \frac{1}{3}$$

Analog für z_2.
Solche Fehler machen jede Rechnung unbrauchbar.

5. Gleitkommadarstellung

5.23 Definition (Gleitkommadarstellung). Die Gleitkommastellung einer Zahl $z \in \mathbb{R}$ ist gegeben durch:

$$z = \pm m \cdot b^e$$

mit einer Matisse m, dem Exponenten e und der Basis b.

Der Einfachheit halber nehmen wir an, dass die Basis für alle Zahlen gleich ist, obwohl sie prinzipiell verschieden sein könnte.

5.24 Beispiel. Die Gleitkommadarstellungen von:

- $(-384.753)_{10} = -3.84753 \cdot 10^2$
- $(0.00042)_{10} = 4.2 \cdot 10^{-4}$
- $(1010.101)_2 = (1.010101)_2 \cdot 2^3$

Achtung: Die Gleitkommadarstellung ist nicht eindeutig!

5.25 Definition. Die Matisse m heißt normalisiert, falls $m = m_1.m_2m_3 \dots m_t$ mit $1 \leq m_1 \leq b$.

Um die eindeutige, normalisierte Gleitkommadarstellung im Computer zu speichern, müssen wir festlegen, wie viele Stellen für Matisse und Exponent zur Verfügung gestellt werden. Dies resultiert in der Menge

$$F = F(b, t, e_{min}, e_{max}) = \{z = \pm m_1.m_2m_3 \dots m_t \cdot b^e | e_{min} \leq e \leq e_{max}\}$$

Da 0 nicht in dieser Form dargestellt werden kann, reserviert man dafür eine spezielle Ziffernfolge in Matisse und Exponent.

5.26 Bemerkung. Das Hidden Bit ist hilfreich

- Für $b = 2$ kann die erste Ziffer m_1 der Matisse weggelassen werden (Hidden Bit)

- Um den Exponenten besser vergleichen zu können, verwendet man oft die Exzess- oder Bias-Darstellung. Durch Addition der Exzesser $|e_{min}| + 1$ wird der Exponetn auf den Bereich $1, 2, \ldots, |e_{min}| + e_{max} + 1$ transformiert.

5.27 **Beispiel** (IEEE 754 Standard). Betrachte binäre Gleitkommazhalen mit 64 Bit auf dem Computer.

- 52 Bit für die Matisse in Hidden Bit Darstellung
- 11 Bit für den Exponenten mit $e_{min} = -1022$ und $e_{max} = 1023$ gespeichert in Exzessdarstellung.
- 1 Bit als Vorzeichen.

Weiter Fälle können Online nachgelesen werden.

Genauigkeit der Gleitkommadarstellung

Da die Menge aller Zahlen in $F = F(b, t, e_{min}, e_{max})$ endlich ist, müssen wir Zahlen in $\mathbb{R} \setminus F$ geeignet annähern.

5.28 **Definition.** Die Rundung ist eine Abbildung $rd\colon \mathbb{R} \to F$ mit

- $rd(a) = a$, $a \in F$
- $rd(z)$, $z \in \mathbb{R}$ ist gegeben so, dass $|z - rd(z)| = \min\limits_{a \in F}(z - a)$ für alle $z \in \mathbb{R}$.

5.29 **Definition** (Maschinengenauigkeit). Den maximalen relativen Rundungsfehler ε_{mach} für $z_{min} \leq |z| \leq z_{max}$ nennt man Maschinengenauigkeit. Die Stellen der Mantisse heißen signfikante Stellen. Wenn die Mantisse t-stellig ist, spricht man von t-stelliger Arithmetik.

5.30 **Satz** (Maschinengenauigkeit von F). Die Maschinengenauigkeit ε_{mach} für $F = F(b, t, e_{min}, e_{max})$ ist:

$$\varepsilon_{mach} = \frac{1}{2} b^{1-t}.$$

Beweis. Betrachte relativen Rundungsfehler ε, der vom Abscheiden nicht signifikanten Stellen herrüht. Sei $z_{min} < x < z_{max}$ und $\tilde{x}$ die durch Abschneiden entstehende Zahl. Dann gilt:

$$\begin{aligned}
\varepsilon &= \frac{|x - \tilde{x}|}{|x|} \\
&= \frac{|x_1.x_2x_3 \ldots x_t x_{t+1} \ldots \cdot b^e - x_1x_2x_3 \ldots x_t \cdot b^e}{|x|} \\
&= \frac{|0.x_{t+1} \ldots | \cdot |b^{e+1-t}|}{|x|}
\end{aligned}$$

Da $|0.x_{t+1} \ldots | < 1$ und $b^e \leq |x|$ gilt:

$$\varepsilon < \frac{b^{e+1-t}}{b^e} = b^{1-t}$$

Da der Rundungsfehler ε_{mach} höchsten halb so groß ist wie der Abschneidefehler folgt die Behauptung. □

Die Maschinengenauigkeit ε_{mach} ist die wichtigste Größe zur Beurteilung der Genauigkeit von Gleitkommarechnungen am Computer. Sie gibt Aufschluss zur Anzahl signifikanter Stellen zur Basis b. Die zugehörige Anzahl s Stellen im Dezimalsystem erhält man durch das Auflösen von:

$$\varepsilon_{mach} = \frac{1}{2} b^{1-t} = \frac{1}{2} 10^{1-s}$$

nach s. Es folgt daher:

$$s = \lfloor 1 + (t-1) \log_{10}(b) \rfloor$$

Für den IEEE 754 Standard mit $b = 2, t = 53$ erhält man $s = 16$ signifikante Stellen.

Fehleranalyse

6. Rechnerarithmetik

6.1 Beispiel. Betrachte $F = F(10, 5, -4, 5)$ und Maschinenzahlen

$$x = 2.5684 \cdot 10^0 = 2.56840000$$
$$y = 3.2791 \cdot 10^{-3} = 0.0032791$$

Es gilt:

$$\left.\begin{aligned} x + y &= 2.5716792 \\ x - y &= 2.5651209 \\ x \cdot y &= 0.00842204044 \\ \tfrac{x}{y} &= 783.2637004 \end{aligned}\right\} \notin F$$

6.2 Bemerkung. Die Menge $F(b, t, e_{min}, e_{max})$ ist nicht abgeschlossen bezüglich der Grundrechenarten und können somit im Allgemeinen nicht im Computer implementiert werden.

Lösung Wir runden das Ergebnis und implementieren so eine Pseudoarithmetik. Das bedeutet, wir ersetzen $\circ \in \{+, -, \cdot, \div\}$ durch $\boxdot \in \{\boxplus, \boxminus, \boxdot, \boxslash\}$ definiert durch:

$$x \boxdot y \coloneqq rd(x \circ y) \tag{6.1}$$

Auf Hardwareebene wird üblicherweise mit einer längeren Matisse gearbeitet und dann normalisiert und gerundet. Dies entspricht dem IEEE 754 Standard.

6.3 Bemerkung. Für $|x|, |y|, |x \circ y| \in [z_{min}, z_{max}]$ impliziert (6.1), dass

$$\frac{|x \boxdot y - x \circ y|}{|x \circ y|} = \frac{|rd(x \circ y) - x \circ y|}{|x \circ y|} \leq \varepsilon_{mach}$$

Das bedeutet, dass $\boxdot$ im Computer bestmöglich umgesetzt ist.

6.4 Beispiel. Betrachte $F = F(10, 5, -4, 5)$

- Setze:
  - $a = 0.98765$

- $b = 0.012424$
- $c = -0.0065432$

Dann gilt:

$$(a + b) + c = a + (b + c) = 0.9925208$$

Numerisch gilt:

$$(0.98765 \boxplus 0.012424) \boxminus 0.0065432 = rd(0.9935568) = 0.99356$$

und

$$0.98765 \boxplus (0.012424 \boxminus 0.0065432) = rd(0.9935308) = 0.99353$$

- Setze
  - $a = 4.2832$
  - $b = -4.2821$
  - 5.7632

  Dann gilt

$$(a + b) \cdot c = a \cdot c + b \cdot c = 0.006339520000001$$

  Numerisch gilt:

$$(4.2832 \boxminus 4.2821) \boxdot 5.7632 = 0.006339$$

  und

$$(4.2832 \boxdot 5.7632) \boxminus (4.2821 \boxdot 5.7632) = 0.006000$$

6.5 Bemerkung. Mathematisch äquivalente Algorithmen auf Fließkommazahlen können je nach Implementierung zu wesentlich unterschiedlichen Ergebnissen führen, selbst wenn die Eingangszahlen exakt dargestellt werden.

6.1. Auslöschung

Unglücklicherweise pflanzen sich numerische Fehler, zum Beispiel durch Rundung im Verlauf eines Algorithmus fort.

6.6 Lemma (Fehlerfortpflanzung). Es seien $x, y \in \mathbb{R}$ mit Datenfehlern $\Delta x, \Delta y \in \mathbb{R}$ behaftet, welche

$$\left|\frac{\Delta x}{x}\right|, \left|\frac{\Delta y}{y}\right| \ll 1$$

erfüllen. Für $\circ \in \{+, -, \cdot, \div\}$ gilt für den fortgepflanzten Fehler

$$\Delta(x \circ y) \coloneqq (x + \Delta x) \circ (y + \Delta y) - x \circ y,$$

dass

$$\frac{\Delta(x \pm y)}{x \pm y} = \frac{x}{x \pm y}\frac{\Delta x}{x} \pm \frac{y}{x \pm y}\frac{\Delta y}{y}$$
$$\frac{\Delta(x \cdot y)}{xy} \approx \frac{\Delta x}{x} + \frac{\Delta y}{y}$$
$$\frac{\Delta(\frac{x}{y})}{\frac{x}{y}} \approx \frac{\Delta x}{x} - \frac{\Delta y}{y}$$

Hierbei bedeutet "≈", dass Terme mit $(\Delta x)^2, (\Delta y)^2, \Delta x \Delta y$ vernachlässigt werden.

Beweis. Übung. □

Für $\cdot, \div$ addieren bzw. subtrahieren sich die Fehler.

Achtung: Ist $|x \pm y|$ wesentlich kleiner als $|x|$ oder $|y|$, kann der relative Fehler massiv verstärkt werden. Dieses Phänomen heißt Auslöschung.
Bei der Konstruktion von Algorithmen sollte Auslöschung möglichst vermieden werden.

6.7 Beispiel. Betrachte

$$x^2 - 2px + q = 0 \tag{6.2}$$

mit Lösungen:

$$x_{1,2} = p \pm \sqrt[2]{p^2 - q}$$

Daten: $p, q \in \mathbb{R}$ so ,dass (6.2) lösbar ist
Ergebnis: Nullstellen x_1, x_2 von (6.2).
$d = \sqrt{p \cdot p - q}$
$x_1 = p + d$
$x_2 = p - d$

Algorithmus 2: naive Nullstellenberechung

Mit $p = 100$, $q = 1$ in dreistelliger, dezimaler Gleitkommaarithmetik ergibt sich:

$$d = \sqrt{10000 - 1} = \sqrt{rd(9999)} = \sqrt{10000} = 100$$
$$x_1 = 100 + 100 = 200$$
$$x_2 = 100 - 100 = 0$$

Die exakten Werte sind $x_1 \approx 199.99, x_2 \approx 0.00500$, welche in dreistelliger, dezimaler Gleitkommaarithmetik als $x_1 = 200, x_2 = 0$ dargestellt werden.
Gemäß 5.30 ist die Maschinengenauigkeit:

$$\varepsilon_{mach} = \frac{1}{2}b^{1-t} = \frac{1}{2}10^{1-3} = 0.005$$

Der relative Fehler $\frac{|0-0.005|}{|0.005|} = 1$ ist also zu 100 Prozent falsch.

Wir können die Genauigkeit von x_2 erhöhen, indem wir den Wurzelsatz von Vieta $x_1 x_2 = q$ verwenden.

Daten: $p, q \in \mathbb{R}$ so ,dass (6.2) lösbar ist.
Ergebnis: Nullstellen x_1, x_2 (6.2).
$d = \sqrt[2]{p \cdot p - q}$
wenn $p \geq 0$ **dann**
  $x_1 = p + d$ **sonst**
    $x_1 = p - d$
  Ende
Ende
$x_2 = \frac{q}{x_1}$

Algorithmus 3: verbesserte Nullstellenberechnung

Wir erhalten nun $d = 100, x_1 = 200, x_2 = \frac{1}{200} = 0.005$

7. Vorwärts- und Rückwertsanalyse

Abstrakt gesehen entspricht das Lösen eines Problems dem Auswerten einer Funktion f. Beim numerischen Auswerten von f können verschiedene Fehler passieren:

7.8 Definition (Fehlerarten). Sei $D \subset \mathbb{R}$ eine Menge von Eingangsdaten und $W \subset \mathbb{R}$ die Menge der möglichen Ergebnisse. Wir unterscheiden folgende Fehlerarten bei der Auswertung von $f\colon D \to W$.

- Datenfehler Typischerweise sind die Eingangsdaten $x \in D$ nicht exakt, sondern mit einem Datenfehler Δx behaftet. Die gestörten Eingangsdaten $\tilde{x} = x + \Delta x$ produzieren einen Fehler $f(x + \Delta x) - f(x)$.
- Verfahrensfehler Exakte Verfahren enden bei exakter Rechnung nach endlich vielen Operationen mit dem exakten Ergebnis. Näherungsverfahren enden in Abhängigkeit bestimmter Kriterien mit einer Näherung $\tilde{y}$ für die Lösung $y \in W$.
- Rundungsfehler Verursacht durch Maschinenzahlen und Rundungen während der Arithmetik.

Ein Teilgebiet der Numerik versucht diese Fehler durch eine Fehleranalyse zu quantifizieren. Hierzu verfolgt man die Auswirkungen von allen Fehlern, die in den einzelnen Schritten vorkommen können.

7.9 Definition (Analysearten). Bei der *Vorwärtsanalyse* wird der Fehler von Schritt zu Schritt verfolgt und der akkumulierte Fehler für jedes Teil-Ergebnis abgeschätzt.
Bei der *Rückwertsanalyse* geschieht die Verfolgung des Fehlers hingegen so, dass jedes Zwischenergebnis als exakt berechnetes Ergebnis zu gestörten Daten interpretiert wird, d.h , der akkumulierte Fehler wird als Datenfehler interpretiert.

7.10 Beispiel. Betrachte $f(x, y) = x + y$

- Vorwärtsanalyse: $\boxed{f} = x \boxplus y = (x+y)(1+\varepsilon)$
- Rückwärtsanalyse: $x \boxplus y = x(1+\varepsilon) + y(1+\varepsilon) = f(x(1+\varepsilon), y(1+\varepsilon))$

mit $|\varepsilon| \leq \varepsilon_{mach}$

In der Praxis ist die Vorwärtsanalyse kaum durchführbar. Für die meisten Algorithmen ist, wenn überhaupt, nur eine Rückwärtsanalyse bekannt.

7.11 Beispiel. Fortsetzung von Beispiel 6.7
Führe Rückwärtsanalyse durch zu Algorithmus 2 für die Betrags-kleinere Nullstelle x_2 durch: Dann ist

$$f(p,q) = p - \sqrt[2]{p^2 - q}$$

mit $p > 0$. Für $|\varepsilon_i| \leq \varepsilon_{mach}, i = 1,2,3,4$ betrachten wir:

$$\begin{aligned}\left(p - \sqrt[2]{(p^2(1+\varepsilon_1) - q)(1+\varepsilon_2)}(1+\varepsilon_3)\right)(1+\varepsilon_4) &= p(1+\varepsilon_4) - \sqrt[2]{(p^2(1+\varepsilon_1) - q)(1+\varepsilon_2)(1+\varepsilon_3)^2(1+\varepsilon_4)^2} \\ &= p^2(1+\varepsilon_1)(1+\varepsilon_3)^2(1+\varepsilon_4)^2 - q(1+\varepsilon_2)(1+\varepsilon_3)^2(1+\varepsilon_4)^2 \\ &= \dots \\ &= p(1+\varepsilon_4) - \sqrt[2]{p^2(1+\varepsilon_4)^2 - q(1+\varepsilon_7)} \\ &= f(p(1+\varepsilon_4), q(1+\varepsilon_7))\end{aligned}$$

Die Abschätzung für ε_7 explodiert, falls $0 < |q| \ll 1 < p$. Dies war in Beispiel 6.7 der Fall.

8. Kondition und Stabilität

Gegeben sei eine stetige und differenzierbare Funktion

$$f\colon \mathbb{R} \to \mathbb{R}, x \mapsto y = f(x)$$

Für fehlerhafte Daten $x + \Delta x$ mit kleinen Fehler Δx gilt:

$$\frac{f(x+\Delta x) - f(x)}{\Delta x} \approx f'(x).$$

Für den absoluten Datenfehler Δy gilt:

$$\Delta y = f(x+\Delta x) - f(x) \approx f'(x) \cdot \Delta x$$

und für den relativen Fehler:

$$\frac{\Delta y}{y} = \frac{f'(x) \cdot \Delta x}{f(x)} = \frac{f'(x)x}{f(x)} \cdot \frac{\Delta x}{x}$$

8.12 Definition (Konditionszahlen). Die Zahl

$$K_{abs} = |f'(x)|$$

heißt *absolute Konditionszahl* des Problems $x \mapsto f(x)$. Für $f(x) \cdot x \neq 0$ heißt

$$K_{rel} = \left| \frac{f'(x) \cdot x}{f\,(x)} \right|$$

die entsprechende *relative Konditionszahl*. Ein Problem heißt schlecht konditioniert, falls eine dieser beiden Konditionszahlen deutlich größer als 1 sind. Ansonsten heißt das Problem gut konditioniert.

8.13 Beispiel. Konditionierung der Addition und Multiplikation

- Für die Addition $f(x) = x + a$ gilt

$$K_{rel} = \left| \frac{f'(x)x}{f(x)} \right| = \left| \frac{x}{x+1} \right|$$

  $\Longrightarrow$ K_{rel} ist groß, falls $|x + a| \ll |x|$.

- Für die Multiplikation $f(x) = x \cdot a$ gilt

$$K_{rel} = \frac{|f'(x)x|}{|f(x)|} = \frac{|ax|}{|ax|} = 1$$

  $\Longrightarrow$ Die absolute Kondition ist schlecht, falls $1 \ll a$. Die relative Kondition ist immer gut.

8.14 Definition. Erfüllt die Implementierung eines Algorithmus $\boxed{f}$ zur Lösung eines Problems $x \mapsto f(x)$ die Abschätzung

$$\left| \frac{\boxed{f} - f(x)}{f(x)} \right| \leq C_V K_{rel} \varepsilon_{mach}$$

mit einem mäßig großen $C_V > 0$, so wird der Algorithmus *vorwärtsstabil* genannt. Ergibt die Rückwärtsanalyse $\boxed{f}(x) = f(x + \Delta x)$ mit

$$\left| \frac{\Delta x}{x} \right| \leq C_R \varepsilon_{mach}$$

mit $C_R > 0$ nicht zu groß, so heißt der Algorithmus $\boxed{f}$ *rückwärtsstabil*

8.15 Satz. Jeder rückwärtsstabile Algorithmus ist auch vorwärtsstabil

Beweis. Übung. □

Oft ist Rückwärtsstabilität einfacher nachzuweisen.

Faustregel zu konditionierten Probleme

- Gut konditioniertes Problem + stabiler Algorithmus
  $\Longrightarrow$ Gute numerische Resultate.
- Schlecht konditioniertes Problem oder instabiler Algorithmus
  $\Longrightarrow$ Fragwürdige Ergebnisse.

8.16 Beispiel. Fortsetzung von Beispiel 7.11
Die Rückwärtsanalyse hat gezeigt: Falls $0 < |q| \ll 1 < p$ wird der numerische Fehler untragbar. Unsere Abbildung ist:

$$f(q) = p - \sqrt[2]{p^2 - 1}$$

Als Konditionszahlen ergeben sich:

$$K_{abs} = |f'(q)| = \left|\frac{1}{2\sqrt[2]{p^2 - q}}\right| < 1$$

$$\begin{aligned} K_{rel} &= \left|\frac{f'(q)q}{f(q)}\right| \\ &= \left|\frac{q}{2\sqrt[2]{p^2-q}(p - \sqrt[2]{p^2-q})(p + \sqrt[2]{p^2-q})}\right| \\ &= \frac{1}{2}\left|\frac{p + \sqrt[2]{p^2-q}}{\sqrt[2]{p^2-q}}\right| \\ &\approx \frac{1}{2}\left|\frac{p+p}{p}\right| \approx 1 \end{aligned}$$

$\Longrightarrow$ Nullstellenberechung ist ein gut konditioniertes Problem, aber Algorithmus 2 muss instabil sein.

Dreitermrekursion

9. Theoretische Grundlagen

9.1 Definition. Für gegebene p_0 und p_1 heißt eine Rekursion der Form

$$p_k = a_k p_{k-1} + b_k p_{k-2} + c_k,\ k = 2, 3, \ldots \tag{9.1}$$

mit $b_k \neq 0$ eine *Dreitermrekursion*. Die zugehörige *Rückwärtsrekursion* ist

$$p_{k-2} = -\frac{a_k}{b_k} p_{k-1} + \frac{1}{b_k} p_k - \frac{c_k}{b_k},\ k = n, n-1, \ldots \tag{9.2}$$

Ist $b_k = 1$, das heißt (9.1) und (9.2) gehen durch vertauschen von p_{k-2} und p_k auseinander hervor, so heißt die Rekursion symmetrisch. Gilt $c_k = 0$ für alle k, so heißt die Rekursion homogen.

9.2 Beispiel. Die Fibonacci-Zahlen sind rekursiv definiert durch:

$$p_k = p_{k-1} + p_{k-2}$$

mit $p_0 = 0, p_1 = 1$. Es gilt $a_k = b_k = 1$.
Sei

$$p_k(x) = 2\cos(x) p_{k-1}(x) - 1 \cdot p_{k-2}(x)$$

mit $p_0 = 1$ und $p_1 = \cos(x)$. Man kann zeigen, dass

$$p_k(x) = 2\cos(kx)$$

ist.
Die Chebychev-Polynome T_k erfüllen

$$T_k(x) = 2xT_{k-1} - T_{k-2}(x)$$

mit $T_0(x) = 1$ und $T_1(x) = x$

Daten: Koeffizienten von $\{a_k\}_{k=2}^n$, $\{b_k\}_{k=2}^n$,$\{c_k\}_{k=2}^n$, Startwerte p_0, p_1
Ergebnis: Werte von $\{p_k\}_{k=2}^n$
für $k \leftarrow 2$ **bis** n **tue**
| $p_k = a_k p_{k-1} + b_k p_{k-2} + c_k$
Ende

Algorithmus 4: Dreitermrekursion

9.3 Beispiel. Betrachte: $p_0 = 1, p_1 = \sqrt[2]{2} - 1, p_k = -2p_{k-1} + p_{k-2}$, $k = 2, 3, \ldots$
Algorithmus 5 liefert:

Die Oszillationen für höhere k sollten uns misstrauische machen und uns motivieren das Problem genauer anzuschauen.

Wir können homogene Dreitermrekursion schreiben als:

$$\begin{bmatrix} p_k \\ p_{k-1} \end{bmatrix} = \begin{bmatrix} a_k p_{k-1} + b_k p_{k-2} \\ p_{k-1} \end{bmatrix} =: \begin{bmatrix} a_k & b_k \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} p_{k-1} \\ p_{k-2} \end{bmatrix} = A_k \begin{bmatrix} p_{k-1} \\ p_{k-2} \end{bmatrix}$$

Rekursiv folgt:

$$\begin{bmatrix} p_k \\ p_{k-1} \end{bmatrix} = A_k \begin{bmatrix} p_{k-1} \\ p_{k-2} \end{bmatrix} = A_k A_{k-1} \begin{bmatrix} p_{k-2} \\ p_{k-3} \end{bmatrix} = \ldots = A_k \ldots A_2 \begin{bmatrix} p_1 \\ p_0 \end{bmatrix}$$

Offensichtlich gilt für alle $\alpha, \beta \in \mathbb{R}$ und Startwerte p_0, p_1, q_0, q_1 und $B_k \coloneqq A_k \ldots A_2$, dass

$$B_k \begin{bmatrix} \alpha p_1 + \beta q_1 \\ \alpha p_0 + \beta q_0 \end{bmatrix} = \alpha B_k \begin{bmatrix} p_1 \\ p_0 \end{bmatrix} + \beta B_k \begin{bmatrix} q_1 \\ q_0 \end{bmatrix} = \alpha \begin{bmatrix} p_k \\ p_{k-1} \end{bmatrix} + \beta \begin{bmatrix} q_k \\ q_{k-1} \end{bmatrix}$$

Das heißt, die Lösungsfolge $\{p_k\}$ hängt *linear* von den Startwerten $\begin{bmatrix} p_1 \\ p_0 \end{bmatrix} \in \mathbb{R}^2$ ab. Im Falle unseres Beispiels gilt: $a_k = a = -2$, $b_k = b = 1$. Das bedeutet, es gilt:

$$B_k = A^{k-1}, A = \begin{bmatrix} a & b \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} -2 & 1 \\ 1 & 0 \end{bmatrix}$$

Aus den Eigenwerten λ_1, λ_2 der Matrix A kann man die Lösungen der homogenen Dreitermrekursion direkt angeben.

9.4 Satz. Seien λ_1, λ_2 die Nullstellen des *charakteristischen Polynoms*

$$q(\lambda) = \lambda^2 - a\lambda - b \tag{9.3}$$

Dann ist die Lösung der homogenen Dreitermrekursion:

$$p_k = ap_{k-1} + bp_{k-2}, k = 2, 3, \ldots$$

gegeben durch:

$$p_k = \alpha\lambda_1^k + \beta\lambda_2^k, k = 2, 3, \ldots$$

mit $\alpha, \beta \in \mathbb{R}$ Lösungen des linearen Gleichungssystems.

$$\alpha + \beta = p_0$$

$$\alpha\lambda_1 + \beta\lambda_2 = p_1$$

Beweis. • Induktion über k

◊ Induktionsanfang: $\underline{k = 0, k = 1}$: $p_0 = \alpha + \beta = \alpha\lambda_1^0 + \beta\lambda_2^0, p_1 = \alpha\lambda_1^1 + \beta\lambda_2^1$

◊ Induktionsschritt: $\underline{k-1, k \to k+1}$:

$$\begin{aligned} p_{k+1} &= ap_k + bp_{k-1} \\ &= a(\alpha\lambda_1^k + \beta\lambda_2^k) + b(\alpha\lambda_1^{k-1} + \beta\lambda_2^{k-1}) \\ &= \alpha(a\lambda_1^k + b\lambda_1^{k-1}) + \beta(a\lambda_2^k + b\lambda_2^{k-1}) \\ &= \alpha\lambda_1^{k-1}(a\lambda_1 + b) + \beta\lambda_2^{k-1}(a\lambda_2 + b) \\ &= \alpha\lambda_1^{k+1} + \beta\lambda_2^{k+1} \end{aligned}$$

Da nach (9.3) λ_1^2 und λ_2^2 Nullstellen sind.

□

Wende 9.4 auf das Beispiel 9.3 an: Mit $a = -2, b = 1$ hat das charakteristische Polynom (9.3) die Nullstellen:

$$\lambda_1 = \sqrt[2]{2} - 1, \lambda_2 = -\sqrt[2]{2} - 1$$

Aus

$$\alpha + \beta = p_0 = 1$$
$$\alpha\lambda_1 + \beta\lambda_2 = p_1 = \sqrt[2]{2} - 1$$

folgt $\alpha = 1, \beta = 0$. Der Satz 9.4 impliziert:

$$p_k = \lambda_1^k > 0, k = 0, 1, 2, \ldots$$

in Beispiel 9.3

Daten: Koeffizienten a,b, Startwerte p_0, p_1
Ergebnis: Werte $\{p_k\}_{k=2}^n$
Initialisiere:

- $\lambda_1 = \frac{a}{2} + \sqrt[2]{\frac{a^2}{4} + b}$
- $\lambda_2 = \frac{a}{2} - \sqrt[2]{\frac{a^2}{4} + b}$
- $\beta = \frac{p_1 - \lambda_1 p_0}{\lambda_2 - \lambda_2}$
- $\alpha = p_0 - \beta$

für $k \leftarrow 2$ **bis** n **tue**
  $p_k = \alpha\lambda_1^k + \beta\lambda_2^k$
Ende

Algorithmus 5: verbesserte Dreitermrekursion

Dieser Algorithmus liefert für Beispiel 9.3 folgende Grafik:
Dies ist in diesem Fall das richtige Ergebnis, da

$$p_k = \lambda_1^k = (\sqrt[2]{2} - 1)^k \ll 1$$

Was geht schief in dem vorherigen Algorithmus 5?
Betrachten wir unser Problem $f\colon \mathbb{R}^2 \to \mathbb{R}$, $f(p_0, p_1) = p_k$ mit gestörten Eingangsdaten: $\hat{p_0} = 1(1+\varepsilon_0)$, $\hat{p_1} = \lambda_1(1+\varepsilon_1)$, $|\varepsilon_0|, |\varepsilon_1| \leq \varepsilon_{mach}$.
Dies ergibt:

$$\tilde{\beta} = \frac{\lambda_1(1+\varepsilon_1) - \lambda_1(1+\varepsilon_0)}{\lambda_2 - \lambda_1} = (\varepsilon_1 - \varepsilon_0)\frac{\lambda_1}{\lambda_2 - \lambda_1}$$

$$\tilde{\alpha} = 1 + \varepsilon_0\frac{\lambda_2 - \lambda_1}{\lambda_2 - \lambda_1} - (\varepsilon_1 - \varepsilon_0)\frac{\lambda_1}{\lambda_2 - \lambda_1} = 1 + \varepsilon_0\frac{\lambda_2}{\lambda_2 - \lambda_1} - \varepsilon_1\frac{\lambda_1}{\lambda_2 - \lambda_1}$$

und somit

$$\begin{aligned}\tilde{p_k} &= \tilde{\alpha}\lambda_1^k + \tilde{\beta}\lambda_2^k \\ &= (1 + \varepsilon_0\frac{\lambda_2}{\lambda_2 - \lambda_1} - \varepsilon_1\frac{\lambda_1}{\lambda_2 - \lambda_1})\lambda_1^k + (\varepsilon_1 - \varepsilon_0)\frac{\lambda_1}{\lambda_2 - \lambda_1}\lambda_2^k\end{aligned}$$

Der relative Fehler von $\tilde{p_k}$ zu $p_k = \lambda_1$ ist somit :

$$\begin{aligned}\left|\frac{\tilde{p_k} - p_k}{p_k}\right| &= \left|\varepsilon_0\frac{\lambda_2}{\lambda_2 - \lambda_1} - \varepsilon_1\frac{\lambda_1}{\lambda_2 - \lambda_1} + (\varepsilon_1 - \varepsilon_0)\frac{\lambda_1}{\lambda_2 - \lambda_1}\left(\frac{\lambda_2}{\lambda_1}\right)^k\right| \\ &= \left|\varepsilon_0 + (\varepsilon_1 - \varepsilon_0)\frac{\lambda_1}{\lambda_2 - \lambda_1}\left(\left(\frac{\lambda_2}{\lambda_1}^k\right) - 1\right)\right|\end{aligned}$$

Beobachte: Falls $\frac{\lambda_2}{\lambda_1} > 1$ explodiert der relative Fehler für wachsende k. Dies war in Beispiel 9.3 der Fall.
Falls wir den Begriff der Konditionszahl aus der Definition 8.12 geeignet für Funktion $\mathbb{R}^2 \to \mathbb{R}$ erweitern, sehen wir, dass das Problem schlecht konditioniert ist.

9.5 Definition (Minimallösung). Die Lösung $\{p_k\}$ der Dreiterm-Rekursion (9.1) zu den Startwerten p_0 und p_1 heißt *Minimallösung*, falls für jede Lösung $\{p_k\}$ zu den von p_0, p_1 linear unabhängigen Startwerten q_0, q_1 gilt, dass:

$$\lim_{k\to\infty}\frac{p_k}{q_k} = 0$$

Die Lösung $\{q_k\}$ wird dominante Lösung genannt.

9.6 Beispiel. In Beispiel 9.3 ist $\{p_k\}$ Minimallösung genau dann, wenn $\beta = 0$.

Die Minimallösung ist nur bis auf einen skalaren Faktor eindeutig. Deshalb normieren wir sie mit der zusätzlichen Bedingung: $p_0^2 + p_1^2 = 1$.

Frage: Wie können wir eine normierte Minimallösung berechnen, obwohl das Problem schlecht konditioniert ist?

10. Miller-Algorithmus

Betrachte die Rückwärtsrekursion zu

$$q_k = a_k q_{k-1} + b_k q_{k-2}, k = 2, 3, \ldots$$

d.h.

$$q_{k-2} = -\frac{a_k}{b_k} q_{k-1} + \frac{1}{b_k} q_k, k = n, n-1, \ldots, 2$$

mit Startwerten $q_n = 0, q_{n-1} = 1$. Für $a_k = a, b_k = b$ besitzt das charakteristische Polynom

$$q(\mu) = \mu^2 + \frac{a}{b}\mu - \frac{1}{b}$$

Die Nullstellen:

$$\mu_{1,2} = \frac{-a \pm \sqrt[2]{a^2 + 4b}}{2b} = \frac{1}{\lambda_{1,2}}$$

Idee: Wende Satz 9.4 "rückwärts" an.
Lösen wir

$$\alpha + \beta = 0$$

und

$$\alpha\mu_1 + \beta\mu_2 = q_{n-1} = 1$$

ergibt sich:

$$\alpha = \frac{\lambda_1\lambda_2}{\lambda_2 - \lambda_1} \neq 0$$

$$\beta = -\frac{\lambda_1\lambda_2}{\lambda_2 - \lambda_1} \neq 0$$

und

$$q_k = \alpha\mu_1^{n-k} + \beta\mu_2^{n-k} = \frac{\alpha}{\lambda_1^{n-k}} + \frac{\beta}{\lambda_2^{n-k}}$$

Für $|\lambda_1| < |\lambda_2|$ folgt, dass

$$\begin{aligned} p_k^{(n)} &:= \frac{q_k}{q_0} \\ &= \frac{\frac{\alpha}{\lambda_1^{n-k}} + \frac{\beta}{\lambda_2^{n-k}}}{\frac{\alpha}{\lambda_1^n} + \frac{\beta}{\lambda_2^n}} \\ &= \frac{\lambda_1^k + \frac{\beta}{\alpha}\lambda_2^k \left(\frac{\lambda_1}{\lambda_2}\right)^n}{1 + \frac{\beta}{\alpha}\left(\frac{\lambda_1}{\lambda_2}\right)^n} \\ &\to \lambda_1^k = p_k \end{aligned}$$

Beobachtung: Für großes n approximiert $p_k^{(n)}$ die Minimallösung.

Daten: n genügend groß $\{a_k\}_{k=2}^n, \{b_k\}_{k=2}^n$
Ergebnis: Approximation von $\{p_k^{(n)}\}_{k=0}^n$ um eine normierte Minimallösung
Setze $\hat{p_n} = 0, \hat{p_{n-1}} = 1$
für $k \leftarrow n$ **bis** 2 **tue**
  $\hat{p_{k-2}} = -\frac{a_k}{b_k}\hat{p_{k-1}} + \frac{1}{b_k}\hat{p_k}$
Ende
für $k \leftarrow 0$ **bis** n **tue**
  $p_k^{(n)} = \frac{\hat{p_n}}{\sqrt{\hat{p_0^2}+\hat{p_n^2}}}$
Ende

Algorithmus 6: Miller-Algorithmus

10.7 Satz (Miller-Algorithmus). Sei $\{p_k\}_{k=0}^\infty$ Minimallösung der normierten, homogenen Dreitermrekursion. Dann gilt für die Lösung des Miller-Algorithmus:

$$\lim_{n\to\infty} p_k^{(n)} = p_k, k = 0, 1, 2, \ldots$$

Beweis. Wir bemerken, dass sich jede Lösung der Dreitermrekursion als Linearkombination einer Minimallösung und einer dominanten Lösung schreiben lässt (Übung). Mit den richtigen α und β folgt (Übung):

$$\begin{aligned}\hat{p_k} &= \alpha p_k + \beta q_k \\ &= \frac{p_k q_n - q_k p_n}{p_{n-1}q_n - q_{n-1}p_n} \qquad = \frac{q_n}{p_{n-1}q_n - q_{n-1}p_n}\left(p_k - \frac{p_n}{q_n}q_k\right)\end{aligned}$$

Aus (Normierungs-Konstante)

$$\begin{aligned}\hat{p_0^2} + \hat{p_1^2} &= \frac{q_n^2}{(p_{n-1}q_n - q_{n-1}p_n)^2}\left(p_0^2 + p_1^2 - 2\frac{p_n}{q_n}(p_0q_0 + p_1q_1) + \frac{p_n^2}{q_n^2}(q_0^2 + q_1^2)\right) \\ &= \frac{q_n^2}{(p_{n-1}q_n - q_{n-1}p_n)^2}\left(1 - 2\frac{p_n}{q_n} + \frac{p_n^2}{q_n^2}\right)\end{aligned}$$

folgt:

$$\begin{aligned}p_k^{(n)} &= \frac{\hat{p_k}}{\sqrt{\hat{p_0^2} + \hat{p_1^2}}} \\ &= \frac{p_k - \frac{p_n}{q_n}q_k}{\sqrt{1 - 2\frac{p_n}{q_n} + \frac{p_n^2}{q_n^2}}} \qquad \to p_k\end{aligned}$$

□

Wir können also auch für schlecht konditionierte Probleme stabile Algorithmen finden.

Sortieren

11. Das Sortierproblem

Gegeben $n \in \mathbb{N}$ verschiedene Zahlen $z_1, \ldots, z_n \in \mathbb{R}$.

Gesucht Permutation $\pi_1, \ldots, \pi_n$, so dass $z_{\pi_1} < \ldots < z_{\pi_n}$

11.1 **Definition** (Permutation). Eine Permutation π von $\{1, 2, \ldots, n\}$ ist eine bijektive Abbildung von $\{1, 2, \ldots, n\}$ auf sich selbst. Wir schreiben $\pi(k) = \pi_k$ für $k = 1, \ldots, n$

11.2 **Bemerkung.** Da wir die Zahlen $z_1, \ldots, z_n$als verschieden annehmen ist das Sortierproblem eindeutig lösbar.

Probiere so lange alle möglichen Permutationen durch, bis die gewünschte Sortierung vorliegt

Algorithmus 7: Brute-Force

11.3 **Satz.** Es gibt $n! = n(n-1)\ldots 2 \cdot 1$ Permutationen der Menge $\{1, 2, \ldots, n\}$

Beweis. Wir haben

- n Möglichkeiten die erste Zahl auszuwählen
- $n-1$ Möglichkeiten die zweite Zahl auszuwählen
- …
- 1 Möglichkeit die letzte Zahl auszuwählen

woraus die Behauptung folgt. □

Im schlimmsten Fall (worst case) muss der Algorithmus 7 also

$$(n-1) \cdot n!$$

Vergleiche durchführen. Da dies extrem aufwändig sein kann, betrachten wir im Folgenden einem Algorithmus, der die transitive Struktur

$$x < y \text{ und } y < z \implies x < z$$

der Ordnungsrelation ausnutzt.

Daten: Menge $S_n = \{z_1, \ldots, z_n\}$
Ergebnis: Sortierte Menge $S^\pi = \{z_{\pi_1}, \ldots, z_{\pi_n}\}$
für $k \leftarrow n$ **bis** 1 **tue**
  $z_{\pi_k} = \max(S_k)$
  $S_{k-1} = S_k \setminus \{z_{\pi_k}\}$
Ende

Algorithmus 8: Bubblesort

11.4 Beispiel. Die zu sortierende Menge ist: $\{4, 1, 2\}$.

- $S_3 = \{4, 1, 2\}$, $k = 3$, $z_{\pi_3} = 4$
- $S_2 = \{1, 2\}$, $k = 2$, $z_{\pi_2} = 2$
- $S_1 = \{1\}$, $k = 1$, $z_{\pi_1} = 1$

Die sortierte Liste ist: $\{1, 2, 4\}$.

Um den Aufwand von Algorithmen zu untersuchen, beschränken wir uns auf das asymptotische Verhalten für große n.

11.5 Definition (Landau-Notation). Wir schreiben

- $f(x) = \mathcal{O}(g(x))$, falls Zahlen $C > 0, x_0 > 0$ existieren, so dass: $|f(x)| \leq Cg(x), \forall_{x>x_0}$.
- $f(x) = \Omega(g(x))$, falls Zahlen $C > 0, x_0 > 0$ existieren, so dass $|f(x)| \geq Cg(x) \forall_{x>x_0}$.
- $f(x) = o(g(x))$, falls für jedes $c > 0$ ein $x_0 > 0$ existiert, so dass $|f(x)| \leq cg(x), \forall_{x>x_0}$.
- $f(x) = \Theta(g(x))$, falls $f(x) = \mathcal{O}(g(x))$, $g(x) = \mathcal{O}(f(x))$

11.6 Bemerkung. Genau dann wenn Beziehung der Landau-Notation

- $f(x) = \mathcal{O}(g(x)) \iff \limsup_{x\to\infty} |\frac{f(x)}{g(x)}| \leq C < \infty$
- $f(x) = \Omega(g(x)) \iff \liminf_{x\to\infty} |\frac{f(x)}{g(x)}| > 0$
- $f(x) = o(g(x)) \iff \lim_{x\to\infty} |\frac{f(x)}{g(x)}| = 0$
- $f(x) = \Theta(g(x)) \iff f(x) = \mathcal{O}(g(x)), f(x) = \Omega(g(x))$

11.7 Beispiel. Es gilt $\sin(x) = \mathcal{O}(1)$, da $|\sin(x)| \leq 1$ für alle $x \in \mathbb{R}$. Bei Polynomen gilt die Laufzeit ist die höchste Potenz, sofern $x \geq 1$.

11.8 Definition. Der Aufwand eines Algorithmus ist die kleinste obere Schranke für das betrachtete Aufwandsmaß

Für uns ist der Speicherbedarf irrelevant und benutzen als Aufwandsmaß für den Rechen die Anzahl der benötigten Vergleiche.
In der k-ten Iteration des Bubblesort-Algortihmus 8 müssen k Vergleiche ausgeführt werden. Das heißt, der Gesamtaufwand des Algorithmus ist:

$$\sum_{k=1}^{n-1} k = \frac{n(n-1)}{2} = \mathcal{O}(n^2)$$

Dies ist eine drastische Verbesserung im Vergleich zu $\mathcal{O}(n(n!))$ von Algorithmus 7

12. Mergesort

Zu erst wollen wir folgendes beobachten:

12.9 Lemma. Gegeben seien zwei sortiere Mengen

- $S_x = \{x_1 < \ldots < x_m\}$
- $S_y = \{y_1 < - \ldots < y_n\}$

Dann lässt sich die Menge $S = S_x \cup S_y$ mit linearem Aufwand sortieren. Genauer werden $m + n - 1$ Vergleiche benötigt.

Beweis. Konstruktiv, durch den entsprechenden Algorithmus. □

Daten: Sortierte Mengen S_x und S_y
Ergebnis: Sortierte Menge $S = S_x \cup S_y$
Initialisiere $i = j = k = 1$
solange $i \leq m$ *und* $j \leq n$ **tue**
  wenn $x_i < y_j$ **dann**
    $z_k = x_i$
    $i = i + 1$
  Ende
  sonst
    $z_k = y_j$
    $j = j + 1$
  Ende
  $k = k + 1$
Ende
für $l \leftarrow 0$ **bis** $m - i$ **tue**
  $z_{k+l} = x_{k+l}$
Ende
für $l \leftarrow 0$ **bis** $n - j$ **tue**
  $z_{k+l} = y_{k+l}$
Ende

Algorithmus 9: Merge

Basierend auf dieser Beobachtung können wir eine divide-and-conquer Strategie angeben um Mengen der Länge $n = 2^m, m \in \mathbb{N}$ zu sortieren.

Daten: Menge $S = \{z_1, \ldots, z_n\}$
Ergebnis: Sortierte Menge $S^\pi = \{z_{\pi_1} < \ldots < z_{\pi_n}\}$
wenn $n = 1$ **dann**
| $S^\pi = S$
Ende
sonst
- Sortiere
  $L = \{z_1, \ldots, z_{\frac{n}{2}}\}$
  $R = \{z_{\frac{n}{2}+1}, \ldots, z_n\}$ mittels Mergesort zu L^π und R^π
- Sortiere $L^\pi \cup R^\pi$ mittels Merge-Algorithmus 9 zu S^π

Ende

Algorithmus 10: Mergesort

12.10 Beispiel. Mergesort der Menge $\{20, 7, 84, 31, 71, 42, 18, 10\}$

12.11 Bemerkung. Da Mergesort sich selbst aufruft, sprechen wir von einem *rekursiven Algorithmus*. Im Allgemeinem ist es schwierig zu beurteilen, ob solche Algorithmen terminieren. Im Fall von Mergesort ist der Fall jedoch klar, da die Rekursion im Fall $n = 1$ abgebrochen wird.

12.12 Satz. Der Aufwand von Mergesort ist $\mathcal{O}(n \log n)$

Beweis. Bezeichne A(n) den Aufwand für das Sortieren einer $n = 2^m$ elementigen Menge mittels Mergesort. Dann gilt:

$$\begin{aligned} A(1) &= 0 \\ A(n) &= n - 1 + 2A(\frac{n}{2}) \end{aligned}$$

Auflösen der Rekursion ergibt:

$$\begin{aligned} A(n) &= n - 1 + 2A(\frac{n}{2}) \\ &= 2n - 1 - 2 + 4A(\frac{n}{4}) \\ &= \ldots \\ &= mn - \sum_{i=0}^{m-1} 2^i \\ &= mn - \frac{1 - 2^m}{1 - 2} \\ &= (m-1)n + 1 \end{aligned}$$

$m = \log_2(n) = \frac{\log(n)}{\log(2)}$ impliziert die Behauptung □

$\log n \ll n$, also ist die Verbesserung von $\mathcal{O}(n^2)$ nach $\mathcal{O}(n \log(n)$ signifikant. Man nennt das Wachstum auch beinahe linear.

12.13 Bemerkung. Die Implementierung von Mergesort als in-place-Algorithmus ist je nach Datenstrukturen trickreich. Deshalb wird oft nicht in-place implementiert, weshalb für jeden "divide"-Schritt zusätzlicher Speicherplatz implementiert werden muss.

13. Quicksort

Idee: Divide-and-Conquer Strategie basierend auf dem Inhalt der zu sortierenden Liste.

Daten: Menge $S = \{z_1, \ldots, z_n\}$
Ergebnis: Sortierte Menge $S^\pi = \{z_{\pi_1}, \ldots, z_{\pi_n}\}$
Wähle ein Pivot-Element $x \in S$
Bestimme eine Permutation π, so dass $x = z_{m_\pi}$
wenn $L = \{z_{\pi_1}, \ldots, z_{\pi_{m-1}} \neq \emptyset\}$ **dann**
| Sortiere L zu L^π mittels Quicksort
Ende
wenn $R = \{z_{\pi_{m+1}}, \ldots, z_{\pi_n}\} \neq \emptyset$ **dann**
| Sortiere R zu R^π mittels Quicksort
Ende
Vereinige $S^\pi = R^\pi \cup \{x\} \cup L^\pi$

Algorithmus 11: Quicksort

13.14 Beispiel. Beispiel anhand der gleichen Menge von Mergesort.

13.15 Lemma. Im schlimmsten Fall ist der Aufwand von Quicksort $\mathcal{O}(n^2)$

Beweis. $A(n)$ aus 12.12 wird umso größer, je unterschiedlicher die Größe der beiden Teilprobleme $A(m-1)$ und $A(n-m)$ ist. $A(n)$ ist also maximal für $m = 1$ oder $m = n$, also wenn das Pivot-Element das größte oder kleinste Element ist. Dann gilt:

$$A(n) = n - 1 + A(n-1)$$

Der Rest erfolgt Analog zu der Aufwandserklärung von Bubblesort 8:

$$A(n) = \frac{n(n-1)}{2} = \mathcal{O}(n^2)$$

Damit ist der Aufwand gezeigt. □

13.16 Satz. Alle Permutationen der Zahlen $\{1, 2, \ldots, n\}$ seien gleich wahrscheinlich. Dann benötigt Quicksort im Durchschnitt $\mathcal{O}(n \log(n)$ Vergleiche zum sortieren von Zahlen.

Beweis. Sie Π die Menge aller Permutationen von $\{1, 2, \ldots, n\}$ und sei $A(\pi), \pi \in \Pi$ die Anzahl Vergleiche um eine Permutation π mittels Quicksort zu sortieren. Der durchschnittliche Aufwand ist:

$$\overline{A}(n) = \frac{1}{n!} \sum_{\pi \in \Pi} A(\pi)$$

O.B.d.A: Sei das erste Element das Pivotelement. Definiere:

$$\Pi_k = \{\pi \in \Pi | \pi_1 = k\}, k = 1, \ldots, n$$

mit

$$|\Pi_k| = (n-1)!, k = 1, \ldots, n$$

Für k fix und $\pi \in \Pi_k$ teilt Quicksort im ersten Aufruf in zwei Mengen

$$\pi_< = \{\text{Permutationen von } 1, 2, 3 \ldots, k-1\}$$
$$\pi_> = \{\text{Permutationen von } k+1, \ldots, n\}$$

Analog zu 12.12 folgt

$$A(\pi) = n - 1 + A(\pi_<) + A(\pi_>)$$

und

$$\sum_{\pi \in \pi_k} A(\pi) = (n-1)!(n-1) + \sum_{\pi \in \Pi_k} A(\pi_<) + \sum_{\pi \in \Pi_k} A(\pi_>) \tag{13.1}$$

Wenn π alle Permutationen aus Π_k durchläuft, entstehen für $\Pi_<$ alle Permutationen von $\{1, 2, \ldots, k-1\}$. Dabei tritt jede Permutation genau $\frac{(n-1)!}{(k-1)!} = \frac{|\Pi_k|}{\Pi_<}$ mal auf.

$$\Longrightarrow \sum_{\pi \in \Pi_k} A(\pi_<) = \frac{(n-1)!}{(k-1)!} \sum_{\pi \in \Pi_k} A(\pi_<) = (n-1)!\overline{A}(k-1) \tag{13.2}$$

Analog:

$$\sum_{\pi \in \Pi_k} A(\pi_>) = (n-1)!\overline{A}(n-k) \tag{13.3}$$

Zusammensetzen:

$$\begin{aligned} \overline{A}(n) &= \frac{1}{n!} \sum_{\pi \in \Pi} A(\pi) \\ &= \frac{1}{n!} \sum_{k=1}^{n} \sum_{\pi \in \Pi_k} \end{aligned}$$

Unter Verwendung der Gleichungen 13.1, 13.2, 13.3 folgt weiter:

$$\begin{aligned}
&= \frac{1}{n!}\sum_{k=1}^{n}(n-1)!(n-1+\overline{A}(k-1)+\overline{A}(n-k) \\
&= n-1+\frac{1}{n}\sum_{k=1}^{n}\overline{A}(k-1)+\overline{A}(n-k) \\
&= n-1+\frac{2}{n}\sum_{k=0}^{n-1}\overline{A}(k)
\end{aligned}$$

mit Startwerten $\overline{A}(0) = \overline{A}(1) = 0$.
Per Induktion folgt:

$$\begin{aligned}
\overline{A}(n) &= 2(n+1)\sum_{i=1}^{n}\frac{1}{i} - 4n \\
&\leq 2(n+1)(1+\int_{1}^{n}\frac{1}{x}dx - 4n \\
&= 2(n+1)(1+\log(n)) - 4n \\
&= 2(n+1)\log(n) - 2(n-1) \\
&= \mathcal{O}(n\log(n))
\end{aligned}$$

□

14. Untere Schranke für das Sortierproblem

Frage: Gibt es einen Sortieralgorithmus, der schneller ist als $\mathcal{O}(n\log(n))$?

14.17 Satz. Jedes deterministische Sortierverfahren, das auf paarweisen Vergleichen basiert und keine Vorkenntnisse über die zu sortierende Menge hat, benötigt im schlimmsten Fall mindestens $\log_2(n!)$ Vergleiche zum Sortieren von n verschiedenen Zahlen.

Bevor wir den Beweis machen gibt es noch einige Hinweise, die wir für den Beweis benötigen.

14.18 Bemerkung. Es gilt:

- $\log_2(n!) = \Theta(n\log(n)$
- $\log_2(n) \leq \log_2(n^n) = n\log_2(n) = \frac{1}{\log(2)}n\log n$
- $\log(n!) \geq \log(\frac{n}{2}\ldots\frac{n}{2}) \geq \log((\frac{n}{2})^{\frac{n}{2}} = \frac{n}{2}\log(\frac{n}{2}) = \frac{n}{2}\log(n) - n\log(2)$

Zum Beweis benötigen wir einen Entscheidungsbaum. Dieser kann jedem Sortieralgorithmus mit paarweisen Vergleichen zugeordnet werden, und illustriert das Verhalten des Algorithmus:

- Innere Knoten des Entscheidungsbaums sind Vergleiche im Algorithmus.
- Ein Weg von der Wurzel zu einem Blatt entspricht der zeitlichen Abfolge der Vergleiche. Ein Weitergeben nach rechts illustriert einen richtigen Vergleich, nach links einen falschen.
- Die $n!$ Blätter des Baumes stellen die Permutationen der zu sortierenden Menge dar, die jeweils zur Abfolge der Vergleiche gehört.

14.19 Beispiel. Baum für $\{z_1, z_2, z_3\}$:

Beweis. 14.17
Idee: Finde eine untere Schranke für die Tiefe des Baumes. Dies ist die Mindestanzahl der im schlimmsten Fall mötigen Vergleiche. Im besten Fall ist ein Baum ausbalanciert, d.h. alle Blätter haben die gleiche Tiefe m und wir haben 2^m Blätter. Da wir $n!$ Permutationen haben (also Blätter), muss $2^m = n!$ gelten. Es folgt $m \log_2(n!)$ □

Satz 14.17 sagt, dass Mergesort in der asymptotischen Laufzeit optimal ist.

Frage: Können wir etwas Ähnliches für Quicksort zeigen?

14.20 Satz. Alle Permutationen der Zahlen $z_1, \ldots, z_n$ seien gleich wahrscheinlich. Dann benötigen Sortierverfahren wie in Satz 14.17 im Mittel $\log_2(n!)$ Vergleiche.

Beweis. Die mittlere Höhe $\overline{H}(z)$ des Entscheidungsbaums τ ist gegeben als der Mittelwert der Tiefen t_τ aller seiner Blätter v. Sei $B(\tau)$ die Menge der Blätter und $\beta(\tau) = |B(\tau)|$ Dann gilt:

$$\overline{H}(\overline{c}) = \frac{1}{\beta(\overline{c})} \sum_{y=B(\overline{c}} t\overline{c}(v)$$

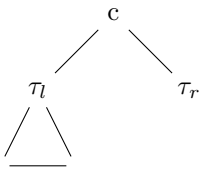
Zu zeigen ist, dass:

$$\overline{H}(\overline{c}) \geq \log_2(\beta(\overline{c}))$$

Vollständige Induktion über die Höhe des Entscheidungsbaums $\overline{c}$.

◊ Induktionsverankerung:$H(\overline{c}) = 0$
$\overline{c}$ besteht nur aus seiner Wurzel. $\implies \overline{H}(c) = \frac{1}{1} \sum_{y \in B(\tau)} t_\tau(v) = 0 = \log_2(1)$

◊ Induktionsschritt: $H(\overline{c}) \leq h - 1 \implies H(\overline{c}) = h$
Seien τ_l, τ_r die Wurzeln des linken bzw. rechten Teilbaums zum Wurzelknoten von c:



Beobachte:

- $H(\tau_l), H(\tau_r) < h$
- $B(\tau) = B(\tau_l) \cup B(\tau_r)$
- $B(\tau_l) \cap B(\tau_r) = \emptyset$
- $\beta(\overline{c}) = \beta(\tau_l) + \beta(\tau_r)$
- $\beta(\tau_l), \beta(\tau_r) \geq 1$
- $t_\tau(v) = t_{\tau_l}(v) + 1$
- $t_\tau(v) = t_{\tau_r}(v) + 1$

Rechne:

$$\begin{aligned}\overline{H}(\tau) &= \frac{1}{\beta(\tau)} \sum_{v \in B(\tau)} t_\tau(v) \\ &= \dots \\ &\geq 1 + \frac{1}{b} \min_{x \in [0,b]} (x \log_2(x) + (b-x) \log_2(b-x))\end{aligned}$$

Da

$$\begin{aligned}f'(x) &= \log_2(x) - \log_2(b-x) \\ &= \log_2(\frac{x}{b-x}) \qquad\qquad = 0\end{aligned}$$

genau dann, wenn $x = \frac{b}{2}$, ist dies die einzige Möglichkeit, wo das Minimum in $(0, b)$ angenommen werden kann. Die anderen Möglichkeiten sind die Intervallrandpunkte, was ausgeschlossen ist. Also gilt:

$$\begin{aligned}\overline{H}(\tau) &\geq 1 + \log_2(\frac{b}{2}) \\ &= 1 + \log_2(\beta(\tau)) - 1 \\ &= \log_2(\beta(\tau))\end{aligned}$$

□

Graphen

15. Grundlagen

15.1 **Definition** (Graph). Ein Graph ist ein Paar $G = (V, E)$, bestehend aus endlichen Mengen V von Knoten (verticles) und E von Kanten (edges). Die Kanten beschreiben die Verbindungen zwischen den Knoten und sind gerichtet oder ungerichtet.

- Für *gerichtete* Graphen oder *Digraphen* gilt:

$$E \subset \{(v, w) \in V \times V | v \neq w\}$$

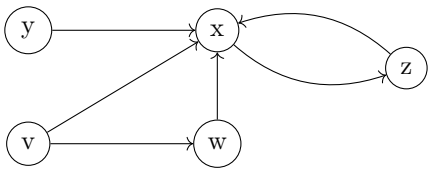
  Ist $e = (v, w) \in E$, so heißt v der Anfangsknoten und w der Endknoten.

- Für *ungerichtete* Graphen ist E eine Menge von ungeordneten Paaren $\{v, w\} \subset E, v \neq w$, das heißt:

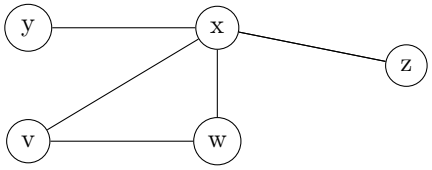
$$E \subset \{X \subset V || X| = 2\}$$

15.2 **Beispiel.** Sei $V = \{v, w, x, y, z\}$

- $G = (V, E), E = \{(v, w), (v, x), (w, x), (x, z), (z, x), (y, x)\}$ ist ein Digraph, der wie folgt illustriert werden kann:



- Der gleiche Graph sieht ungerichtet wie folgt aus:



  Wir bemerken insbesondere, dass $\{v, w\} = \{w, v\}$ im ungerichteten Graphen ist.

15.3 **Bemerkung.** In Digraphen werden manchmal auch Kanten der Form $\{v, v\}, v \in V$ zugelassen.

15.4 Definition. Sei $G = (V, E)$ ein Graph. Ein *Weg* oder *Pfad* von $v = v_0$ nach $w = v_r$ in G ist eine Kantenfolge

$$\pi = v_0, v_1, \ldots, v_r$$

mit $r \geq 1$ und

- $(v_i, v_{i+1}) \in E, i = 0, \ldots, r-1$im Falle von Digraphen.
- $\{v_i, v_{i+1}\} \in E, i = 0, \ldots, r-1$ im Falle von ungerichteten Graphen.

Der Weg heißt *einfach*, falls

- $v_0, \ldots, v_r$ paarweise verschieden sind oder
- $v_0, \ldots, v_r$ paarweise verschieden mit $v_0 = v_r$.

Die *Länge* von π ist gegeben als $|\pi| = r$, ist also die Anzahl Knoten, die in π durchlaufen werden. Ein Knoten w heißt von Knoten v *erreichbar*, falls ein Weg von v nach w existiert.

15.5 Beispiel. Bezogen auf Beispiel 15.2 ist:

- $\pi_1 = v, w$ ein einfacher Weg der Länge 1
- $\pi_2 = v, w, x$ ein einfacher Weg der Länge 2
- $\pi_3 = v, w, x, z, x, z$ ein Weg der Länge 5

Im ungerichteten Fall ist

- $\pi_4 = v, w, x, v$ ein einfacher Weg der Länge 3.

In beiden Fällen ist z von v erreichbar, im ungerichteten Fall gilt dies auch umgekehrt. Im gerichteten Fall ist v nicht von z erreichbar.

15.6 Definition. Sei $G = (V, E)$ ein Digraph und $v \in V$. Wir definieren:

- die Menge der (direkten) *Nachfahren* von v:

$$post(v) = \{w \in V | (v, w) \in E\}$$

- die Menge der (direkten) *Vorfahren* von v:

$$pre(v) = \{w \in V | v \in post(w)\}$$

- die Menge der *erreichbaren Knoten* von v:

$$post^*(v) = \{w \in V | \text{ es gib einen Weg von v nach w}\}$$

- die Menge aller Knoten die v erreichen können:

$$pre^*(v) = \{w \in V | v \in post^*(w)\}$$

- die Nachbarn als die Menge $post(v) \cup pre(v)$ von v
- und den Knotengrad $deg(v)$, welcher der Anzahl seiner Kanten entspricht.

15.7 Bemerkung. Mit den offensichtlichen Modifikationen kann die Definition 15.6 auch auf ungerichtete Graphen angewendet werden. Wir erhalten in diesem Fall für $v \in V$, dass

$$post(v) = pre(v)$$
$$post^*(v) = pre^*(v)$$

15.8 Beispiel. Für den Digraphen aus Beispiel 15.2 gilt:

- $post(v) = \{w, x\}$
- $post^*(v) = \{w, x, z\}$
- $pre(v) = \emptyset$
- $pre^*(v) = \emptyset$
- $deg(v) = 2$

Für den ungerichteten Graphen gilt geändert:

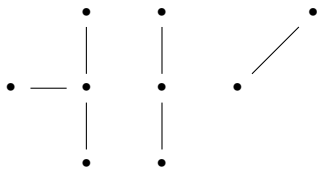
- $pre(v) = \{w, x\}$
- $pre^*(v) = \{w, x, y, z\}$
- $post^* = \{w, x, y, z\}$

16. Zusammenhang

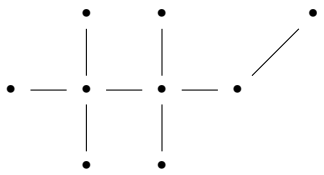
16.9 Definition. Sei $G = (V, E)$ ein ungerichteter Graph $\emptyset \neq C \subset V$

- Die Menge C heißt *zusammenhängend*, falls je zwei Knoten $v, w \in C, v \neq w$, voneinander erreichbar sind, das heißt, $v \in post^*(w), w \in post^*(v)$.
  Ist G ein Digraph, so heißt C zusammenhängend, falls C im zugrundeliegenden ungerichteten Graphen zusammenhängend ist.
- Eine zusammenhängende Knotenmenge heißt *Zusammenhangskomponente*, falls sie *maximal* ist. Das bedeutet, es gibt keine weitere zusammenhängende Knotenmenge $D \subset V$ mit $C \subsetneq V$.
- Ein Graph heißt zusammenhängend, falls V zusammenhängend ist.

16.10 Beispiel. Dies ist ein unzusammenhängender Graph mit drei Zusammenhangskomponenten:



Analog hierzu ist der folgende Graph zusammenhängend:



Wir können beobachten, dass die Zusammenhangskomponenten eines ungerichteten Graphen die Äquivalenzklassen der Knotenmenge V unter der Äquivalenzrelation

$$v = w \iff \{v\} \cup post^*(v) = \{w\} post^*(w)$$

ist.
Insbesondere zerfällt G in paarweise disjunkte Zusammenhangskomponenten $C_1, \ldots, C_r$ mit

$$V = \bigcup_{i=1}^{r} C_i$$
$$E = \bigcup_{i=1}^{r} E_i$$

wobei $E_i = E \cap \{X \subset C_i : |X| = 2\}$.

16.11 Satz. Sei $G = (V, E)$ ein ungerichteter Graph mit $n = |V| \geq 1$ Knoten und $m = |E|$ Kanten. Ist G zusammenhängend, so folgt für die Anzahl Knoten und Kanten, dass

$$m \geq n - 1$$

Beweis. □

Per vollständiger Induktion:

◊ IV $\underline{n=1}$
$m = 0 = n - 1$

◊ IS $\underline{n \geq 3}$
Wähle $v \in V$ mit:

$$deg(v) = \min_{v \in V} deg(v) =: k$$

Da G zusammenhängend ist, muss $k > 0$ gelten.
$\underline{k \geq 2}$

$$\begin{aligned} 2m &= 2|E| \\ &= \sum_{w \in V} |degw \\ &\geq 2|V| \\ &= 2n \qquad\qquad \Longrightarrow\ m \geq n \geq n-1 \end{aligned}$$

$\underline{k = 1}$
Sei $G'(V', E')$ derjenige Graph, der durch das Streichen von v und der zugehörigen Kante entsteht. G' ist zusammenhängend, weil G zusammenhängend ist. Die Induktionsannahme impliziert:

$$m - 1 = |E'| \geq |V'| - 1 = (n-1) - 1 = n - 1 \implies m \geq n - 1$$

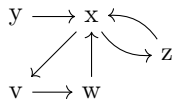
17. Zyklische Graphen

17.12 Definition. Ein *einfacher Zyklus* oder ein *einfacher Kreis* in einem Graphen $G = (V, E)$ ist ein einfacher Weg $\pi = v_0, \ldots, v_r$ mit $v_0 = v_r$ und $r \geq 2$ (im Falle von Digraphen) oder $r \geq 3$ (im Falle von ungerichteten Graphen). Ein *Zyklus* oder *Kreis* ist ein Weg, ,der sich aus einfachen Zyklen zusammensetzt.

17.13 Bemerkung. Aus der Definition 17.12 folgt:

- Jeder einfache Zyklus ist ein Zyklus
- Sind $\pi_1 = v_0, \ldots, v_r$, $\pi_2 = w_0, \ldots, w_s$ mit $v_i = w_0 = w_r$ so ist auch $\pi = v_0, \ldots, v_{i-1}, w_0, \ldots, w_s, v_{i+1}, \ldots, v_r$ ein Zyklus
- Es gibt keine andere Möglichkeit um Zyklen zu generieren.

17.14 Beispiel. Der Digraph



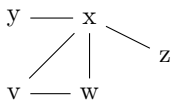
besitzt die einfachen Zyklen:

$$\pi_1 = x, z, x$$
$$\pi_2 = v, w, x, w$$

und die nicht einfachen Zyklen:

$$\pi_3 = x, z, x, z, x$$
$$\pi_4 = v, w, x, z, x, v$$

Der dazugehörige ungerichtete Graph:



besitzt den einfachen Zyklus

$$\pi_1 = v, w, x, v$$

Die Wege

$$\pi_2 = x, y, x$$
$$\pi_3 = v, w, x, y, x, v$$

sind keine Zyklen.

17.1. Eulergraphen

Im Folgenden betrachten wir exemplarisch das Königsberger Brückenproblem.

17.15 Das Königsbergerbrückenproblem ist ein von Leonhard Euler gelöstes mathematisches Problem. Es handelt konkret um die Stadt Königsberg und um die Frage, ob es einen Rundweg gibt, bei dem man alle sieben Brücken der Stadt über den Pregel exakt einmal überqueren kann und wieder zum Ausgangspunkt gelangt.
Euler zeigte, dass es keinen solchen Rundweg gibt.

17.16 **Definition.** Ein *Eulerscher Rundweg* in einem Graphen $G = (V, E)$ ist ein Zyklus, der jede Kante $e \in E$ genau einmal enthält. Im ungerichteten Fall nennen wir G *Eulersch*, falls der Grad jedes Knotens gerade ist.
Ein Digraph ist Eulersche, falls $indeg(v) = outdeg(v), v \in V$.

17.17 **Satz.** Ein zusammenhängender Graph $G = (V, E)$ besitzt genau dann einen Eulerschen Rundweg, falls der Graph Eulersch ist.

Beweis. Hin- und Rückrichtung

- $\underline{\Longrightarrow}$ Ein Knoten $v \in V$ in einem Eulerschen Graph, der k-mal in einem Eulerschen Rundweg vorkommt, muss im gerichteten Fall

  $$indeg(v) = outdeg(v)$$

  und im ungerichteten Fall

  $$deg(v) = 2k$$

  erfüllen.

- $\underline{\Longleftarrow}$ Sei G Eulersch und $\pi = v_0, \ldots, v_r$ der längste Weg, in dem jede Kante aus E höchstens einmal vorkommt. Dies bedeutet, dass jede ausgehende Kante von v_r im Weg enthalten sein muss (sonst wäre es nicht der längste Weg). Die Bedingung an den Knotengrad impliziert: $v_0 = v_r$.

Annahme:

- Digraphen: Es gibt eine Kante $e = (w_1, w_2) \in E$ mit $e \neq (v_i, v_{i+1}, i = 0, \ldots, r-1$
- ungerichtete Graphen: Es gibt eine Kante $e = \{w_1, w_2\} \in E$ mit $e \neq \{v_i, v_{i+1}\}, i = 0, \ldots, r-1$.

Fall 1: w_1 oder w_2 sind in π enthalten.
Ohne Beschränkung der Allgemeinheit: $v_r = w_1$

$$\Longrightarrow \tilde{\pi} = v_0, \ldots, v_r, w_2$$

Dies ist ein Widerspruch zur Annahme, da π nun nicht mehr der längste Weg ist.
Fall 2: w_1 und w_2 sind beide nicht in π enthalten. Da G zusammenhängend ist, gibt es einen Weg von v_0 nach w_1. Entlang dieses Weges muss es eine Kante geben, die einen Knoten in π und einen nicht in π hat.
Analog zum ersten Fall führt dies zum Widerspruch.

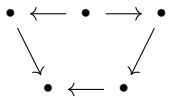
□

Wir beachten: Satz 17.17 ist die Antwort auf das Königsberger Brückenproblem. Demnach gibt es keinen solchen Weg und es ist nicht mögliche einen Euler-Weg zu finden.

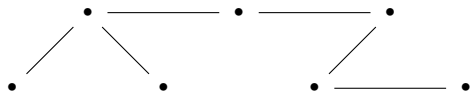
18. Bäume

18.18 Definition. Ein Graph G heißt *azyklisch* oder *zyklenfrei*, falls es keine Zyklen in G gibt. Ein ungerichteter, azyklischer und zusammenhängender Graph ist ein *Baum*

18.19 Beispiel. Ein azyklischer Graph kann wie folgt aussehen:



Sollte dieser ungerichtet sein könnte er so aussehen:



Dieser Graph ist sogar zusammenhängend.

18.20 Satz. Sei $G = (V, E)$ ein ungerichteter Graph mit $n = |V|$ knoten. Dann sind die folgenden Aussagen äquivalent.

a) G ist ein Baum

b) G hat n-1 Kanten und ist zusammenhängend

c) G hat n-1 Kanten und ist azyklisch

d) G ist azyklisch und das Hinzufügen einer beliebigen, noch nicht vorhandenen Kante erzeugt einen Zyklus.

e) G ist zusammenhängend und das Entfernen einer beliebigen Kante macht G unzusammenhängend.

f) Jedes Paar verschiedener Knoten in G ist durch genau einen einfachen Weg miteinander verbunden.

Beweis. Wir zeigen nicht jede Äquivalenz einzeln, sondern folgern aus anderen Aussagen.

- $\underline{a \implies f}$ Falls es für ein gegebenes Paar $v, w \in V$, verschiedene Wege gäbe, so würde die Vereinigung dieser beiden Wege einen Zyklus beinhalten (Widerspruch zu azyklisch in Baum).
- $\underline{f \implies e}$ "zusammenhängend" folgt per Definition. Falls wir eine beliebige Kante entfernen, so sind die beiden Endpunkte nicht mehr voneinander erreichbar ($\implies$ G wird unzusammenhängend).
- $\underline{e \implies d}$ G ist azyklisch, da wir sonst eine (ausgewählte) Kante entfernen könnten, so dass g immer noch zusammenhängend wäre. Seien $v, w \in V, w \neq v$, dann gibt es einen Weg von v nach w. Hinzufügen der Kante $\{v, w\}$ macht diesen Weg zum Zyklus.
- $\underline{d \implies c}$ Wir zeigen per Induktion über $m = |E|$, dass für einen azyklischen, ungerichteten Graph
$$n = m + p \tag{18.1}$$
gilt, wobei p die Anzahl Zusammenhangskomponenten ist.
  ◊ IV $\underline{m = 0}$ trivial

  ◊ IS $\underline{m \to m + 1}$ Beim Hinzufügen einer Kante muss eine Zusammenhangskomponente verschwinden, da sonst ein Zyklus entstehen würde.
  Da das Hinzufügen einer beliebigen neuen Kante einen Zyklus entstehen lässt, muss $p = 1$ gelten.
$$\implies |E| = m = n - 1$$
- $\underline{c \implies b}$ Folgt aus (18.1)
- $\underline{b \implies a}$ Zu zeigen: G ist azyklisch.
  Angenommen G hat k verschiedene einfache Zyklen. Dann können wir G durch entfernen von k Kanten zu einem azyklischen Graphen machen. (18.1) ist andwendbar, un impliziert, dass
$$\begin{aligned} n &= (m - k) + 1 \\ m + 1 &= (m - k) + 1 \qquad \implies k = 0 \end{aligned}$$
  d.h. G keine Zyklen.

□

18.21 Bemerkung. Fixieren wir in einem Baum $G = (V, E)$ einen Wurzelknoten, so wird automatisch eine Richtung in den Kanten festgelegt. Für $v \in V$ heißt $pre(v), |pre(v)| = 1$ der *Elternteil* und $post(v)$ die *Kinder*.

19. Implementierung von Graphen

19.22 Definition. Ein Graph $G = (V, E)$ mit $V = \{1, \dots, n\}$ kann durch einen *Adjazenzmatrix* oder *Nachbarschftsmatrix*

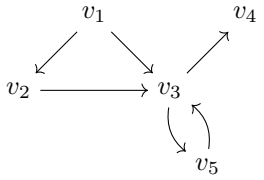
$$A = [a_{i,j}]_{i,j=1}^{n} \in \mathbb{R}^{n,n}$$

mit

$$a_{i,j} = \begin{cases} 1 & , \text{ falls } (i,j) \in E, \text{ bzw. falls } \{i,j\} \in E \\ 0 & , \text{ sonst} \end{cases}$$

dargestellt werden.

19.23 Beispiel. Der gerichtete Graph



besitzt die Adjazenzmatrix

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

und falls der Graph als ungerichteter Graph aufgefasst wird:

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

19.24 Bemerkung. Bei ungerichteten Graphen ist die Adjazenzmatrix immer symmetrisch. Der Speicherbedarf einer Adjazenzmatrix ist $|V|^2$, unabhängig von $|E|$. Für größere Graphen ist dies ineffizient. Wir werden beim Behandeln von dünnbesetzten Matrizen ein alternatives Speicherformat kennenlernen.
Zum Abschluss bemerken wir, dass sich viele graphentheoretische Konzepte mir Hilfe der Adjazenzmatrix in die Sprach der linearen Algebra übersetzten lässt.

Algorithmen auf Graphen

20. Graphendurchmusterung

Graphen müssen häufig durchmustert, d.h systematisch durchsucht, werden. Im Folgenden werden wir die "Tiefensuche" und die "Breitensuche" betrachten, welche sich beide auf den folgenden Algorithmus zurückführen lassen:

Daten: Graph $G = (V, E)$, Startknoten $s \in V$
Ergebnis: Zusammenhängender, azyklischer Graph $G'(R, T)$ mit $R = \{s\} \cup post^*(s)$ und $T \subset E$

- $R = \{s\}$
- $Q = \{s\}$
- $T = \emptyset$
- (a) **wenn** $Q = \emptyset$ **dann**
  | stop
  Ende
  sonst
  | Wähle $v \in Q \subset R$
  Ende
- Wähle $w \in V \setminus R \cap post(v)$ **wenn** *kein solches w exisitert* **dann**
  | setze $Q = Q \setminus \{v\}$ und gehe zu (a)
  Ende
- $R = R \cup \{w\}$
- $Q = Q \cup \{w\}$
- $T = T \cup \{e\}$ mit $e = (v, w)$ oder $e = \{v, w\}$
- Gehe zu (a)

Algorithmus 12: Algorithmische Suche

Je nachdem wie $v \in Q$ gewählt wird, bezeichnen wir den Suchalgorithmus unterschiedlich:

20.1 Definition. Bei der *Tiefensuche* oder *Depth-First-Search* (DFS) wird derjenige Knoten in $v \in Q$ ausgewählt, der zuletzt zu Q hinzugefügt wurde. Bei der *Breitensuche* oder *Breadth-First-Search* (BSF) wird derjenige Knoten $v \in Q$ ausgewählt, der zuerst zu Q hinzugefügt wurde.

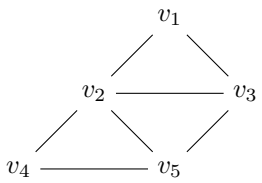
20.2 Satz. Algorithmus 12 liefert einen zusammenhängenden, azyklischen Graphen $G'(R, T)$, mit $R = \{s\} \cup post^*(s)$ und $T \subset E$.

Beweis. Per Konstruktion ist (R, T) zu jedem Zeitpunkt im Algorithmus zusammenhängend.
(R, T) ist außerdem zu jedem Zeitpunkt azyklisch, denn per Konstruktion gilt $Q \subset R$ und e verbindet jeweils $v \in Q \subset R$ mit $w \in V \backslash R$. Wir müssen also zeigen: $R = \{s\} \cup post^*(v)$ ($T \subset E$ ist klar).
Annahme: Nach Ende des Algorithmus gibt es $w \in V \setminus R$ von s erreichbar. Dann gibt es einen Weg

$$\pi = s, v_1, \ldots, v_r, w$$

und darin eine Kante $e = (x, y) \in E$ bzw. $e > \{x, y\} \in E$, mit $x \in R$ und $y \notin R$, $x, y \in \{s, v_1, \ldots, v_r, w\}$. Da $x \in R$, muss zu einem gewissen Zeitpunkt im Algorithmus auch $x \in Q$ gelten. Der Algorithmus terminiert, aber nur, falls x aus Q entfernt wurde, was nur geschieht, wenn $y \notin V \setminus R \cap post(x)$, also falls $e \notin E$ ist.
Dies ist ein Widerspruch zu unserer Annahme. □

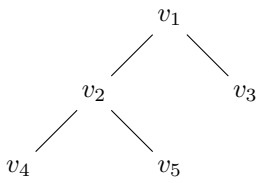
20.3 Beispiel. Betrachte



$s = \{v_1\}$. Bei der Tiefensuche entsteht folgender Baum:

$$v_1 \text{ — } v_2 \text{ — } v_4 \text{ — } v_5 \text{ — } v_3$$

Die Breitensuche hingegen ergibt:



20.4 Bemerkung. Ist G ein ungerichteter Graph, so ist das Resultat G' von Algorithmus 12 ein Baum mit Wurzel s, der *Spannbaum*

20.5 Satz. Die Ausführung von "wähle $v \in Q$" und "wähle $w \in V \setminus R \cap post(v)$" sei in $\mathcal{O}(1)$ durchführbar. Dann besitzt Algorithmus 12 den linearen Aufwand $\mathcal{O}(|V| + |E|)$.

Beweis. "wähle $v \in Q$" wird maximal $(|post(v)| + 1)$ mal aufgerufen.
"wähle $w \in V \setminus R \cap post(v)$" wird für jede Kante maximal ein mal aufgerufen.
Daraus folgt direkt die Behauptung. □

Wir hatten gezeigt, (Satz 16.11), dass $|V| - 1 \leq |E|$, d.h, es gilt $\mathcal{O}(|V| + |E|)$ kann nach oben beschränkt werden durch $\mathcal{O}(2|E|)$.

Beobachtung: Die Breitensuche beschreibt einen "kürzesten" Weg von Startknoten zu jedem anderen Knoten.

20.6 Satz. Sie $G = (V, E)$ ein ungerichteter Graph, $s, v \in V$ und
$dist_G(s, v) = \min\{|\pi| : \pi = s, u_1, \ldots, u_r, v$ ein Weg in G$\}$ mit $dist_G(s, v) = \infty$, falls es keinen Weg von s nach v gibt. Dann enthält der durch die Breitensuche erzeugte Graph $G' = (R, T)$ zum Startknoten $s \in V$ einen kürzesten Weg zu jedem Knoten $v \in post^*(s)$.Dies bedeutet, es gib einen einfachen Weg $\pi = s, u_1, \ldots, u_r, v$ in G' mit $|\pi| = dist_{G'}(s, v) = dist_G(s, v)$ minimal.

Beweis. Wir bemerken zuerst, dass G' azyklisch ist und somit π in G' eindeutig bestimmt ist. Modifiziere Algorithmus 12 nun wie folgt:

- In 1: l(s)=0
- In 4: l(w) = l(v)+1

Dies bedeutet, zu jedem Zeitpunkt im Algorithmus gilt:

$$l(v) = dist_{(R,T)}(s, v), v \in R$$

Insbesondere gibt es für kein in 2 ausgewähltes $v \in Q$ ein $w \in R$ mit:

$$l(w) > l(v) + 1 \tag{20.1}$$

Zu Zeigen: $dist_{R,T}(s, v) = dist_{G'}(s, v) = dist_G(s, v)$
Annahme: Nach Abbruch des Algorithmus gibt es ein $w \in V$ mit

$$dist_G(s, w) < dist_{G'}(s, w) \tag{20.2}$$

Falls es mehr als ein solches w gibt , wähle dasjenige mit den kleinsten Abstand $dist_G(s, w)$.
Sei $\pi = s, u_1, \ldots, u_r, v, w \implies dist_G(s, v) = dist_{G'}(s, v)$, da sonst v ein Knoten mit kleineren Abstand wäre, der (20.2) erfüllt.

$$\begin{aligned} \implies l(w) &= dist_{G'}(s, w) \\ &> dist_G(s, w) \\ &= dist_G(s, v) + 1 \\ &= dist_{G'}(s, v) + 1 \\ &= l(v) + 1 \end{aligned}$$

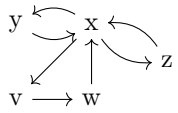
Dies ist ein Widerspruch zu (20.1). □

20.7 **Bemerkung.** Die Breitensuche berechnet also die Wege aller Knoten zur Wurzel in $\mathcal{O}(|V| + |E|)$.

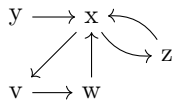
21. Starker Zusammenhang

21.8 **Definition.** Ein Digraph heißt *stark zusammenhängend*, falls für jedes Knotenpaar $v, w \in V$, $v \neq w$, sowohl $v \in post^*(w)$, als auch $w \in post^*(w)$ gilt. Es gibt also einen Weg von v nach w und umgekehrt.
Die *starken Zusammenhangskomponenten* sind die maximalen stark zusammenhängenden Teilgraphen.

21.9 **Beispiel.** Der Graph:



ist stark zusammenhängend.



Dieser Graph, wo der Weg von x nach y entfernt wurde, ist nicht stark zusammenhängend. Die starken Zusammenhangskomponenten sind y und der restliche Graph G'.

Im folgenden konstruieren wir einen Algorithmus zur Bestimmung solcher starker Zusammenhangskomponenten.

Daten: Digraph $G = (V, E)$
Ergebnis: Abbildung $comp\colon V \to \mathbb{N}$, welche die Zugehörigkeit zu einer starken Zusammenhangskomponente signalisiert.

- $R = \emptyset$
- $N = 0$
- **für** $v \in V$ **tue**
  wenn $v \notin R$ **dann**
  FirstVisit(v)
  Ende
  Ende
- $R = \emptyset$
- $K = 0$
- **für** $j \leftarrow |V|$ **bis** *1* **tue**
  wenn $\psi^{-1}(j) \notin R$ **dann**
  $K = K + 1$
  SecondVisit($\psi^{-1}(j)$)
  Ende
  Ende

Function `FirstVisit(`v`)`

- $R = R \cup \{v\}$
- **für** $w \in V \setminus R$, $(v, w) \in E$ **tue**
  FirstVisit(w)
  Ende
- $N = N + 1$
- $\psi(v) = N$
- $\psi^{-1}(N) = V$

zurück
Function `SecondVisit(`v`)`

- $R = R \cup \{w\}$
- **für** $w \in V \setminus R$, $(w, v) \in E$ **tue**
  SecondVisit(w)
  Ende
- $comp(v) = K$

zurück

Algorithmus 13: Bestimmung starker Zusammenhangskomponenten

21.10 Beispiel. Wir betrachten den Graph:

Der Startknoten v_1 für FirstVisit ergibt die Besuchsreihenfolge: v_1, v_7, v_2, v_4, v_5. Die Tiefensuche mittels FirstVisit bricht für v_3 und v_6 direkt ab. Wir erhalten:

$$\begin{array}{lll} \psi(v_2) = 1 & \psi(v_7) = 4 & \psi(v_6) = 7 \\ \psi(v_5) = 2 & \psi(v_1) = 5 & \\ \psi(v_4) = 3 & \psi(v_3) = 6 & \end{array}$$

SecondVisit für $v_6 = \psi^{-1}(7)$ bricht sofort ab $\implies$ $comp(v_6) = 1$
SecondVisit für $v_3 = \psi^{-1}(6)$ bricht sofort ab $\implies$ $comp(v_3) = 2$
SecondVisit für $v_1 = \psi^{-1}(5)$ ergibt die Besuchsreihenfolge: $v_2, v_7 \implies comp(v_1) = 3, v \in \{v_1, v_2, v_7\}$
SecondVisit für $v_4 = \psi^{-1}(3)$ ergibt die Besuchsreihenfolge: $v_5 \implies comp(v_4) = 4$.
Demnach ergibt sich:

$$\{v_6\}, \{v_3\}, \{v_1, v_2, v_7\}, \{v_4, v_5\}$$

21.11 Satz. Algorithmus 13 identifiziert die starken Zusammenhangskomponenten eines Digraphen $G = (V, E)$ in linearem Aufwand $\mathcal{O}(|V| + |E|)$

Beweis. Aufwand: Analog zur Tiefensuche (Satz 20.5)

- Gleiche starke Zusammenhangskomponente *implies* gleicher comp-Wert
  Seien $v, w \in V$ Knoten in der gleichen , starken Zusammenhangskomponente
  - $\implies$ Es gibt einen Weg von v nach w und umgekehrt.
  - $\implies$ SecondVisit weist $comp(v) = comp(w)$ zu
- Gleicher comp-Wert $\implies$ Gleiche starke Zusammenhangskomponente
  Seien $v, w \in V$ mit $comp(v) = comp(w)$. Definiere $r(v)$ ist derjenige von v erreichbare Knoten, der den höchsten ψ-Wert hat.
  Beobachte:
  - $comp(v) = comp(w) \implies v$ und w sind im gleichen, von SecondVisist erzeugten Subbaum. $\implies r(v) = r(w) =: r$ (d.h r ist von v, w erreichbar)
  - $\psi(r) > \psi(v) \implies v$ wurde in FirstVisit vor r nummeriert. $\implies$ Der von FirstVisit erzeugte Baum hat einen Weg von r nach v , d.h v ist von r erreichbar.
  - Analog: w ist von r erreichbar. $\implies v$ ist über r von w aus erreichbar und umgekehrt.

□

21.12 Definition. Sei $G = (V, E)$ ein Digraph. Zieht man die starken Zusammenhangskomponenten zu einem Knoten zusammen, entsteht ein *kondensierter Graph.*
Eine *Nummerierung* $v = \{v_1, \ldots, v_n\}$ der Knoten heißt *topologische Ordnung*, falls $(v_i, v_j) \in E$ nur, wenn $i < j$.

21.13 Lemma. Der Digraph $G = (V, E)$ besitzt eine topologische Ordnung genau dann, wenn er azyklisch ist.

Beweis. Übung □

21.14 Satz. Zu jedem Digraphen $G = (V, E)$ bestimmt Algorithmus 13 eine topologische Ordnung in linearem Aufwand $\mathcal{O}(|V| + |E|)$. Gibt es eine solche Ordnung nicht, so sagt uns das der Algorithmus ebenfalls in linearem Aufwand.

Beweis. Seien $V_i, V_j \subset V$ die (disjunkten) Knotenmengen zweier stabiler Zusammenhangskomponenten mit $comp(v_i) = i, comp(v_j) = j$ für $v_i \in V_i, v_j \in V_j$.
Ohne Beschränkung der Allgemeinheit: $i < j$
Annahme: Es gibt eine Kante $e = (v_j, v_i)$ mit $v_j \in V_j, v_i \in V_i$. *Beobacht:* In SecondVisit werden alle Knoten in V_i vor derjenigen in V_j zu R hinzugefügt. $\Longrightarrow$

- Beim Überprüfen von $e = (v_j, v_i)$ in SecondVisit gilt: $v_i \in R, v_j \notin R$
- v_j wird zu R hinzugefügt, bevor K erhöht wird.
- $comp(v_i) = comp(v_j)$

Dies ist ein Widerspruch.
$\Longrightarrow$ Algorithmus 13 erzeugt eine topologische Ordnung, falls diese existiert.
Aus Lemma 21.13 folgt:
Es gibt genau dann eine topologische Ordnung auf G, falls G azyklisch ist.
$\Longleftrightarrow$ alle starken Zusammenhangskomponenten sind eindeutig.
$\Longleftrightarrow$ Am Ende des Algorithmus ist $K = |V|$ □

21.15 Bemerkung. Der Beweis zeigt, dass Algorithmus 13 auch benutzt werden kann, um in linearer Zeit zu überprüfen, ob ein Graph Zyklen hat.

22. Kürzeste Wege Probleme

22.16 Definition (gewichteter Graph). Sei $G = (V, E)$ ein Graph. Eine *Gewichtsfunktion* für die Kanten von G ist eine Abbildung $w\colon E \to R$. Ist $\pi = v_0, v_1, \ldots, v_r$ ein Weg, so heißt

$$w(\pi) = \sum_{i=1}^{r-1} w(v_i, v_{i+1})$$

die *Weglänge* von π bezüglich w. Das Tripple $G = (V, E, w)$ heißt *gewichteter Graph.*

22.17 Definition. Sei $G = (V, E, w)$ ein gewichteter Graph und $v, \tilde{w} \in V$. Ein *kürzester Weg* von v nach $\tilde{w}$ in G bezüglich w ist ein v-$\tilde{w}$-Weg π mit der Eigenschaft $w(\pi) \leq w(\pi')$ für alle anderen v-w-Wege π' . Die kürzeste Weglänge $\delta(v, \tilde{w})$ von v nach $\tilde{w}$ ist definiert durch:

$$\delta(v, \tilde{w}) \begin{cases} \min\{w(\pi) | \pi \text{ ist } v\text{-}\tilde{w}\text{-Weg } \} & \text{, falls ein solcher Weg existiert} \\ \infty & \text{, sonst} \end{cases}$$

22.18 Beispiel. Betrachte den gewichteten Graph $G = (V, E, w)$ mit

Mögliche Wege von v_3 nach v_1 sind:

$$\begin{aligned} \pi_1 &= v_3, v_1 && \Longrightarrow\ w(\pi_1) = 6 \\ \pi_2 &= v_3, v_2, v_1 && \Longrightarrow\ w(\pi_2) = 5 \\ \pi_3 &= v_3, v_4, v_2, v_1 && \Longrightarrow\ w(\pi_3) = 9 \end{aligned}$$

22.19 Notation. Wir werden im Folgenden nur Digraphen behandeln, da sich ungerichtete Graphen immer auch als gerichtete Graphen interpretieren lassen. Sei $G = (V, E, w)$ ein gewichteter Digraph.

Wir unterscheiden drei verschiedene Varianten des *kürzeste-Wege-Problems:*

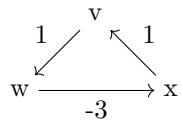
a) Einzelpaar-kürzeste-Wege-Problem
Gegeben $v, u \in V$, suche einen kürzesten Weg von v nach u.

b) Einzelquelle-kürzeste-Wege-Problem
Gegeben $v \in V$, suche einen kürzesten Weg von v zu allen $u \in post^*(v)$.

c) Alle-Paare-kürzeste-Wege-Problem
Finde alle Paare $v, u \in V$ einen kürzesten v-u-Weg

Wir bemerken, dass das Problem c die Probleme a und b löst, daher betrachten wir im Folgenden nur Problem c.
Negative Gewichte sind zwar nach Definition 22.17 zugelassen, können aber Probleme bereiten:

22.20 Beispiel. Betrachte:



In diesem Graphen gibt es keinen kürzesten Weg.

$$\begin{aligned} w(v, w, x) &= -2 \\ w(v, w, x, v, w, x) &= -3 \end{aligned}$$

Man kann den Weg also immer weiter verlängern und der Weg wird immer kürzer.

22.21 Lemma. Sei $G = (V, E, w)$ ein gewichteter Digraph. Falls es in G keine Zyklen mit negativer Weglänge gibt, dann gibt es für $v, u \in V, u \in post^*(v)$ einen kürzesten Weg π mit:

$$\delta(v, w) = w(\pi) > -\infty$$

Beweis. Da es keine negativen Zyklen gibt, genügt es alle einfachen Wege von v nach u zu betrachten. Da $|V|$ und $|E|$ endlich sind, sind dies endlich viele, so dass π durch einen Minimierer gegeben ist. □

22.22 Lemma. Sei $G = (V, E, w)$ ein gewichteter Digraph ohne negativen Zyklen. Ist $\pi = v_0, \ldots, v_r$ ein kürzester Weg von v_0 nach v_r dann ist der Teilweg

$$\pi_{i,j} = v_i, \ldots, v_j, 0 \leq i < j \leq r$$

ein kürzester Weg von v_i nach v_j.

Beweis. Angenommen: $\pi'_{i,j}$ ist ein kürzester Weg von v_i nach v_j mit $w(\pi'_{i,j} < w(\pi_{i,j})$

$$\begin{aligned} \implies \pi' &= \pi_{0,i}\pi'_{i,j}\pi_{j,r} \text{ erfüllt:} \\ w(\pi') &= w(\pi_{0,i}) + w(\pi'_{i,j}) + w(\pi_{j,r}) \\ &< w(\pi_{0,i}) + w(\pi_{i,j}) + w(\pi_{j,r}) \\ &= w(\pi') \\ &\implies w(\pi') < w(\pi) \end{aligned}$$

Dies ist ein Widerspruch, da π kürzester Weg nach Annahme ist. □

22.23 Lemma. Sei $G = (V, E, w)$ ein gewichteter Digraph ohne negative Zyklen und sei $\pi = v_0, v_1, \ldots, v_r$ ein kürzester Weg von v_0 nach v_r . Dann gilt:

$$\delta(v_0, v_r) = \delta(v_0, v_{r-1} + w(v_{r-1}, v_r)$$

Beweis. Aus Lemma 22.22 folgt:

- $\pi' = v_0, \ldots, v_{r-1}$ ist ein kürzester Weg von v_0 nach v_{r-1}
- $\delta(v_0, v_{r-1} = w(\pi')$
- $\delta(v_0, v_r) = w(\pi)$

$w(\pi)$ lässt sich umschreiben:

$$\begin{aligned} w(\pi) &= w(\pi') + w(v_{r-1}, v_r) \\ &= \delta(v_0, v_{r-1} + w(v_{r-1}, v_r) \end{aligned}$$

□

Daten: Gewichteter Graph $G = (V, E, w)$ mit nicht negativen Gewichten
Ergebnis: Kürzeste Wege von s nach $v \in post^*(s)$ samt Weglänge $l(v) = \delta(s, v)$

- $l(s) = 0$
  $l(v) = \infty$, $v \in V \setminus \{s\}$
  $R = \emptyset$
- Finde $u \in V \setminus R$ mit $l(n) = \min_{v \in V \setminus R} l(v)$
- $R = R \cup \{u\}$
- **für** $v \in V \setminus R$ *mit* $(u, v) \in E$ **tue**
  wenn $l(v) > l(u) + w(u, v)$ **dann**
  $l(v) = l(u) + w(u, v)$
  $p(v) = u$
  Ende
  Ende
- $R \neq V$ gehe zu 2.

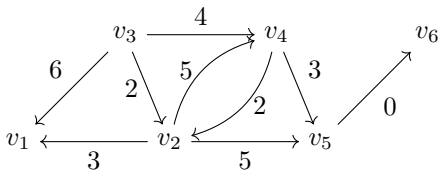
Algorithmus 14: Dijkstra Algorithmus

22.24 Bemerkung. Die kürzesten Wege im Output des Algorithmus sind durch die Abbildung $p\colon V \to V$ gegeben, denn

$$\pi = s, \ldots, p(p(p(v))), p(p(v)), p(v), v$$

ist ein kürzester s-v-Weg.

22.25 Beispiel. Der Graph



mit Startknoten v_3 hat nach Ausführung von Algorithmus 14 folgende Ausgabe:

Tabelle 1.: Dijkstra

Iteration	$l(v_1), l(v_2), \ldots, l(v_6)$						u, $l(u)$, $p(u)$		
0	∞	∞	0	∞	∞	∞	v_3	0	$-$
1	6	2		4	∞	∞	v_2	2	v_3
2	5				7	∞	v_4	4	v_3
3						∞	v_1	5	v_2
4						∞	v_5	7	v_2
5						7	v_6	7	v_5

22.26 **Satz.** Der Algorithmus von Dijkstra 14 arbeitet korrekt, mit einer Laufzeit von $\mathcal{O}(n^2), n = |V|$.

Beweis.

22.27 **Notation.** Schreibe in der k-ten Iteration $^{(k)}$ an alle Größen, die im Algorithmus vorkommen

Zeige: Bei jeder Ausführung von 2. gilt:

a) $l^{(k)}(v) \leq l^{(k)}(y)$ für alle $v \in R^{(k)}, y \in V \setminus R^{(k)}$

b) $l^{(k)}(v) = \delta(s, v)$ für alle $v \in R^{(k)}$
$l^{(k)}(v) < \infty, v \neq s \implies$ Es existiert ein kürzester s-v-Weg mit Knoten in $R^{(k)}$ und letzter Kante $(p(v), v)$

c) $v \in V \setminus R^{(k)} \implies l^{(k)}(v)$ ist die kürzeste Weglänge von s nach v im Teilgraphen mit Knoten $R^{(k)} \cup \{v\}$ in G
$v \in V \setminus R^{(k)}, l^{(k)}(v) < \infty \implies p^{(k)}(v) \in R^{(k)}$ und $l^{(k)}(v) = l^{(k)}(p(v)) + w(p^{(k)}(v), v)$

Per vollständiger Induktion:

$\Diamond$ IV $k = 0$ Trivialerweise erfüllt

$\Diamond$ IS $k \to k + 1$ Sei u der in 2. ausgewählte Knoten.

a) Für $v \in R^{(k)}, y \in V \setminus R^{(k)}$ gilt:

$$\begin{aligned} l^{(k+1)}(v) &= l^{(k)}(v) \\ &\leq l^{(k)}(u) \\ &= \min_{u \in V \setminus R^{(k)}} l^{(k)}(u) \\ &\leq l^{(k+1)}(y) \end{aligned}$$

Da $R^{(k+1)} = R^{(k)} \cup \{u\} \implies$ Behauptung a) für $k + 1$ gilt.

b) Wir müssen die Behauptung über die kürzeste Weglänge nur für u überprüfen.

c) für $k \implies l^{(k)}(u)$ ist die kürzeste Weglänge zum Weg π im Teilgraphen $R^{(k)} \cup \{u\}$.
zu Zeigen $l^{(k+1)}(u) = l^{(k)}(u) = w(\pi) = \delta(s,u)$ in G
Angenommen: Es gibt einen Weg π' in G mit mindestens einem Knoten $y \in V \setminus R^{(k)}$ und $w(\pi' < w(\pi)$
$\pi' = s, v_1, \ldots, v_{r-1}, y, v_{r+1}, \ldots, v_{r+m}, u$

$$\begin{aligned} \implies l^{(k)}(y) &\leq w(s, v_1, \ldots, v_{r-1}, y) \\ &\leq w(\pi') \\ &< w(\pi) \\ &= l^{(k)}(u) \end{aligned}$$

Dies ist ein Widerspruch zu $l^{(k)}(u) = \min_{v \in V \setminus R^{(k)}} l^{(k)}(v)$
$\implies$ b für $k+1$

c) Zeige, dass 3. und 4. die Aussage c) erhalten. Sei $v \in V \setminus R^{(k+1)}$ in 4.

$$\begin{aligned} \implies p^{(k+1)}(v) &= u \\ l^{(k+1)}(v) &= l^{(k+1)}(u) + w(u,v) \end{aligned}$$

c) für $k \implies$ Es gibt einen s-v-Weg im von $R^{(k+1)} \cup \{v\}$ aufgespannten Teilgraph $G'(v)$ in G , mit Länge $l^{(k+1)}(v) = l^{(k+1)}(u) + w(u,v)$
zu Zeigen Dieser Weg ist ein kürzester Weg in G
Angenommen: Für ein $v \in V \setminus R^{(k+1)}$ in 4. gibt es einen s-v-Weg π in $G'(v)$ mit $w(\pi) < l^{(k+1)}(v)$.
Beobachte: $l^{(k+1)}(v) \leq l^{(k)}(v)$
$\implies$ u muss π enthalten sein, da dies die einzige Veränderung zum k-ten Schritt ist und $l^{(k)}(v)$ nach c) im k-ten Schritt ein kürzester Weg in $R^{(k)} \cup \{u\}$ ist.
Sei x der Vorgänger von v in π.

$$\begin{aligned} l^{(k+1)}(v) &= l^{(k+1)}(x) + w(x,v) \\ &\leq l^{(k+1)}(u) + w(x,u) \\ &\leq w(\pi) \end{aligned}$$

Dies ist ein Widerspruch $\implies$ c) für $k+1$. Für $k = n$ impliziert b) die Behauptung.

Aufwand: $\mathcal{O}(n^2)$ ist trivial. □

22.28 Bemerkung. Clevere Implementierungen können den Dijkstra-Algorithmus 14 in $\mathcal{O}(m + n\log(n), n = |V|, m = |E|$ durchführen.

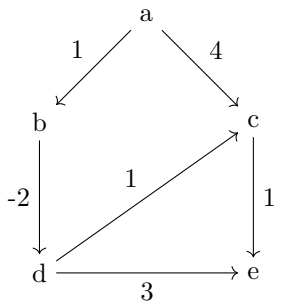
Für die Verallgemeinerung für negative Gewichte betrachten wir folgend den Moore-Bellmann-Ford Algorithmus:

Daten: Gewichteter Digraph $G = (V, E, w)$ ohne negative Zyklen, Startknoten $s \in V$
Ergebnis: kürzeste Wege von s nach v, $v \in post^*(s)$ samt Weglänge $l(v) = \delta(s, v)$

- $l(s) = 0$
  $l(v) = \infty, v \in V \setminus \{s\}$
- **für** $k = 1, ..., n-1$ **tue**
  für $e = (u, v) \in E$ **tue**
  wenn $l(v) > l(u) + w(u, v)$ **dann**
  $l(v) = l(u) + w(u, v)$
  $p(v) = u$
  Ende
  Ende
  Ende

Algorithmus 15: Moore-Bellmann-Ford

22.29 Beispiel. Betrachte den Graphen:



mit Startknoten a.

Tabelle 2.: Iteration 1

Kante	$l(a), \ldots, l(e)$					$p(b), \ldots, p(e)$			
(a,b)	0	1	∞	∞	∞	a	-	-	-
(a,c)			4	∞	∞		a	-	-
(b,d)			4	-1	∞		a		c
(c,e)			4		5		a		c
(d,c)			0		5		d		c
(d,e)					2				d

Tabelle 3.: Iteration 2

Kante	$l(a), \ldots, l(e)$					$p(b), \ldots, p(e)$			
(a,b)	0	1	0	-1	2	a	d	b	d
(a,c)									
(b,d)									
(c,e)									
(d,c)					1				c
(d,e)									

Keine Veränderung mehr ab der 3. Iteration.

22.30 **Satz.** Der Moore-Bellman-Ford Algorithmus 15 arbeitet korrekt mit Aufwand $\mathcal{O}(m \cdot n), m = |E|, n = |V|$.

Beweis. Bezeichne zu jedem Zeitpunkt:

$$R = \{v \in V | l(v) < \infty\}$$
$$F = \{(u,v) \in E | u = p(v)\}$$

Zeige zu erst:

a) $l(v) \geq l(u) + w(u,v), (u,v) \in E$

b) (R,F) ist ein azyklischer Graph

c) (R,F) ist ein gerichteter Baum mit Wurzel s, d.h jeder Knoten in R ist über genaue einen Weg von s aus erreichbar.

a) Bei $p(v) = u$ gilt:

$$l(v) = l(u) + w(u,v)$$

Danach wird $l(u)$ höchstens verkleinert.

b) Angenommen: Es gibt einen Zyklus $\pi = v_0, v_1, \ldots, v_{r-1}, v_r$ mit $v_0 = v_r$
Ohne Beschränkung der Allgemeinheit entstanden durch Setzen von $p(v_r) = v_{r-1}$
$\Longrightarrow$ Davor galt:

$$l(v_0) = l(v_r) > l(v_{r-1}) + w(v_{r-1}, v_r)$$

Wegen a) gilt:

$$l(v_{i+1}) \geq l(v_i) + w(v_i, v_{i+1}, i = 0, \ldots, r-2)$$

$$\begin{aligned} \Longrightarrow \sum_{i=1}^{r} w(v_{i-1}, v_i) &= \left(\sum_{i=1}^{r-1} w(v_{i-1}, v_i) \right) + w(v_{r-1}, v_r) \\ &< \sum_{i=1}^{r} (l(v_i) - l(v_{i-1}) \\ &= 0 \end{aligned}$$

Widerspruch dazu, dass es keine negativen Zyklen gibt.

c) $x \in R \setminus \{s\} \Longrightarrow p(x) = R$ Gehe von Blättern zu Wurzel dann folgt c)
$a, b, c \Longrightarrow l(y)$ ist mindestens die Länge des (eindeutigen) s-y-Wegs in (R, F)
Behauptung: Nach k Iterationen ist $l(y)$ auch die maximale Länge eines kürzesten s-y-Wegs in G mit maximal k Kanten.
Zeige durch Induktion:

$\Diamond$ IV $k = 1$ klar

$\Diamond$ $k - 1 \to k$ Sei

$$\pi = s, v_1, \ldots, v_r, x, y$$

ein kürzester s-y-Weg in G mit höchsten k Kanten und $r \leq k - 2$.
Dann folgt aus Lemma 22.23:

- $\pi' = s, v_1, \ldots, v_r, x$ ist ein kürzester s-k-Weg mit k-1 Kanten
- $l(x) \leq w(\pi')$
-

$$\begin{aligned} l(y) &\leq l(x) + w(x, y) \\ &\leq w(\pi') + w(x, y) \\ &= w(\pi) \end{aligned}$$

Wir haben gezeigt: $w(\pi) = \delta(s, y)$, wobei π der eindeutige Weg in (R, T) mit maximal k Kanten.
Daraus folgt die Behauptung.

Aufwand: Trivial. □

Die letzte Frage dieses Kapitels: Algorithmus für das Alle-Paare-kürzeste-Wege Problem lässt eine naive Antwort zu. Diese lautet: Wende Dijkstra oder Moore-Belmann-Ford auf jede Quelle an.Das Resultat ist jedoch, dass man keine brauchbaren Datenstrukturen mehr hat. Demnach gibt es folgenden Algorithmus:
Ohne Beschränkung der Allgemeinheit sei $V = \{1, \ldots, n\}$

Daten: Gewichteter Digraph $G = (V, E, w), V = \{1, \ldots, n\}$ ohne negative Zyklen
Ergebnis: Matrizen $[l_{i,j}]_{i,j=1}^{n}, [p_{i,j}]_{i,j=1}^{n}$

- $$\begin{aligned} l_{i,j} &= w(i,j) && ,(i,j) \in E \\ l_{i,j} &= \infty && ,\forall (i,j) \in (V \times V) \backslash E, i = j \\ l_{i,i} &= 0, i \in V \\ p_{i,j} &= i && ,i,j \in V \end{aligned}$$

- **für** $j = 1, ..., n$ **tue**
  - **für** $i = 1, ..., n, i \neq j$ **tue**
    - **für** $k = 1, ..., n, k \neq j$ **tue**
      - **wenn** $l_{i,k} > l_{i,j} + l_{j,k}$ **dann**
        $l_{i,k} = l_{i,j} + l_{j,k}$ $p_{i,k} = p_{j,k}$
      - **Ende**
    - **Ende**
  - **Ende**

Ende

Algorithmus 16: Floyd-Warshall

22.31 **Satz.** Der Algorithmus 16 arbeitet korrekt, mit Aufwand $\mathcal{O}(n^3), n = |V|$

Beweis.

22.32 **Notation.** Im Iterationsschritt j_0 ist $l_{ik}^{(j_0)}$ die Länge eines kürzesten i-k-Weges im von $\{1, \ldots, j_0\}$ aufgespannten Teilgraphen mit Endkante $(p_{ik}^{(j_0)}, k)$.

□

23. Netzwerkflussprobleme

23.33 **Beispiel.** Ausgehend von einer Bananenplantage s sollen alle geernteten Bananen zum Lagerhaus t transportiert werden. Für den Transport stehen Straßen mit $r_1, \ldots, r_p$ $\frac{kg}{n}$ Transportkapazität zu den Seehäfen $A_1, \ldots, A_p$ zu Verfügung. Von den Zielhäfen

$B_1, \ldots, B_q$ stehen Transportkapazitäten von $d_1, \ldots, d_q$ $\frac{kg}{n}$ zum Supermarkt t bereit. Die Transportkapazität zwischen den Seehäfen werden mit $c(A_i, A_j), 1 \leq i \leq p, 1 \leq j \leq p$ bezeichnet.

Fragen:

- Ist es mögliche, alle Transportkapazitäten auszuschöpfen?
- Falls nein, was ist die maximal mögliche Transportkapazität?
- Wie sollen die Bananenladungen aufgeteilt werden?

Konstruiere einen gewichteten Digraph $G = (V, E, w)$ mit

- $V = \{s, A_1, \ldots, A_p, B_1, \ldots, q, t\}$
- $E = \{(s, A_i), (A_i, B_j), (B_j, t); 1 \leq i \leq p, 1 \leq j \leq p\}$
- $w(e) = \begin{cases} r_i & , e = (s, A_i) \\ c(A_i, B_j) & , e(A_i, B_j) \\ d_i & , e(B_i, t) \end{cases}$

23.34 Definition. Ein *Netzwerk* ist ein Tupel $N = (V, E, c, s, t)$ bestehend aus

- einem gewichteten Digraphen $G = (V, E, c)$
- einer *Kapazitätsfunktion* $c\colon E \to R_{\geq 0}$
- einer Quelle $s \in V$ mit $pre(s) = \emptyset$
- einer Senke $t \in V$ mit $post(t) = \emptyset$

Ein Fluss $f\colon E \to R_{\geq 0}$ ist eine Funktion, die folgende Bedingungen erfüllt:

a) *Kapazitätsbedingung:*

$$f(v, w) \leq c(v, w)$$

b) *Kirchhoffsches Gesetz:*

$$\sum_{u \in pre(v)} f(u, v) = \sum_{w \in post(v)} f(v, w)$$

für alle $v \in V \setminus \{s, t\}$.

Der *Wert* des Flusses ist:

$$flow(f) = \sum_{w \in post(s)} f(s, w)$$

Der *maximale FLuss* von N wird bezeichnet als:

$$MaxFlow(N) = \max\{flow(f) \text{ ist Fluss für N}\}$$

Eine Flussfunktion f wird *optimal* genannt, falls

$$flow(f) = MaxFlow(N)$$

ist.
Ein *Schnitt* für N ist eine Knotenmenge $S \subset V$ mit $s \in S, t \notin S$.
Die *Kapazität* eines Schnittes ist gegeben durch:

$$cap(S) = \sum_{v \in S w \in post(v) \setminus S} c(v, w)$$

Die minimale Schnittkapazität von N ist:

$$MinCut(N) = \min\{cap(S) | \text{ S ist Schnitt für N}\}$$

23.35 Lemma. Sei S ein Schnitt für $N = (V, E, c, s, t)$. Dann gilt für jeden Fluss f, dass

$$flow(f) = \sum_{w \in post(v) \setminus S} f(v, w) - \sum_{u \in pre(v)} f(u, v)$$
$$flow(f) \leq cap(S)$$

Beweis. Rechne:

$$\begin{aligned} flow(f) &= \sum_{w \in post(s)} f(s, w) \\ &= \sum_{v \in S} \left(\sum_{w \in post(v)} f(v, w) - \sum_{u \in pre(v)} f(u, v) \right) \\ &= \sum_{w \in post(v)} f(v, w) - \sum_{u \in pre(v) \setminus S} f(u, v) \end{aligned}$$

Für die nächste Behauptung können wir ebenfalls nachrechnen:

$$\begin{aligned} flow(f) &= \sum_{w \in post(v) \setminus S} f(v, w) - \sum_{u \in pre(v) \setminus S} f(u, v) \\ &\leq \sum_{w \in post(v) \setminus S} c(v, w) \\ &= cap(S) \end{aligned}$$

□

23.36 Satz (Max-FLow-Min-Cut Theorem). Sei $N = (V, E, c, s, t)$ ein Netzwerk, dann gilt:

$$MaxFlow(N) = MinCut(N)$$

Beweis. Zu zeigen: Es gibt einen Schnitt für den die Gleichheit gilt.
Idee: Gegeben sei ein Fluss f mit flow(f)=MaxFlow(N), konsturiere einen Schnitt S mit flow(f)=cap(S).

Daten: Netzwerk N, Fluss f mit $flow(f) = MaxFlow(N)$.
Ergebnis: S mit $flow(f) = cap(S)$

- $S = \{s\}$
- **solange** $x \in S, y \in V \setminus S$ *existieren mit* $\begin{cases} f(x,y) < c(x,y) & , \textit{falls } (x,y) \in E \\ 0 < f(y,x) & , \textit{falls } (y,x) \in E \end{cases}$

  dann **tue**
  | $S = S \cup \{y\}$

  Ende

Behauptung Das Resultat S des Algorithmus ist ein Schnitt von N

Beweis. Zu Zeigen: $t \notin S$
Angenommen: $t \in S, v_r = t$
$\Longrightarrow$ Es gibt $v_{r-1} \in S$ mit $f(v_{r-1}, v_r) < c(v_{r-1}, v_r)$.
Iterativ: Es gibt einen ungerichteten Weg π mit $v_0 = s, v_r = t$
Definiere für $i = 0, \ldots, r-1$:

$$\varepsilon_i = \begin{cases} c(e) - f(e) & , \text{ falls } e = (v_i, v_{i+1}) \in E, e^{-1} = (v_{i+1}, v_i) \notin E \\ f(e^{-1}) & , \text{ falls } e \notin E, e^{-1} \in E \\ \max\{c(e) - f(e), f(e^{-1})\} & , \text{ falls } e \in E, e^{-1} \in E \end{cases}$$

Beobachte: $\varepsilon_i > 0$

$$\varepsilon = \min_{\leq i \leq r-1} \varepsilon_i > 0 \tag{23.1}$$

Zeige: Es gib einen Fluss f^* in N mit $flow(f^*) = flow(f) + \varepsilon$
Konstruiere:

$$f^*(e) = \begin{cases} f(e) + \varepsilon & , \text{ falls } e = (v_i, v_{i+1} \in E, e^{-1} \notin E \\ f(e) - \varepsilon & , \text{ falls } e = (v_{i+1}, v_i) \in E, e^{-1} \notin E \\ f(e) + \varepsilon & , \text{ falls } e = (v_i, v_{i+1}), e^{-1} \in E \text{ und } c(e) - f(e) > f(e^{-1}) \\ f(e) - \varepsilon & , \text{ falls } e = (v_{i+1}, v_i), e^{-1} \in E \text{ und } c(e) - f(e) \leq f(e^{-1}) \\ f(e) & , \text{ sonst} \end{cases}$$

Bemerke:

- Kapazitätsbedingung bleibt erfüllt.
- Kirchpffsches Gesetz bleibt erfüllt, da f^* weiterhin ein Fluss ist.

Das heißt:

$$\begin{aligned} flow(f^*) &= \sum_{v \in post(s)} f^*(s,v) \\ &= \sum_{v \in pre(t)} f^*(v,t) \\ &= \sum_{v \in pre(t) \setminus \{v_{r-1}\}} f^*(v,t) + f^*(v_{r-1},t) \\ &= flow(f) + \varepsilon \end{aligned}$$

Dies ist ein Widerspruch dazu, dass $flow(f) = MaxFlow(N) \implies S$ ist ein Schnitt. □

S ist ein Schnitt in N mit:

$$f(x,y) = c(x,y)$$
$$f(y,x) = 0$$

für alle $x \in S, y \in V \setminus S \implies flow(f) = cap(S)$ nach Kirchoff. □

Wir formalisieren nun weiter:

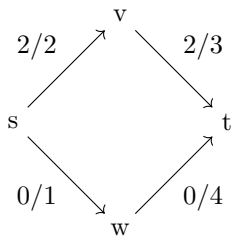
23.37 Definition. Sei f ein Fluss im Netzwerk $N = (V, E, c, s, t)$. Eine Kante $e = (x, y) \in E$ heißt *Vorwärtskante*, falls $f(e) < c(e)$
Eine Kante $e^{-1}(y, x) \in E$ heißt *Rückwärtskante*, falls $f(e^{-1}) > 0$.
Der *Restgraph* für f ist der Digraph $G' = (V, E')$ mit

$$E' = \{(x,y) \in V \times V | (x,y) \text{ ist Vorwärts- oder Rückwärts-Kante}\}$$

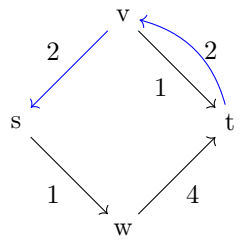
$c(e) - f(e)$ bzw. $f(e^{-1})$ heißen *Restkapazitäten*. Ein *augmentierender Weg* ist ein s-t-Weg im Restgraphen.

Im vorherigen Beweis haben wir einen solchen augmentierenden Weg konstruiert.

23.38 Beispiel. Betrachte



Der Restgraph mit Restkapazitäten ist:



Blau: Rückwärtskante, Rot: Vorwärtskante

Der Beweis von Satz 23.36 gibt uns außerdem einen Algorithmus, wie wir, für einen nicht-optimalen Fluss, ein Fluss mit höheren Wer finden können. Idee: Benutze dies um einen optimalen Fluss zu finden.

Daten: Netzwerk $N = (V, E, c, s, t)$
Ergebnis: Fluss f mit $flow(f) = MaxFlow(N)$

- $f(e) = 0, e \in E$
- Suche einen augmentierenden Weg von s nach t
- **wenn** *keiner existiert* **dann**
  | stop

  Ende
- Berechne ε (23.1)
- Augmentiere f um ε, gehe zu 2

Algorithmus 17: Ford-Fulkerson

23.39 **Bemerkung.** Im Falle von irrationalen Kapazitäten muss der Algorithmus nicht notwendigerweise terminieren (Beweis durch Gegenbeispiel)

23.40 **Satz** (Integral-Flow-Theorem). Sei $N = (V, E, c, s, t)$ ein Netzwerk mit ganzzahligen Kapazitäten. Dann terminiert der Algorithmus 18 nach maximal $\sum_{e \in E} c(e)$ Augmentierungsschritten mit einem ganzzahligen, optimalen Fluss.

Beweis. Betrachte:

Ganzzahlig: Wir starten mit $f(e) = 0, e \in E$. Die Restkapazitäten im Algorithmus und somit auch ε sind zu jedem Zeitpunkt ganzzahlig.

Optimal: Der Algorithmus terminiert, falls kein augmentierender Weg mehr gefunden werden kann. Der Beweis des Max-Flow-Min-Cut-Theorem zeigt, dass dies den maximalen Fluss impliziert.

Anzahl Schritte Im schlimmsten Fall wird pro Iteration der Fluss von genau einer Kante um 1 erhöht $\implies$ Behauptung $\square$

Es gibt eine Möglichkeit, die Laufzeit des Algorithmus zu verbessern, indem man immer den kürzesten augmentierenden Weg wählt. Dies bringt uns zum nächsten Algorithmus:

Daten: Netzwerk $N = (V, E, c, s, t)$
Ergebnis: Fluss f mit $flow(f) = MaxFlow(N)$

- $f(e) = 0, e \in E$
- Suche einen kürzesten augmentierenden Weg von s nach t
- **wenn** *keiner existiert* **dann**
  | stop
  Ende
- Berechne ε (23.1)
- Augmentiere f um ε und gehe zu 2

Algorithmus 18: Edmonds-Karp

23.41 Beispiel. Betrachte

23.42 Lemma. Sei $(f_0, \pi_0), (f_1, \pi_1) \dots$ die Folge von Flüssen f_i mit zugehörigen augmentierenden Weg π, die vom Edmonds-Karps-Algorithmus 19 erzeugt wird. Dann gilt:

$$|\pi_i \leq |\pi_{i+1}| \text{ für alle } i$$

und falls $e = (v, w)$ in π_i und $e^{-1} = (w, v)$ in $\pi_j, i < j$ dann gilt:

$$|\pi_i| + 2 \leq |\pi_j|$$

Beweis. Sei $l_i(x, y)$ die Länge des kürzesten x-y-Wegs im Restgraphen von $f_i \implies |\pi_i| = e_i(s, t)$
Behauptung 1 $l_{i+1}(s, v) \geq l_i(s, v)$ für alle $v \in V$
Falls $l_{i+1}(s, v) = \infty$ ist Behauptung 1 trivial.
Sonst: v ist von s im Restgraphen von f_{i+1} erreichbar, d.h

$$\infty > l_{i+1}(s, v) = r$$

Sei also $\pi = s, v_1, \dots, v_r$ ein kürzester Weg von s nach $v = v_r$ im Restgraphen von f_{i+1}.
Behauptung 2 $l_i(s, v_{j+1}) \leq l_i(s, v_j) + 1, 1 \leq j < r$

Beweis. Behauptung 2

- *Fall 1* (v_j, v_{j+1}) hat eine Kante im Restgraphen $f_i \to$ trivial

- *Fall 2* $(v_j, v_{j+1}) = e$ ist keine Kante im Restgraphen von f_i.

  $\Longrightarrow$ $e^{-1} = (v_{j+1}, v_j)$ ist Kante im Restgraphen von f_i
  $\Longrightarrow$ Der Flusswert vone^{-1} muss sich von f_i zu f_{i+1} verändert haben.
  $\Longrightarrow$ $(v_{j+1}), v_j)$liegt auf π_i
  $\Longrightarrow$ $l_i(s, v_{j+1}) = l_i(s, v_j) - 1 \leq l_i(s, v_j) + 1$

□

Beweis. Behauptung 1. Rechne:

$$\begin{aligned} l_i(s, v) &= l_i(s, v_r) \\ &\leq l_i(s, v_{r-1}) + 1 \\ &\leq \dots \\ &\leq l_i(s, s) + r \\ &= l_{i+1}(s, v) \end{aligned}$$

Behauptung 1 impliziert 1. mit $v = t$. □

Behauptung 3 $l_{i+1}(v, t) \geq l_i(v, t)$ für alle i
Beweis Analog zu Behauptung 1.
Nun können wir alles zusammenführen und rechnen:

$$\begin{aligned} |\pi_j| &= l_j(s, t) \\ &= l_j(s, w) + 1 + l_j(v, t) \\ &\geq l_i(s, w) + 1 + l_i(v, t) \\ &= l_i(s, v) + l_i(v, t) + 2 \\ &= |\pi_i| + 2 \end{aligned}$$

□

23.43 **Satz.** Der Algorithmus von Edmonds-Karp 19 terminiert nach höchsten $\frac{m \cdot n}{2}$ Augmentierungsschritten, $m = |E|, n = |V|$.

Beweis. Sei $(f_0, \pi_0), (f_1, \pi_1), \dots$ die vom Algorithmus erzeugte Folge von Flussfunktionen f_i und zugehörigen augementierenden Wegen π_i im Restgraphen von f_i .
Beobachte 1: In jedem Augmentierungsschritt wird mindestens eine Kante $e \in (v, w)$ in π_i voll ausgeschöpft:

- e Vorwärtskante im Restgraph zu f_i

$$f_i(e) < c(e) \to f_{i+1}(e) = c(e)$$

- e Rückwärtskante im Restgraph zu f_i

$$f_i(e) > 0 \to f_{i+1}(e) = 0$$

$\Longrightarrow$ e ist im $i+1$ Schritt nicht im Restgraph.
Beobachte 2: In Augmentierungsschritt $f_k \to f_{k+1}$ kann e nur in π_k vorkommen und voll ausgeschöpft werden, falls $e^{-1} = (w, v)$ im Weg $\pi_j, i < j < k$ vorgekommen ist.

$$\Longrightarrow |\pi_i| \leq |\pi_j| - 2 \leq |\pi_k| - 4$$

Also: Falls e in $\pi_{i_0}, \ldots, \pi_{i_e}$ voll ausgeschöpft wird, gibt es $j_0, \ldots, j_{e-1}$ so, dass

- $i_0 < j_0 < i_1 < j_1 < \ldots < j_{e-1} < i_e$
- e^{-1} kommt in $\pi_{j_0}, \ldots, \pi_{j_{e-1}}$ vor
- $1 \leq |\pi_{i_0}| \leq \ldots \leq |\pi_{i_k}| - 4l$

Beobachte: π_{i_k} ist eine kürzester Weg im Restgraphen von f_{i_k}

$$\begin{aligned}
&\Longrightarrow \pi_{i_k} \text{ ist einfach} \\
&\Longrightarrow |\pi_{i_k}| < n \\
&\Longrightarrow l < \frac{n}{4} \text{ für jede Kante} \\
&\Longrightarrow \text{ maximal } 2m\frac{n}{4} = \frac{m \cdot n}{2} \text{ Augmentierungsschritte}
\end{aligned}$$

□

23.44 Bemerkung. Dies beweist die Existenz einer Lösung des Netzwerkflussproblems für Kapazitäten in $R_{\geq 0}$.

23.45 Korollar. Für $m = |E|, n = |V|$, ist der Aufwand des Edmonds-Karp Algorithmus $\mathcal{O}(m^2 n)$.

Beweis. Ein kürzeste-Wege-Problem pro Schritt lösbar in $\mathcal{O}(m)$ mit Breitensuche. □

Numerische Lösung linearer Gleichungssysteme

Motivation

Für $\mathbb{K} \in \{\mathbb{R}, \mathbb{C}\}$ bezeichnen wir $\mathbb{K}^n$ den Vektorraum der $\mathbb{K}$-wertigen Vektoren

$$\vec{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}, x_i \in \mathbb{R}, i = 1, \ldots, n$$

und mit $\mathbb{K}^{m \times n}$ den Vektorraum der $\mathbb{K}$-wertigen Matrizen

$$A = \begin{bmatrix} a_{1,1} & \ldots & a_{1,n} \\ \vdots & & \vdots \\ a_{m,1} & \ldots & a_{m,n} \end{bmatrix}, a_{i,j} \in \mathbb{K}, i = 1, \ldots, m, j = 1, \ldots, n$$

In vielen Anwendungen in Physik, Ökonomie, Life Sciences, Informatik, etc. müssen lineare Gleichungssysteme gelöst werden, also System der Form:

$$Ax = b, A \in \mathbb{R}^{n \times n}, x, b \in \mathbb{R}^n$$

Hierbei ist n oft sehr groß.

24. Vektoren- und Matrixnormen

Im folgendem werden wir uns damit beschäftigen, ob lineare Gleichungssysteme gut oder schlecht konditioniert sind, d.h kann man stabile Algorithmen zur Lösung solcher Systeme zur Lösung angeben.

24.1 Definition. Sei X ein $\mathbb{K}$-Vektorraum. Eine Abbildung $\| \bullet \| \colon X \to (0, \infty)$ heißt *Norm* auf X, falls:

a) $\|x\| > 0$ für alle $x \in X \setminus \{0\}$

b) $\|\alpha \cdot x\| = |\alpha| \cdot \|x\|$ für alle $\alpha \in \mathbb{R}, x \in X$

c) $\|x + y\| \leq \|x\| + \|y\|$, für alle $x, y \in X$

24.2 Bemerkung. Jede Norm induziert auch einen Distanzbegriff, den durch die Norm induzierte Metrik

$$dist(x, y) = \|x - y\|$$

Wegen $x = x - 0$ kann $\|x\|$ deshalb als Distanz zum Nullpunkt aufgefasst werden, welche von der gewählten Norm abhängt.

24.3 Beispiel. Normen:

- Für $X = \mathbb{K}^n$
  - Betragssummennorm: $\|x\|_1 = \sum_{i=1}^n |x_i|$
  - Euklidische Norm: $\|x\|_2 = \sqrt[2]{\sum_{i=1}^n |x_i|^2}$
  - Maximumsnorm: $\|x\|_\infty = \max_{1\leq i\leq n} |x_i|$
  - Schatten p-Norm- $\|x\|_p = \sqrt[p]{\sum_{i=1}^n |x_i|^p}$
- Für $X = \mathbb{K}^{m\times n}$
  - Spaltensummennorm: $\|A\|_1 = \max_{1\leq j\leq n} \sum_{i=1}^m |a_{i,j}|$
  - Zeilensummennorm: $\|A\|_\infty = \max_{1\leq i\leq m} \sum_{j=1}^n |a_{i,j}|$
  - Frobeniusnorm $\|A\|_F = \sqrt[2]{\sum_{i=1}^m \sum_{j=1}^n |a_{i,j}|^2}$

Für $A = \begin{bmatrix} 2 & -3 \\ 1 & 1 \end{bmatrix}$ gilt:

$$\|A\|_1 = 4, \|A\|_\infty = 5, \|A\|_F = \sqrt[2]{15}$$

24.4 Satz. Alle Normen auf $\mathbb{K}^n$ sind äquivalent, d.h für zwei Normen $\|\bullet\|_a, \|\bullet\|_b$ auf $\mathbb{K}^n$ gilt:

$$\underline{c}\|x\|_a \leq \|x\|_b \leq \overline{c}\|x\|_a$$

für $\underline{c}, \overline{c} > 0$, unabhängig von x.

Beweis. Es genügt die Behauptung für $\|\bullet\|_a = \|\bullet\|_\infty$ und $\|\bullet\|_b$ beliebig zu zeigen. Wegen:

$$x - y = \sum_{i=1}^n (x_i - y_i)e_i$$

gilt:

$$\begin{aligned}
|\|x\| - \|y\|| &\leq \|x - y\| \\
&= \|\sum_{i=1}^{n}(x_i - y_i)e_i\| \\
&\leq \sum_{i=1}^{n}\|(x_i - y_i)e_i\| \\
&= \sum_{i=1}^{n}|x_i - y_i|\|e_i\| \\
&\leq \|x - y\|_\infty \sum_{i=1}^{n}\|e_i\|
\end{aligned}$$

Also ist $\|\bullet\|\colon \mathbb{K}^n \to \mathbb{R}$ eine Lipschitz-stetige Funktion mit Konstante L.
$\implies \|\bullet\|$ nimmt auf jeder kompakten Menge sowohl Minimum $\underline{c}$ als auch Maximum $\overline{c}$ an. Insbesondere auch auf:

$$S^{n-1} = \{x \in \mathbb{K}^n : \|x\|_\infty = 1\}$$

$$\begin{aligned}
&\implies 0 < \underline{c} \leq \overline{c} \\
&\implies \underline{c} \leq \|\frac{x}{\|x\|_\infty}\| \leq \overline{c} \\
&\implies \underline{c}\|x\|_\infty \leq \|x\| \leq \overline{c}\|x\|_\infty
\end{aligned}$$

□

24.5 Korollar. Analog für Matrizen.

24.6 Beispiel. Für $x \in \mathbb{K}^n$

$$\|x\|_\infty^2 = \max_{1\leq i\leq n}|x_i|^2 \leq \sum_{i=1}^{n}|x_i|^2 \leq n\max|x_i|^2 = n\cdot\|x\|_\infty^2$$

$\implies \|x\|_\infty \leq \|x\|_2 \leq \sqrt[2]{n}\|x\|_\infty$
Die deutet darauf hin, dass im Unendlichdimensionalen nicht alle Normen äquivalent sind.

Da Matrizen miteinander multipliziert werden können, können Normen auf Matrizen spezielle Eigenschaften haben, die diese Eigenschaft wiederspiegeln.

24.7 Definition. Eine Matrixnorm $\|\bullet\|_M$ heißt *submultiplikativ*, falls

$$\|A\cdot B\|_M \leq \|A\|_M \cdot \|B\|_M$$

für alle $A, B \in \mathbb{K}^{n\times n}$.
Eine Matrixnorm $\|\bullet\|_M$ auf $\mathbb{K}^{n\times n}$ heißt *verträglich* mit einer Vektornorm $\|\bullet\|_V$ auf $\mathbb{K}^n$, falls:

$$\|Ax\|_V \leq \|A\|_M \cdot \|x\|_V$$

für alle $A \in \mathbb{K}^{n\times n}, x \in \mathbb{K}^n$.

24.8 Beispiel. • $\|A\| = \max |a_{i,j}$ ist nicht submultiplikativ auf $\mathbb{K}^{n\times n}$.

$$A = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \implies \|A\| = 1$$

jedoch

$$A^2 = \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix} \implies \|A^2\| = 2 \neq 1 = \|A\|^2$$

- Die Frobenius Norm ist submultiplikativ und mit der Euklischen Norm verträglich (Übung)

24.9 Definition. Sei $\|\bullet\|_V$ eine Vektornorm auf $\mathbb{K}^n$, dann ist

$$|||A||| = \sup_{x\neq 0} \frac{\|Ax\|_V}{\|x\|_V} = \max_{\|x\|=1} \|Ax\|_V$$

eine Norm auf $\mathbb{K}^{n\times n}$, die von $\|\bullet\|_V$ induzierte Matrixnorm.

24.10 Beispiel. Die Spaltensummennorm ist von $\|\bullet\|_1$ und die Zeilensummennorm ist von $\|\bullet\|_\infty$ induziert.

24.11 Lemma. Die von $\|\bullet\|_V$ induzierte Matrixnorm ist submultiplikativ und ist mit $\|\bullet\|_V$ verträglich. Ist $\|\bullet\|_M$ eine mit $\|\bullet\|_V$ verträgliche Matrixnorm so gilt:

$$|||A||| \leq \|A\|_M$$

für alle $A \in \mathbb{K}^{n\times n}$.

Beweis. Unterteile:

Submultiplikativität $A = 0$ oder $B = 0 \implies$ klar.
Seien $A, B \neq 0$:

$$\begin{aligned} |||AB||| &= \sup_{x\neq 0} \frac{\|ABx\|_V}{\|x\|_V} \\ &= \sup_{Bx\neq 0} \frac{\|ABx\|_V}{\|x\|_V} \\ &= \sup_{Bx\neq 0} \left(\frac{\|ABx\|_V}{\|x\|_V} \frac{\|Bx\|_V}{\|x\|_V} \right) \\ &\leq \left(\sup_{Bx\neq 0} \frac{\|ABx\|_V}{\|x\|_V} \right) \left(\sup_{Bx\neq 0} \frac{\|Bx\|_V}{\|x\|_V} \right) \\ &= |||A||| \cdot |||B||| \end{aligned}$$

Verträglichkeit $A = 0$ oder $x = 0$ klar.
Seien $A \neq 0, x \neq 0$.

$$\begin{aligned} &\Longrightarrow \frac{\|Ax\|_V}{\|x\|_V} \leq \sup \frac{\|Ax\|_V}{\|x\|_V} = |||A||| \\ &= \text{Behauptung.} \end{aligned}$$

Rechne nun:

$$|||A||| = \sup \frac{\|Ax\|_V}{\|x\|_V} \leq \|A\|_M$$

da $\|Ax\|_V \leq \|A\|_M \|x\|_v$ □

24.12 Bemerkung. Mit den offensichtlichen Modifikationen gelten die Definitionen über Submultiplikativität und Verträglichkeit auch für rechteckige Matrizen.

25. Kondition linearer Gleichungssysteme

25.13 Wiederholung. Die Kondition beschreibt, wie sehr Fehler in den Eingangsdaten eines Problems verstärkt oder gedämpft werden und sich auf einen Fehler der Ausgangsdaten übertragen.

Zum Lösen von $Ax = b$, A invertierbar, bezeichnen wir Δb als einen Eingangsfehler in b.

$$\begin{aligned} &\Longrightarrow x + \Delta x = A^{-1}(b + \Delta b) = A^{-1}b + A^{-1}\Delta b \\ &\Longrightarrow \Delta x = A^{-1}\Delta b \end{aligned}$$

Für ein verträgliches Matrix-Vektorraum-Paar gilt dann:

$$\|\Delta x\| = \|A^{-1}\Delta b\| \leq \|A^{-1}\|\|\Delta b\|$$

und

$$\frac{\|\Delta x\|}{\|x\|} \leq \|A^{-1}\| \frac{\|\Delta b\|}{\|x\|} \leq \|A^{-1}\|\|A\| \frac{\|\Delta b\|}{\|b\|}$$

25.14 Definition. Der Faktor

$$cond_M(A) = \|A^{-1}\|_M \|A\|_M$$

wird als *Kondition* der Matrix A bezüglich der Matrixnorm $\| \bullet \|_M$ bezeichnet.

25.15 Beispiel. Betrachte

$$A = \begin{bmatrix} 10^{-3} & 1 \\ 1 & 1 \end{bmatrix}$$

$$\begin{aligned} &\Longrightarrow A^{-1} \approx \begin{bmatrix} -1.001 & 1.001 \\ 1.001 & -0.001 \end{bmatrix} \\ &\Longrightarrow \|A\|_\infty = 2 \qquad\qquad \|A^{-1}\|_\infty \approx 2 \\ &\Longrightarrow cond_\infty(A) \approx 4 \end{aligned}$$

A ist also bezüglich $\| \bullet \|_\infty$ gut konditioniert.

26. LR-Zerlegung

Man betrachte das Lösen von linearen Gleichungssystemen

$$\begin{bmatrix} a_{1,1} & \dots a_{1,n} & \\ \vdots & & \vdots \\ a_{n,1} & \dots & a_{n,n} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ n \end{bmatrix}$$

mittels Gauss-Elimination. Im i-ten Schritt:

$$\begin{bmatrix} [ccccc|c]* & \dots & \dots & \dots & * & * \\ 0 & * & \dots & \dots & * & * \\ \vdots & 0 & a_{i,i}^{(i)} & \dots & a_{i,n}^{(i)} & b_i^{(i)} \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & a_{n,i}^{(i)} & \dots & a_{n,n}^{(i)} & b_n^{(i)} \end{bmatrix} \begin{matrix} \\ \\ | \cdot (-\tau_{i+1}^{(i)}) \\ \\ | \cdot (-\tau_n^{(i)}) \end{matrix} \quad \tau_j^{(i)} = \frac{a_{j,i}^{(i)}}{a_{i,i}^{(i)}}$$

Die Elimination im i-ten Schritt lässt sich als Matrix-Matrix-Produkt darstellen:

$$L_i[A_i|b_i] = [A_{i+1}|b_{i+1}]$$

wobei

$$L_i = \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & -\tau_{i+1}^{(i)} & 1 & \\ & & -\tau_n^{(i)} & & 1 \end{bmatrix}$$

In der Matrix-Schreibweise ist der Gauss-Algorithmus also als

$$L_{n-1} \cdot L_{n-2} \cdot \ldots \cdot L_1[A|b] = [A_n|b_n] = [R|c] = \begin{bmatrix} * & \dots & * \\ & \ddots & \vdots \\ 0 & & * \end{bmatrix}$$

darstellbar, wobei R eine *obere Dreiecksmatrix* ist.
Insbesondere folgt aus

$$L_{n-1} \cdot \ldots \cdot L_1 A = R,$$

dass

$$A = L_1^{-1} \ldots L_{n-1}^{-1} R = LR$$

ist.

26.16 Lemma. Es gilt:

a) $L_i = I - l_i e_i^t = I - \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \tau_{i+1}^{(i)} \\ \vdots \\ \tau_n^{(i)} \end{bmatrix}$

b) $L_i^{-1} = I + l_i e_i^t = \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & -\tau_{i+1}^{(i)} & 1 & \\ & & -\tau_n^{(i)} & & 1 \end{bmatrix}$

c) $L = I + l_1 e_1^t + l_2^t e_2 + \ldots + l_{n-1} e_{n-1}^t = \begin{bmatrix} 1 & & & & \\ \tau_2^{(1)} & 1 & & & \\ \tau_3^{(1)} & \tau_3^{(2)} & 1 & & \\ \vdots & \vdots & \ddots & \ddots & \\ \tau_n^{(1)} & \tau_n^{(2)} & \ldots & \tau_n^{n-1} & 1 \end{bmatrix}$

Beweis. Nacheinander:

a) Durch hinschreiben.

b) Rechne:

$$(I - l_i e_i^t)(I + l_i e_i^t) = [\ldots 1 \ldots] \cdot \begin{bmatrix} 0 \\ \vdots \\ 0 \\ * \\ \vdots \\ * \end{bmatrix} = 0$$

c) Zeige per Induktion über i, dass: $L_1^{-1} \ldots L_i^{-1} = I + l_1 e_1^t + \ldots + l_i e_i^t$

◊ IV $\underline{i = 1}$ aus b.

◊ IS $\underline{i \to i+1}$

$$\begin{aligned} L_1^{-1} \ldots L_i^{-1} L_{i+1}^{-1} &= (I + l_1 e_1^t + \ldots + l_i e_i^t)(I + l_{i+1} e_{i+1}^t) \\ &= I + l_1 e_1^t + l_2 e_2 + \ldots + l_i e_i^t + l_{i+1} e_{i+1}^t \end{aligned}$$

□

Wir können L also direkt aus den im Gauss-Algorithmus vorkommenden Größen zusammensetzen. Der Algorithmus bricht ab, wenn eins der Pivotelemente null wird.

26.17 **Satz.** Falls kein Pivotelement null wird, bestimmt der Gauss-Algorithmus neben der Lösung x von $Ax = b$ eine LR-Zerlegung $A = LR$ in eine obere Dreiecksmatrix R und eine normierte, untere Dreiecksmatrix L.

26.18 **Beispiel.** Betrachte den Gauss-Algorithmus angewandt auf:

$$A = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 10 \end{bmatrix} \begin{matrix} \\ \mid -2 \\ \mid -3 \end{matrix}$$

$$\rightarrow \begin{bmatrix} 1 & 4 & 7 \\ 0 & -3 & -6 \\ 0 & -6 & -11 \end{bmatrix} \begin{matrix} \\ \mid -2 \\ \\ \end{matrix}$$

$$\rightarrow \begin{bmatrix} 1 & 4 & 7 \\ 0 & -3 & -6 \\ 0 & 0 & 1 \end{bmatrix} = R \qquad L = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 2 & 1 \end{bmatrix}$$

26.19 **Bemerkung.** In der Praxis werden die ursprünglichen Einträge der Matrix A durch die modifizierten Einträge $a_{i,j}^{(k)}$ überschrieben. Die Matrix L lässt sich sukzessive in die nicht mehr benötigten untere Hälfte von A schreiben. Man spricht von "in place"-Algorithmen, die keinen zusätzlichen Speicher benötigen.

Mit Hilfe der LR-Zerlegung wollen wir im folgenden das Lösen von linearen Gleichungssystemen ermöglichen.

1 Berechne $A = LR$

2 Löse $Ax = LRx = b$ durch

- Löse $Ly = b$ mittels Vorwärtssubstitution
- Löse $Rx = y$ mittels Rückwärtssubsititution

Daten: LR-Zerlegung $A = LR$ invertierbar, $A \in \mathbb{R}^{n \times n}, b \in \mathbb{R}^n$
Ergebnis: Lösung x von $Ax = b$
für $i \leftarrow 1$ **bis** n **tue**
  $y_i = b_i - \sum_{j=1}^{i-1} L_{i,j} y_j$
Ende
für $i \leftarrow n$ **bis** 1 **tue**
  $x_i = \frac{1}{R_{i,i}} \left(y_i - \sum_{j=i+1}^{n} R_{i,j} x_j \right)$
Ende

Algorithmus 19: Vorwärts- und Rückwärtssubsitution

26.20 **Beispiel.** Löse $Ax = b$ mit A aus Beispiel 26.18:

$$A = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 5 & 10 \end{bmatrix} = \begin{bmatrix} 1 & & \\ 2 & 1 & \\ 3 & 2 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 4 & 7 \\ & -3 & -6 \\ & & 1 \end{bmatrix} \qquad b = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Vorwärtssubstitution:

$$Ly = \begin{bmatrix} 1 & & \\ 2 & 1 & \\ 3 & 2 & 1 \end{bmatrix} \cdot \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = b \implies \begin{cases} y_1 = 1 \\ y_2 = 1 - 2 = -1 \\ y_3 = 1 - 3 - 2 \cdot (-1) = 0 \end{cases}$$

Rückwärtssubstitution:

$$Rx = \begin{bmatrix} 1 & 4 & 7 \\ & -3 & -6 \\ & & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix} = y \implies \begin{cases} x_3 = 0 \\ x_2 = \frac{1}{-3}(-1 - (-6)0) = \frac{1}{3} \\ x_1 = 1(1 - \frac{4}{3} - 7 \cdot 0 = -\frac{1}{3} \end{cases}$$

26.21 **Bemerkung.** Die LR-Zerlegung und Vorwärts- und Rückwärtssubstitution sind zum klassischen Gauss-Algorithmus äquivalent, es werden sogar die genau gleichen Operationen durchgeführt. Die LR-Zerlegung kann jedoch für andere rechte Seiten wiederverwendet werden.

Aufwand: Anzahl der Multiplikationen im i-ten Schirtt:

$$n - i + (n - i)(n - i) = (n - i)^2 + n - i$$

Also

$$\sum_{i=1}^{n-1} \left((n-i)^2 + n - i\right) = \sum_{i=1}^{n-1} \left(i^2 + i\right) = \frac{1}{3}n^3 + \mathcal{O}(n^2)$$

Multiplikationen zur Berechnung der *LR*-Zerlegung. Der Aufwand der Substitutionen beträgt: $\mathcal{O}(n^2)$.
Daraus resultiert ein Gesamtaufwand zum Lösen eines linearen Gleichungssystem von : $\frac{1}{3}n^3 + \mathcal{O}(n^2)$. Der Aufwand im Speicher ist $\mathcal{O}(n^2)$.

27. LR-Zerlegung mit Spaltenpivotisierung

27.22 **Beispiel.** Fortsetzung von Beispiel 25.15

$$A = \begin{bmatrix} 10^{-3} & 1 \\ 1 & 1 \end{bmatrix}, b = \begin{bmatrix} 1/2 \end{bmatrix} \qquad \implies x = \begin{bmatrix} 1.001001 \\ 0.998999 \end{bmatrix}$$

Löse in dreistelliger Arithmetik, mit kleiner Störung in der rechten Seite:

$$\begin{bmatrix} 0.001 & 1 & 1.01 \\ 1 & 1 & 2.01 \end{bmatrix} \begin{array}{l} \\ | \cdot (-1000) \\ \longleftarrow \end{array} \to \left[\begin{array}{cc|c} 0.001 & 1 & 1.01 \\ 0 & -999 & -1010 \end{array}\right] \begin{array}{l} \\ \\ + \end{array}$$

$$\implies \begin{cases} x_2 = -1010 \boxslash (-999) = 1.01 \\ x_1 = (1.01 \boxminus (1 \boxdot 1.01)) \boxslash 0.001 = 0 \end{cases}$$

Der Algorithmus muss also instabil sein.

Grund: Das Inverse des Pivot-Elemts (1000) ist sehr groß und verstärkt den Fehler

27.23 Definition. Sei $1 \leq i < j \leq n$

$$T^{\{i,j\}} = \begin{bmatrix} 1 & & & & & & & & & \\ & \ddots & & & & & & & & \\ & & 1 & & & & & & & \\ & & & 0 & & & 1 & & & \\ & & & & 1 & & & & & \\ & & & & & \ddots & & & & \\ & & & & & & 1 & & & \\ & & & 1 & & & & 0 & & \\ & & & & & & & & 1 & \\ & & & & & & & & & \ddots \\ & & & & & & & & & & 1 \end{bmatrix}$$

heißt Transpositionsmatrix.
Eine Permutationsmatrix ist das Produkt mehrerer Transpositionsmatrizen.

27.24 Lemma. Sei $T^{\{i,j\}}$ eine Transpositionsmatrix. Es gilt:

a) Multiplikation von links mit $T^{\{i,j\}}$ vertauscht i-te und j-te Zeile.

b) Multiplikation von rechts mit $T^{\{i,j\}}$ vertauscht i-te und j-te Spalte.

c) $T^{\{i,j\}}$ ist symmetrisch.

d) $\left(T^{\{i,j\}}\right)^2 = I$

Ist P eine Permutationsmatrix, so gilt: $P^T P = I$

Beweis. a-d sind elementar. Letzter Punkt: Übung. □

Umseztung: Anstatt im i-ten Schritt des Gauss-Algorithmus

$$A_i \to L_i A_i = A_{i+1}$$

zu berechnen, berechnen wir:

$$A_i \to L_i P_i A_i = A_{i+1} \tag{27.1}$$

27.25 Lemma. Sei $1 \leq i < j \leq n$ und $P_i = T^{\{i,k\}}$ mit $i < k \leq n$. Dann gilt: $P_i L_j = \tilde{L_j} P_i$, wobei $\tilde{L_j}$ und L_j bis auf ein Vertauschen der $\tau_k^{(j)}$ und $\tau_i^{(j)}$ gleich sind.

Beweis. Durch nachrechnen. Also:

$$\tilde{L_j} = P_i L_j P i$$

□

27.26 **Satz.** Sei A nicht singulär. Dann bestimmt der Gauss-Algorithmus mit Spaltenpivotisierung eine Zerlegung der Matrix $PA = \tilde{L}R$, wobei R die obere Dreiecksmatrix A_n ,$P = P_{n-1} \dots P_1$ eine Permutationsmatrix und $\tilde{L} = \tilde{L_1^{-1}} \dots \tilde{L_{n-1}^{-1}}$ eine untere Dreiecksmatrix ist mit:

$$\begin{aligned}
&\tilde{L_{n-1}} = L_{n-1} \\
&\tilde{L_{n-2}} = P_{n-1} L_{n-2} P_{n-1} \\
&\dots\dots \\
&\tilde{L_1} = P_{n-1} P_{n-2} \dots P_2 P_1 L_1 P_2 \dots P_{n-2} P_{n-1}
\end{aligned}$$

Beweis. Falls der Algorithmus nicht zusammenbricht:

$$\begin{aligned}
R = A_n &= L_{n-1} P_{n-1} A_{n-1} \\
&= \tilde{L_{n-1}} P_{n-1} L_{n-2} P_{n-2} A_{n-2} \\
&= \tilde{L_{n-1}} \tilde{L_{n-2}} P_{n-1} P_{n-2} L_{n-3} P_{n-3} A_{n-3} \\
&\dots \\
&= \tilde{L_{n-1}} \dots \tilde{L_1} P_{n-1} \dots P_1 A \\
&= \tilde{L^{-1}} \cdot P
\end{aligned}$$

Sollte der Algorithmus abbrechen, folgt, dass ein Pivot-Element null wird. Daraus folgt, dass die Determinante von der Restmatrix null ist. Demnach ist auch die Determinante von der Matrix A null. Das ist ein Widerspruch dazu, dass A invertierbar ist. □

27.27 **Beispiel.** Wir betrachten die $PA = \tilde{L}R$ Zerlegung.

$$\begin{bmatrix} 1 & 1 & 0 & 2 \\ \frac{1}{2} & \frac{1}{2} & 2 & -1 \\ -1 & 0 & -\frac{1}{8} & -5 \\ 1 & -7 & 9 & 10 \end{bmatrix} \begin{matrix} \\ \left| -\frac{1}{2} \right. \\ \left| \\ \left| \\ \end{matrix} \begin{matrix} \\ \leftarrow \\ \leftarrow \\ \leftarrow \end{matrix} \begin{matrix} \\ \\ \left| -1 \right. \\ \left| \\ \end{matrix} \begin{matrix} \\ \\ + \\ + \\ + \end{matrix} \rightarrow \begin{bmatrix} 1 & 1 & 0 & 2 \\ 0 & 0 & 2 & -2 \\ 0 & 1 & -\frac{1}{8} & -3 \\ 0 & -8 & 9 & 8 \end{bmatrix} \begin{matrix} \\ \leftarrow \\ \\ \leftarrow \end{matrix} \xrightarrow{P_2}$$

$$\begin{bmatrix} 1 & 1 & 0 & 2 \\ 0 & -8 & 9 & 8 \\ 0 & 1 & -\frac{1}{8} & -3 \\ 0 & 0 & 2 & -2 \end{bmatrix} \begin{matrix} \\ \\ \left| \frac{1}{8} \right. \\ \\ \end{matrix} \begin{matrix} \\ \\ \leftarrow \\ \\ \end{matrix} \begin{matrix} \\ \\ + \\ \\ \end{matrix} \rightarrow \begin{bmatrix} 1 & 1 & 0 & 2 \\ 0 & -8 & 9 & 8 \\ 0 & 0 & 1 & -2 \\ 0 & 0 & 2 & -2 \end{bmatrix} \begin{matrix} \\ \\ \leftarrow \\ \leftarrow \end{matrix} \xrightarrow{P_3}$$

$$\begin{bmatrix} 1 & 1 & 0 & 2 \\ 0 & -8 & 9 & 8 \\ 0 & 0 & 2 & -2 \\ 0 & 0 & 1 & -2 \end{bmatrix} \begin{matrix} \\ \\ \\ \left| -\frac{1}{2} \right. \\ \end{matrix} \begin{matrix} \\ \\ \\ \leftarrow \\ \end{matrix} \begin{matrix} \\ \\ \\ + \\ \end{matrix} \rightarrow \begin{bmatrix} 1 & 1 & 0 & 2 \\ 0 & -8 & 9 & 8 \\ 0 & 0 & 2 & -2 \\ 0 & 0 & 0 & -1 \end{bmatrix} = R$$

$$PA = \begin{bmatrix} 1 & 1 & 0 & 2 \\ 1 & -7 & 9 & 10 \\ \frac{1}{2} & \frac{1}{2} & 2 & -1 \\ -1 & 0 & -\frac{1}{8} & -5 \end{bmatrix} L = \begin{bmatrix} 1 & & & \\ \frac{1}{2} & 1 & & \\ -1 & -\frac{1}{8} & 1 & \\ 1 & 0 & \frac{1}{2} & 1 \end{bmatrix} \rightarrow \tilde{L} = \begin{bmatrix} 1 & & & \\ 1 & 1 & & \\ \frac{1}{2} & 0 & 1 & \\ -1 & -\frac{1}{8} & \frac{1}{2} & 1 \end{bmatrix}$$

27.28 **Bemerkung.** Um $\tilde{L}$ zu bekommen, berechne zuerst L, und wende dann die Transpositionsmatrizen P_j auf die i-te Spalte an.

27.29 **Beispiel.** Betrachte

$$A = \begin{bmatrix} 10^{-3} & 1 \\ 1 & 1 \end{bmatrix}, b = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \implies x = \begin{bmatrix} 1.001001 \\ 0.998999 \end{bmatrix}$$

in 3-stelliger Arithmetik wie zuvor.

$$\left[\begin{array}{cc|c} 0.001 & 1 & 1.01 \\ 1 & 1 & 2.01 \end{array}\right] \to \left[\begin{array}{cc|c} 1 & 1 & 2.01 \\ 0.001 & 1 & 1.01 \end{array}\right] \to \left[\begin{array}{cc|c} 1 & 1 & 2.01 \\ 0 & -0.999 & 1.01 \end{array}\right] \implies \begin{cases} x_2 = 1.01 \boxslash (-0.999) = 1.01 \\ x_1 = (2.01 \boxminus 1.01) = 1 \end{cases}$$

27.30 **Bemerkung.** Die Spaltenpivotisierung reicht in der Praxis meistens aus. Eine stabilere Idee ist die totale Pivotisierung. Hierbei werden Zeilen und Spalten so permutiert, dass das betragsmäßig größte Element von A_i im i-ten Schritt das Pivot-Element ist. Der Aufwand ist jedoch nicht mehr vernachlässigbar.

28. Cholesky-Zerlegung

In der Praxis sind viele Matrizen symmetrisch, positiv definit.

28.31 **Definition.** Eine symmetrische Matrix $A \in \mathbb{R}^{n\times n}$ heißt *positiv definit*, falls

$$x^t A x > 0 \text{ für alle } x \in \mathbb{R}^n \setminus 0 \tag{28.1}$$

28.32 **Lemma.** Eine positiv definite symmetrische Matrix $A \in R^{n\times n}$ ist invertierbar.

Beweis. Angenommen: A ist nicht invertierbar.
$\implies$ Es gibt $x \in \mathbb{R}^n, x \neq 0$ so, dass $Ax = 0$
$\implies$ $x^t A x = 0$
Dies ist schon ein Widerspruch zu (28.1) □

Betrachte eine Matrix

$$A = \left[\begin{array}{c|c} A_{1,1} & A_{1,2} \\ \hline A_{2,1} & A_{2,2} \end{array}\right] \in \mathbb{R}^{n\times n}$$

mit $1 \leq p \leq n-1$ und das lineare Gleichungssystem:

$$Ax = \left[\begin{array}{c|c} A_{1,1} & A_{1,2} \\ \hline A_{2,1} & A_{2,2} \end{array}\right] \cdot \left[\begin{array}{c} x_1 \\ \hline x_2 \end{array}\right] = \left[\begin{array}{c} b_1 \\ \hline b_2 \end{array}\right] \tag{28.2}$$

Falls $A_{1,1}$ invertierbar ist, können wir eine "Block"-Gauss-Elimination durchführen:

$$\begin{bmatrix} A_{11} & A_{12} & b_1 \\ A_{21} & A_{22} & b_2 \end{bmatrix} \begin{array}{l} \mid -A_{21}A_{11}^{-1} \urcorner \\ \longleftarrow\!\!\!-\!\!\!-\!\!\!\lrcorner_{+} \end{array} \to \left[\begin{array}{cc|c} A_{11} & A_{12} & b_1 \\ 0 & A_{22} - A_{21}A_{11}^{-1}A_{12} & b_2 - A_{21}A_{11}^{-1}b_1 \end{array}\right]$$

28.33 Definition. Sei $S \coloneqq A_{22} - A_{11}^{-1}A_{12}$ dann ist S das Schurkomplement von A bezüglich A_{11}. Falls S invertierbar ist gilt:

$$x_2 = S^{-1}(b_2 - A_{21}A_{11}^{-1}b_1)$$
$$x_1 = A_{11}^{-1}(b_1 - A_{12}x_2)$$

28.34 Lemma. Sei $A \in \mathbb{R}^{n\times n}$ symmetrisch, positiv definit. Dann ist A_{11} dies ebenfalls und somit invertierbar. Das Schurkomplement S bezüglich A_{11} ist wohldefiniert und symmetrisch, positiv definit.

Beweis. Zeige:

A spd: Da A spd gilt:

$$0 \leq \begin{bmatrix} x_1^t & 0 \end{bmatrix} \cdot \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ 0 \end{bmatrix} = \begin{bmatrix} x_1^t & 0 \end{bmatrix} \cdot \begin{bmatrix} A_{11}x_1 \\ A_{21}x_1 \end{bmatrix} = x_1^t A_{11} x_1$$

mit Gleichheit genau dann, wenn $x_1 = 0 \implies A_{11}$ ist spd und S wohldefiniert.

S symmetrisch :

$$\begin{aligned} S^t &= \left(A_{22} - A_{21}A_{11}^{-1}A_{12}\right)^t \\ &= A_{22}^t - A_{12}^t A_{11}^{-t} A_{21}^t \\ &= A_{22} - A_{21}A^{-1}A_{12} \\ &= S \end{aligned}$$

S positiv definit: Für x_2 beliebig setze $x_1 = -A_{11}^{-1}A_{12}x_2$:

$$\begin{aligned} 0 &\leq \begin{bmatrix} x_1^t & x_2^t \end{bmatrix} \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\ &= \begin{bmatrix} A_{11}x_1 + A_{12}x_2 \\ A_{21}x_1 + A_{22}x_2 \end{bmatrix} \\ &= \begin{bmatrix} 0 \\ -A_{21}A_{11}^{-1}A_{12}x_2 + A_{22}x_2 \end{bmatrix} \\ &= \begin{bmatrix} 0 \\ Sx_2 \end{bmatrix} \end{aligned}$$

mit Gleichheit genau dann, wenn $x_2 = 0$ und $x_1 = 0 \implies S$ ist positiv definit und symmetrisch. □

28.35 Satz. Sei $A \in \mathbb{R}^{n\times n}$ und symmetrisch, positiv definit. Dann existiert eine *Cholesky-Zerlegung* $A = LL^t$, wobei $L \in \mathbb{R}^{n\times n}$ eine untere Dreiecksmatrix ist.

Beweis. Per Induktion.

□

28.36 Korollar. A hat eine Cholesky-Zerlegung genau dann, wenn A spd ist.

Beweis. Übung. □

28.1. Berechnung der Cholesky-Zerlegung

Wir suchen Zahlen $l_{i,j}$ so, dass :

$$\begin{bmatrix} a_{1,1} & \dots & a_{n,1} \\ \vdots & & \vdots \\ a_{n,1} & \dots & an,n \end{bmatrix} = \begin{bmatrix} l_{11} & & \\ l_{21} & l_{22} & \\ \vdots & \vdots & \ddots \\ l_{n,1} & l_{n,2} & l_{n,n} \end{bmatrix} \cdot \begin{bmatrix} l_{11} & l_{21} & \dots & l_{n,1} \\ & l_{22} & & \vdots \\ & & \ddots & l_{n,n} \end{bmatrix}$$

Also:

$$\begin{aligned} a_{11} = l_{11}^2 &\implies l_{11} = \sqrt[2]{a_{11}} \\ a_{21} = l_{21}l_{11} &\implies l_{21} = \frac{a_{21}}{l_{11}} \\ \vdots &\implies \vdots \\ a_{22} = l_{21}^2 + l_{22}^2 &\implies l_{22} = \sqrt[2]{a_{22} - l_{21}^2} \\ a_{32} = l_{31}l_{21} + l_{32}l_{22} &\implies l_{32} = \frac{a_{32} - l_{31}l_{21}}{l_{22}} \end{aligned}$$

Allgemein:

$$\left.\begin{aligned} l_{j,j} &= \sqrt[2]{a_{jj} \textstyle\sum_{k=1}^{j-1} l_{j,k}^2} & j = 1, \dots, n \\ l_{i,j} &= \frac{1}{l_{j,j}\left(a_{i,j} - \sum_{k=1}^{j-1} l_{i,k}l_{j,k}\right)} & j < i \leq n \end{aligned}\right\} \tag{28.3}$$

28.37 Bemerkung. Es muss $l_{jj} \neq 0$ gelten, da anderenfalls ein Widerspruch zur Existenzaussage über die Cholesky-Zerlegung besteht.

28.38 Korollar. Werden die Vorzeichen der Diagonalelemente der Cholesky-Faktoren eindeutig festgelegt, dann ist die Cholesky-Zerlegung eindeutig.

Aufwand: (28.3) sagt, dass zur Berechnung von $l_{i,j}$, j Multiplikationen bzw. Divisionen / Wurzeln benötigt werden. Der Gesamtaufwand beträgt demnach:

$$\sum_{j=1}^{n}(n - j + 1)j) = \frac{1}{6}n^3 + \mathcal{O}(n^2)$$

Dies ist circa doppelt so schnell wie die LR-Zerlegung.

Graphenbasierte Löser

Motivation

Betrachte die Temperatur in einem Raum, modelliert durch $\Omega = (0,1)^2$. Die Temperatur in jedem Punkt im Raum kann als Funktion $u\colon \Omega \to \mathbb{R}$ aufgefasst werden. Gegeben sei außerdem eine Wärmequelle, modelliert als $f\colon \Omega \to \mathbb{R}$ im Raum und eine Temperatur der Wände von $u(x) = 0, x \in \partial\Omega = (\{0,1\} \times [0,1]) \cup ([0,1] \times \{0,1\}$. Nach einer gewissen (unendlichen) Zeit wird die Temperatur ein Gleichgewicht annehmen, welches die Lösung einer *partiellen Differentialgleichung*

$$\begin{cases} -\Delta u(x) = f(x) & x \in \Omega \\ u(x) = 0 & x \in \partial\Omega \end{cases} \tag{28.1}$$

geschrieben werden.

28.1 Definition. Der Laplace-Operator Δ ist definiert als:

$$\Delta u(x) = \frac{\partial^2 u}{\partial x_1^2}(x_1, x_2) + \frac{\partial^2 u}{\partial x_2^2}(x_1, x_2, x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \in \Omega$$

falls u genügend oft differenzierbar ist.

Das Ziel ist es, u zuberechnen, wenn f gegeben ist. Das Problem im Allgemeinen ist, dass (28.1) nicht von Hand lösbar ist, daher erfolgt die Lösung mit Hilfe des Computers. Dies ist generell nur approximativ möglich.

Beobachte: Für eine genügend oft differenzierbare Funktion $g\colon \mathbb{R} \to \mathbb{R}$, gilt

$$g'(x) = \lim_{h \to 0} \frac{g(x+h) - g(x)}{h} \approx \frac{g(x) - g(x-h)}{h} \approx \frac{g(x+h) - g(x)}{h}$$

für ein kleines h. Also:

$$\begin{aligned} g"(x) &\approx \frac{g'(x+h) - g'(x)}{h} \\ &\approx \frac{\frac{g(x+h)-g(x)}{h} - \frac{g(x)-g(x-h)}{h}}{h} \\ &\approx \frac{g(x+h) - 2g(x) + g(x-h)}{h^2} \end{aligned}$$

Unsere Idee ist nun, die zweiten Ableitungen im Laplace-Operator durch diese Approximationen zu ersetzen:

$$\frac{\partial^2 u}{\partial x_1^2}(x_1, x_2) \approx \frac{u(x_1 + h, x_2) - 2u(x_1, x_2) + u(x_1 - h, x_2)}{h^2}$$
$$\frac{\partial^2 u}{\partial x_2^2}(x_1, x_2) \approx \frac{u(x_1, x_2 + h) - 2u(x_1, x_2) + u(x_1, x_2 - h)}{h^2}$$

Für eine systematische Approximation der Ableitungen führen wir ein Gitter ein. Wähle $n \in \mathbb{N}$ und eine Schrittweite $h = \frac{1}{n}$.

Wir betrachten u nur noch an den Gitterpunkten $x_{i,j}$ und suchen Approximationen $u_{i,j} \approx u(x_{i,j})$. Wir erhalten:

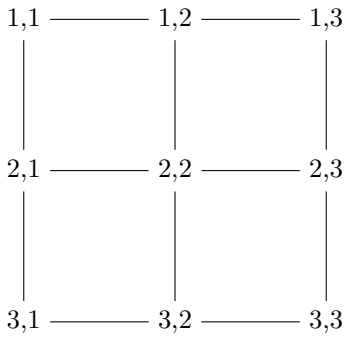
$$\left.\begin{array}{l} \frac{1}{h^2}(4u_{i,j} - u_{i-1,j} - u_{i+1,j} - u_{i,j-1} - u_{i,j+1} = f_{i,j} = f(x_{i,j}), x_{i,j} \in \Omega \\ u_{i,j} = 0, x_{i,j} \in \partial\Omega, \text{ also } i, j \in \{0, n\} \end{array}\right\} \tag{28.2}$$

Wir erhalten also ein lineares Gleichungssystem $An = f$ mit $N = (n-1)^2$ Unbekannten. Gemäß (28.2) hat A nur fünf Einträge pro Zeile bzw. Spalte, insgesamt also $\mathcal{O}(N)$ Einträge. A ist demnach extrem dünn besetzt.

28.2 Beispiel. Sei $n = 4$. Dann ist $h = \frac{1}{4}$ und

Beobachtung: Wir haben das Lösen einer partiellen Differentialgleichung auf das Lösen eines linearen Gleichungssystem reduziert. Jedoch ist für die Lösung u von (28.1) zu erhalten, oft $1 \ll n$ nötig. Die Größe des linearen Gleichungssystem ist $\mathcal{O}(n^2) = \mathcal{O}(N)$. Das Lösen des linearen Gleichungssystem ist demnach $\mathcal{O}(N^3)$, was in der Realität praktisch nicht durchführbar ist. Dies liegt daran, dass die LR-Zerlegung von A im ALlgemeinem viel mehr nicht-null Einträge hat als A.
Um die Dünnbesztheit von A besser auszunutzen, sodass die Faktoren der (modifizierten LR-Zerlegung ebenfalls dünnbestetzt sind betrachten wir die Systemmatrix A als ADjazentmatrix eines Graphen G_A.

28.3 Beispiel. Der Graph G_A zur Matrix aus Beispiel 28.2 ist:



29. Cholesky-Zerlegung und Graphen

Beobachtung: Da Umnummerieren von Knoten verändert einen Graphen nicht. In der zugehörigen Adjazenzmatrix müssen aber Spalten und Zeilen entsprechend permutiert werden.

Idee: Permutiere Zeilen und Spalten der Adjazenzmatrix so, dass möglichst wenig fill-in bei der Berechnung von LR- oder Cholesky-Zerlegung generiert wird.
Die Matrix aus dem vorherigen Kapitel ist symmetrisch positiv definit. Die Einträge der Cholesky-Zerlegung sind dann gegeben durch:

$$l_{j,j} = \sqrt{a_{\partial(j),\partial(j)} - \sum_{k=1}^{j-1} l_{j,\partial(k)}}$$
$$l_{i,j} = \frac{1}{l_{jj}} \left(a_{\partial(i),\partial(j)} - \sum_{k=1}^{j-1} l_{i,\partial(k)} l_{j,\partial(k)} \right)$$

wobei $\partial(1), \partial(2), \ldots, \partial(N)$ eine Permutation ist.

Setzen wir $l_i = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ l_{i,i} \\ \vdots \\ l_{n,i} \end{bmatrix} \in \mathbb{R}^N$ und schreiben

$$A_1 = A, A_{i+1} = A_i - l_i l_i^t = [a_{l,k}^{(i)}]_{l,k=1}^N$$

Wir können l_i auch berechnen durch:

$$l_i = \frac{1}{\sqrt{a_{\partial(i),\partial(i)}^{(i)}}} \begin{bmatrix} a_{1,\partial(i)}^{(i)} \\ \vdots \\ a_{N,\partial(i)}^{(i)} \end{bmatrix} \tag{29.1}$$

Dies entspricht der sukzessiven Elimination der Zeilen und Spalten $\partial(1), \ldots, \partial(N)$. Der zugehörige Graph verändert sich in denn Sinne, dass in jedem Schritt ein Knoten eliminiert wird.

29.4 Beispiel. Betrachte A aus Beispiel 28.2.

29.5 Lemma. Sei $v_i = nnz(l_i)$ die Anzahl nicht-null Einträge in l_i. Dann ist der Aufwand um die Cholesky-Zerlegung $A = LL^t$ zu bestimmen gegeben durch:

$$\Theta = \sum_{i=1}^{N} \frac{v_i(v_i + 3)}{2}$$

Die Anzahl nicht-null Einträge $nnz(L)$ in L ist:

$$\eta = \sum_{i=1}^{N} v_i$$

Beweis. (**??**) impliziert, dass l_i mit v_i Multiplikationen berechnet werden kann. $l_i l_i^t$ ist symmetrisch, kann also in $\frac{v_i(v_i+1)}{2}$ Multiplikationen berechnet werden.

$$\Theta = \sum_{i=1}^{N} \left(\frac{v_i(v_i+1)}{2} + v_i \right) = \sum_{i=1}^{N} \frac{v_i(v_i+3)}{2}$$

Die Formel für η folgt aus $L = [l_1 | \dots | l_N]$. □

Um v_i möglichst klein zu halten eliminieren wir nun in einer Reihenfolge.

29.6 Definition. Eine *Eliminationsreihenfolge* auf einem Graphen $G = (V, E)$ ist eine Numerierung $\sigma \colon \{1, \dots, |V|\} \to V$ der Knoten. Diese führt auf einen *geodneten Graphen* $G^\sigma = (V, E, \sigma)$. Für G^σ ist das *Defizit* eines Knotens gegeben durch:

30. Nested Dissection

31. Generalized nested dissection

Teil IV

Algorithmische Mathematik 2

Wahrscheinlichkeitsräume

1. Zufällige Ereignisse

Die "Zutaten" für ein zufälliges Ereignis sind:

- Zufallssituation, die (gedanklich) beliebig oft wiederholt werden kann
- Ein Versuch / eine Realisierung einer Zufallssituation
- Eine Menge aller möglichen Ereignisse

Dabei ist die Annahme, dass das Ergebnis jedes Versuch Element in der Ereignismenge ist.

1.1 **Definition** ((zufälliges) Ereignis). Sei Ω eine *Eregnismenge*. Ein *(zufälliges) Ereignis* ist eine Teilmenge $A \subset \Omega$. Für ein Ergebnis eines Versuchs $\omega \in \Omega$ sagt man, dass A eingetreten ist, falls $\omega \in A$.

1.2 **Bemerkung.** Ω kann Ereignisse enthalten, die nie eintreten! Umgekehrt müssen alle eintretenden Ereignisse Teilmenge von Ω sein.

1.3 **Beispiel.** Typische Alltagssituationen sind Zufallssituationen:

- Werfen eines Würfels : $\Omega = \{1, \ldots, 6\}$
  Eine gerade Zahl würfeln: $A = \{2, 4, 6\} \subset \Omega$
- Lebensdauer eines Elektrogeräts: $\Omega = \{\omega \in \Omega | \omega \geq 0\}$
  Lebensdauer beträgt 3-4 Jahre: $A = \{\omega \in \Omega | 3 \leq \omega \leq 4\}$
- Überprüfen der Funktionsfähigkeit von Elektrogeräten: $\Omega = \{(\omega_1, \ldots, \omega_n) \in \Omega | w_i \in \{0, 1\}\}$
  Es funktionieren mindestens $k \in \mathbb{N}$ Geräte davon: $A = \{(\omega_1, \ldots, \omega_n) \in \Omega | \sum_{i=1}^{n} w_i \geq k\}$

2. Rechnen mit zufälligen Ereignissen

2.4 **Notation.** Sei Ω eine Ereignismenge. Für einen Versuch $\omega \in \Omega$ und Ereignisse $A, B \in \Omega$ sagen wir:

- A oder B, falls $\omega \in A \cup B$.
- A und B, falls $\omega \in A \cap B$
- nicht A, falls $\omega \notin A \iff \omega \in \Omega \setminus A =: \overline{A}$

- sicheres Ereignis, falls $A = \Omega$
- unmögliches Ereignis, falls $A = \overline{\Omega} = \emptyset$
- Elementarereignis A, falls $A = \{\tilde{\omega}\}$ für $\tilde{\omega} \in \Omega$
- unvereinbare Ereignisse, falls $A \cap B = \emptyset$
- A zieht B nach sich, falls $A \subset B$

2.5 Bemerkung. Die Begriffe gelten auch für $\bigcup_{i=1}^{n} A_i$ und $\bigcap_{i=1}^{n} A_i$ bzw. $\bigcup_{i=1}^{\infty} A_i$ und $\bigcap_{i=1}^{\infty} A_i$ mit $A_i \subset \Omega, i \in \mathbb{N}$

2.6 Satz. Sei Ω eine Menge mit $A, B, C \subset \Omega$. Dann gilt:

- Kommutativgesetz:
  - $A \cup B = B \cup A$
  - $A \cap B = B \cap A$
- Assoziativgesetz:
  - $(A \cup B) \cup C = A \cup (B \cup C)$
  - $A \cap B) \cap C = A \cap (B \cap C)$
- Distributivgesetz:
  - $(A \cap B) \cup C = (A \cup C) \cap (B \cup C)$
  - $(A \cup B) \cap C = (A \cap C) \cup (B \cap C)$
- Morgansche Regeln:
  - $\overline{A \cap B} = \overline{A} \cup \overline{B}$
  - $\overline{A \cup B} = \overline{A} \cap \overline{B}$
- ebenfalls gilt:
  - $\overline{\bigcup_{i \in I} A_i} = \bigcap_{i \in I} \overline{A_i}$
  - $\overline{\bigcap_{i \in I} A_i} = \bigcup_{i \in I} \overline{A_i}$
  - $A \cup \emptyset = A$ und $A \cup \Omega = \Omega$
  - $A \cap \emptyset = \emptyset$ und $A \cap \Omega = A$

Beweis. Analysis 1 □

2.7 Definition. Sei Ω eine Ereignismenge. Eine Menge $\mathcal{A}$ von Teilmengen von Ω heißt σ-Algebra falls:

a) $\Omega \in \mathcal{A}$

b) $A \in \mathcal{A} \implies \overline{A} \in \mathcal{A}$

c) $A_i \in \mathcal{A} \forall_{i \in \mathbb{N}} \implies \bigcup_{i=1}^{\infty} A_i \in \mathcal{A}$

Das Tupel $(\Omega, \mathcal{A})$ heißt *messbarer Raum.*

2.8 Beispiel. Die Potenzmenge $\mathcal{P}(\Omega) = \{A | A \subset \Omega\}$ ist die "größtmögliche" σ-Algebra.
Münzwurf: $\Omega = \{0, 1\}$.
$\mathcal{P}(\Omega) = \{\emptyset, \{0\}, \{1\}, \{0, 1\}\}$
$\mathcal{A} = \{\emptyset\} \subset \mathcal{P}(\Omega)$ ist keine σ-Algebra!

2.9 Korollar. Sei $(\Omega, \mathcal{A})$ ein messbarer Raum. Dann gilt:

a) $\emptyset \in \mathcal{A}$

b) $A, B \in \mathcal{A} \implies A \setminus B \in \mathcal{A}$

c) $A_i \in \mathcal{A}, i \in \mathbb{N} \implies \bigcap_{i=1}^{\infty} A_i \in \mathcal{A}$

Beweis. Übungsaufgabe □

3. Rechnen mit Wahrscheinlichkeiten

Sei $(\Omega, \mathcal{A}$ ein messbarer Raum und $A \subset \mathcal{A}$ ein Ereignis. Betrachte $n \in \mathbb{N}$ sich nicht beeinflussende, d.h. unabhängige Versuche.

3.10 Definition (Häufigkeiten). Wir unterscheiden in:

- Die *absolute Häufigkeit* $h_n(A)$, welche an gibt, wie oft A bei n Versuchen eintritt.
- Die *relative Häufigkeit*: $H_n(A) = \frac{h_n(A)}{n}$

3.11 Satz. Sei $(\Omega, \mathcal{A})$ ein messbarer Raum. Dann gilt:

a) $H_n(\Omega) \geq 0$ für alle $A \subset \mathcal{A}, n \in \mathbb{N}$

b) $H_n(\Omega) = 1$

c) $H_n(A \cup B) = \frac{h_n(A) + h_n(B)}{n} = H_n(A) + H_n(B)$

3.12 Wiederholung. $(\Omega, \mathcal{A})$ ist ein messbarer Raum. Dabei ist Ω die Ereignismenge und $\mathcal{A}$ eine σ-Algebra. Für diese müssen folgende Dinge erfüllt sein:

a) $\Omega \in \mathcal{A}$

b) $A \in \mathcal{A} \implies \overline{A} \in \mathcal{A}$

c) $A_i \in \mathcal{A}, i \in \mathbb{N} \implies \bigcup_{i=1}^{\infty} A_i \in \mathcal{A}$

Außerdem haben wir die absolute Häufigkeit $h_n(A)$ und die relative Häufigkeit $H_n(A) = \frac{h_n(A)}{n}$ eingeführt. Die Eigenschaften der relativen Häufigkeit sind:

1) $0 \leq H_n(A)$

2) $H_n(\Omega) = 1$

3) A, B unvereinbar $\implies H_n(A \cup B) = H_n(A) + H_n(B)$

3.13 Notation. Die relative Häufigkeit konvergiert oft für $n \to \infty$ gegen eine feste Zahl.
In diesem Fall schreiben wir $P(A) = \lim_{n\to\infty} H_n(A)$ und sprechen von der Wahrscheinlichkeit von A.

Außerdem motiviert Satz 3.11 zu einer genauen Definition:

3.14 Definition (Axiomsystem von Kolmogorov). Sei $(\Omega, \mathcal{A}$ ein messbarer Raum, der ein Zufallsexperiment beschreibe. Dann erfülle eine Abbildungen die Axiome:

(i) Positivität : $0 \leq P(A)$ für alle $A \in \mathcal{A}$

(ii) Normierung: $P(\Omega) = 1$

(iii) σ-Additivität: $A_i \in \mathcal{A}, i \in \mathbb{N} : A_i \cap A_j = \emptyset, i \neq j$ dann gilt: $P(\bigcup_{i=1}^{\infty}) = \sum_{i=1}^{\infty} P(A_i)$.

Diese Axiome implizieren die Abbildung: $P\colon \mathcal{A} \to [0, 1]$.

3.15 Notation. Wir nennen:

- P eine Wahrscheinlichkeitsverteilung
- $P(A)$ als die Wahrscheinlichkeit
- $(\Omega, \mathcal{A}, P)$ einen Wahrscheinlichkeitsraum.

3.16 Satz. Sei $(\Omega, \mathcal{A}, P)$ ein Wahrscheinlichkeitsraum dann gilt:

1) $P(\emptyset) = 0$

2) Für endlich viele Mengen $A_i \in \mathcal{A}, i = 1, \ldots, n \in \mathbb{N}$ paarweise unvereinbar gilt: $P(\bigcup_{i=1}^{n} A_i) = \sum_{i=1}^{n} P(A_i)$

3) $P(A) = 1 - P(\overline{A})$ für alle $A \in \mathcal{A}$

4) Für $A, B \in \mathcal{A}$ gilt: $P(A \cup B) = P(A) + P(B) - P(A \cap B)$

5) Für $A_i \in \mathcal{A}$, paarweise unvereinbar mit $\bigcup_{i=1}^{\infty} A_i = \Omega$ gilt: $\sum_{i=1}^{\infty} P(A_i) = 1$.

Beweis. 1) Setze

$$\begin{cases} A_i = \Omega & \text{, falls } i = 1 \\ A_i = \emptyset & \text{, sonst} \end{cases}$$

Daraus folgt, dass $A_i, i \in \mathbb{N}$ paarweise unvereinbar sind.

$$\implies 1 = P(\Omega) = P(\bigcup_{i=1}^{\infty} A_i) = \sum_{i=1}^{\infty} P(A_i) = 1 + \sum_{i=2}^{\infty} P(\emptyset)$$

$$\implies P(\emptyset) = 0$$

2) Setze $A_i \coloneqq \emptyset$ so folgt dies direkt aus (iii).

3) $A \cup \overline{A} = \Omega$ und $A \cap \overline{A} = \emptyset$ nach $(ii), (iii)$ folgt: $1 = P(\Omega) = P(A) + P(\overline{A})$

4) Wir definieren: $D = A \cap B, C = B \setminus A = B \cap \overline{A}$. Dann impliziert Cm dass $A \cup B = A \cup C$, da A, C unvereinbar sind. Das heißt:

$$P(A \cup B) = P(A) + P(C)$$

Weiter gilt nun:

$$\begin{aligned} &\implies B = C \cup D \\ &\implies P(B) = P(C \cup D) = P(C) + P(D) \\ &\implies P(B) = P(C) + P(A \cap B) \\ &\implies \text{Behauptung} \end{aligned}$$

5) $1 = P(\Omega) = P(\bigcup_{i=1})^n A_i) = \sum_{i=1}^n P(A_i)$

□

3.17 Satz. Sei $(\Omega, \mathcal{A}, P)$ ein Wahrscheinlichkeitsraum.

- Für eine aufsteigende Familie $A_1 \subset A_2 \subset A_3 \subset \ldots \subset A_i \in \mathcal{A}$ gilt:

$$\lim_{n\to\infty} P(A_n) = P(\lim_{n\to\infty} A_n) = P\left(\bigcup_{i=1}^{\infty} A_i\right)$$

- Für eine absteigende Familie $A_1 \supset A_2 \ldots \supset A_i \in \mathcal{A}$ gilt:

$$\lim_{n\to\infty} P(A_n) = P(\lim_{n\to\infty} A_n) = P\left(\bigcap_{i=1}^{\infty} A_i\right)$$

Beweis. Übung. □

Die einfachsten Modelle sind Laplace-Modelle:

3.18 Satz (Laplace-Experiment). Sei $(\Omega, \mathcal{A}, P)$ ein Wahrscheinlichkeitsraum mit $|\Omega| = n \in \mathbb{N}$ und jedes Elementarereignis hat die gleiche Wahrscheinlichkeit: $P(\{\omega\}) = \frac{1}{n} \forall_{\omega\in\Omega}$. Dann ist:

$$P(A) = \frac{m}{n}, \text{ für alle } A \in \mathcal{A} \text{ mit } |A| = m \in \mathbb{N}$$

Beweis. Elementarereignisse sind unvereinbar. Es gilt:

$$P(A) = P(\{w \in \mathcal{A}\}) = \sum_{i=1}^{m} \frac{1}{n} = \frac{m}{n}$$

□

3.19 Bemerkung. Das Laplace-Modell sagt: $P(A) = \frac{\text{Anzahl Elementarereignisse für A}}{\text{Anzahl aller Elementarereignisse}}$

3.20 Beispiel. Das einmalige Werfen eines Würfels hat den Ereignisraum: $\Omega = \{1, \ldots, 5\}$ und ist ein Laplace-Modell.
Für $A = \{\text{durch 3 teilbar}\} = \{3, 6\}$ gilt:

$$P(A) = \frac{|A|}{|\Omega|} = \frac{2}{6} = \frac{1}{3}$$

4. Kombinatorik

Wir bleiben bei endlichen Ereignismengen mit $|\Omega| = n \in \mathbb{N}$ Elementen. Solche Zufallsexperimente können als Urnen mit Kugeln (den Ereignissen) aufgefasst werden. Ein Elementarereignis entspricht den ziehen einer Kugel.

Varianten:

- Werden die entnommenen Kugeln vor dem nächsten Experiment wieder zurückgelegt?
- Ist die Reihenfolge der entnommenen Kugeln relevant?
  - Falls ja: Variationen
  - Sonst: Kombinationen

4.21 Satz. Sei Ω eine Menge mit $|\Omega| = n \in \mathbb{N}$.

1) Die Anzahl aller Kombinationen der Länge m mit zurücklegen ist:

$$|\{\{\omega_1, \ldots, \omega_n\} | \omega_1, \ldots, \omega_n \in \Omega\}| = \binom{n+m-1}{m}$$

wobei $\binom{n}{k} = \frac{n!}{(n-m)!m!}$ für $n \in \mathbb{N}, 0 \leq k < n$.

2) Die Anzahl aller Kombinationen der Länge $m \leq n$ ohne Zurücklegen ist:

$$|\{\{\omega_1, \ldots, \omega_n\} | \omega_1, \ldots, \omega_n \in \Omega\} = \binom{n}{m}$$

Beweis. Ohne Beschränkung der Allgemeinheit: Nehme an, dass die Elemente in Ω von 1 bis n durchnummeriert sind.Zähle Elemente in

$$\{(a_1, \ldots, a_n) | 1 \leq a_1 \leq a_2 \leq \ldots \leq a_m \leq n\} = A$$

Definiere $b_i = a_i + i - 1, i = 1, \ldots, m$. Dies definiert eine Bijektion auf A mit Bild:

$$\{(b_1, \ldots, b_m) | 1 \leq b_1 \leq \ldots \leq b_m \leq n + m - 1\} = B$$

Zähle nun Elemente in B. Bei der Vernachlässigung der Sortierung sind das

$$\frac{(n+m-1)!}{(n-1)!}$$

Elemente. Es interessiert uns nur eines dieser $m!$ Möglichkeiten, daraus folgt die Behauptung.
Der zweite Teil erfolgt Analog zu 1. □

4.22 **Beispiel** (Geburtenparadoxon). Bestimme die Wahrscheinlichkeit, dass zwei Personen im selben Raum am gleichen Tag Geburtstag haben. Dazu sei dies der Modellrahmen:

- $n \in \mathbb{N}$ die Personen im Raum
- keine Geburtstage am 29. Februar
- Alle restlichen 365 Tage im Jahr sind als Geburtstage alle gleich wahrscheinlich.

Demnach handelt es sich um ein La-Place Modell.
Betrachtetes Ereignis:

$$A = \{\text{mindesten 2 Personen haben am gleichen Tag Geburtstag}\}$$

Um diese Wahrscheinlichkeit zu bestimmen, betrachte man das Komplementärereignis:

$$\overline{A} = \{\text{alle Personen haben an verschiedenen Tagen Geburtstag}\}$$

Wir können dies wie folgt bestimmen:

$$P(\overline{A}) = \frac{\text{Anzahl günstiger Ergebnisse}}{\text{Anzahl möglicher Ergebnisse}}$$

Wir bestimmen nun das richtige "Urnenmodell":

- Für $\overline{A}$: Ziehen ohne zurücklegen und Beachtung der Reihenfolge.
- Für Ω : Ziehen mit zurücklegen und Beachtung der Reihenfolge.

Daraus ergibt sich:

$$P(A) - P(\overline{A}) = 1 - \frac{365 \cdot 364 \cdot \ldots \cdot (365-n+1)}{365^n}$$

Wir betrachten ein paar Beispielwerte:

$$\begin{aligned} n = 23 &\implies P(A) \approx 0,507 \\ n = 70 &\implies P(A) \approx 0,999 \\ n = 160 &\implies P(A) \approx 1 \end{aligned}$$

Bedingte Wahrscheinlichkeiten und Unabhängigkeit

5. Bedingte Wahrscheinlichkeit

Betrachte die Ereignisse $A, B, A \cap B$.
Absolute Häufigkeit: $h_n(A)$
Relative Häufigkeit: $H_n(A) = \frac{h_n(A)}{n}$
Wie oft tritt A ein, unter Bedingung, dass B schon eingetreten ist?

5.1 Notation. Wir schreiben $A|B$ für "A gegeben B"

$$H_n(A|B) = \frac{h_n(A \cap B)}{h_n(B)} = \frac{H_n(A \cap B)}{H_n(B)}$$

5.2 Definition (bedingte Wahrscheinlichkeit). Sei $(\Omega, \mathcal{A}, P)$ ein Wahrscheinlichkeitsraum und $A, B \in \mathcal{A}$ mit $P(B) > 0$. Die *bedingte Wahrscheinlichkeit* von A gegeben B ist definiert als :

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

5.3 Beispiel. Würfle zwei mal mit einem idealisierten Würfel und betrachte die Ereignisse:

$$A = \{\text{Beim ersten Wurf wird eine 6 gewürfelt}\}$$
$$B = \{\text{die Summe beider Würfel ist 8}\}$$

Betrachte dazu die Tabelle:

$$\implies P(B) = \frac{5}{36}, P(A \cap B) = \frac{1}{36}, P(A|B) = \frac{1}{5}$$

Dies ist Konsistent zur Definition.

Faustregel: Die Rechenregel für bedingte Wahrscheinlichkeit sind die für Wahrscheinlichkeit, falls das zweite Argument fest gelassen wird.

5.4 Satz. Sei $(\Omega, \mathcal{A}, P)$ ein Wahrscheinlichkeitsraum und $A, B, C \in \mathcal{A}$ mit $P(C) > 0$ Dann gilt:

a) $P(C|C) = 1$

b) $P(A|C) = 1 - P(\overline{A}|C)$

c) $P(A \cup B|C) = P(A|C) + P(B|C) - P(A \cap B|C)$

Beweis. Wir zeigen:

a) $P(C|C) = \frac{P(C \cap C)}{P(C)} = \frac{P(C)}{P(C)} = 1$

b) Beobachte: $A \cup \overline{A} = \Omega \implies A, \overline{A}$ unvereinbar.

$$\implies (A \cap C) \cup (\overline{A} \cap C) = C$$

Ebenfalls unvereinbar.

$$\begin{aligned} \implies 1 = \frac{P(C)}{P(C)} &= \frac{P((A \cap C) \cup (\overline{A} \cap C)}{P(C)} \\ &= \frac{P(A \cap C) + P(\overline{A} + C)}{P(C)} \\ &= P(A|C) + P(\overline{A}|C) \end{aligned}$$

c)

$$\begin{aligned} P(A \cup B|C) &= \frac{P((A \cup B) \cap C)}{P(C)} \\ &= \frac{P((A \cap C) \cup (B \cap C)) \cup}{P(C)} \\ &= \frac{P(A \cap C) + P(B \cap C) - P((A \cap C) \cap (B \cap C))}{P(C)} \\ &= P(A|C) + P(B|C) - P(A \cap B|C) \end{aligned}$$

□

5.5 Beispiel (Stapelsuchproblem). Durchsuche 7 gleich große Bücherstapel nach einem Buch. Die Wahrscheinlichkeit, dass das Buch überhaupt in den Stapeln ist, ist 0.8. Angenommen wir haben die ersten 6 Stapel durchsucht und das Buch nicht gefunden. Wie groß ist die Wahrscheinlichkeit, dass das Buch im letzten Stapel ist?
Ereignisse:

$$A_i = \{\text{Buch ist im i-ten Stapel}\}, i = 1, \ldots, 7$$

mit $P(A_i) = p \in (0, 1)$ unbekannt.
Wir wissen:

$$\begin{aligned} 0.8 &= P(A_1 \cup \ldots \cup A_7) \\ &= P(A_1) + \ldots + P(A_7) \\ &= 7p \\ \implies p &= \frac{0.8}{7} \approx 0.114 \end{aligned}$$

$$\begin{aligned} P(A_7|\overline{A_1 \cup \ldots \cup A_6}) &= P(A_7|\overline{A_1} \cap \ldots \cap \overline{A_6}) \\ &= \frac{P(A_7}{1 - P(A_1 \cup \ldots \cup A_6} \\ &= \frac{p}{1-6p} \\ &\approx 0.3636 \end{aligned}$$

6. Multiplikationsregeln

Umstellen der Definition der bedingten Wahrscheinlichkeit liefert:

$$P(A \cap B) = \begin{cases} P(A|B) \cdot P(B) \\ P(B|A) \cdot P(A) \end{cases}$$

6.6 Beispiel (Notenverteilung). Es sei:

- 70 Prozent aller Studierenden haben eine Note, die besser als 3 ist
- Davon sind 25 Prozent, die eine bessere Note als 2 haben.

Berechne die Wahrscheinlichkeit, dass eine beliebige Person in der Menge mit einer besseren Note als 2 abschließt. Ereignisse:

$$\begin{aligned} A &= \{\text{Note} \leq 2\} \\ B &= \{\text{Note} \leq 3\} \end{aligned}$$

Nach der Multiplikationsregel:

$$P(A) = P(A \cap B) = P(A|B) \cdot P(B) = 0.175$$

6.7 Satz (erweiterte Multiplikationsregel). Sei $(\Omega, \mathcal{A}, P$ ein Wahrscheinlichkeitsraum und $A_1, \ldots, A_n \in \mathcal{A}$ mit $P(A_1 \cap \ldots \cap A_{n-1} > 0$. Dann gilt

$$P(A_1 \cap \ldots \cap A_n) = P(A_1) \cdot P(A_2|A_1) \cdot P(A_3|A_1 \cap A_2) \ldots P(A_n|A_1 \cap \ldots \cap A_{n-1})$$

Beweis. Induktion. □

Die erweiterte Multiplikationsregel erlaubt die Darstellung eines mehrstufigen Zufallsexperiment als Baumdiagramm.

6.8 Beispiel (Ziegenproblem). Wähle eine aus drei Türen. Hinter einer der Türen befindet sich ein Auto, hinter den anderen beiden nur Ziegen. Nun öffnet ein anderer eine der Türen, in dem das Auto nicht steht.
Man darf sich nun erneut entscheiden. Ist dies Vorteilhaft?

6.9 Satz (Satz von der totalen Wahrscheinlichkeit). Sei $(\Omega, \mathcal{A}, P)$ ein Wahrscheinlichkeitsraum und $B_1, B_2, \ldots$ A eine Zerlegung von Ω das heißt

$$\bigcup_{i=1}^{\infty} B_i \Omega \text{ mit } B_i \cap B_j, i \neq j.$$

Dann gilt für ein beliebiges Ereignis $A \in \mathcal{A}$, dass

$$P(A) = \sum_{i=1}^{\infty} P(A|B_i)P(B_i).$$

Dabei werden alle Terme mit $P(B_i) = 0$ weggelassen.

Beweis. Rechne:

$$A = A \cap \Omega = A \cap (\bigcup_{i=1}^{\infty} B_i) = \bigcup_{i=1}^{\infty} (A \cap B_i).$$

Es gilt weiter:

$$\begin{aligned} P(A) = P(\bigcup_{i=1}^{\infty}) &= \sum_{i=1}^{\infty} P(A \cap B_i) \\ &= \sum_{i=1}^{\infty} P(A|B_i)P(B_i). \end{aligned}$$

□

6.10 Satz (Bayesche Formel). Mit den gleichen Voraussetzungen wie in Satz 6.9 gilt die Bayesche Formel:

$$P(B_j|A) = \frac{P(A|B_j)P(B_j)}{P(A)} = \frac{P(A|B_j)P(B_j)}{\sum_{i=1}^{\infty} P(A|B_i)P(B_i)}, j \in \mathbb{N}.$$

Beweis. Mittels der Multiplikationsregeln und dem Satz der totalen Wahrscheinlichkeit folgt die Behauptung direkt. □

6.11 Beispiel. Betrachte die Statisk einer Netzbetreibers für Störungen innerhalb eines Jahres:

- 50 % aller Störungen sind an Switches
- 30 % aller Störungen sind an Routern
- 20 % aller Störungen sind nicht klar zugeordnet

Kurzfristig können gelöst werden:

- 50 % der Störungen an Switches
- 30 % der Störungen an Routern
- 5 % der Störungen von den nicht klar zugeordneten Störungen.

Wie viel Prozent aller Störungen können spontan gelöst werden? Wir betrachten die Ereignisse:

$$B_1 = \{\text{Störung an einem Switch}\}$$
$$B_2 = \{\text{ Störungen an einem Router }\}$$
$$B_3 = \{\text{sonstige Störung}\}.$$

Die Wahrscheinlichkeiten dazu sind:

$$P(B_1) = 0.5$$
$$P(B_2) = 0.3$$
$$P(B_3) = 0.2$$

mit $B_1 \cup B_2 \cup B_3 = \Omega$. Außerdem ist $A = \{\text{Störung konnte spontan behoben werden}\}$. Nach dem Satz der totalen Wahrscheinlichkeit gilt:

$$\begin{aligned} P(A) &= P(A|B_1)P(B_1) + P(A|B_2)P(B_2) + P(A|B_3)P(B_3) \\ &= 0.5 \cdot 0.5 + 0.3 \cdot 0.3 + 0.05 \cdot 0.2 \\ &\implies 35\% \text{ aller Störungen konnten spontan Behoben werden.} \end{aligned}$$

7. Stochastische Unabhängigkeit

7.12 Definition. Sei $(\Omega, \mathcal{A}, P)$ Ein Wahrscheinlichkeitsraum. $A, B \in \mathcal{A}$ zwei Ereignisse sind stochastisch unabhängig falls gilt:

$$P(A \cap B) = P(A)P(B).$$

7.13 Bemerkung.

$$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{P(A)P(B)}{P(B)} = P(A)$$

mit dem Analog $P(B|A) = P(B)$ sind Äquivalent.

7.14 Satz. Falls A und B stochastisch unabhängig sind, sind auch die Ereignisse $\overline{A}$ und $\overline{B}$, A und $\overline{B}$ stochastisch unabhängig.

Beweis.

$$\begin{aligned} P(A \cap \overline{B}) &= P(A) - P(A \cap B) = P(A) - P(A)P(B) \\ &= P(A)(1 - P(B)) = P(A)P(\overline{B}). \end{aligned}$$

Der Rest erfolgt Analog. □

7.15 Definition. Sei $(\Omega, \mathcal{A}, P)$ ein Wahrscheinlichkeitsraum. Die Familie von Ereignissen $A_1, \ldots \in \mathcal{A}$ heißt stochastisch unabhängig, wenn für jede endliche Auswahl von Mengen von Ereignissen $A_{i_1}, \ldots, A_{i_n}$ gilt, dass :

$$P(A_{i_1)\cap \ldots \cap A_{i_n}}) = P(A_{i_1}) \cdot \ldots \cdot P(A_{i_n}).$$

7.16 **Bemerkung.** Engliche Familien $A_1, \ldots, A_n$ von Ereignissen sind ebenfalls abgedeckt setze $A_i = \Omega$ für $i > 1$. Es genügt nicht nur $P(\bigcup_{i=1}^{\infty} A_i) = \prod_{i=1}^{\infty} P(A_i)$ zu fordern. Insbesondere können die Ereignisse $A_1, A_2, \ldots$ paarweise stochstische unabhängig sein, ohne dass die ganze Familiie stochastisch unabhängig ist.

7.17 **Beispiel.** Sei $\Omega = \{1, 2, 3, 4\}$ ein Laplace-Modell, das heißt $P(\{1\}) = P(\{2\}) \ldots = \frac{1}{4}$. Wir betrachten die Ereignisse:

$$A = \{1, 1\}, B = \{1, 3\}, C = \{2, 2\}$$
$$\implies P(A) = P(B) = P(C) = \frac{1}{2}.$$

Wir betrachten die Wahrscheinlichkeiten:

$$P(A \cap B) = P(\{1\}) = \frac{1}{4} = P(A)P(B)$$
$$P(A \cap C) = P(\{2\}) = \frac{1}{4} = P(A)P(C).$$

$\implies A, B, C$ sind paarweise stochastisch unabhängig.

Aber: $P(A \cap B \cap C) = P(\emptyset) = 0$ und somit $P(A)P(B)P(C) = \frac{1}{8}$.

7.18 **Proposition.** Für stochastisch unabhängige Ereignisse gilt:

$$P(\bigcup_{i=1}^{\infty} A_i) = 1 - \prod_{i=1}^{\infty} P(\overline{A_i}).$$

8. Punktexperimente

Frage: Wie können wir das Hintereinanderausführen von Zufallsexperimenten beschreiben? Betrachte Wahrscheinlichkeitsräume $(\Omega_i, \mathcal{A}_i, P_i)$, $i = 1, \ldots, n$.

Idee: Kopple die Wahrscheinlichkeitsräume stochastisch unabhängig $\implies \Omega = \Omega_1 \times \ldots \times \Omega_n$

Frage: Wie sieht eine passende σ-Algebra aus?. Der kanonische Kanditat

$$Q = \{A_1 \times \ldots \times A_n | A_i \in \mathcal{A}, i = 1, \ldots, n\}$$

ist keine σ-Algebra.

8.19 **Satz.** Sei Ω eine Ereignismenge und $S \subset \mathcal{P}(\Omega)$. Dann ist $\sigma(S) = \bigcap_{S \subset A \subset \mathcal{P}} \mathcal{A}$ die kleinste σ-Algebra mit $S \subset \sigma(S)$. Sie wird die von S erzeugte σ-Algebra genannt.

Beweis. Betrachte

$$M = \{S \subset \mathcal{A} \subset \mathcal{P} | \mathcal{A} \text{ ist } \sigma\text{-Algebra}\}.$$

Dann ist $\sigma(S)$ wohldefiniert. Überprüfe die Bedingungen einer σ-Algebra :

a) $\Omega \in \mathcal{A} \forall_{A \in M} \implies \Omega \in \sigma(S)$

b)

$$\begin{aligned} A \in \sigma(S) &\implies A \in \mathcal{A}, \forall_{A \in M} \\ &\implies \overline{A} \in \mathcal{A} \\ &\implies \overline{A} \in \sigma(S) \end{aligned}$$

c) Sei $A_i \in \sigma(S), i \in \mathbb{N}$.

$$\begin{aligned} &\implies A_i \in \mathcal{A} \\ &\implies \bigcup_{i=1}^{\infty} A_i \in \mathcal{A} \\ &\implies \bigcup_{i=1}^{\infty} A_i \in \sigma(S) \end{aligned}$$

Daraus folgt, dass $\sigma(S)$ eine σ-Algebra ist. Nach Definition sogar die kleinste. $\square$