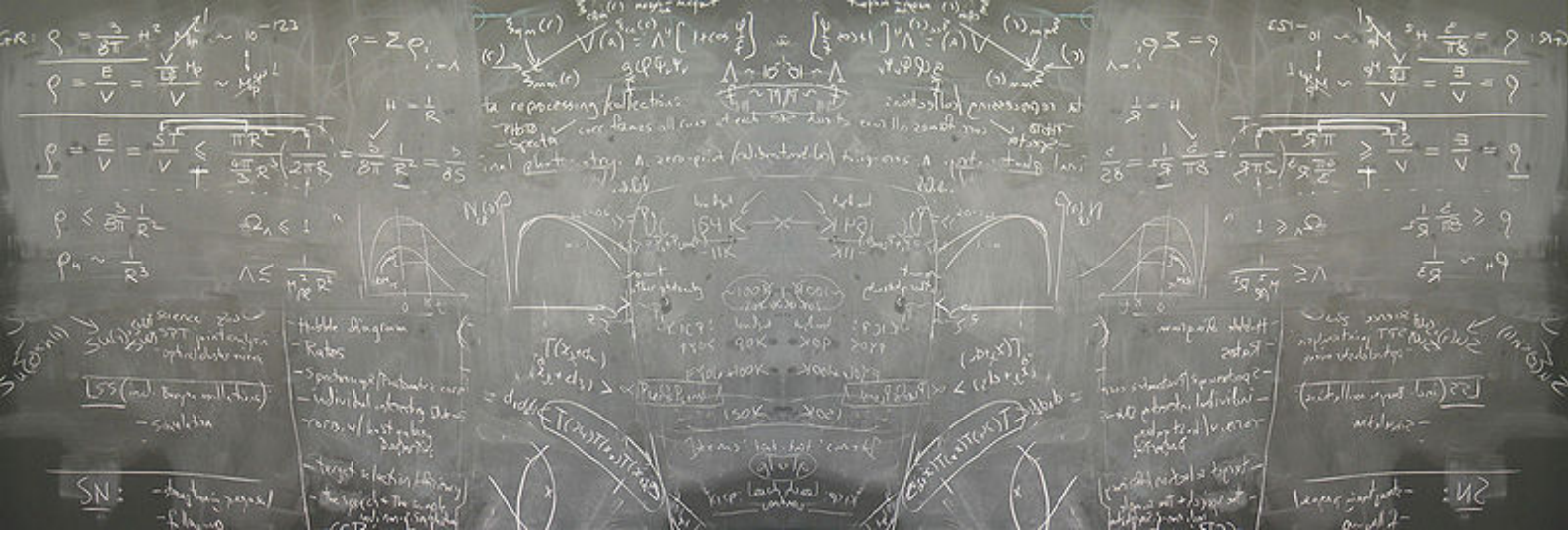
Aufgaben zur Mathematik für ADS

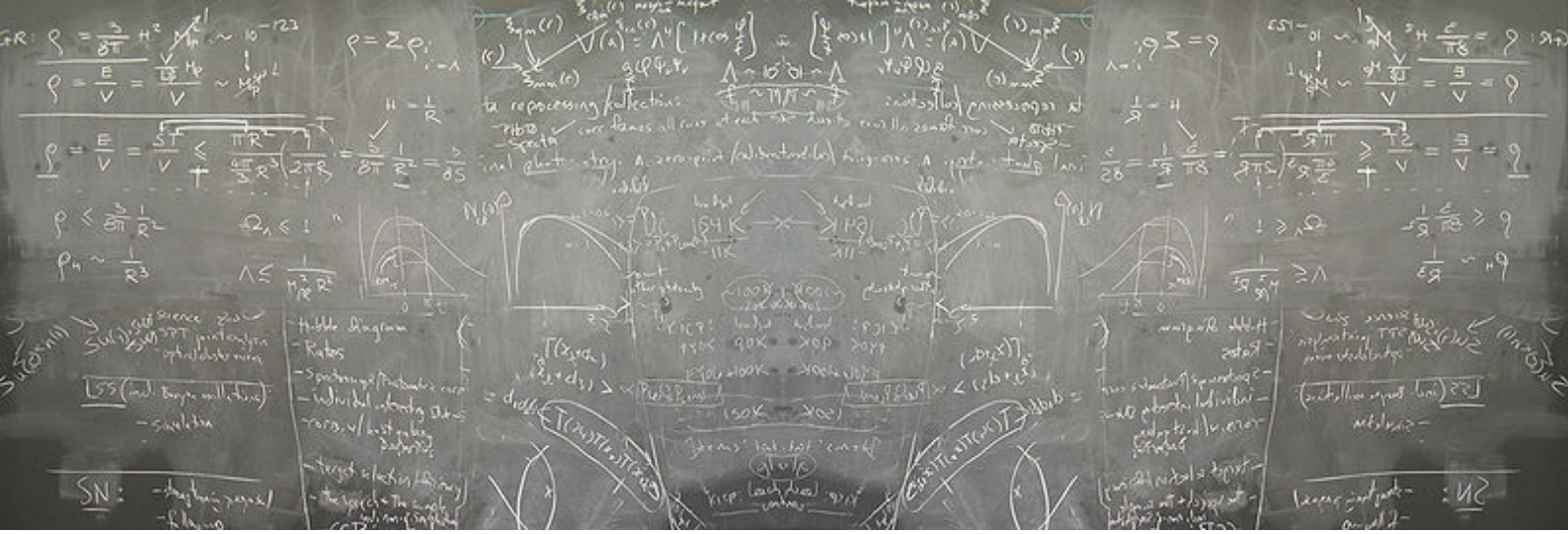
Formales Denken für angehende AlgorithmikerInnen

Sebastian Berndt
Alexandra Lassota

Stand: 1. Dezember 2020, 08:58:59 Uhr



Inhaltsverzeichnis



Herzlich Willkommen!

Herzlich Willkommen zur kurzen Mathe-Einführung, besonders geeignet für Zwei-Fach-Studierende, die im nächsten Semester *Algorithmen und Datenstrukturen (ADS)* hören werden. Wir werden versuchen, im Laufe des Semesters die dafür nötigen formalen Grundlagen der Hochschul-Mathematik zu entwickeln. Richtig Mathematik lernen kann man (wie auch Programmieren, Schwimmen oder Feuerbälle werfen) nur durch reichlich Übung. Daher werden wir die kostbare Vorlesungszeit nicht dafür nutzen, Ihnen irgendwelche abstrakten Vorträge zu halten. Ganz im Gegenteil: Wir werden diese gemeinsame Zeit sinnvoll nutzen und versuchen, so viele Aufgaben wie möglich zu bearbeiten. Dafür müssen Sie sich natürlich im Vorhinein mit dem Stoff beschäftigt haben. Also gibt es zu jedem Themengebiet eine Kapitel-Auswahl, die *vor dem Vorlesungstermin* von Ihnen gelesen und verstanden werden sollte. Wir klären zwar zu Anfang der Vorlesung etwaige Fragen, wollen uns aber so schnell wie möglich auf die Aufgaben stürzen.

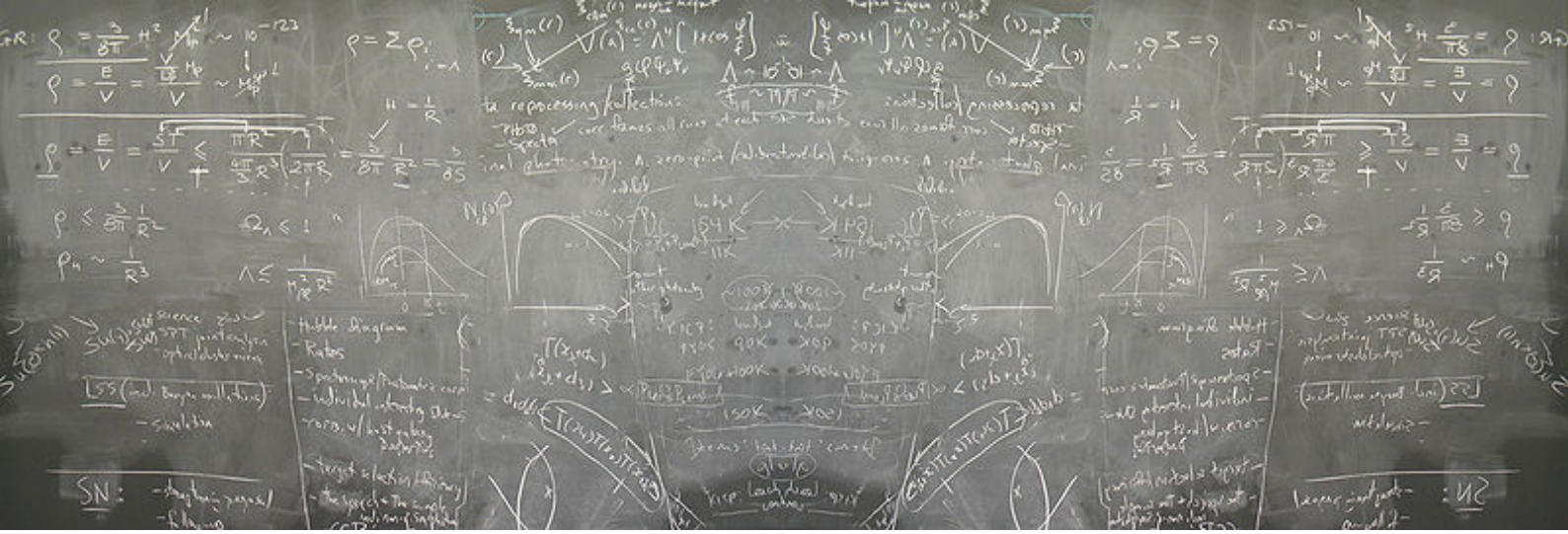
Über Hinweise und Verbesserungsvorschläge freuen wir uns natürlich immer sehr.

Im Text gibt es an einigen Stellen Kommentare. Diese sehen wie folgt aus und geben ergänzende Informationen.

Zum Ablauf im Wintersemester 2020/2021

Im Verlaufe des Semesters werden wir uns an sieben Terminen treffen: 19./20.11., 26./27.11., 3./4.12., 10./11.12., 17./18.12., 07./08.01., 14./15.01.

Am Ende des Semesters wird es eine Prüfung über die Inhalte der Veranstaltung geben. Diejenigen, die diese Prüfung bestehen, erhalten dann in *Algorithmen und Datenstrukturen* eine inhaltlich reduzierte Klausur. Für die Prüfungs gibt es keine formale Teilnahmevoraussetzung, aber wir empfehlen stark, an allen Terminen teilzunehmen.



1. Aussagen und Mengen

Zum Lesen:
Kapitel 2.1, 2.2, 2.4 in [Wol17].

Stichworte:

- Sachverhalt: Kontext; Wahrheitswert
- Definition: Begriffsbildung; *Textersetzung*
- Menge: Urelemente; Zusammenfassung von bestimmten, wohlunterschiedenen Objekten; Leere Menge $\emptyset$; Explizite Darstellung $\{0,1,2,3\}$; Deskriptive Darstellung $\{x \mid x \in \mathbb{N}, x \leq 3\}$; Mengen, die Mengen enthalten; $\in$-Symbol; Operatoren $\cup$, $\cap$, $\setminus$

Definition 1.1 Wir definieren uns ein paar sehr nützliche Hilfsmengen:

- Die Menge der natürlichen Zahlen $\mathbb{N}_0 = \{0,1,2,\ldots\}$[a].
- Die Menge der ganzen Zahlen $\mathbb{Z} = \{0,-1,1,-2,2\} = \mathbb{N}_0 \cup \{-x | x \in \mathbb{N}_0\}$.
- die Menge der rationalen Zahlen $\mathbb{Q} = \{p/q \mid p,q \in \mathbb{Z}\}$.
- die Menge der reellen Zahlen $\mathbb{R}$.

[a] Ob die 0 eine natürliche Zahl ist, führt immer zu Diskussionen, insbesondere zwischen MathematikerInnen und InformatikerInnen. Um solchen Problemen aus dem Weg zu gehen, schreiben wir häufig $\mathbb{N}_0$ um klar zu machen, dass im Informatik-nahen Kontext die 0 üblicherweise eine natürliche Zahl ist.

Aufgaben

Aufgabe 1.1 — Vorrechnen. Wir betrachten die beiden Mengen $M = \{x \mid x \in \mathbb{N}_0, 2^x \leq 10\}$ und $N = \{y \mid y \in \mathbb{N}_0, \text{Es gibt } z \in \mathbb{N}_0 \text{ mit } y = z^2 \text{ und } z \leq 5\}$. Geben Sie die beiden Mengen M und N sowie $M \cup N$, $M \cap N$ und $M \setminus N$ explizit an, d. h. durch die in Mengenklammern eingeschlossene Aufzählung ihrer Elemente. ■

Beweis. Wir gehen durch die natürlichen Zahlen x durch, um zu sehen, welche von ihnen zu M gehören. Dabei sehen wir, dass $M = \{0,1,2,3\}$ gilt, da $2^x > 10$ für alle $x \geq 4$ gilt.

Wir gehen durch die natürlichen Zahlen y durch, um zu sehen, welche von ihnen zu N gehören. Dabei sehen wir, dass $N = \{0,1,4,9,16,25\}$ gilt, da wir entsprechende $y = 0,1,2,3,4,5$ einsetzen

können.

Es gilt $M \cup N = \{0,1,2,3,4,9,16,25\}$, $M \cap N = \{0,1\}$ und $M \setminus N = \{2,3\}$. ■

Aufgabe 1.2 — Vorrechnen. Welche der folgenden Sätze sind Sachverhalte?

(i) Die Stadt Kiel liegt am 01.01.2019 in Schleswig-Holstein.
(ii) Die Stadt Kiel liegt am 01.01.2019 im Freistaat Bayern.
(iii) Mathe ist schwer.
(iv) Sebastian (ein Autor dieser Unterlagen) findet, dass Mathe schwer ist.
(v) $x^2 = 4$.
(vi) $42 = 2 \cdot 3 \cdot 7$.
(vii) $43 = 2 \cdot 3 \cdot 7$.
(viii) $43 \in 0$.

Beweis. (i) Dies ist ein Sachverhalt, der wahr ist.
(ii) Dies ist ein Sachverhalt, der falsch ist.
(iii) Dies ist kein Sachverhalt, weil wir keinen Wahrheitswert zuordnen können.
(iv) Dies ist ein Sachverhalt (der wahr ist).
(v) Dies ist kein Sachverhalt, da der Wahrheitswert vom Wert von x (also einem Kontext) abhängt.
(vi) Dies ist ein Sachverhalt, der wahr ist.
(vii) Dies ist ein Sachverhalt, der falsch ist.
(viii) Dies ist kein Sachverhalt, da die Aussage nicht klar definiert ist.

■

Aufgabe 1.3 Sei $n \in \mathbb{N}_0$ mit $n > 0$. Spezifizieren Sie deskriptiv eine Menge T_n, die genau die positiven ganzzahligen Teiler von n enthält, d .h. jede Zahl $i \in \mathbb{N}_0$ mit $i > 0$, für die $n/i \in \mathbb{N}_0$ gilt.

Beweis. $T_n = \{i \mid i \in \mathbb{N}_0, n/i \in \mathbb{N}_0, i > 0\}$

Wichtig ist hier, dass die Zahl n schon gebunden ist. Wir sollen solch eine Menge ja für alle $n \in \mathbb{N}_0$ konstruieren. ■

Aufgabe 1.4 Für eine Menge M ist $\mathcal{P}(M)$ die *Potenzmenge* von M, also die Menge, die alle Teilmengen von M enthält: $\mathcal{P}(M) = \{N \mid N \subseteq M\}$.

(i) Geben Sie $\mathcal{P}(\{1,2,3\})$ an.
(ii) Geben Sie $\mathcal{P}(\{\{1,2,3\}\})$ an.

Beweis. (i) Es gilt $\mathcal{P}(\{1,2,3\}) = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1,2\}, \{1,3\}, \{2,3\}, \{1,2,3\}\}$.
(ii) Es gilt $\mathcal{P}(\{\{1,2,3\}\}) = \{\emptyset, \{\{1,2,3\}\}\}$.

■

Aufgabe 1.5 Für welche der folgenden Mengen A und B gilt $A = B$?

(i) $A = \{1,2,3\}$, $B = \{1,3,2\}$
(ii) $A = \{x \mid x \in \mathbb{N}_0, x^2 + x = 2\}$, $B = \{1,-2\}$
(iii) $A = \{x \mid x \in \mathbb{R}, x^2 + x = 2\}$, $B = \{1,-2\}$

Beweis. Es gilt
(i) $A = B$, denn die Reihenfolge der Elemente spielt keine Rolle bei Mengen.

(ii) $A \neq B$, denn B enthält -2, aber -2 ist keine nicht-negative Zahl.
(iii) $A = B$, denn 1 und -2 sind die eindeutigen Lösungen von $x^2 + x = 2$.

■

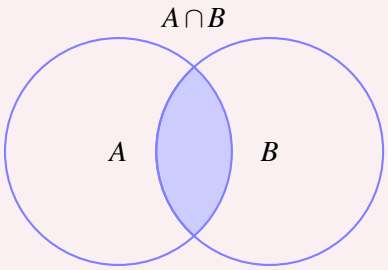
Aufgabe 1.6 Welche der folgenden Aussagen sind wahr?
(i) $\emptyset = \{0\}$
(ii) $x \in \{x\}$
(iii) $\emptyset = \{\emptyset\}$
(iv) $\emptyset \in \{\emptyset\}$
(v) $\emptyset \in \emptyset$

■

Beweis. Es gilt
(i) Falsch, die linke Menge $\emptyset$ enthält kein Element, die rechte Menge $\{0\}$ enthält die 0.
(ii) Wahr für alle x.
(iii) Falsch, die linke Menge $\emptyset$ enthält kein Element, die rechte Menge $\{\emptyset\}$ enthält die Menge $\emptyset$.
(iv) Wahr (und ein Spezialfall von $x \in \{x\}$ für $x = \emptyset$).
(v) Falsch, denn $\emptyset$ enthält kein Element.

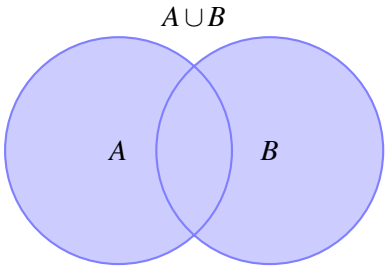
■

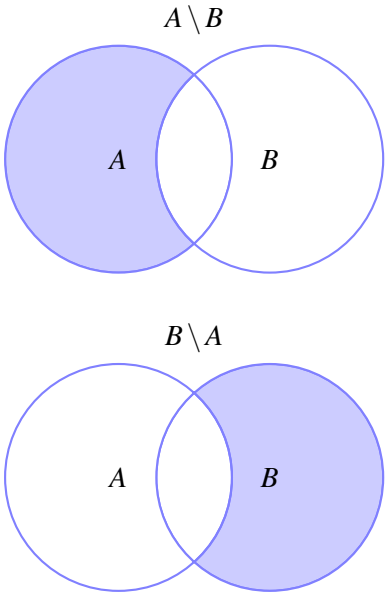
Aufgabe 1.7 Ein *Venn-Diagramm* kann genutzt werden, um zwei oder drei Mengen zu visualisieren. Zum Beispiel zeigt das folgende Bild die Schnittmenge $A \cap B$ für zwei beliebige Mengen A und B:



Zeichnen Sie Venn-Diagramme für die Mengen $A \cup B$, $A \setminus B$ und $B \setminus A$. ■

Beweis.





■

Aufgabe 1.8 Wir betrachten das folgende Zwei-SpielerInnen-Spiel: Gegeben ist eine endliche Menge M von natürlichen Zahlen. Abwechselnd darf nun jeder Spielende eine Teilmenge M' von M wählen, wobei gelten muss, dass (a) $M' \neq M$, (b) $M' \neq \emptyset$ und (c) M' darf keine Obermenge einer bisher gewählten Menge sein. Wurde also zum Beispiel $\{1,3\}$ bereits gewählt, dürfen $\{1,3,4\}$ oder $\{1,2,3\}$ nicht mehr gewählt werden. Der erste Spieler/ die erste Spielerin, der/die keine solche Menge M' mehr wählen darf, verliert.

Hier ein kurzes Beispiel für den Ablauf mit $M = \{1,2,3\}$:

1. SpielerIn 1 wählt nun als Menge $M' = \{1,3\}$.
2. SpielerIn 2 wählt $M' = \{2,3\}$, da dies keine Obermenge von $\{1,3\}$ ist.
3. SpielerIn 1 wählt die Menge $M' = \{1\}$, die keine Obermenge von $\{1,3\}$ oder $\{2,3\}$ ist.
4. SpielerIn 2 wählt die Menge $\{2\}$, die keine Obermenge von $\{1,3\}$, $\{2,3\}$ oder $\{1\}$ ist.
5. Nun kann SpielerIn 1 nur noch die $\{3\}$ wählen, die keine Obermenge von $\{1,3\}$, $\{2,3\}$, $\{1\}$ oder $\{2\}$ ist.
6. SpielerIn 2 verliert nun, da er keine gültige Menge mehr wählen kann.

Enthält M nur ein Element, gewinnt SpielerIn 2, denn SpielerIn 1 hat keinen gültigen Zug. Enthält M zwei Elemente, gewinnt auch SpielerIn 2: SpielerIn 1 muss irgendeine ein-elementige Menge, SpielerIn 2 nimmt die andere ein-elementige Menge und SpielerIn 1 kann keinen Zug durchführen.

Welcher Spielende gewinnt, wenn $|M| = 3$? ■

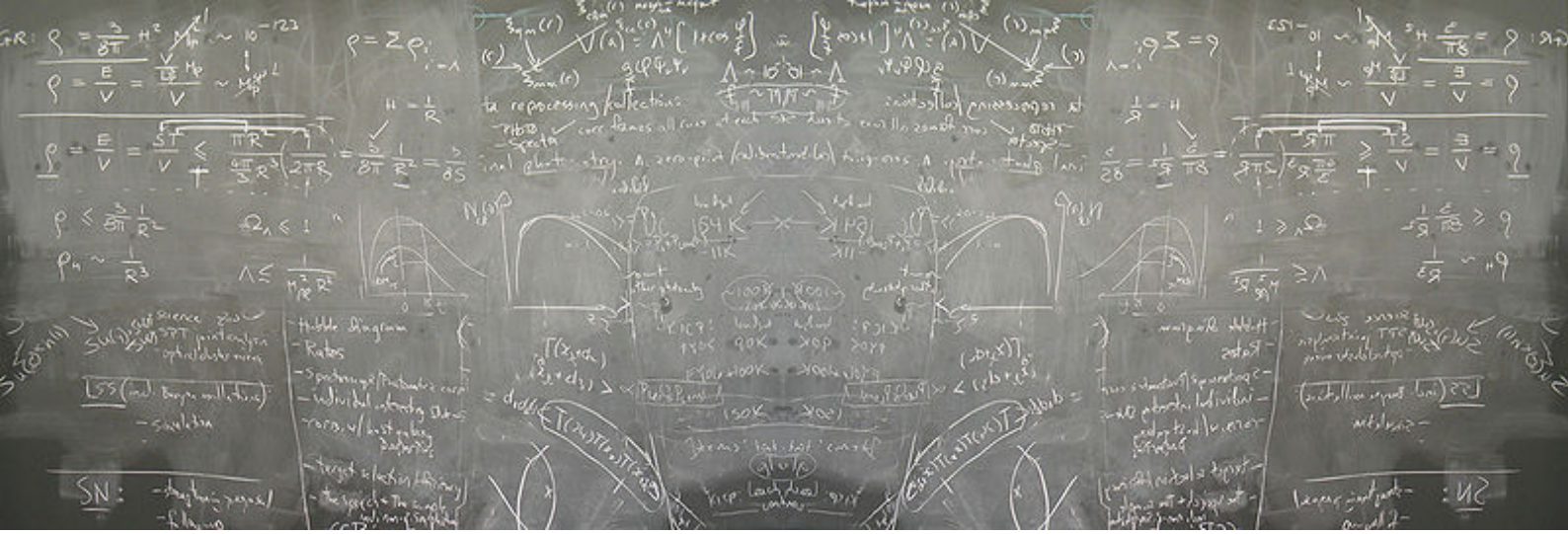
Beweis. Sei $M = \{a,b,c\}$. SpielerIn 1 hat zwei Möglichkeiten: Er/ Sie kann eine einelementige Menge oder einer Menge mit zwei Elementen wählen.

- Wählt er/sie eine einelementige Menge (sagen wir $\{a\}$), so wählt SpielerIn 2 die zweielementige Menge, die die anderen Elemente enthält (also $\{b,c\}$). Nun sind wir in der Situation von $|M| = 2$, denn SpielerIn 1 kann nur noch $\{b\}$ oder $\{c\}$ wählen und SpielerIn 2 nimmt die jeweils andere Menge.
- Wählt er/sie eine zweielementige Menge (sagen wir $\{a,b\}$), so wählt SpielerIn 2 die einelementige Menge, die das andere Element enthält (also $\{c\}$). Nun sind wir in der Situation von $|M| = 2$, denn SpielerIn 1 kann nur noch $\{a\}$ oder $\{b\}$ wählen und SpielerIn 2 nimmt die jeweils andere Menge.

In jedem Fall gewinnt SpielerIn 2. ■

Lernziele:

- Grundlegendes Verständnis für das Konzept einer Menge
- Sicherer Umgang mit expliziter und deskriptiver Mengendarstellung
- Sicherer Umgang mit grundlegenden Mengenoperationen wie $\cup$, $\cap$ und $\setminus$
- Beurteilen können, wann es sich um einen Sachverhalt handelt



2. Beweise und Aussagenlogik

Zum Lesen:
Kapitel 2.3, 2.5, 2.6 in [Wol17].

Stichworte:

- Beweis: Zwingende Argumentation; *Akt der Kommunikation*; Annahme und Beweisverpflichtung; Finden (VerfechterIn); Verifizieren (SkeptikerIn)
- Zusammenfügung von Aussagen
- Kontext und Wahrheitsgehalt; Extensionalität; Verschiedene Welten
- Operatoren: *und*, *oder*, *nicht*, *entweder-oder*, *genau-dann-wenn*, *wenn-dann*
- Wahrheitstabellen

Definition 2.1 Die Operationen im Buch sind häufig auch durch andere Symbole dargestellt:

- Die *oder*-Verknüpfung "or" schreibt man auch als $\vee$, also
- Die *und*-Verknüpfung "and" schreibt man auch als $\wedge$.
- Die *nicht*-Verknüpfung "not" schreibt man auch als $\neg$.
- Die *entweder-oder*-Verknüpfung "xor" schreibt man auch als $\oplus$.
- Die *genau-dann-wenn*-Verknüpfung schreibt man auch als $\Leftrightarrow$.
- Die *wenn-dann*-Verknüpfung schreibt man auch als $\Rightarrow$.

Aufgabe 2.1 — Vorrechnen. Zeigen Sie, dass $\sqrt{3} \notin \mathbb{Q}$, also irrational ist. ∎

Beweis. Angenommen, dass $\sqrt{3} \in \mathbb{Q}$. Also gibt es ganze Zahlen $p, q \in \mathbb{Z}$, so dass $p/q = \sqrt{3}$. Die eindeutige Primfaktorisierung von p und q impliziert, dass es ganze Zahlen $a, b \in \mathbb{Z}$ und natürliche Zahlen $i, j \in \mathbb{N}_0$ gibt mit

(i) $p = a \cdot 3^i$,
(ii) $q = b \cdot 3^j$,
(iii) a ist nicht durch 3 teilbar und
(iv) b ist nicht durch 3 teilbar.

Nach Definition von $\sqrt{3}$, wir haben

$$3 = \sqrt{3} \cdot \sqrt{3} = (\sqrt{3})^2 = (p/q)^2 = \left(\frac{a \cdot 3^i}{b \cdot 3^j}\right)^2 = \frac{a^2 \cdot 3^{2i}}{b^2 \cdot 3^{2j}}.$$

Multiplizieren mit $b^2 \cdot 3^{2j}$ gibt uns $b^2 \cdot 3^{2j+1} = a^2 \cdot 3^{2i}$. Da a und b nicht durch 3 teilbar sind, sind auch a^2 und b^2 nicht durch 3 teilbar. Die Primfaktorzerlegung von $a^2 \cdot 3^{2i}$ enthält also genau $2i$ Dreien. Analog enthält die Primfaktorzerlegung von $b^2 \cdot 3^{2j+1}$ also genau $2j+1$ Dreien. Da aber $a^2 \cdot 3^{2i} = b^2 \cdot 3^{2j+1}$ gilt, muss somit $2i = 2j+1$ gelten. Da $2i$ gerade und $2j+1$ ungerade ist, haben wir einen Widerspruch erreicht. Somit gilt $\sqrt{3} \notin \mathbb{Q}$.

Ab hier folgt die schematische Darstellung des Beweises. Diese ist *nicht* Teil des Beweises, sondern hilft nur beim Verständnis.

1.

Annahme	Beweisverpflichtung
	$\sqrt{3}$ ist nicht rational

2.

Annahme	Beweisverpflichtung
$\sqrt{3}$ ist rational	Widerspruch

3.

Annahme	Beweisverpflichtung
Es gibt $p,q \in \mathbb{Z}$ mit $p/q = \sqrt{3}$	Widerspruch

4.

Annahme	Beweisverpflichtung
Es gibt $p,q \in \mathbb{Z}$ mit $p/q = \sqrt{3}$	Widerspruch
Es gibt $a,b \in \mathbb{Z}$ und $i,j \in \mathbb{N}_0$ mit (i) $p = a \cdot 3^i$, (ii) $q = b \cdot 3^j$, (iii) a ist nicht durch 3 teilbar und (iv) b ist nicht durch 3 teilbar	

5.

Annahme	Beweisverpflichtung
Es gibt $p,q \in \mathbb{Z}$ mit $p/q = \sqrt{3}$	Widerspruch
Es gibt $a,b \in \mathbb{Z}$ und $i,j \in \mathbb{N}_0$ mit (i) $p = a \cdot 3^i$, (ii) $q = b \cdot 3^j$, (iii) a ist nicht durch 3 teilbar und (iv) b ist nicht durch 3 teilbar	
$a^2 \cdot 3^{2i} = b^2 \cdot 3^{2j+1}$	

6.

Annahme	Beweisverpflichtung
Es gibt $p,q \in \mathbb{Z}$ mit $p/q = \sqrt{3}$	Widerspruch
Es gibt $a,b \in \mathbb{Z}$ und $i,j \in \mathbb{N}_0$ mit (i) $p = a \cdot 3^i$, (ii) $q = b \cdot 3^j$, (iii) a ist nicht durch 3 teilbar und (iv) b ist nicht durch 3 teilbar	
$a^2 \cdot 3^{2i} = b^2 \cdot 3^{2j+1}$	
$2i = 2j+1$	

■

Aufgabe 2.2 — Vorrechnen. Beweisen Sie: Wenn A und B Aussagen sind, so gilt: WG(Wenn A, dann B) = WG(Nicht A oder B). ∎

Beweis. Es gilt nach Definition

$$\mathrm{WG}(\text{Wenn } A, \text{ dann } B) = \mathrm{WG}(\mathrm{WG}(A) \Rightarrow \mathrm{WG}(B)).$$

Nun betrachten wir die zugehörige Wahrheitstabelle

WG(A)	WG(B)	WG(WG(A) $\Rightarrow$ WG(B))
w	w	w
w	f	f
f	w	w
f	f	w

.

Besonders wichtig ist hier die Zeile $f\ w\ w$, aus etwas Falschem darf etwas Wahres folgen. Betrachten wir kurz folgendes Beispiel. Aus $4 < 5$ und $5 < 6$ kann ich $4 < 6$ folgern. Hierbei ist $\mathrm{WG}(4 < 5 \wedge 5 < 6) = w$ und $\mathrm{WG}(4 < 6) = w$. Füge ich nun der Prämisse eine falsche Behauptung hinzu $(1 > 2)$, so kann ich ja weiterhin logisch $4 < 6$ argumentieren. Also gilt $\mathrm{WG}(4 < 5 \wedge 5 < 6 \wedge 1 > 2) = f$, aber $\mathrm{WG}((4 < 5 \wedge 5 < 6 \wedge 1 > 2) \Rightarrow (4 < 6)) = w$.

Andererseits gilt

$$\mathrm{WG}(\text{Nicht } A \text{ oder } B) = \mathrm{WG}(\neg \mathrm{WG}(A) \vee \mathrm{WG}(B)).$$

Nun betrachten wir auch hier die zugehörige Wahrheitstabelle

WG(A)	WG(B)	$\neg$WG(A)	WG($\neg$WG(A) $\vee$ WG(B))
w	w	f	w
w	f	f	f
f	w	w	w
f	f	w	w

.

Die letzten Spalten der Tabellen sind identisch, also gilt

$$\mathrm{WG}(\neg \mathrm{WG}(A) \vee \mathrm{WG}(B)) = \mathrm{WG}(\neg \mathrm{WG}(A) \vee \mathrm{WG}(B))$$

und somit

$$\mathrm{WG}(\text{Wenn } A, \text{ dann } B) = \mathrm{WG}(\text{Nicht } A \text{ oder } B).$$ ■

Aufgabe 2.3 Seien M, N und K Mengen mit $M \subseteq N$. Beweisen Sie, dass (i) $M \cup K \subseteq N \cup K$ und (ii) $M \cap K \subseteq N \cap K$ gilt. ∎

Beweis. (i) Wir müssen zeigen, dass jedes Element in $M \cup K$ auch Element von $N \cup K$ ist. Sei also $x \in M \cup K$. Wenn $x \in M$, so ist auch $x \in N$, denn $M \subseteq N$. Da $M \subseteq M \cup K$, gilt auch $x \in M \cup K$. Gilt $x \in K$, so folgt $x \in M \cup K$, da $K \subseteq M \cup K$.

(ii) Wir müssen zeigen, dass jedes Element in $M \cap K$ auch Element von $N \cap K$ ist. Sei also $x \in M \cap K$. Somit ist auch $x \in M$ und $x \in K$. Da $M \subseteq N$ gilt, folgt $x \in N$ und somit $x \in N \cap K$. ■

Aufgabe 2.4 Beweisen Sie, dass jede durch 14 teilbare natürliche Zahl auch bereits durch 7 teilbar ist. ∎

Beweis. Sei $n \in \mathbb{N}_0$ eine durch 14 teilbare Zahl. Also gibt es eine natürliche Zahl $k \in \mathbb{N}_0$ mit $n = 14 \cdot k$. Um zu zeigen, dass n auch durch 7 teilbar ist, müssen wir zeigen, dass es eine natürliche Zahl $k' \in \mathbb{N}_0$ mit $n = 7k'$ gibt. Setzen wir $k' = 2k$, so folgt $k' \in \mathbb{N}_0$ und $n = 14k = 7 \cdot 2k = 7k'$. Somit ist n durch 7 teilbar. ■

Aufgabe 2.5 Entscheiden Sie, ob die folgenden Aussagen wahr oder falsch sind, indem Sie entweder einen Beweis oder ein Gegenbeispiel angeben:
(i) Für alle Mengen M, N und K gilt $(M \setminus N) \setminus K = M \setminus (N \setminus K)$.
(ii) Für alle Mengen M, N und K mit $M \subseteq K$ und $N \subseteq K$ gilt $M \cup N \subseteq K$. ■

Beweis. (i) Betrachte $M = \{1,2\}$, $N = \{2\}$ und $K = \{2\}$. Dann gilt $(M \setminus N) \setminus K = (\{1,2\} \setminus \{2\}) \setminus \{2\} = \{1\} \setminus \{2\} = \{1\}$. Auf der anderen Seite gilt $M \setminus (N \setminus K) = \{1,2\} \setminus (\{2\} \setminus \{2\}) = \{1,2\} \setminus \emptyset = \{1,2\}$. Somit ist die Aussage also nicht wahr.
(ii) Sei also $x \in M \cup N$. Ist $x \in M$, so gilt auch $x \in K$, denn $M \subseteq K$. Gilt hingegen $x \in N$, so gilt auch $x \in K$, denn $N \subseteq K$.

■

Aufgabe 2.6 Beweisen Sie: Wenn A und B Aussagen sind, so gilt: WG(A genau dann, wenn B) = WG((Wenn A, dann B) und (Wenn B, dann A)). ■

Beweis. Es gilt nach Definition

$$\mathrm{WG}(A \text{ genau dann, wenn } B) = \mathrm{WG}(\mathrm{WG}(A) \Leftrightarrow \mathrm{WG}(B)).$$

Die zugehörige Wahrheitstabelle ist

WG(A)	WG(B)	WG(WG(A) $\Leftrightarrow$ WG(B))
w	w	w
w	f	f
f	w	f
f	f	w

.

Weiterhin gilt

$$\mathrm{WG}((\text{Wenn } A, \text{ dann } B) \text{ und } (\text{Wenn } B, \text{ dann } A)) =$$
$$\mathrm{WG}(\mathrm{WG}(\mathrm{WG}(A) \Rightarrow \mathrm{WG}(B)) \wedge \mathrm{WG}(\mathrm{WG}(B) \Rightarrow \mathrm{WG}(A))).$$

Die zugehörigen Tabellen sind

WG(A)	WG(B)	WG(WG(A) $\Rightarrow$ WG(B))	WG(WG(B) $\Rightarrow$ WG(A))
w	w	w	w
w	f	f	w
f	w	w	f
f	f	w	w

und

WG(WG(WG(A) $\Rightarrow$ WG(B)) $\wedge$ WG(WG(B) $\Rightarrow$ WG(A)))
w
f
f
w

.

Da die letzten Spalten gleich sind, gilt

$$\mathrm{WG}(\mathrm{WG}(A) \Leftrightarrow \mathrm{WG}(B)) = \mathrm{WG}(\mathrm{WG}(\mathrm{WG}(A) \Rightarrow \mathrm{WG}(B)) \wedge \mathrm{WG}(\mathrm{WG}(B) \Rightarrow \mathrm{WG}(A)))$$

und somit

$$\mathrm{WG}(A \text{ genau dann, wenn } B) = \mathrm{WG}((\text{Wenn } A, \text{ dann } B) \text{ und } (\text{Wenn } B, \text{ dann } A)).$$ ■

Aufgabe 2.7 Seien n und m natürliche Zahlen. Beweisen Sie, dass jede durch $n \cdot m$ teilbare natürliche Zahl auch durch n teilbar ist. ■

Beweis. Sei $x \in \mathbb{N}_0$ eine durch $n \cdot m$ teilbare Zahl. Also gibt es eine ganze Zahl $k \in \mathbb{N}_0$ mit $x = n \cdot m \cdot k$. Um zu zeigen, dass x auch durch n teilbar ist, müssen wir zeigen, dass es eine natürliche Zahl $k' \in \mathbb{N}_0$ mit $x = nk'$ gibt. Setzen wir $k' = mk$, so folgt $k' \in \mathbb{N}_0$ und $x = n \cdot m \cdot k = n \cdot k'$. Somit ist x durch n teilbar.

Hier kann man schön sehen, dass wir nur den Beweis für $n = 7, m = 2$ verallgemeinert haben. ■

Aufgabe 2.8 Sei p eine Primzahl. Zeigen Sie, dass $\sqrt{p} \notin \mathbb{Q}$, also irrational ist. ■

Beweis. Angenommen, dass $\sqrt{p} \in \mathbb{Q}$. Also gibt es ganze Zahlen $r, s \in \mathbb{Z}$, so dass $r/s = \sqrt{p}$. Die eindeutige Primfaktorisierung von r und s impliziert, dass es ganze Zahlen $a, b \in \mathbb{Z}$ und natürliche Zahlen $i, j \in \mathbb{N}_0$ gibt mit

(i) $r = a \cdot p^i$,
(ii) $s = b \cdot p^j$,
(iii) a ist nicht durch p teilbar und
(iv) b ist nicht durch p teilbar.

Nach Definition von $\sqrt{p}$, wir haben

$$p = \sqrt{p} \cdot \sqrt{p} = (\sqrt{p})^2 = (r/s)^2 = \left(\frac{a \cdot p^i}{b \cdot p^j}\right)^2 = \frac{a^2 \cdot p^{2i}}{b^2 \cdot p^{2j}}.$$

Multiplizieren mit $b^2 \cdot p^{2j}$ gibt uns $b^2 \cdot p^{2j+1} = a^2 \cdot p^{2i}$. Da a und b nicht durch p teilbar sind, sind auch a^2 und b^2 nicht durch p teilbar. Die Primfaktorzerlegung von $a^2 \cdot p^{2i}$ enthält also genau $2i$ Kopien von p. Analog enthält die Primfaktorzerlegung von $b^2 \cdot p^{2j+1}$ also genau $2j+1$ Kopien von p. Da aber $a^2 \cdot p^{2i} = b^2 \cdot p^{2j+1}$ gilt, muss somit $2i = 2j+1$ gelten. Da $2i$ gerade und $2j+1$ ungerade ist, haben wir einen Widerspruch erreicht. Somit gilt $\sqrt{p} \notin \mathbb{Q}$.

Auch hier sieht man schön, wie man den speziellen Fall ganz einfach verallgemeinern kann. ■

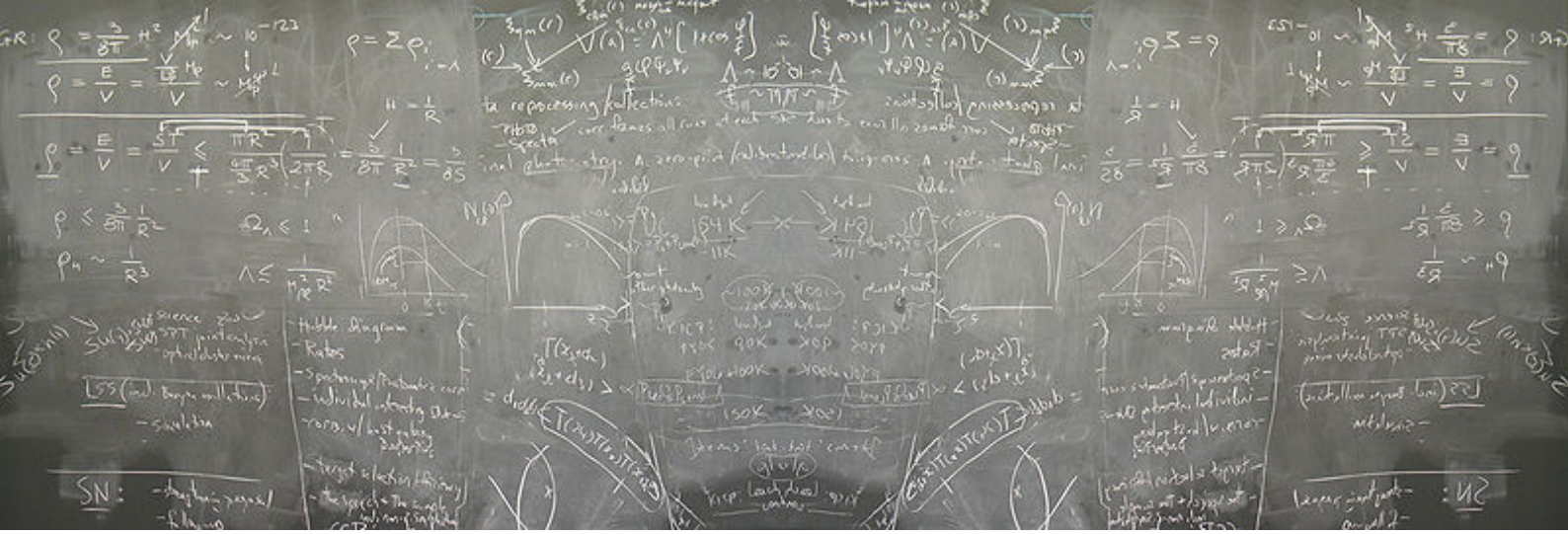
Aufgabe 2.9 Für eine Menge M ist $\mathscr{P}(M)$ die *Potenzmenge* von M, also die Menge, die alle Teilmengen von M enthält: $\mathscr{P}(M) = \{N \mid N \subseteq M\}$. Beweisen Sie, dass für alle Mengen M und N gilt, dass $\mathscr{P}(M \cap N) = \mathscr{P}(M) \cap \mathscr{P}(N)$. ■

Beweis. Sei $X \in \mathscr{P}(M \cap N)$. Somit gilt also $X \subseteq M \cap N$ und somit auch $X \subseteq M$ und $X \subseteq N$. Also gilt auch $X \in \mathscr{P}(M) \cap \mathscr{P}(N)$. ■

Lernziele:

- Das Prinzip eines Beweises verstanden haben
- Erste Beweise alleine führen können
- Grundlegendes Verständnis für die elementaren Operatoren zum Verknüpfen von Aussagen

- Wahrheitstabellen interpretieren können
- Die Welt der Aussagen und die Welt der Wahrheitswerte unterscheiden können



3. Beweismuster

Zum Lesen:
Kapitel 2.8 in [Wol17]
Ausarbeitung von Sebastian zu Beweisrezepten (also dieses Dokument)

Stichworte:

- Offensichtliche Beweiszüge
- Beweismuster für *und*, *oder*, *wenn-dann*, *genau-dann-wenn*
- Technik der Fallunterscheidung
- Widersprüchliche Annahmen
- Definitionsanwendung

3.1 Beweisrezepte

Im Folgenden stellen wir einfache Rezepte vor, an denen Sie sich für bestimmte Arten von Beweisen orientieren können. Diese Rezepte entstammen dem Skript zur Vorlesung *Einführung in die Logik* an der Universität zu Lübeck von Till Tantau aus dem Jahr 2017.

Titel: All-Aussagen beweisen
Ziel: Es soll gezeigt werden »für alle Dinge, die so und so sind, gilt blah« oder auch »jedes Ding, das so und so ist, hat die Eigenschaft blah«.
Vorgehen:

1. Beginne den Beweis mit »Sei x ein beliebiges Ding, das so und so ist.«
2. Zeige nun, dass x die Eigenschaft blah hat.

Titel: Beweise strukturieren
Ziel: Der Leser/ die Leserin soll immer genau wissen, was bereits gezeigt wurde und was noch zu zeigen ist, denn Beweise werden sehr schnell »unübersichtlich«.
Vorgehen:

1. Benutzen Sie Wendungen wie »Damit wurde gezeigt, dass . . . « oder »Es bleibt zu zeigen, dass . . . « oder »Im Folgenden zeigen wir zunächst . . . , . . . zeigen wir hingegen später.«
2. Geben Sie am Anfang eines langen Beweises eine Übersicht und teilen Sie den Beweis in Abschnitte wie »Die Konstruktion« oder »Die Rückrichtung der Korrektheit der zweiten Unterkonstruktion«.
3. Formulieren Sie Zwischenbehauptung als Lemmata, die Sie zu erst beweisen.

Titel: Details weglassen
Ziel: Der Beweis soll kurz und knapp bleiben.
Vorgehen:

1. Man lässt »langweilige« Teile des Beweises weg.
2. Freundlicherweise schreibt stattdessen »Man kann zeigen, dass. . . « oder »Auf den Nachweis, dass . . . gilt, wurde verzichtet«.

Vermeiden sollte man »Trivialerweise gilt . . . « oder »Offensichtlich gilt . . . «, dies reizt den Leser/ die Leserin eher zu argumentieren, dass dies doch nicht so trivial ist.

Titel: Fallunterscheidung
Ziel: Es soll gezeigt werden, dass eine beliebige Behauptung gilt, die sich gut in überschaubar viele Fälle zerlegen lässt.
Vorgehen:

1. Leite den Beweis mit »Wir machen eine Fallunterscheidung.« ein.
2. Man benennt zwei oder mehr beliebige Annahmen (»Fälle« genannt), die mit der Behauptung nichts zu tun haben brauchen, von denen aber in jeder Situation mindestens (besser: genau) eine zutreffen muss.
3. Für jede Annahme X schreibt man »Fall X:«, gefolgt von einem Beweis, dass X die Behauptung impliziert.

Titel: Konstruktiver Beweis
Ziel: Es soll gezeigt werden, dass es ein Ding mit bestimmten Eigenschaften *gibt*.
Vorgehen:

1. Man gibt an, wie ein bestimmtes Ding *konstruiert* werden soll.
2. Man zeigt nun, dass das konstruierte Ding die behaupteten Eigenschaften hat.

Der zweite Schritt wird gerne vergessen!

Titel: Kreisschluss
Ziel: Es soll gezeigt werden, dass mehrere Behauptungen gleichwertig sind. Dazu zeigt man dass die Aussagen sich kreisförmig implizieren.
Vorgehen:

1. Nimm an, dass die erste Aussage gilt. Folgere, dass nun auch die zweite gelten muss.
2. Beginne nun eine neue Argumentation. Nimm an, dass die zweite Aussage gilt. Folgere, dass dann die dritte gelten muss.
3. Und so weiter bis zur letzte, für die man zeigt, dass sie die erste impliziert.

Falls nun eine Aussage gilt, so gelten folglich alle.

Titel: Namen für Teile vergeben
Ziel: Man möchte über die Teile eines Dings »reden«.
Vorgehen:

1. Ist das Ding ein Tupel mit einer festen Anzahl Komponenten (wie zum Beispiel eine Grammatik, die immer aus vier Teilen besteht), so schreibt man »Sei $G = (N,T,S,E)$ eine Grammatik…« oder »Sei $G = (V,E)$ ein Graph…«.
2. Ist das Ding ein Tupel mit einer variablen Anzahl Komponenten, so schreibt man »Sei $(x_1,x_2,x_3,\ldots,x_n)$ das Tupel…«. Nebenbei hat man mit n auch einen Namen für die Länge des Tupels eingeführt.
3. Ist das Ding eine abzählbare Menge, so schreibt man »Sei $M = \{m_1,m_2,m_3,\ldots,m_n\}$« (bei endlichen Mengen) oder »Sei $M = \{m_1,m_2,m_3,\ldots\}$« (bei abzählbar unendlichen Mengen).

Titel: Trennung der Ebenen
Ziel: Die zwei Beweisebenen sollen dem Leser/ der Leserin klar werden.
Vorgehen:

1. Für die »Objekte der Logik« benutzen wir *Formeln* und *Symbole*.
2. Für die Metaebene benutzen wir *normalsprachlichen Text*.

Gutes Beispiel: … Falls $\hat{\beta}(\psi) = 0$, so gilt $\hat{\beta}(\psi \to \phi) = 1$. …
Schlechtes Beispiel: … $\hat{\beta}(\psi) = 0 \to \hat{\beta}(\psi \to \phi) = 1$. …

Titel: Widerspruchsbeweis
Ziel: Eine Behauptung beweisen, indem man zeigt, dass die Annahme des Gegenteils zu einem Widerspruch führt.
Vorgehen:

1. Leite den Beweis mit »Wir führen einen Widerspruchsbeweis.« oder »Zum Zwecke des Widerspruchs nehmen wir an, dass XYZ nicht gilt.«
2. Führe nun einen Beweis, in dem die Annahme »nicht XYZ« beliebig benutzt werden darf.
3. Beende den Beweis, wenn doch »XYZ« hergeleitet wurde oder wenn ein offensichtlich falsche Behauptung wie $1 = 2$ hergeleitet wurde, mit »Widerspruch.« oder »Dies ist aber ein Widerspruch zur Annahme, dass nicht XYZ gilt, weshalb doch XYZ gelten muss«.

Titel: Zwei Beweisrichtungen
Ziel: Es soll gezeigt werden, dass eine Behauptungen A genau dann gilt, wenn eine andere Behauptung B gilt.
Vorgehen:

1. Beginne mit »Es sind zwei Richtungen zu zeigen.«
2. Fahre fort mit »Für die erste Richtung nehmen wir an, dass A gilt.« Folgere, dass dann auch B gelten muss.
3. Beginne einen neuen Absatz mit »Für die Rückrichtung nehmen wir an, dass B gilt.« Folgere, dass dann auch A gelten muss.

Aufgaben

Aufgabe 3.1 — Vorrechnen:. Beweisen Sie: Jede Primzahl, die größer als 5 ist, lässt bei Division durch 10 den Rest 1, 3, 7 oder 9. ∎

Beweis. Sei x eine Primzahl und $x > 3$. Lässt x bei Division durch 10 den Rest 1, 3, 7 oder 9, gilt

die Behauptung offensichtlich. Wir betrachten die restlichen Fälle einzeln:

- Hinterlässt x bei Division durch 10 den Rest 0, so gibt es eine natürliche Zahl $k \in \mathbb{N}_0$ mit $x = 10k$. Somit ist x durch 2 und 5 teilbar, was ein Widerspruch dazu ist, dass x prim ist.
- Hinterlässt x bei Division durch 10 den Rest 2, so gibt es eine natürliche Zahl $k \in \mathbb{N}_0$ mit $x = 10k+2$. Nun können wir $10k+2 = 2(5k+1)$ schreiben. Somit ist x durch 2 teilbar. Da $x > 5$, ist dies ein Widerspruch dazu, dass x prim ist.
- Hinterlässt x bei Division durch 10 den Rest 4, so gibt es eine natürliche Zahl $k \in \mathbb{N}_0$ mit $x = 10k+4$. Nun können wir $10k+4 = 2(5k+2)$ schreiben. Somit ist x durch 2 teilbar. Da $x > 5$, ist dies ein Widerspruch dazu, dass x prim ist.
- Hinterlässt x bei Division durch 10 den Rest 5, so gibt es eine natürliche Zahl $k \in \mathbb{N}_0$ mit $x = 10k+5$. Nun können wir $10k+5 = 5(2k+1)$ schreiben. Da $x > 5$, gilt auch $k \geq 1$. Somit ist x durch 5 und $2k+1 > 1$ teilbar. Dies ein Widerspruch dazu, dass x prim ist.
- Hinterlässt x bei Division durch 10 den Rest 6, so gibt es eine natürliche Zahl $k \in \mathbb{N}_0$ mit $x = 10k+6$. Nun können wir $10k+6 = 2(5k+3)$ schreiben. Somit ist x durch 2 teilbar. Da $x > 5$, ist dies ein Widerspruch dazu, dass x prim ist.
- Hinterlässt x bei Division durch 10 den Rest 8, so gibt es eine natürliche Zahl $k \in \mathbb{N}_0$ mit $x = 10k+8$. Nun können wir $10k+8 = 2(5k+4)$ schreiben. Somit ist x durch 2 teilbar. Da $x > 5$, ist dies ein Widerspruch dazu, dass x prim ist.

Somit haben wir alle Fälle abgehandelt und für alle Reste ungleich 1, 3, 7 und 9 gezeigt, dass diese nicht vorkommen können. Somit ist die Behauptung bewiesen. ■

Aufgabe 3.2 — Vorrechnen:. Betrachten Sie folgenden Beweis zur Aussage, dass n^2+n für alle ganzen Zahlen n eine gerade Zahl ist:

n gerade $\Longrightarrow (2k)^2 + 2k = 4k^2 + 2k = 2(2k^2 + k)$

n ungerade $\Longrightarrow (2r+1)^2 + 2r + 1 = 4r^2 + 4r + 1 + 2r + 1 = 2(2r^2 + 2r + 1)$

Schreiben Sie den Beweis neu, indem Sie aus ihm einen kompletten Text machen und versuchen, alle Unklarheiten und Ungeschicktheiten zu beseitigen. ■

Beweis. Sei $n \in \mathbb{Z}$ eine ganze Zahl. Dann ist n entweder gerade oder ungerade. Wir unterscheiden die beiden Fälle:

- Ist n gerade, so gibt es eine ganze Zahl $k \in \mathbb{Z}$ mit $n = 2k$. Weiterhin gilt dann

$$n^2 + n = (2k)^2 + 2k = 4k^2 + 2k = 2(2k^2 + k)$$

  und somit ist n^2+n durch 2 teilbar.
- Ist n ungerade, so gibt es eine ganze Zahl $k \in \mathbb{Z}$ mit $n = 2k+$. Weiterhin gilt dann

$$n^2 + n = (2k+1)^2 + 2k + 1 = 4k^2 + 4k + 1 + 2k + 1 = 4k^2 + 6k + 2 = 2(2k^2 + 3k + 1)$$

  und somit ist n^2+n durch 2 teilbar.

Da wir nun alle möglichen Fälle für n betrachtet haben, gilt die Aussage. ■

Aufgabe 3.3 Seien a und b reelle Zahlen mit $b \neq 0$. Beweisen Sie, dass $\frac{|a|}{|b|} = \left|\frac{a}{b}\right|$.

Hinweis: Es gilt $|a| = a$, wenn $a \geq 0$ und $|a| = -a$ für $a < 0$. ■

Beweis. Seien a, b reelle Zahlen mit $b \neq 0$. Nun haben wir vier mögliche Fälle:

- Gilt $a \geq 0$ und $b \geq 0$, so folgt $|a| = a$ und $|b| = b$ und somit $\frac{|a|}{|b|} = \frac{a}{b}$. Da $a \geq 0$ und $b \geq 0$, gilt auch $\frac{a}{b} \geq 0$ und somit ist $\frac{a}{b} = \left|\frac{a}{b}\right|$.

- Gilt $a \geq 0$ und $b < 0$, so folgt $|a| = a$ und $|b| = -b$ und somit $\frac{|a|}{|b|} = \frac{a}{-b}$. Da $a \geq 0$ und $b < 0$, gilt auch $\frac{a}{-b} \geq 0$ und somit ist $\frac{a}{-b} = |\frac{a}{b}|$.
- Gilt $a < 0$ und $b \geq 0$, so folgt $|a| = -a$ und $|b| = b$ und somit $\frac{|a|}{|b|} = \frac{-a}{b}$. Da $a < 0$ und $b \geq 0$, gilt auch $\frac{-a}{b} \geq 0$ und somit ist $\frac{-a}{b} = |\frac{a}{b}|$.
- Gilt $a < 0$ und $b < 0$, so folgt $|a| = -a$ und $|b| = -b$ und somit $\frac{|a|}{|b|} = \frac{-a}{-b} = \frac{a}{b}$. Da $a < 0$ und $b \geq 0$, gilt auch $\frac{a}{b} \geq 0$ und somit ist $\frac{a}{b} = |\frac{a}{b}|$.

■

Aufgabe 3.4 Diese Aufgabe entstammt dem sehr lesenswerten Buch von Beutelspacher [Beu99].

Betrachten Sie folgenden Beweis zur Aussage, dass das Quadrat jeder ungeraden natürlichen Zahl n bei Division durch 4 den Rest 1 ergibt:

n Quadrat $\Longrightarrow$ n^2

$(2n+1)^2 = 4n^2 + 4n + 1 \underbrace{=}_{\text{modulo } 4}$ Rest 1

Schreiben Sie den Beweis neu, indem Sie aus ihm einen kompletten Text machen und versuchen, alle Unklarheiten und Ungeschicktheiten zu beseitigen. ■

Beweis. Sei n eine ungerade natürliche Zahl. Dann gibt es eine natürliche Zahl k, so dass $n = 2k+1$. Somit gilt

$$n^2 = (2k+1)^2 = 4k^2 + 4k + 1 = 4(k^2+k) + 1.$$

Bei Division durch 4 bleibt also der Rest 1. ■

Aufgabe 3.5 Seien x und y reelle Zahlen. Beweisen Sie, dass $|x+y| \leq |x| + |y|$. Diese Eigenschaft nennt man auch die *Dreiecksungleichung* und sie ist sehr nützlich. ■

Beweis. Seien x und y reelle Zahlen. Wir haben vier Fälle:

- Gilt $x \geq 0$ und $y \geq 0$, so ist auch $x+y \geq 0$ und somit

$$|x+y| = x+y \leq x+y = |x| + |y|.$$

- Gilt $x < 0$ und $y < 0$, so ist auch $x+y < 0$ und somit

$$|x+y| = (-x)+(-y) \leq (-x)+(-y) = |x| + |y|.$$

- Gilt $x \geq 0$ und $y < 0$, so müssen wir wieder zwei Fälle unterscheiden:
  - Gilt $x+y \geq 0$, so haben wir

$$|x+y| = x+y \leq_{y<0} x+(-y) = |x| + |y|.$$

  - Gilt $x+y < 0$, so haben wir

$$|x+y| = -(x+y) = (-x)+(-y) \leq_{x\geq 0} x+(-y) = |x| + |y|.$$

- Gilt $x < 0$ und $y \geq 0$, so müssen wir wieder zwei Fälle unterscheiden:
  - Gilt $x+y \geq 0$, so haben wir

$$|x+y| = x+y \leq_{x<0} (-x)+y = |x| + |y|.$$

– Gilt $x+y<0$, so haben wir

$$|x+y| = -(x+y) = (-x)+(-y) \leq_{y\geq 0} (-x)+y = |x|+|y|.$$

Damit haben wir alle Fälle untersucht. Wie man sieht, funktioniert der vierte Fall genauso wie der dritte. Häufig kürzt man solche Argumente dann ab. ■

Aufgabe 3.6 Diese Aufgabe entstammt dem sehr lesenswerten Buch von Beutelspacher [Beu99].

Betrachten Sie folgenden Beweis zur Aussage, dass für alle natürlichen Zahlen n und alle Primzahlen p gilt, dass p genau dann ein Teiler von n ist, wenn p ein Teiler von n^2 ist.

Zu zeigen: p teilt n^2 $\Leftrightarrow$ wenn p teilt n

Klar: notwendig

Umgekehrt: Wenn p nicht n teilt, dann teilt p auch nicht n^2 wg. Eindeutigkeit PFZ.

Schreiben Sie den Beweis neu, indem Sie aus ihm einen kompletten Text machen und versuchen, alle Unklarheiten und Ungeschicktheiten zu beseitigen.

■

Beweis. Sei n eine natürliche Zahl und p eine Primzahl. Wir möchten nun zeigen, dass p genau dann ein Teiler von n ist, wenn p ein Teiler von n^2 ist. Dazu zeigen wir die beiden Implikationen einzeln.

"$\Rightarrow$" Ist p ein Teiler von n, so gibt es eine natüriche Zahl k mit $n = p\cdot k$. Dann gilt auch $n^2 = (pk)^2 = p^2k^2 = p(pk^2)$. Somit ist p auch tein Teiler von n^2.

"$\Leftarrow$" Wir betrachten die Kontraposition und nehmen also an, dass p kein Teiler von n ist. Wir wissen, dass jede natürliche Zahl n eine *eindeutige* Primfaktorzerlegung besitzt. Somit gibt es eindeutige Primzahlen $q_1,\ldots,q_r$ und natürliche Exponenten $k_1,\ldots,k_r \geq 1$ mit

$$n = q_1^{k_1}\cdot q_2^{k_2}\cdot\cdots\cdot q_r^{k_r}.$$

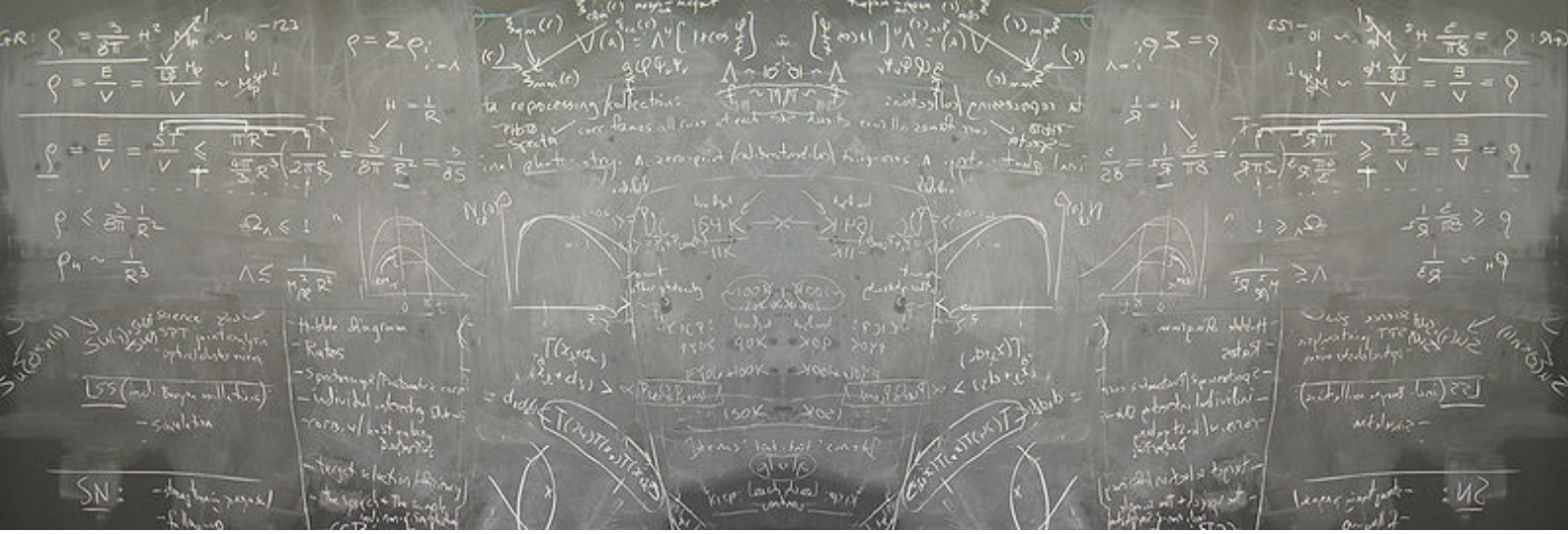
Da p kein Teiler von n ist, gilt $q_i \neq p$ für alle $i = 1,\ldots,r$.
Es gilt nun auch

$$n^2 = (q_1^{k_1}\cdot q_2^{k_2}\cdot\cdots\cdot q_r^{k_r})^2 = q_1^{2k_1}\cdot q_2^{2k_2}\cdot\cdots\cdot q_r^{2k_r}.$$

Da die q_i alle Primzahlen sind, ist dies auch die eindeutige Primfaktorzerlegung von n^2. Da $q_i \neq p$ für alle $i = 1,\ldots,r$, ist somit p kein Teiler von n^2. ■

Lernziele:

- Grundlegende Beweisrezepte und Beweismuster kennen
- Das Beweismuster »Fallunterscheidung« kennen
- Beweise klar und verständlich aufschreiben können
- Typische Konventionen für den Aufschrieb von Beweisen kennen



4. Prädikatenlogik

Zum Lesen:
Kapitel 3.1, 3.3, 3.4 in [Wol17].

Stichworte:

- Quantoren: *Für-alle*, *Es-gibt-ein*; Quantorenwechsel; Reihenfolge
- Unbestimmte Symbole vs. konkrete Werte

Definition 4.1 Die Quantoren im Buch sind häufig auch durch andere Symbole dargestellt:

- Der *Für-alle*-Quantor wird auch als $\forall$ geschrieben.
- Der *Es-gibt-ein*-Quantor wird auch als $\exists$ geschrieben.

Aufgaben

Aufgabe 4.1 — Vorrechnen. Beweisen Sie: Es gibt eine Menge M, so dass für alle x gilt, dass $x \in M$ impliziert, dass $x < 4$. ■

Beweis. Mithilfe logischer Operatoren und Quantoren kann man die Aussage auch schreiben als

$$\exists M \forall x : x \in M \Rightarrow x < 4$$

oder als

$$\exists M \forall x \in M : x < 4.$$

Da an vorderster Stelle ein Es-gibt-ein-Quantor steht, dürfen wir M frei wählen. Sei also $M = \{1\}$. Nun sei $x \in M$. Da M nur aus einem Element besteht, gilt also $x = 1 < 4$. ■

Aufgabe 4.2 Beweisen Sie: Für alle Mengen M, N und K gilt, dass $(M \cup N) \cap K = (M \cap K) \cup (N \cap K)$. ■

Beweis. Als Formel kann man die Aussage wie folgt aufschreiben:

$$\forall M \forall N \forall K : [(\forall x \in (M \cup N) \cap K : x \in (M \cap K) \cup (N \cap K)) \wedge (\forall x \in (M \cap K) \cup (N \cap K) : x \in (M \cup N) \cap K)].$$

Seien also M, N und K Mengen. Wir zeigen zuerst die Richtung $(M \cup N) \cap K \subseteq (M \cap K) \cup (N \cap K)$. Sei dafür $x \in (M \cup N) \cap K$. Somit gilt auch $x \in K$. Nun unterscheiden wir zwei Fälle:

(i) Ist $x \in M$, so gilt $x \in M \cap K$. Da $M \cap K \subseteq (M \cap K) \cup (N \cap K)$ ist, folgt $x \in (M \cap K) \cup (N \cap K)$.

(ii) Ist $x \in N$, so gilt $x \in N \cap K$. Da $N \cap K \subseteq (M \cap K) \cup (N \cap K)$ ist, folgt $x \in (M \cap K) \cup (N \cap K)$.

Nun zeigen wir $(M \cap K) \cup (N \cap K) \subseteq (M \cup N) \cap K$. Sei also $x \in (M \cap K) \cup (N \cap K)$. Wir unterscheiden wieder zwei Fälle:

(i) Ist $x \in (M \cap K)$, so gilt $x \in M$ und $x \in K$. Da $M \subseteq M \cup N$, ist somit $x \in M \cup N$ und somit auch $x \in (M \cup N) \cap K$.

(ii) Ist $x \in (N \cap K)$, so gilt $x \in N$ und $x \in K$. Da $N \subseteq M \cup N$, ist somit $x \in M \cup N$ und somit auch $x \in (M \cup N) \cap K$.

■

Aufgabe 4.3 Beweisen Sie: Für alle $x \in \mathbb{N}_0$ gibt es ein $y \in \mathbb{N}_0$, so dass $y > x$. ■

Beweis. Mit Quantoren steht dort $\forall x \in \mathbb{N}_0 \exists y \in \mathbb{N}_0 : y > x$. An vorderster Stelle steht ein für-alle-Quantor, also sei $x \in \mathbb{N}_0$. Nun kommt ein es-gibt-ein-Quantor, also dürfen wir y konkret wählen, zum Beispiel mit $y = x + 1$. Da $x < x + 1 = y$, folgt also die Aussage. ■

Aufgabe 4.4 Beweisen Sie: Es gibt eine Menge M, so dass für alle Mengen N gilt, dass $M \cup N = N$. ■

Beweis. Sei $M = \emptyset$ die leere Menge und sei N eine beliebige Menge. Dann gilt $M \cup N = \emptyset \cup N = N$. ■

Aufgabe 4.5 Widerlegen Sie: Es gibt eine Menge M, so dass für alle Mengen N gilt, dass $M \cup N = M$. ■

Beweis. Betrachte $M = \{1\}$ und $N = \{1,2\}$. Dann gilt $M \cup N = \{1,2\} \neq \{1\} = M$, also $M \cup N \neq M$. ■

Aufgabe 4.6 Beweisen Sie: Es gibt eine natürliche Zahl $n_0 \in \mathbb{N}_0$, so dass für alle natürlichen Zahlen $n \in \mathbb{N}_0$ gilt: wenn $n \geq n_0$, dann $4n^3 + 1 \leq n^4$. ■

Beweis. Sei $n_0 = 5$ und $n \in \mathbb{N}_0$ mit $n \geq 5$. Dann gilt

$$4n^3 + 1 \underbrace{\leq}_{1 < 5 \leq n < n^3} 4n^3 + n^3 = 5n^3 \underbrace{\leq}_{5 \leq n} n \cdot n^3 = n^4.$$

■

Aufgabe 4.7 Beweisen Sie: Für alle Mengen M und N gibt es eine Menge K mit $M \subseteq K$ und $N \subseteq K$. ■

Beweis. Seien M und N Mengen. Wähle $K = M \cup N$. Da $M \subseteq M \cup N = K$, gilt $M \subseteq K$. Weiterhin ist $N \subseteq M \cup N = K$, also $N \subseteq K$. ■

Aufgabe 4.8 Beweisen Sie: Für alle Zahlen $x \in \mathbb{N}_0$ gibt es eine natürliche Zahl $y \in \mathbb{N}_0$ so dass aus $x > 0$ folgt, dass $y < x$. ■

Beweis. Sei $x \in \mathbb{N}_0$ mit $x > 0$. Mit $y = 0$ folgt $y = 0 < x$, also $y < x$. ■

Aufgabe 4.9 Beweisen Sie: Für alle $x \in \mathbb{N}_0$ und alle $y \in \mathbb{N}_0$ gibt es eine Menge M, so dass für alle $z \in \mathbb{N}_0$ gilt: Ist $z \neq x$ und $z \neq y$, so gilt $x \in M$, $y \in M$ und $z \notin M$. ■

Beweis. Seien $x, y \in \mathbb{N}_0$. Wähle $M = \{x, y\}$. Sei nun $z \in \mathbb{N}_0$ mit $z \neq x$ und $z \neq y$. Nach Definition gilt $x \in \{x, y\} = M$, also $x \in M$ und $y \in \{x, y\} = M$, also auch $y \in M$. Da nun $z \neq x$ und $z \neq y$, folgt $z \notin M$. ■

Aufgabe 4.10 Beweisen Sie: Für alle Mengen M und N gilt, dass $M \subseteq N$ genau dann wenn $M \cap N = M$. ■

Beweis. Wir zeigen also $\forall M, N : (M \subseteq N \Leftrightarrow M \cap N = M)$. Aus Aufgabe 2.6 können wir schließen, dass wir die genau-dann-wenn-Beziehung auch durch zwei Implikationen darstellen können. Wir zeigen also

$$\forall M, N : [(M \subseteq N \Rightarrow M \cap N = M) \wedge (M \cap N = M \Rightarrow M \subseteq N)].$$

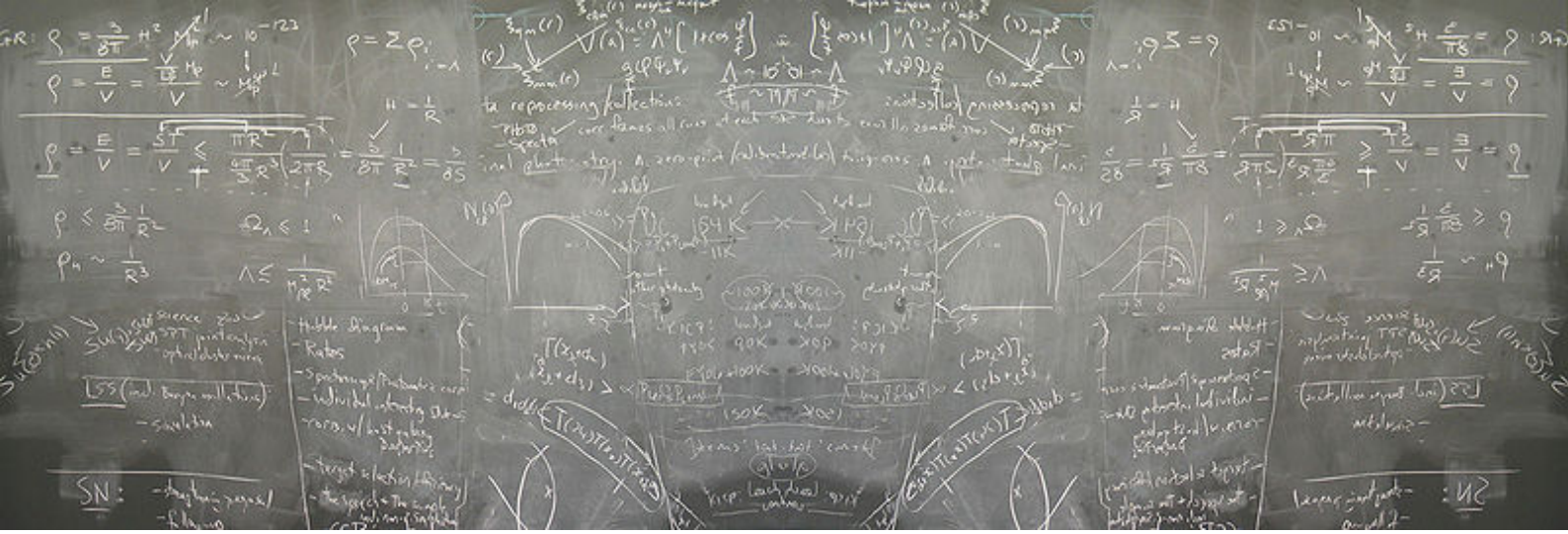
Seien also M und N Mengen. Offensichtlich gilt $M \cap N \subseteq M$ auch ohne weitere Annahmen. Wir müssen also nur noch zeigen, dass $(M \subseteq N) \Leftrightarrow M \subseteq M \cap N$ gilt, indem wir die zwei entsprechenden Implikationen beweisen.

- Gilt $M \subseteq N$, so gilt für alle $x \in M$, dass auch $x \in N$. Also folgt $x \in M \cap N$ für alle $x \in M$ und somit $M \subseteq M \cap N$.
- Gilt $M \subseteq M \cap N$, so gilt für alle $x \in M$, dass sie auch in $M \cap N$ und somit auch in N sind. Somit gilt $M \subseteq N$.

■

Lernziele:

- Grundlegendes Verständnis für Quantoren und deren Reihenfolge
- Sicherer Umgang mit dem Beweisen von Aussagen, die Quantoren enthalten



5. Folgen, Induktion, Rekurrenzen

Zum Lesen:
Kapitel 3.7 in [Wol17]
Kapitel 8.1, 8.2, 8.3, 8.4, 8.5 in [Bel18]

Stichworte:

- Aussagen über alle hinreichend große natürliche Zahlen
- Induktionsanfang, Induktionsvoraussetzung, Induktionsbehauptung, Induktionsschritt
- Fehlerhafte Argumentation bei "untypischen" Fällen
- Sequenzen von natürlichen Zahlen: Rekursive Definition / Rekurrenzgleichung vs. geschlossene Form
- Initiale Werte für Rekurrenzen
- Typischer Induktionsbeweis für Rekurrenzen
- Fibonacci-Zahlen F_n mit $F_0 = 1$, $F_1 = 1$ und $F_n = F_{n-1} + F_{n-2}$

Definition 5.1 In [Bel18] wird die *Summennotation* $\sum_{i=1}^{n} a_i$ genutzt. Dies ist einfach eine kompakte Schreibweise für $a_1 + a_2 + \ldots + a_n$. Zum Beispiel ist $\sum_{i=1}^{10} i = 1 + 2 + \ldots + 10$ oder $(1^2 - 1) + (2^2 - 1) + (3^2 - 1) + \ldots + (42^2 - 1) = \sum_{i=1}^{42} (i^2 - 1)$.

Definition 5.2 In beiden Büchern wird eine leicht vereinfachte Form der Induktion genutzt, die man manchmal auch *schwache Induktion* nennt. Um $H(k+1)$ zu beweisen, wird hierbei $H(k)$ vorausgesetzt. In der *starken Induktion* (manchmal auch *Noethersche Induktion*), nimmt man statt $H(k)$ *alle Aussagen* $H(0), H(1), \ldots, H(k)$ als Induktionsvoraussetzung. Häufig kann man dadurch seine Beweise etwas vereinfachen.

Aufgaben

Aufgabe 5.1 — Vorrechnen. Beweisen Sie: Für alle $n \in \mathbb{N}_0$ gilt, dass $\sum_{i=0}^{n} i = \frac{n(n+1)}{2}$. ∎

Beweis. Wir beweisen die Aussage per Induktion:

Induktionsanfang: Für $n = 0$ gilt

$$\sum_{i=0}^{n} i = \sum_{i=0}^{0} i = 0 = \frac{0 \cdot 1}{2} = \frac{n(n+1)}{2}.$$

Induktionsvoraussetzung: Es gelte $\sum_{i=0}^{k} i = \frac{k(k+1)}{2}$ für ein $k \in \mathbb{N}_0$.
Induktionsschritt: Es gilt

$$\sum_{i=0}^{k+1} i = [\sum_{i=1}^{k} i] + (k+1) \underbrace{=}_{\text{IV}} \frac{k(k+1)}{2} + (k+1) =$$
$$\frac{k(k+1)+2(k+1)}{2} = \frac{k^2+k+2k+2}{2} = \frac{(k+1)(k+2)}{2}.$$

■

Aufgabe 5.2 — Vorrechnen. Gegeben sei die Folge a_n mit $a_0 = 7$ und $a_n = a_{n-1} + 2$. Finden Sie eine geschlossene Form für a_n. ■

Beweis. Der Bottom-Up-Ansatz gibt uns $a_1 = a_0 + 2 = 7 + 2$, $a_2 = a_1 + 2 = 7 + 2 + 2$, $a_3 = a_2 + 2 = 7 + 2 + 2 + 2$. Es sieht also nach $a_n = 7 + 2n$ aus.

Der Top-Down-Ansatz gibt uns

$$a_n = a_{n-1} + 2 = a_{n-2} + 2 + 2 = a_{n-3} + 2 + 2 + 2 = \ldots = a_0 + 2n = 7 + 2n.$$

Wir behaupten also $a_n = 2n + 7$.

Wir beweisen die Aussage per Induktion:
IA: Für $n = 0$ gilt $a_n =_0 = 7 = 2 \cdot 0 + 7 = 2n + 7$.
IV: Es gelte $a_k = 2k + 7$ für ein $k \in \mathbb{N}_0$.
IS: Es gilt

$$a_{k+1} \underbrace{=}_{\text{Def.}} a_k + 2 \underbrace{=}_{\text{IV}} 2k + 7 + 2 = 2(k+1) + 7.$$

■

Aufgabe 5.3 Beweisen Sie: Für alle $n \in \mathbb{N}_0$ und alle reellen Zahlen $q \in \mathbb{R}$ mit $q \neq 1$ gilt, dass $\sum_{i=0}^{n} q^i = \frac{q^{n+1}-1}{q-1}$. ■

Beweis. Sei $q \in \mathbb{R}$ mit $q \neq 1$. Wir beweisen per Induktion:
IA: Für $n = 0$ gilt $\sum_{i=0}^{n} q^i = \sum_{i=0}^{0} q^i = q^0 = 1 = \frac{q-1}{q-1} = \frac{q^{0+1}-1}{q-1} = \frac{q^{n+1}-1}{q-1}$.
IV: Es gelte $\sum_{i=0}^{k} q^i = \frac{q^{k+1}-1}{q-1}$ für ein $k \in \mathbb{N}_0$.
IS: Es gilt

$$\sum_{i=0}^{k+1} q^i = [\sum_{i=0}^{k} q^i] + q^{k+1} \underbrace{=}_{\text{IV}} \frac{q^{k+1}-1}{q-1} + q^{k+1} = \frac{q^{k+1}-1+(q-1)q^{k+1}}{q-1} =$$
$$\frac{q^{k+1}-1+q^{k+2}-q^{k+1}}{q-1} = \frac{q^{k+2}-1}{q-1} = \frac{q^{(k+1)+1}-1}{q-1}.$$

■

Aufgabe 5.4 Gegeben sei die Folge a_n mit $a_0 = 3$ und $a_n = 2a_{n-1}$. Finden Sie eine geschlossene Form für a_n. ■

Beweis. Der Bottom-Up-Ansatz gibt uns $a_1 = 2 \cdot a_0 = 2 \cdot 3$, $a_2 = 2 \cdot a_1 = 2 \cdot 2 \cdot 3$ und $a_3 = 2a_2 = 2 \cdot 2 \cdot 2 \cdot 3$, also vermuten wir $a_n = 3 \cdot 2^n$.

Der Top-Down-Ansatz gibt uns

$$a_n = 2a_{n-1} = 2 \cdot 2a_{n-2} = 2 \cdot 2 \cdot 2a_{n-3} = \ldots = 2^n a_0 = 3 \cdot 2^n.$$

Wir behaupten also, dass $a_n = 3 \cdot 2^n$.

Wir beweisen per Induktion:

IA: Für $n = 0$ gilt $a_n = a_0 = 3 = 3 \cdot 1 = 3 \cdot 2^0 = 3 \cdot 2^n$.

IV: Es gelte $a_k = 3 \cdot 2^k$ für ein $k \in \mathbb{N}_0$.

IS: Es gilt $a_{k+1} \underbrace{=}_{\text{Def.}} 2 \cdot a_k \underbrace{=}_{\text{IV}} 2 \cdot 3 \cdot 2^k = 3 \cdot 2^{k+1}$.

■

Aufgabe 5.5 Beweisen Sie: Für alle $n \in \mathbb{N}_0$ gilt, dass $F_n \leq 2^n$. ■

Beweis. Wir beweisen per Induktion. Da wir in der rekursiven Definition $F_n = F_{n-1} + F_{n-2}$ auf zwei vorherige Werte zugreifen, müssen wir den Induktionsanfang für $n = 0$ und $n = 1$ zeigen. Außerdem benutzen wir starke Induktion.

IA: Für $n = 0$ gilt $F_n = F_0 = 1 \leq 2^0 = 2^n$ und für $n = 1$ gilt $F_n = F_1 = 1 \leq 2 = 2^1 = 2^n$.

IV: Es gelte $F_{k'} \leq 2^{k'}$ für ein $k \in \mathbb{N}_0$ mit $k \geq 1$ und alle $k' \in \mathbb{N}_0$ mit $k' \leq k$. So kann man alle Voraussetzungen $H(k), H(k-1), \ldots, H(0)$ kompakt aufschreiben.

IS: Es gilt

$$F_{k+1} \underbrace{=}_{\text{Def.}} F_k + F_{k-1} \underbrace{\leq}_{\text{IV}} 2^k + F_{k-1} \underbrace{\leq}_{\text{IV}} 2^k + 2^{k-1} \leq 2^k + 2^k = 2 \cdot 2^k = 2^{k+1}.$$

■

Aufgabe 5.6 Gegeben sei die Folge a_n mit $a_0 = 1$, $a_1 = 2$ und $a_n = a_{n-1} + 2a_{n-2}$. Finden Sie eine geschlossene Form für a_n. ■

Beweis. Der Bottom-Up-Ansatz gibt uns $a_2 = a_1 + 2a_0 = 2 + 2 \cdot 1 = 4 = 2^2$, $a_3 = a_2 + 2a_1 = 4 + 2 \cdot 2 = 8 = 2^3$, $a_4 = a_3 + 2a_2 = 8 + 2 \cdot 4 = 16 = 2^4$. Wir vermuten also $a_n = 2^n$.

Der Top-Down-Ansatz ist hier wenig hilfreich. Zumindest sehe ich keinen hilfreichen Ansatz.

Wir beweisen per Induktion, wobei wir wieder zwei Anfänge brauchen:

IA: Für $n = 0$ gilt $a_n = a_0 = 1 = 2^0 = 2^n$ und für $n = 1$ gilt $a_n = a_1 = 2 = 2^1 = 2^n$.

IV: Es gelte $a_{k'} = 2^{k'}$ für ein $k \in \mathbb{N}_0$ mit $k \geq 1$ und alle $k' \in \mathbb{N}_0$ mit $k' \leq k$.

IS: Es gilt

$$a_{k+1} \underbrace{=}_{\text{Def.}} a_k + 2a_{k-1} \underbrace{=}_{\text{IV}} 2^k + 2a_{k-1} \underbrace{=}_{\text{IV}} 2^k + 2 \cdot 2^{k-1} = 2^k + 2^k = 2^{k+1}.$$

■

Aufgabe 5.7 Beweisen Sie: Für alle $n \in \mathbb{N}_0$ gilt, dass $2n^3+3n^2+n$ durch 6 teilbar ist. ■

Beweis. Wir beweisen per Induktion:
IA: Für $n=0$ gilt $2n^3+3n^2+n=0$, und die 0 ist offenbar durch 6 teilbar.
IV: Es sei $2k^3+3k^2+k$ durch 6 teilbar für ein $k \in \mathbb{N}_0$.
IS: Es gilt

$$\begin{aligned}
&2(k+1)^3+3(k+1)^2+k+1 = \\
&2(k+1)\cdot(k+1)\cdot(k+1)+3(k+1)\cdot(k+1)+k+1 = \\
&2\cdot(k^2+2k+1)(k+1)+3(k^2+2k+1)+k+1 = \\
&2(k^3+2k^2+k+k^2+2k+1)+3(k^2+2k+1)+k+1 = \\
&2k^3+9k^2+13k+6 = \\
&(2k^3+3k^2+k)+(6k^2+12k+6).
\end{aligned}$$

Nach Induktionsvoraussetzung ist $2k^3+3k^2+k$ durch 6 teilbar. Offenbar gilt $6k^2+12k+6 = 6(k^2+2k+1)$. Da k^2+2k+1, ist also auch $6k^2+12k+6$ durch 6 teilbar. Die Summe zweier durch 6 teilbarer Zahlen ist auch durch 6 teilbar und somit ist $2(k+1)^3+3(k+1)^2+k+1$ durch 6 teilbar. Hier sieht man gut die generelle Idee: Wenn man nicht gut weiterkommt, erst einmal alles ausrechnen.

■

Aufgabe 5.8 Gegeben sei die Folge a_n mit $a_0=2$ und $a_n = 2a_{n-1}+1$. Finden Sie eine geschlossene Form für a_n. ■

Beweis. Der Bottom-Up-Ansatz gibt uns $a_1 = 2a_0+1 = 2\cdot 2+1$, $a_2 = 2a_1+1 = 2\cdot(2\cdot 2+1)+1 = 2\cdot 2\cdot 2+2+1$, $a_3 = 2a_2+1 = 2(2\cdot 2\cdot 2+2+1)+1 = 2\cdot 2\cdot 2\cdot 2+2\cdot 2+2+1$. Hier sehen wir nicht direkt ein, wie sich die Folge verändert.

Der Top-Down-Ansatz gibt uns

$$\begin{aligned}
&a_n = 2a_{n-1}+1 = 2\cdot(2a_{n-2}+1)+1 = 2\cdot 2\cdot a_{n-2}+3 = 2\cdot 2\cdot(2a_{n-3}+1)+3 = \\
&2\cdot 2\cdot 2a_{n-3}+4+3 = 2\cdot 2\cdot 2\cdot(2a_{n-4}+1)+7 = 2\cdot 2\cdot 2\cdot 2\cdot a_{n-4}+15 = \ldots = \\
&2^n a_0+2^{n-1}-1 = 2^{n+1}+2^n-1.
\end{aligned}$$

Wir vermuten also $a_n = 2^{n+1}+2^n-1 = 2^n(2+1)-1 = 3\cdot 2^n-1$. Hier sieht man, dass wir fast die gleiche geschlossene Form wie oben haben, aber die Struktur viel schwerer aussieht.

Wir beweisen per Induktion:
IA: Für $n=0$ gilt $a_n = a_0 = 2 = 3\cdot 1-1 = 3\cdot 2^0-1 = 3\cdot 2^n-1$.
IV: Es gelte $a_k = 3\cdot 2^k-1$ für ein $k \in \mathbb{N}_0$.
IS: Es gilt

$$a_{k+1} \underbrace{=}_{\text{Def.}} 2a_k+1 \underbrace{=}_{\text{IV}} 2(3\cdot 2^k-1)+1 = 3\cdot 2^{k+1}-2+1 = 3\cdot 2^{k+1}-1.$$

■

Aufgabe 5.9 Beweisen Sie: Für alle $n \in \mathbb{N}_0$ gilt, dass $\sum_{i=0}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$. ■

Beweis. Wir beweisen per Induktion:
IA: Für $n=0$ gilt $\sum_{i=0}^{n} i^2 = \sum_{i=0}^{0} i^2 = 0 = \frac{0\cdot 1\cdot 1}{6} = \frac{n(n+1)(2n+1)}{6}$.

IV: Es gelte $\sum_{i=0}^{k} i^2 = \frac{k(k+1)(2k+1)}{6}$ für ein $k \in \mathbb{N}_0$.
IS: Es gilt

$$\sum_{i=0}^{k+1} i^2 = [\sum_{i=0}^{k} i^2] + (k+1)^2 \underbrace{=}_{\text{IV}} \frac{k(k+1)(2k+1)}{6} + (k+1)^2 =$$
$$\frac{k(k+1)(2k+1)+6(k+1)^2}{6} = \frac{(k^2+k)(2k+1)+6(k+1)^2}{6} =$$
$$\frac{2k^3+k^2+2k^2+k+6k^2+12k+6}{6} = \frac{2k^3+9k^2+13k+6}{6}.$$

Außerdem gilt

$$(k+1)(k+2)(2(k+1)+1) = (k+1)(k+2)(2k+3) = (k^2+2k+k+2)(2k+3) =$$
$$2k^3+3k^2+4k^2+6k+2k^2+3k+4k+6 = 2k^3+9k^2+13k+6.$$

Somit gilt $\sum_{i=0}^{k+1} i^2 = \frac{(k+1)(k+2)(2(k+1)+1)}{6}$.

■

Aufgabe 5.10 Gegeben sei die Folge a_n mit $a_0 = 1$ und $a_n = a_{n-1} + 6n - 3$. Finden Sie eine geschlossene Form für a_n. ■

Beweis. Der Bottom-Up-Ansatz gibt $a_1 = a_0 + 6 - 3 = 1 + 6 - 3 = 4$, $a_2 = a_1 + 12 - 3 = 4 + 12 - 3 = 13$, $a_3 = a_2 + 18 - 3 = 13 + 18 - 3 = 28$. Hier sieht man nicht so viel.

Der Top-Down-Ansatz ist hilfreicher und gibt

$$a_n = a_{n-1} + 6n - 3 = a_{n-2} + 6(n-1) - 3 + 6n - 3 = a_{n-2} + 6(n + (n-1)) - 2 \cdot 3 =$$
$$a_{n-3} + 6(n + (n-1) + (n-2)) - 3 \cdot 3 = \ldots = a_0 + 6(\sum_{i=1}^{n} i) - 3n = 1 + 6(\sum_{i=1}^{n} i) - 3n.$$

Wir wissen, dass $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$, also gilt $1 + 6(\sum_{i=1}^{n} i) - 3n = 1 + 3n(n+1) - 3n = 3n^2 + 1$. Also vermuten wir, dass $a_n = 3n^2 + 1$ gilt.

Wir beweisen per Induktion:
IA: Für $n = 0$ gilt $a_n = a_0 = 1 = 3 \cdot 0^2 + 1 = 3n^2 + 1$.
IV: Es gelte $a_k = 3k^2 + 1$ für ein $k \in \mathbb{N}_0$.
IS: Es gilt

$$a_{k+1} \underbrace{=}_{\text{Def.}} a_k + 6(k+1) - 3 \underbrace{=}_{\text{IV}} 3k^2 + 1 + 6k + 6 - 3 = 3k^2 + 6k + 4$$

Nun gilt auch $3(k+1)^2 + 1 = 3k^2 + 6k + 3 + 1 = 3k^2 + 6k + 4$, also $a_{k+1} = 3(k+1)^2 + 1$.

■

Aufgabe 5.11 Beweisen Sie: Für alle $n \in \mathbb{N}_0$ mit $n \geq 6$ gilt, dass $F_n \geq (3/2)^n$. ■

Beweis. Wir beweisen per Induktion. Auch hier brauchen wir zwei Induktionsanfänge, nämlich für $n = 6$ und $n = 7$. Es gilt $F_6 = 13$ und $F_{12} = 21$.
IA: Für $n = 6$ gilt $F_n = F_6 = 13 \geq 12 \geq 729/64 = (3/2)^6 = (3/2)^n$ und für $n = 7$ gilt $F_n = F_7 = 21 \geq 18 \geq 2187/128 = (3/2)^7 = (3/2)^n$.
IV: Es gelte $F_{k'} \geq (3/2)^{k'}$ für ein $k \in \mathbb{N}_0$ mit $k \geq 7$ und alle $k' \in \mathbb{N}_0$ mit $k' \leq k$.

IS: Es gilt

$$F_{k+1} = F_k + F_{k-1} \underbrace{\geq}_{\text{IV}} (3/2)^k + F_{k-1} \underbrace{\geq}_{\text{IV}} (3/2)^k + (3/2)^{k-1} =$$
$$(3/2)^{k-1}(3/2+1) = (3/2)^{k-1}(5/2) = (3/2)^{k-1}(10/4) \geq$$
$$(3/2)^{k-1}(9/4) = (3/2)^{k-1}(3/2)^2 = (3/2)^{k+1}.$$

■

Aufgabe 5.12 Gegeben sei die Folge a_n mit $a_0 = 2$ und $a_n = a_{n-1} + 21n^2 - 21n + 7$. Finden Sie eine geschlossene Form für a_n. ■

Beweis. Der Bottom-Up-Ansatz gibt $a_1 = 2 + 21 - 21 + 7 = 9$, $a_2 = 9 + 21 \cdot 4 - 21 \cdot 2 + 7 = 58$, $a_3 = 58 + 21 \cdot 9 - 21 \cdot 3 + 7 = 191$. Hier sehen wir nicht allzuviel.

Beim Top-Down-Ansatz ergibt sich:

$$a_n = a_{n-1} + 21n^2 - 21n + 7 = a_{n-2} + 21(n-1)^2 - 21(n-1) + 7 + 21n^2 - 21n + 7 =$$
$$a_{n-3} + 21(n-2)^2 - 21(n-2) + 7 + 21(n-1)^2 - 21(n-1) + 7 + 21n^2 - 21n + 7 = \ldots =$$
$$a_0 + 21(\sum_{i=1}^{n} i^2) - 21(\sum_{i=1}^{n} i) + 7n = 2 + 21(\sum_{i=1}^{n} i^2) - 21(\sum_{i=1}^{n} i) + 7n.$$

Nun wissen wir, dass $\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$ und $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$ gilt. Also können wir den obigen Term auch schreiben als

$$2 + 21(\sum_{i=1}^{n} i^2) - 21(\sum_{i=1}^{n} i) + 7n = 2 + 21 \cdot \frac{n(n+1)(2n+1)}{6} - 21 \cdot \frac{n(n+1)}{2} + 7n =$$
$$2 + 21 \cdot \frac{2n^3 + 3n^2 + n}{6} - 21 \cdot \frac{n^2 + n}{2} + 7n =$$
$$2 + \frac{42n^3 + 63n^2 + 21n}{6} - \frac{21n^2 + 21n}{2} + 7n =$$
$$2 + 7n^3 + (63/6)n^2 + (21/6)n - (21/2)n^2 - (21/2)n + 7n =$$

Nun gilt $63/6 = 21/2$, also fallen die quadratischen Terme weg. Weiterhin gilt $(21/6) + 7 = (21/6) + (42/6) = (63/6) = (21/2)$, also fallen auch die linearen Terme weg und es bleibt $7n^3 + 2$ übrig. Wir vermuten also $a_n = 7n^3 + 2$.

Wir beweisen per Induktion:

IA: Für $n = 0$ gilt $a_n = a_0 = 2 = 7 \cdot 0^3 + 2 = 7n^3 + 2$.

IV: Es gelte $a_k = 7k^3 + 2$ für ein $k \in \mathbb{N}_0$.

IS: Es gilt

$$a_{k+1} \underbrace{=}_{\text{Def.}} a_k + 21(k+1)^2 - 21(k+1) + 7 \underbrace{=}_{\text{IV}} 7k^3 + 2 + 21(k+1)^2 - 21(k+1) + 7 =$$
$$7k^3 + 2 + 21(k^2 + 2k + 1) - 21(k+1) + 7 = 7k^3 + 2 + 21k^2 + 42k + 21 - 21k - 21 + 7 =$$
$$7k^3 + 21k^2 + 21k + 9.$$

Andererseits gilt

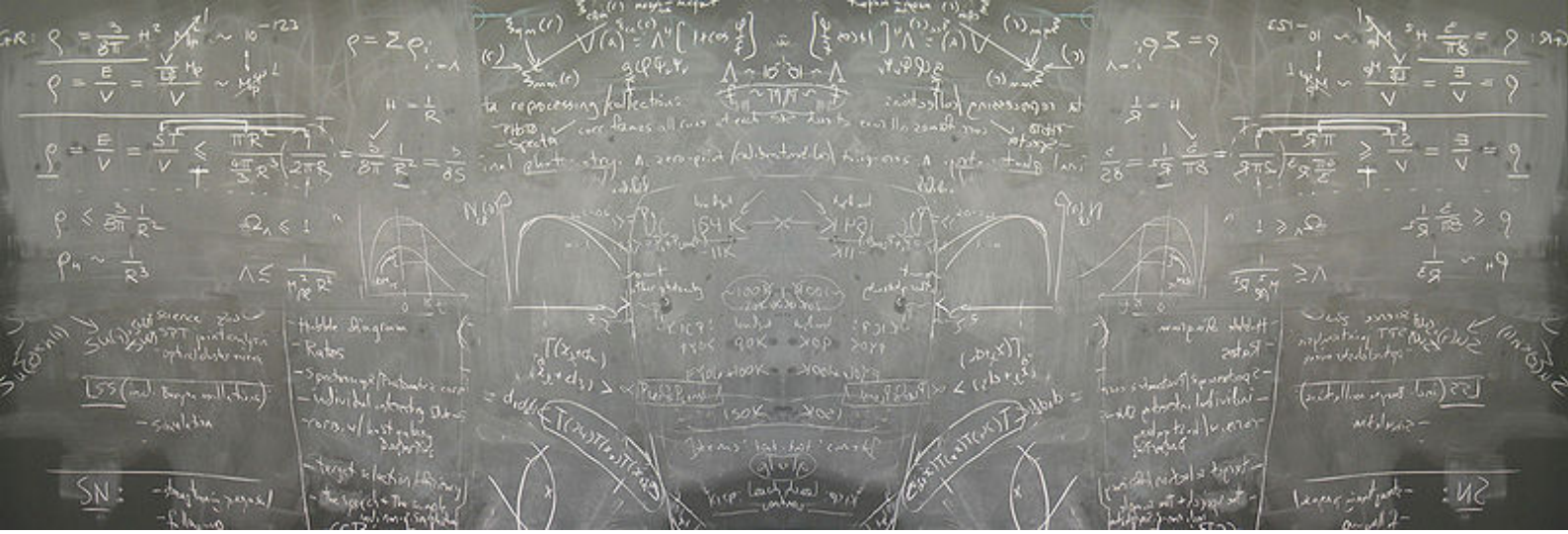
$$7(k+1)^3 + 2 = 7(k+1)(k^2 + 2k + 1) + 2 =$$
$$7(k^3 + 2k^2 + k + k^2 + 2k + 1) + 2 = 7k^3 + 21k^2 + 21k + 9.$$

Somit gilt $a_{k+1} = 7(k+1)^3 + 2$.

■

Lernziele:

- Sicherer Umgang mit Induktionsbeweisen
- Gutes Verständnis für die Unterteilung von Induktionsbeweisen in Anfang, Voraussetzung und Schritt
- Grundlegendes Verständnis von Folgen und Reihen
- Sicherer Umgang mit rekursiv definierten Folgen
- Rekursive Definition und geschlossene Form unterscheiden können
- Erste Übungen im systematischen Lösen von Rekurrenzgleichungen



6. Funktionen, O-Notation, Modulare Arithmetik

Zum Lesen:
Kapitel 3.1, 3.2, 5.3 in [Bel18]
Ausarbeitung von Sebastian zu Funktionen (also dieses Dokument)

Stichworte:

- Funktionen; Domain (Definitionsmenge, Urbildmenge); Target (Zielmenge); Range (Bildmenge)
- Eigenschaften von Funktionen: injektiv, surjektiv, bijektiv
- Grundlegende Funktionen: Polynome, Exponentialfunktion, Logarithmus; Rechenregeln
- Modulare Arithmetik; Teilbarkeit; Rest
- Äquivalenzklassen; Symmetrie; Transitivität; Reflexivität; Partition
- O-Notation zur Bestimmung des Wachstums von Funktionen

Definition 6.1 Für eine reelle Zahl $x \in \mathbb{R}$ ist $\lfloor x \rfloor$ die größte ganze Zahl kleiner oder gleich x und $\lceil x \rceil$ die kleinste ganze Zahl größer oder gleich x. Also gilt $\lfloor 4{,}5 \rfloor = 4$ und $\lceil 4{,}5 \rceil = 5$. Eselsbrücke: $\lceil \cdot \rceil$ sieht aus wie ein Dach und $\lfloor \cdot \rfloor$ wie der Fußboden.

Definition 6.2 Für die Modulo-Operation wird in [Bel18] die Notation $a \equiv b \pmod{m}$ benutzt, wenn a geteilt durch n den gleichen Rest hinterlässt wie b geteilt durch n. Dies ist die typische Schreibweise in der Mathematik, aber in der Informatik schreibt man auch $a \bmod m = b$, und interpretiert $\bmod m$ also als Funktion.

6.1 Funktionen und O-Notation

Im folgenden Abschnitt stellen wir zunächst die wichtigsten speziellen Funktionen vor, die wir so brauchen werden. Zusätzlich werden wir die wichtigsten Rechenregeln für diese Funktionen wiederholen. Danach geht es darum, das Wachstum dieser Funktionen zu vergleichen. Dieser Vergleich geschieht mithilfe der sogenannten *O-Notation*.

6.1.1 Wichtige Funktionen

Polynome

Eine einfache Klasse von Funktionen sind die sogenannten *Polynome*. Eine solche Funktion $f\colon \mathbb{R} \to \mathbb{R}$ lässt sich darstellen als $f(x) = \sum_{i=0}^{n} a_i x^i$, also haben wir $f(x) = a_0 + a_1 x^1 + a_2 x^2 + \ldots + a_n x^n$. Hierbei nennen wir die reellen Zahlen a_i die *Koeffizienten* des Polynoms.

■ Beispiel 6.3

- Die Funktion $f\colon \mathbb{R} \to \mathbb{R}$ mit $f(x) = 2x+1$ ist ein Polynom. Hierbei ist $n = 1$, $a_0 = 1$ und $a_1 = 2$.
- Auch die Funktion $f\colon \mathbb{R} \to \mathbb{R}$ mit $f(x) = x^3 + 4x^2 - 7x + (1/2)$ ist ein Polynom mit $n = 3$, $a_0 = 1/2$, $a_1 = -7$, $a_2 = 4$ und $a_3 = 1$.

◇

Polynome sind die typischen Funktionen, mit denen in der Schule gerechnet wird. Der *Grad* eines Polynoms $f\colon \mathbb{R} \to \mathbb{R}$ mit $f(x) = \sum_{i=0}^{n} a_i x^i$ ist n und somit die höchste vorkommende Potenz von x. Multiplikation und Addition funktionieren genauso, wie wir es bereits aus der Schule gewohnt sind: Bei der Addition addieren wir einfach die Koeffizienten; bei der Multiplikation müssen wir für die Potenz x^i alle Koeffizienten a_j und b_{i-j} multiplizieren und dann aufaddieren.

■ Beispiel 6.4 — Polynom-Multiplikation. Mit $f(x) = 2x+1$ und $g(x) = 7x^2 + x + 3$ gilt also $f(x) + g(x) = 7x^2 + 3x + 4$ und

$$f(x) \cdot g(x) = (2x+1)(7x^2+x+3) = 14x^3 + 2x^2 + 6x + 7x^2 + x + 3 = 14x^3 + 9x^2 + 7x + 3.$$

Der Term $9x^2$ ergibt sich also daraus, dass wir den konstanten Term 1 von f mit dem quadratischen Term $7x^2$ von g multiplizieren und den linearen Term $2x$ von f mit dem linearen Term x von g multiplizieren. Letztendlich addieren wir dann die beiden Terme $1 \cdot 7x^2 + 2x \cdot x = 7x^2 + 2x^2 = 9x^2$ und erhalten so den quadratischen Term von $f(x) \cdot g(x)$.

Der Term a_0 heißt *konstanter* Term, der Term $a_1 x$ heißt *linearer* Term, der Term $a_2 x^2$ heißt *quadratischer* Term und $a_3 x^3$ heißt *kubischer* Term. ◇

Polynome sind auch die wichtigsten Funktionen, die uns bei der Analyse von Algorithmen und Datenstrukturen interessieren werden. Ein sicherer Umgang mit ihnen ist also unerlässlich. Wer sich also noch nicht sicher im Umgang damit fühlt, der möge sich ein paar Polynome ausdenken und mit ihnen Rechnen üben. Auch findet man weitere interessante Beispiele und Informationen bei Wikipedia.

Zusätzlich zu den Polynomen gibt es jedoch noch zwei andere Arten von Funktionen, auf die wir immer wieder treffen werden, die *Exponentialfunktion* und ihr Inverses, der *Logarithmus*.

Exponentialfunktion

Für eine reelle Zahl $b \in \mathbb{R}$ (die sogenannte *Basis*) und eine beliebige Zahl $x \in \mathbb{R}$ (der *Exponent*) können wir die Zahl b^x berechnen. In der Informatik ist x meistens eine natürliche Zahl und wir können uns dann b^x als x-fache Multiplikation von b vorstellen, es gilt also

$$b^x = \underbrace{b \cdot b \cdot \ldots \cdot b}_{x\text{-mal}}.$$

Damit können wir sehr leicht eine der grundlegenden Regeln für die Exponentialfunktion sehen:

$$b^x \cdot b^y = \underbrace{b \cdot b \cdot \ldots \cdot b}_{x\text{-mal}} \cdot \underbrace{b \cdot b \cdot \ldots \cdot b}_{y\text{-mal}} = \underbrace{b \cdot b \cdot \ldots \cdot b}_{x+y\text{-mal}} = b^{x+y}.$$

Aber auch für nicht-natürliche x kann man den Ausdruck b^x definieren und es gelten folgende wichtige Rechenregeln:

(i) Es gilt $b^0 = 1$ für alle $b \neq 0$.
(ii) Es gilt $0^x = 0$ für all $x \neq 0$.
(iii) Der Wert 0^0 ist nicht sinnvoll definiert, also ignorieren wir ihn einfach.
(iv) Es gilt $b^x \cdot b^y = b^{x+y}$ für alle Zahlen x, y.
(v) Es gilt $(b^x)^y = b^{x \cdot y}$ für alle Zahlen x, y. Wichtig hierbei ist, dass $(b^x)^y$ etwas anderes als $b^{x^y} = b^{(x^y)}$ ist. Für $b = 2$, $x = 2$ und $y = 3$ ist $(b^x)^y = (2^2)^3 = 4^3 = 64$, aber $b^{x^y} = 2^{2^3} = 2^8 = 256$.

Logarithmus

Wenn man nun eine positive reelle Basis b und eine positive reelle Zahl y gegeben hat, stellt sich die Frage, ob es denn immer eine Zahl x gibt, so dass $b^x = y$ gilt. Und in der Tat gibt es immer eine solche Zahl, die wir als $\log_b(y)$ den *Logarithmus von y zur Basis b*. Es gilt also immer $b^{\log_b(x)} = x$. Behält man diese Definition im Kopf, kommt man schon ziemlich weit. Zum Beispiel können wir dann auch die wichtigste Rechenregel des Logarithmus sehr einfach herleiten:

$$\log_b(b^x \cdot b^y) = \log_b(b^{x+y}) = x + y = \log_b(b^x) + \log_b(b^y).$$

Also gilt $\log_b(r \cdot s) = \log_b(r) + \log_b(s)$ für alle Zahlen r, s. Daraus kann man dann auch leicht sehen, dass $\log_b(x^y) = y \cdot \log_b(x)$ gilt, Exponenten kann man also vor den Logarithmus ziehen.

Die wichtigste Basis in der Informatik ist natürlich $b = 2$, weshalb man auch häufig einfach log statt $\log_2$ schreibt. Diese Funktion nennt man den *binären Logarithmus* (selten auch: logarithmus dualis). Wie beschrieben, rechnen wir in der Informatik meist mit ganzen Zahlen. Das heißt, wir betrachten nicht den Wert $\log_2(15) \approx 3.9068905956085187$, sondern vielmehr $\lceil \log_2(15) \rceil = 4$. Diese Funktion hat zwei andere, einfache Interpretationen. Zunächst einmal beschreibt $\lceil \log_2(x+1) \rceil$ die Anzahl an Bits, die man braucht, um x in Binärdarstellung zu schreiben. Denn 15 in Binärdarstellung ist $(1111)_2$ und hat somit Länge $\lceil \log(15+1) \rceil = \lceil \log(16) \rceil = 4$. Die 16 hat hingegen die Binärdarstellung (10000) und somit Länge $\lceil \log(16+1) \rceil = \lceil \log(17) \rceil = 5$. Andererseits gibt $\lceil \log_2(x) \rceil - 1$ an, wie häufig man x ganzzahlig durch 2 teilen muss, bis man bei der 1 angekommen ist. Zum Beispiel gibt einem ganzzahliges Teilen von 15 durch 2 die Zahl 7. Ganzzahliges Teilen von 7 durch 2 gibt 3 und ganzzahliges Teilen von 3 durch 2 ergibt 1. Also mussten wir die 15 dreimal teilen. Es gibt sehr, sehr viele Algorithmen, in denen wir ein Problem in zwei gleichgroße Teilprobleme zerlegen und diese dann einzeln lösen. Die *binäre Suche* ist sicherlich das bekannteste solche Verfahren. Aufgrund der Häufigkeit dieses Teilen-und-Herrschen-Ansatzes taucht auch der Logarithmus in vielen Laufzeitabschätzungen auf.

6.1.2 O-Notation

Im Folgenden betrachten wir nur noch Funktionen, die natürliche Zahlen auf natürliche Zahlen abbilden. Diese sind für die Analyse von Algorithmen besonders interessant. Wenn wir einen Algorithmus A haben, der irgendein Problem löst, möchten wir gerne wissen, wieviele Schritte er dafür braucht. Natürlich hängt die Anzahl dieser Schritte von der Größe der Eingabe ab. Um ein Array der Länge 20 zu sortieren, wird man viel weniger Schritte brauchen als für ein Array der Länge 42000000000000000000. Somit weisen wir also einer Eingabegröße, beschrieben durch eine natürliche Zahl n, die Anzahl an Schritten $T_A(n)$ zu, die der Algorithmus A bei der Lösung eines Problemes mit Eingabegröße n braucht. Dies nennen wir die *Laufzeit* von A und wir werden uns noch intensiver damit beschäftigen. Wenn wir nun zwei Algorithmen A und B gegeben haben, möchten wir natürlich gerne entscheiden, welcher der beiden Algorithmen schneller ist, also ob $T_A(n)$ kleiner als $T_B(n)$ gilt. Häufig ist die Antwort darauf jedoch nicht so leicht. Für einige Fälle ist das völlig klar: Gilt $T_A(n) = 3n + 1$ und $T_B(n) = 2n$, so gilt $T_B(n) \leq T_A(n)$ für alle $n \in \mathbb{N}_0$, also ist B schneller als A. Aber wie sieht es aus mit den Laufzeiten $T_A(n) = n^2$ und $T_B(n) = 4n + 7$? Für $n \in \{0, 1, \ldots, 5\}$ gilt $T_A(n) < T_B(n)$ und für $n \geq 6$ gilt $T_B(n) \leq T_A(n)$. Aus einem intuitiven Standpunkt heraus betrachten wir T_B als die kleinere Funktion, denn der höchste auftretende

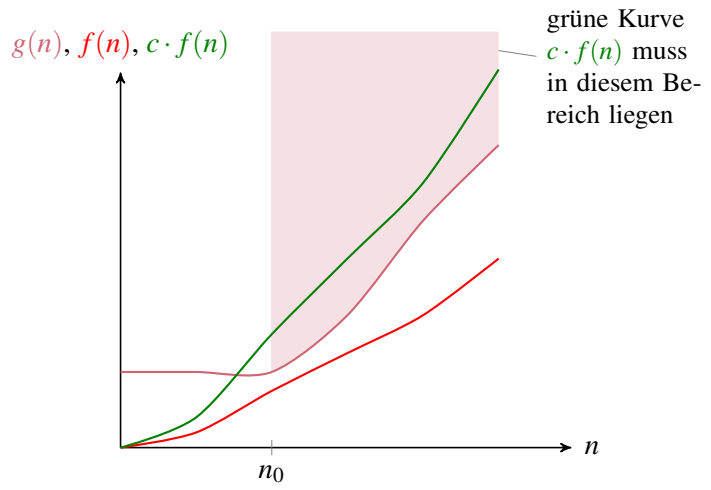


Abbildung 6.1: Skizze zur Definition der O-Notation

Exponent ist kleiner und ab einem bestimmten Zeitpunkt n_0 ist T_B auch wirklich immer kleiner als T_A. Die Einsicht, dass es einen solche Zeitpunkt n_0 gibt, führt uns zur Definition der O-Notation. Wir werden jedoch sogar noch etwas mehr Flexibilität einführen. Da die O-Notation für allgemeine Funktionen funktioniert (nicht nur für solche, die die Laufzeit von Algorithmen messen), werden wir uns im Folgenden wieder auf beliebige Funktionen, die natürliche Zahlen auf natürliche Zahlen abbilden, beschäftigen.

Definition 6.5 — O-Notation. Sei $f\colon \mathbb{N}_0 \to \mathbb{N}_0$ eine Funktion. Die Menge $O(f)$ ist definiert als die Menge aller Funktionen $g\colon \mathbb{N} \to \mathbb{N}_0$, so dass es eine positive Zahl $c \in \mathbb{R}_{>0}$ und eine natürliche Zahl $n_0 \in \mathbb{N}_0$ gibt mit $g(n) \leq c \cdot f(n)$ für alle $n \geq n_0$.

In Formeln ausgedrückt, gilt also

$$O(f) = \{g\colon \mathbb{N}_0 \to \mathbb{N}_0 \mid \exists c \in \mathbb{R}_{>0}\ \exists n_0 \in \mathbb{N}_0\ \forall n \in \mathbb{N}_0, n \geq n_0 : g(n) \leq c \cdot f(n)\}.$$

Dies ist wahrscheinlich eine der kompliziertesten Formeln, die uns begegnen wird. Es ist aber sehr wichtig, sie genau zu verstehen!

Somit heißt $g \in O(f)$, dass es eine Konstante c gibt, so dass $c \cdot f(n)$ für genügend große n immer größer als $g(n)$ ist. Das Prinzip wird im Abbildung 6.1 noch einmal verdeutlicht: Es kann durchaus sein, dass $f(n) \leq g(n)$ gilt, solange $g(n) \leq c \cdot f(n)$ gilt.

Gucken wir uns zunächst einmal ein paar positive Beispiele an:

■ **Beispiel 6.6 —** $g \in O(f)$.

- Für $g(n) = 2n$ und $f(n) = 3n+1$ gilt $g \in O(f)$, denn wir können $c = 1$ und $n_0 = 0$ wählen und somit gilt $g(n) = 2n \leq 3n+1 = f(n) = c \cdot f(n)$ für alle $n \geq n_0 = 0$.
- Für $g(n) = 4n$ und $f(n) = n^2$ gilt $g \in O(f)$, denn wir können $c = 4$ und $n_0 = 0$ wählen und somit gilt $g(n) = 4n \leq 4 \cdot n^2 = c \cdot f(n)$ für alle $n \geq n_0 = 0$.
- Für $g(n) = 5n+2$ und $f(n) = n^2$ gilt $g \in O(f)$, denn wir können $c = 7$ und $n_0 = 1$ wählen und somit gilt $g(n) = 5n+2 \leq 5n+2n = 7n \leq 7n^2 = c \cdot f(n)$ für alle $n \geq n_0 = 0$.
- Für $g(n) = 100000000000n^2$ und $f(n) = n^3$ gilt $g \in O(f)$, denn wir können $c = 100000000000$ und $n_0 = 1$ wählen und somit gilt $g(n) = 100000000000n^2 \leq 100000000000n^3 = c \cdot f(n)$.
- Für $g(n) = n^2$ und $f(n) = 2^n$ gilt $g \in O(f)$. Wähle $c = 1$ und $n_0 = 4$. Nun müssen wir also zeigen, dass $n^2 \leq 2^n$ für alle $n \geq n_0$ gilt. Dies werden wir mithilfe von vollständiger Induktion beweisen:

Induktionsanfang: Für $n = 4$ gilt $n^2 = 4^2 = 16 \leq 16 = 2^4 = 2^n$, also $n^2 \leq 2^n$.
Induktionsvoraussetzung: Für ein festes $k \in \mathbb{N}_0$ mit $k \geq 4$ gelte $k^2 \leq 2^k$.
Induktionsschritt: Wir müssen nun zeigen, dass $(k+1)^2 \leq 2^k$. Es gilt $(k+1)^2 = k^2 + 2k + 1$. Aus den Aufgaben der letzten Woche wissen wir, dass $2k+1 \leq k^2$ für $k \geq 3$ gilt. Also folgt $(k+1)^2 \leq k^2 + k^2 \leq 2k^2$. Nach Induktionsvoraussetzung gilt $k^2 \leq 2^k$ und somit $2k^2 \leq 2 \cdot 2^k = 2^{k+1}$. Insgesamt gilt also $(k+1)^2 \leq 2^{k+1}$.

◇

Nun betrachten wir ein paar Beispiele, bei denen $g \notin O(f)$ liegt. Dazu müssen wir uns zunächst einmal überlegen, was dass denn bedeutet. Betrachten wir formal die Definition der O-Notation und negieren sie, gilt $g \notin O(f)$ genau dann, wenn

$$\neg[\exists c \in \mathbb{R}_{>0}\ \exists n_0 \in \mathbb{N}_0\ \forall n \in \mathbb{N}_0, n \geq n_0 : g(n) \leq c \cdot f(n)] =$$
$$\forall c \in \mathbb{R}_{>0}\ \forall n_0 \in \mathbb{N}_0\ \exists n \in \mathbb{N}_0, n \geq n_0 : g(n) > c \cdot f(n).$$

Also müssen wir zeigen, dass es *für alle* positiven Zahlen c und *für alle* natürlichen Zahlen n_0 *ein* $n \geq n_0$ gibt, so dass $g(n) > c \cdot f(n)$.

■ **Beispiel 6.7 —** $g \notin O(f)$.

- Für $g(n) = n^2$ und $f(n) = n$ gilt $g \notin O(f)$. Sei c eine beliebige positive Zahl und n_0 eine beliebige natürliche Zahl. Wähle $n = n_0 + c$. Dann gilt $g(n) = n^2 = (n_0 + c)^2 > c \cdot (n_0 + c) = c \cdot n = c \cdot f(n)$.
- Für $g(n) = 2n^2$ und $f(n) = n + 7$ gilt $g \notin O(f)$. Sei c eine beliebige positive Zahl und n_0 eine beliebige natürliche Zahl. Wähle $n = n_0 + c + 7$. Dann gilt $g(n) = 2n^2 = 2(n_0 + c + 7)^2 > 2c(n_0 + c + 7) = 2cn = 2c \cdot f(n) > c \cdot f(n)$.
- Für $g(n) = n^3$ und $f(n) = 100000000000n^2$ gilt $g \notin O(f)$. Sei c eine beliebige positive Zahl und n_0 eine beliebige natürliche Zahl. Wähle $n = n_0 + 100000000000 \cdot c + 1$. Dann gilt $g(n) = n^3 = (n_0 + 100000000000 \cdot c + 1)^3 > 100000000000 \cdot c \cdot (n_0 + 100000000000 \cdot c + 1)^2 > 100000000000 \cdot c \cdot (n_0 + 100000000000 \cdot c)^2 >= 100000000000 \cdot c \cdot n^2 = c \cdot f(n)$.

◇

Nach unserer Intuition sollte ein Polynom vom Grad k immer kleiner als ein Polynom vom Grad $k+1$ sein. Mithilfe unserer Definition der Größe durch die O-Notation können wir dies nun auch formal beweisen.

Theorem 6.8 Sei $k \in \mathbb{N}_0$ und $g\colon \mathbb{N}_0 \to \mathbb{N}_0$ ein Polynom mit $g(n) = \sum_{i=0}^{k} a_i n^i$ vom Grad k mit natürlichen Koeffizienten $a_0, \ldots, a_k \in \mathbb{N}_0$ und $a_k > 0$. Für jedes Polynom $f\colon \mathbb{N}_0 \to \mathbb{N}_0$ vom Grad $k+1$ mit $f(n) = \sum_{i=0}^{k+1} b_i n^i$ und natürlichen Koeffizienten $b_0, \ldots, b_{k+1} \in \mathbb{N}_0$ mit $b_{k+1} > 0$ gilt $g \in O(f)$.

Beweis. Sei $a_{\max} = \max_{0 \leq i \leq k}\{a_i\}$ der größte Koeffizient von g. Dann gilt $g(n) = \sum_{i=0}^{k} a_i n^i \leq \sum_{i=0}^{k} a_{\max} n^i \leq (k+1) a_{\max} n^k$. Für $c = (k+1)a_{\max}$ und $n_0 = 1$ gilt dann

$$g(n) \leq (k+1)a_{\max} n^k = c \cdot n^k \leq c \cdot n^{k+1} \leq c \cdot b_{k+1} \cdot n^{k+1} \leq g(n).$$

Also gilt $g \in O(f)$. ■

Ein sehr wichtiger Spezialfall tritt dann ein, wenn $g \in O(f)$ und $f \in O(g)$ gilt. In diesem Fall wachsen also f und g etwas gleich schnell.

Definition 6.9 — Θ-Notation. Sei $f\colon \mathbb{N}_0 \to \mathbb{N}_0$ eine Funktion. Die Menge $\Theta(f)$ ist die Menge

aller Funktionen $g\colon \mathbb{N}_0 \to \mathbb{N}_0$, so dass $g \in O(f)$ und $f \in O(g)$ gilt, also

$$\Theta(f) = \{g \in O(f) \mid f \in O(g)\}.$$

Diese Definition ist symmetrisch: Gilt also $g \in \Theta(f)$, so gilt auch $f \in \Theta(g)$.

■ **Beispiel 6.10 —** $g \in \Theta(f)$**.**

- Für $g(n) = 2n$ und $f(n) = 3n+1$ gilt $g \in \Theta(f)$.
- Für $g(n) = 43n^3 + 27n^2 + (n/21) + 9$ und $f(n) = n^3$ gilt $g \in \Theta(f)$.

◇

Besonders praktisch an der O-Notation ist es, dass es egal ist, ob wir $f(n) = 43n^3 + 23n^2 + 7n + 1$ oder $f'(n) = n^3$ haben, denn $O(f) = O(f')$. Um also knapp anzugeben, dass ein Term $T(n)$ maximal quadratisch in n wächst, können wir einfach $T \in O(n^2)$ schreiben, wobei wir n^2 als Funktion verstehen, die n auf n^2 abbildet. In der Literatur wird manchmal auch "$g = O(f)$" statt "$g \in O(f)$" geschrieben.

Aufgaben

Aufgabe 6.1 — Vorrechnen. Beweisen oder widerlegen Sie die folgenden Aussagen über die Funktion $f\colon \mathbb{Q} \to \mathbb{Q}$ mit $f(x) = 2x$:

(i) Die Funktion f ist injektiv.

(ii) Die Funktion f ist surjektiv.

■

Beweis. Wir zeigen, dass f bijektiv ist und somit injektiv und surjektiv.

(i) Seien $a_1, a_2 \in \mathbb{Q}$ mit $f(a_1) = f(a_2)$. Somit gilt dann $2a_1 = f(a_1) = f(a_2) = 2a_2$ und somit auch $a_1 = a_2$. Also ist f injektiv.

(ii) Sei $b \in \mathbb{Q}$. Wähle $a = b/2$. Dann gilt $a \in \mathbb{Q}$ und weiterhin $f(a) = f(b/2) = 2 \cdot b/2 = b$ und somit ist f surjektiv. Wäre $f\colon \mathbb{N}_0 \to \mathbb{N}_0$, funktioniert der obige Beweis nicht, da $b/2$ nicht unbedingt eine natürliche Zahl sein muss.

Da f injektiv und surjektiv ist, ist f auch bijektiv. ■

Aufgabe 6.2 — Vorrechnen. Sei $g\colon \mathbb{N}_0 \to \mathbb{N}_0$ mit $g(n) = 3n+2$ und $f\colon \mathbb{N}_0 \to \mathbb{N}_0$ mit $f(n) = (1/2)n^2$. Zeigen Sie, dass $g \in O(f)$. ■

Beweis. Variante 1. Sei $c = 10$ und $n_0 = 1$. Dann gilt für alle $n \geq n_0$, dass

$$c \cdot f(n) = 10 \cdot (1/2) \cdot n^2 = 5n^2 \underbrace{\geq}_{n \geq n_0 = 1} 5n = 3n + 2n \underbrace{\geq}_{n \geq n_0 = 1} 3n + 2 = g(n).$$

■

Beweis. Variante 2. Sei $c = 1$ und $n_0 = 7$. Wir beweisen per vollständiger Induktion, dass $c \cdot f(n) \geq g(n)$ für alle $n \geq n_0$.

IA: Für $n = 7$ gilt

$$c \cdot f(n) = c \cdot (1/2)n^2 = (1/2)7^2 = 49/2 = 24{,}5 \geq 23 = 3 \cdot 7 + 2 = 3n + 2 = g(n).$$

IV: Für ein $k \in \mathbb{N}_0$ mit $k \geq 7$ gelte $g(n) \leq c \cdot f(n)$.

IS: Es gilt

$$c \cdot f(k+1) = f(k+1) = (1/2)(k+1)^2 = (1/2)(k^2+2k+1) = (1/2)k^2 + (1/2)(2k+1) = f(k) + (1/2)(2k+1) \underbrace{\geq}_{\text{IV}} g(k) + (1/2)(2k+1) = 3k+2+(1/2)(2k+1).$$

Da $k \geq 7$, gilt $(1/2)(2k+1) \geq 15/2 \geq 3$ und somit

$$3k+2+(1/2)(2k+1) \geq 3k+2+3 = 3(k+1)+2 = g(k+1).$$

■

Aufgabe 6.3 Beweisen Sie, dass für alle $a, b > 0$ und alle $x \in \mathbb{R}$ mit $x > 0$ gilt, dass $\log_a(x) = \log_b(x)/\log_b(a)$.
Hinweis: Betrachten Sie den Ausdruck $\log_b(a^{\log_a(x)})$. ■

Beweis. Aus der Rechenregel $a^{\log_a(x)} = x$ erhalten wir

$$\log_b(a^{\log_a(x)}) = \log_b(x).$$

Andererseits gilt $\log_b(x^y) = y \cdot \log_b(x)$, also gilt

$$\log_b(a^{\log_a(x)}) = \log_a(x) \cdot \log_b(a).$$

Somit gilt $\log_b(x) = \log_a(x) \cdot \log_b(a)$. Division durch $\log_b(a)$ gibt $\log_a(x) = \log_b(x)/\log_b(a)$. ■

Aufgabe 6.4 Beweisen oder widerlegen Sie die folgenden Aussagen über die Funktion $f\colon \mathbb{N}_0 \to \mathbb{N}_0$ mit $f(n) = n+1$:
(i) Die Funktion f ist injektiv.
(ii) Die Funktion f ist surjektiv. ■

Beweis. Wir zeigen, dass f injektiv und *nicht* surjektiv ist.
(i) Seien $a_1, a_2 \in \mathbb{N}_0$ mit $f(a_1) = f(a_2)$. Dann gilt $a_1 + 1 = f(a_1) = f(a_2) = a_2 + 1$ und somit auch $a_1 = a_2$. Also ist f injektiv.
(ii) Angenommen, dass f surjektiv wäre. Für $b = 0$ müsste es dann $a \in \mathbb{N}_0$ mit $f(a) = b$ geben. Da aber $f(a) = a+1$, gilt $a+1 = b = 0$ und somit $a = -1 \notin \mathbb{N}_0$. Somit ist f nicht surjektiv.
■

Aufgabe 6.5 Sei $g\colon \mathbb{N}_0 \to \mathbb{N}_0$ mit $g(n) = n^4$ und $f\colon \mathbb{N}_0 \to \mathbb{N}_0$ mit $f(n) = n^5$. Zeigen Sie, dass $g \in O(f)$. ■

Beweis. Sei $c = 1$ und $n_0 = 1$. Dann gilt für $n \geq n_0$, dass

$$g(n) = n^4 \underbrace{\leq}_{1 = n_0 \leq n} n \cdot n^4 = 1 \cdot n \cdot n^4 = 1 \cdot n^5 = c \cdot f(n).$$

Also gilt $g \in O(f)$. ■

Aufgabe 6.6 Beweisen Sie, dass für alle geraden natürlichen Zahlen $n \geq 1$ gilt, dass

$$(n/2)[\log(n) - 1] \leq \sum_{i=1}^{n} \log(i) \leq n \cdot \log(n).$$

Beweis. Da die Funktion $x \mapsto \log(x)$ monton steigend ist, gilt $\log(i) \leq \log(n)$ für $i \leq n$. Somit folgt

$$\sum_{i=1}^{n} \log(i) \leq \sum_{i=1}^{n} \log(n) = n \cdot \log(n).$$

Andererseits gilt auch

$$\sum_{i=1}^{n} \log(i) \geq \sum_{i=n/2}^{n} \log(i) \geq \sum_{i=n/2}^{n} \log(n/2) = (n/2)\log(n/2).$$

Aus den Rechenregeln für den Logarithmus erhalten wir $\log(a/b) = \log(a) - \log(b)$ und somit $\log(n/2) = \log(n) - \underbrace{\log(2)}_{=1}$. ■

Aufgabe 6.7 Beweisen Sie: Für alle natürlichen Zahlen $a, m, b \in \mathbb{N}_0$ mit $a \equiv b \pmod{m}$ gibt es eine Zahl $q \in \mathbb{Z}$, so dass $b = a + qm$.

Beweis. Nach Definition ist $a \equiv b \pmod{m}$ äquivalent zu $(b-a)/m \in \mathbb{Z}$. Sei also $q = (b-a)/m$. Dann gilt

$$a + q \cdot m = a + [(b-a)/m] \cdot m = a + (b-a) = b.$$

■

Aufgabe 6.8 Sei $g\colon \mathbb{N}_{>0} \to \mathbb{N}_{>0}$ mit $g(n) = 1 \cdot 2 \cdot 3 \cdot \ldots \cdot n$ (das schreibt man auch als $n!$) und $f\colon \mathbb{N}_{>0} \to \mathbb{N}_{>0}$ mit $f(n) = n^n$. Beweisen Sie $g \in O(f)$.

Beweis. Sei $c = 1$ und $n_0 = 1$. Für alle $n \geq n_0$ gilt $g(n) = n! = \underbrace{1}_{\leq n} \cdot \underbrace{2}_{\leq n} \cdot \ldots \cdot \underbrace{n}_{\leq n} \leq \underbrace{n \cdot \ldots \cdot n}_{n\text{-mal}} = n^n = f(n) = c \cdot f(n)$. ■

Aufgabe 6.9 Sei $g\colon \mathbb{N}_0 \to \mathbb{N}_0$ mit $g(n) = 3^n$ und $f\colon \mathbb{N}_0 \to \mathbb{N}_0$ mit $f(n) = 2^n$. Beweisen Sie $g \notin O(f)$.

Beweis. Sei also $c \in \mathbb{R}_{>0}$ und $n_0 \in \mathbb{N}_0$. Sei $n = \max\{n_0, \log_{3/2}(c)\} + 1$. Dann gilt

$$g(n) = 3^n = (3/2)^n \cdot 2^n \underbrace{>}_{n > \log_{3/2}(c)} (3/2)^{\log_{3/2}(c)} \cdot 2^n = c \cdot 2^n = c \cdot f(n).$$

Also ist $g \notin O(f)$. ■

Aufgabe 6.10 Sei $a \in \mathbb{R}$ eine nicht-negative reelle Zahl und $b \in \mathbb{R}$. Zeigen Sie, dass die Funktion $f\colon \mathbb{R} \to \mathbb{R}$ mit $f(x) = ax + b$ bijektiv ist. ∎

Beweis. Wir zeigen, dass f bijektiv, also injektiv und surjektiv ist.

(i) Seien $x_1, x_2 \in \mathbb{R}$ mit $f(x_1) = f(x_2)$. Dann gilt auch $ax_1 + b = f(x_1) = f(x_2) = ax_2 + b$ und durch Subtraktion von b und Division durch a somit auch $x_1 = x_2$. Also ist f injektiv.

(ii) Sei $y \in \mathbb{R}$. Wähle $x = (y-b)/a$. Somit ist $x \in \mathbb{R}$ und wir haben $f(x) = [(y-b)/a] \cdot a + b = (y-b) + b = y$. Also ist f surjektiv.

Da f injektiv und surjektiv ist, ist f auch bijektiv. ■

Aufgabe 6.11 Wenn zwei ganze Zahlen x, y den gleichen Betrag haben, also $|x| = |y|$ gilt, so schreiben wir $x \sim y$.

(i) Beweisen Sie, dass $\sim$ eine Äquivalenzrelation ist.

(ii) Bestimmen Sie die Äquivalenzklassen von $\sim$. ∎

Beweis. (i) Wir müssen also zeigen, dass $\sim$ reflexiv, symmetrisch und transitiv ist.

Reflexivität: Wir müssen also zeigen, dass $x \sim x$ für alle x gilt. Offensichtlich gilt $|x| = |x|$ und somit auch $x \sim x$.

Symmetrie: Wir müssen also zeigen, dass $x \sim y$ auch immer $y \sim x$ impliziert. Gilt nun $x \sim y$, so folgt $|x| = |y|$ und somit auch $|y| = |x|$. Also gilt $y \sim x$.

Transitivität: Wir müssen also zeigen, dass aus $x \sim y$ und $y \sim z$ auch $x \sim z$ folgt. Aus $x \sim y$ folgt also $|x| = |y|$. Ebenso folgt aus $y \sim z$, dass $|y| = |z|$. Also gilt $|x| = |y| = |z|$ und somit $|x| = |z|$. Damit folgt $x \sim z$.

(ii) Für jede ganze Zahl x gibt es eine eindeutige natürliche Zahl x^+ mit $|x| = |x^+|$ (nämlich entweder $x^+ = x$ oder $x^+ = -x$). Somit sind die natürlichen Zahlen $\mathbb{N}_0$ die Äquivalenzklassen von $\sim$.

■

Aufgabe 6.12 Sei $f\colon \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{N}_0$ eine Funktion mit $f(n,m) = n + (1/2)(n+m)(n+m+1)$. Zeigen Sie:

(i) Die Funktion f ist surjektiv.

(ii) Die Funktion f ist injektiv.

Hinweis: Was ist der kleinste Wert, den $f(n,m)$ annehmen kann, wenn $n+m = K$ gilt? Was ist der größte Wert? ∎

Beweis. Gilt $n + m = K$, so ist $f(n,m) = n + (1/2)K(K+1)$. Somit nimmt $f(n+m)$ also K verschiedene Werte für $n+m = K$ an. Für $n = 0$ und $m = K$ ist der Ausdruck minimal und maximal für $n = K$ und $m = 0$. Im ersten Fall ($n = 0$, $m = K$) gilt $f(n,m) = (1/2)K(K+1)$ und im zweiten Fall ($n = K$, $m = 0$) gilt $f(n,m) = K + (1/2)K(K+1) = (1/2)2K + (1/2)K(K+1) = (1/2)[2K + K(K+1)] = (1/2)[2K + K^2 + K] = (1/2)[K(K+3)]$. Für $K \in \mathbb{N}_0$ sei $g_{\min}(K) = (1/2)K(K+1)$ und $g_{\max}(K) = (1/2)[K(K+3)]$. Nun gilt

$$g_{\min}(K+1) = (1/2)[(K+1)(K+2)] = (1/2)[K^2 + 2K + K + 2] = \quad (*)$$
$$(1/2)[K(K+3) + 2] = (1/2)[K(K+3)] + 1 = g_{\max}(K) + 1.$$

Somit schließen die Intervalle von $g_{\min}(K)$ bis $g_{\max}(K)$ nahtlos an die Intervalle $g_{\min}(K+1)$ bis $g_{\max}(K+1)$ an.

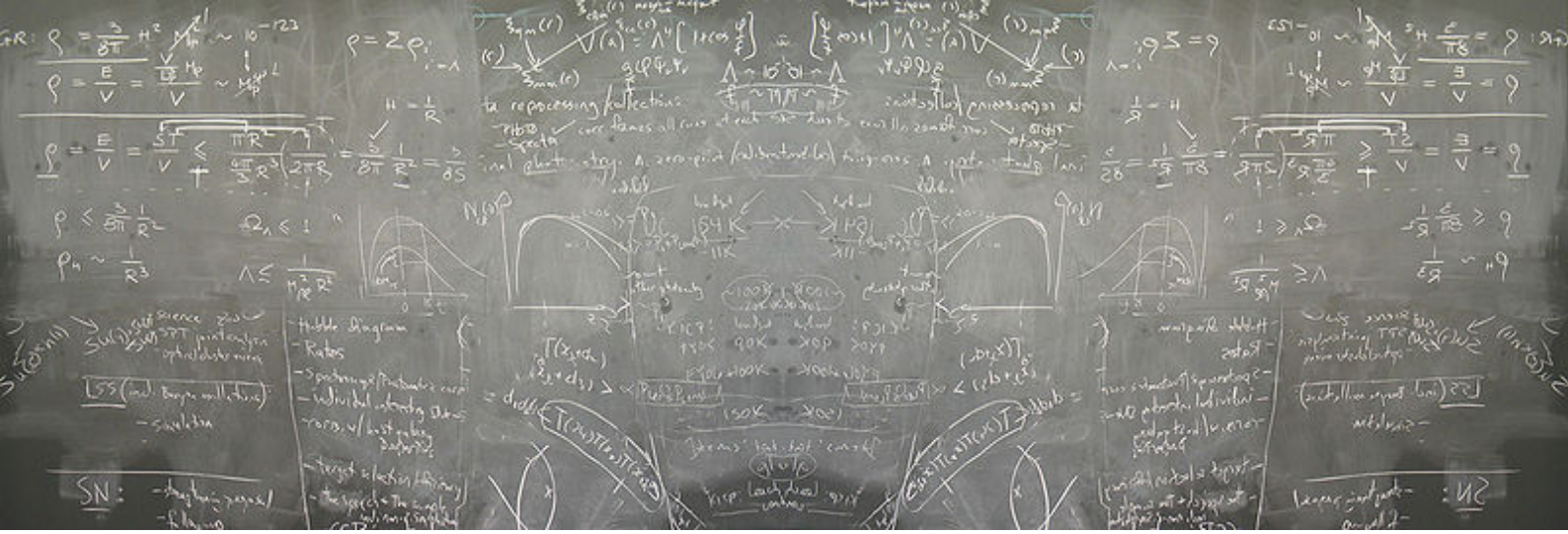
(i) Seien $m_1, m_2, n_1, n_2 \in \mathbb{N}_0$ mit $f(n_1, m_1) = f(n_2, m_2)$. Wenn $n_1 + m_1 < n_2 + m_2$, folgt aus der Gleichung $(*)$ sofort, dass $f(n_1, m_1) < f(n_2, m_2)$. Analog folgt aus $n_1 + m_1 > n_2 + m_2$ auch sofort, dass $f(n_1, m_1) > f(n_2, m_2)$. Somit muss also $n_1 + m_1 = n_2 + m_2 = K$ gelten. Somit ist $f(n_1, m_1) = n_1 + (1/2)K(K+1)$ und $f(n_2) = n_2 + (1/2)K(K+1)$. Also gilt $n_1 = n_2$ und somit auch $m_1 = m_2$. Also ist f injektiv.

(ii) Sei $b \in \mathbb{N}_0$. Nach der obigen Überlegung muss es dann ein K geben mit $g_{\min}(K) \leq b \leq g_{\max}(K)$. Also gibt es ein i mit $b = i + (1/2)K(K+1)$. Mit $n = i$ und $m = K - n$ folgt dann $f(n, m) = i + (1/2)K(K+1) = b$. Somit ist f surjektiv.

■

Lernziele:

- Sicherer Umgang mit Funktionen und deren Eigenschaften
- Beherrschen der Rechenregeln für die wichtigsten Funktionen
- Grundlegendes Verständnis für Modulare Arithmetik, den mod-Operator und Teilbarkeitsregeln
- Äquivalenzklassen verstehen
- Verständnis der O-Notation
- Sicherer Umgang mit dem Vergleich zweier Funktionen durch die O-Notation



7. Kombinatorik, Laufzeiten

Zum Lesen:
Kapitel 6.1, 6.2, 6.5, 6.7, 6.8 in [Bel18]
Ausarbeitung von Sebastian zu Laufzeiten (also dieses Dokument)

Stichworte:

- Binomialkoeffizienten $\binom{n}{k}$; Interpretationen; Identitäten
- Permutationen; $n!$
- Binomischer Lehrsatz; Produkt von Binomen
- k-to-one correspondence
- Kombinatorische Beweise mittels Bijektionen
- Laufzeit von Algorithmen; Konkatenation; for-Schleife; while-Schleife

7.1 Laufzeiten

Wie bereits zu Anfang des letzten Kapitels erklärt, möchten wir jedem Algorithmus A eine *Worst-Case-Laufzeit* T_{A} zuordnen. Anders ausgedrückt: Wenn A eine Eingabe der Länge n erhält, rechnet er für höchstens $T_{\mathsf{A}}(n)$ Schritten. Nehmen wir uns beispielhaft den folgenden Algorithmus, der das maximale Element in einem Array X der Länge n sucht:

```
FindMax(X)
1 :  n := X.length();
2 :  max := X[0];
3 :  for i = 1,...,n-1 :
4 :      if X[i] > max :
5 :          max := X[i];
6 :  if max < 0 :
7 :      max := 0
8 :  return max
```

Nun ist es ersteinmal sehr schwer, eine konkrete Aussage über die wirkliche Laufzeit T_{FindMax} zu machen: Wir wissen nicht, wieviele Operationen der Rechner wirklich für die Zeile » **if** $X[i] >$

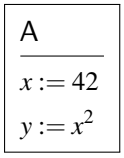
max «benötigt. Es könnte sogar so sein, dass diese Anzahl vom konkreten Rechner abhängt, denn verschiedene Prozessoren können verschiedene Operationen verschieden schnell ausführen. Selbst wenn wir die Zahl auf unserem Rechner genau wüssten, hängt diese Zahl natürlich auch vom Compiler ab, der die Zeile in Maschinencode verwandelt. Somit ist es sehr schwer, die genaue Laufzeit zu bestimmen. Wenn wir jedoch realistisch annehmen, dass jede der Zeilen in konstanter Laufzeit auszuführen ist, können wir eine sinnvolle obere Schranke geben. Sei c die maximale Ausführungszeit einer einzelnen Zeile im obigen Code. Offenbar werden die Zeilen 1, 2, 6, 7 und 8 genau einmal ausgeführt. Die Zeilen 3 und 4 werden jeweils $n-1$-mal ausgeführt, wenn das Eingabearray die Länge n hat. Wie häufig Zeile 5 genau ausgeführt wird, ist nicht klar und hängt von der Eingabe ab: Beim Array $X = [1,2,3,4,\ldots,n]$ wird die Zeile auch $n-1$-mal ausgeführt, beim Array $X = [n,n-1,\ldots,1]$ jedoch niemals. Aber man sieht leicht, dass im schlimmsten Fall nur $n-1$ Wiederholungen auftreten können. Insgesamt können wir also abschätzen, dass $T_{\mathsf{FindMax}}(n) \leq 3c(n-1)+5c$ ist. Da wir jedoch im Wesentlichen nur am groben Wachstum der Funktion interessiert sind, können wir das in eine einfachere Form bringen. Da $3c(n-1)+3c \in O(n)$, haben wir also $T_{\mathsf{FindMax}} \in O(n)$. Die Laufzeit ist also linear. Wenn wir also die Eingabelänge verdoppeln, erwarten wir, dass wir ungefähr doppelt so lange rechnen müssen. Hätten wir zum Beispiel das folgende Verfahren $\mathsf{FindMax}_2$, welches ebenso das Minimum eines Arrays X berechnet, so sehen wir, dass die Zeile 6 hier $(n-1)\cdot(n-1)$-mal ausgeführt wird. Also gilt $T_{\mathsf{FindMax}_2} \in O(n^2)$ und $T_{\mathsf{FindMax}_2} \notin O(n)$.

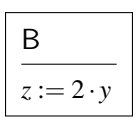
```
FindMax2(X)
 1 :  n := X.length();
 2 :  max := X[0];
 3 :  for i = 1,...,n-1 :
 4 :     maxi := X[i];
 5 :     for j = 1,...,n-1 :
 6 :        if X[j] > X[i] :
 7 :           maxi = X[j];
 8 :     if maxi > max :
 9 :        max := maxi
10 :  return max
```

Wenn wir also die Länge X des Arrays verdoppeln, erwarten wir hier, dass wir ungefähr viermal so lange rechnen müssen. Zusammengefasst wollen wir also die Worst-Case-Laufzeit eines Algorithmus mithilfe der O-Notation nach oben abschätzen. Es gibt drei wichtige Grundoperationen, wie man Algorithmen miteinander verbinden kann, die wir uns im Folgenden genauer anschauen wollen.

Konkatenation

Sind A und B Algorithmen, so erhalten wir den Algorithmus $\mathsf{A} \circ \mathsf{B}$, indem wir zuerst A und dann B ausführen. Betrachten wir das folgende Beispiel:





Dementsprechend erhalten wir den folgenden Algorithmus $\mathsf{A} \circ \mathsf{B}$ als *Konkatenation* oder *Hintereinanderausführung* der beiden Algorithmen.

```
A ∘ B
---------
x := 42
y := x^2
z := 2 · y
```

Wie man sehr leicht einsieht, gilt $T_{\mathsf{A}\circ\mathsf{B}}(n) = T_{\mathsf{A}}(n) + T_{\mathsf{B}}(n)$, also brauchen wir nur die beiden Laufzeiten zu addieren, um die Gesamtlaufzeit zu erhalten.

for-Schleife

Ist A ein Algorithmus, der eine natürliche Zahl i als Eingabe erhält, so können wir mittels einer for-Schleife den Algorithmus wiederholt ausführen:

```
B(n)
---------
for i = 1,...,n :
  A(j)
```

Auch hier kann man sich leicht ableiten, dass für die Laufzeit $T_{\mathsf{B}}(n) \in O(\sum_{i=1}^{n} T_{\mathsf{A}}(i))$ gilt. Wir müssen uns also nur überlegen, wie lange die Summe der Ausführungszeiten von A ist.

Ein relativ häufig auftretender Fall ist, dass man k ineinander geschachtelte Schleifen hat, die jeweils ungefähr n Schritte durchlaufen. Die Gesamtlaufzeit ist in diesem Fall einfach $O(n^k)$ (siehe auch unser obiges Beispiel mit $\mathsf{FindMax}_2$ mit $k = 2$). Die Analyse wird deutlich komplizierter, wenn die Schleifen voneinander abhängen: Läuft zum Beispiel der Schleifenzähler i der äußeren Schleife von 1 bis n und der Schleifenzähler j der inneren Schleife von i bis n, ist die Laufzeit weiterhin $O(n^2)$, aber die Analyse wird deutlich aufwändiger. Es gibt natürlich auch Schleifen, die nur konstant häufig durchlaufen. Diese vergrößern unsere Laufzeit entsprechend nicht.

while-Schleife

Ist A ein Algorithmus, der eine natürliche Zahl i als Eingabe erhält und update ein Algorithmus, der ebenfalls i als Eingabe erhält und eine natürliche Zahl zurückgibt, so können wir mittels einer while-Schleife den Algorithmus auch wiederholt ausführen:

```
B(n)
---------
i := n
while i > 1 :
  A(i)
  i := update(i)
```

Die Laufzeit T_{B} hier zu bestimmen, ist schon schwieriger und hängt enorm vom Verhalten von $\mathsf{update}(i)$ ab. Die einfachsten und häufigsten Fälle sind zum Beispiel, dass $\mathsf{update} = i+1$ oder $\mathsf{update} = i/2$ gilt. Im ersteren Fall haben wir nur eine for-Schleife anders geschrieben. Im zweiten Fall (und allgemein) können wir wie folgt argumentieren: Wir bekommen ja nun Werte $i_1 = n$, $i_2 = \mathsf{update}(i_i) = \mathsf{update}(n)$, $i_3 = \mathsf{update}(i_2) = \mathsf{update}(\mathsf{update}(n))$, und so weiter. Allgemein gilt $i_k = \mathsf{update}^k(n)$, wobei $\mathsf{update}^k(n) = \underbrace{\mathsf{update}(\mathsf{update}(\ldots(\mathsf{update}}_{k\text{-mal}}(n)$ ist. Nun möchten wir das kleinste k bestimmen, so dass $\mathsf{update}^k(n) \leq 1$ gilt. Dementsprechend häufig wird nämlich unsere Schleife ausgeführt. Im Falle $\mathsf{update}(i) = i/2$ kennen wir diesen Wert übrigens schon: Es ist $\lceil \log_2(n) \rceil - 1$. So häufig müssen wir nämlich n durch 2 teilen, bis wir bei der 1 angekommen sind. Im Allgemeinen kann es aber durchaus schwerer sein, die Laufzeit zu analysieren. Haben wir jedoch solch ein k bestimmt, so können wir einfach $T_{\mathsf{B}}(n) \leq k \cdot T_{\mathsf{A}}(n)$ benutzen, um die Laufzeit von B abzuschätzen. Wir werden uns in »Algorithmen und Datenstrukturen «noch sehr intensiv mit der Abschätzung komplizierterer update-Algorithmen auseinandersetzen.

Aufgaben

Aufgabe 7.1 — Vorrechnen. Beweisen Sie: Für alle $n,k \in \mathbb{N}_0$ gilt

$$\sum_{m=0}^{n} \binom{m}{k} = \binom{n+1}{k+1}.$$

Beweis. Wir benutzen einen kombinatorischen Beweis. Auf der rechten Seite haben wir die Anzahl der Teilmengen der Größe $k+1$ einer $n+1$-elementigen Menge. Wir nehmen an, dass es sich bei der Menge um die Zahlen $1,2,\ldots,n+1$ handelt. Für jede Teilmenge der Größe $k+1$ gibt es eine kleinste Zahl m, so dass m in der Teilmenge enthalten ist und $1,2,\ldots,m-1$ nicht. Für ein festes m müssen wir also noch k Zahlen aus den verbleibenden $n+1-m$ Zahlen wählen. Dafür haben wir offenbar $\binom{n+1-m}{k}$ Möglichkeiten. Aufsummieren aller Fälle gibt uns

$$\sum_{m=1}^{n+1} \binom{n+1-m}{k} = \binom{0}{k} + \binom{1}{k} + \ldots + \binom{n}{k} = \sum_{m=0}^{n} \binom{m}{k}.$$

■

Aufgabe 7.2 — Vorrechnen. Geben Sie eine möglichst gute obere Schranke für die Laufzeit des folgenden Algorithmus A, der eine natürliche Zahl n als Eingabe erhält:

```
A(n)
1 : s := 0;
2 : for i = 1,...,n :
3 :     s := s + i;
4 : return s
```

Beweis. Wir nehmen wieder an, dass jede Zeile des Algorithmus in konstanter Zeit läuft und c das Maximum dieser Konstanten ist. Dann laufen die Zeilen 1 und 4 in Zeit $2c$. Da die Zeilen 2 und 3 genau n-mal ausgeführt wird, sind die entsprechenden Kosten also $2c \cdot n$. Insgesamt haben wir also eine Laufzeit von $2c + 2c \cdot n = c \cdot (2n+2) \in O(n)$. ■

Aufgabe 7.3 Beweisen Sie: Für alle $n,k \in \mathbb{N}_0$ gilt

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

Beweis. Wir benutzen wieder einen kombinatorischen Beweis. Auf der rechten Seite nehmen wir zuerst eine beliebige Zahl $i \in \{1,\ldots,n\}$, erstellen dann aus den restlichen $n-1$ Elementen $\{1,\ldots,n\} \setminus \{i\}$ eine $k-1$-elementige Teilmenge und fügen i als letzte Zahl zu diesen Teilmengen hinzu, um eine k-elementige Teilmenge zu erhalten. Auf diesem Wege erhalten wir alle Teilmengen der Größe k von $\{1,\ldots,n\}$, aber wir haben einige doppelt gezählt. Konkret kriege ich jede Teilmenge genau k-mal, denn wir können jede Zahl, die in ihr vorkommt, als letzte Zahl ergänzen. Also gibt es pro Teilmenge k Möglichkeiten, sie zu konstruieren. ■

Aufgabe 7.4 Geben Sie eine möglichst gute obere Schranke für die Laufzeit des folgenden Algorithmus A, der eine natürliche Zahl n als Eingabe erhält:

```
A(n)
1 :  s := 0;
2 :  for i = 1,...,n :
3 :      for j = 1,...,n :
4 :          s := s + i · j;
5 :  return s
```

Beweis. Wieder sei c die maximale Laufzeit einer einzelnen Zeile. Die Zeilen 1 und 5 laufen somit in Zeit $2c$. Für jedes $i \in \{1,\dots,n\}$ und jedes $j \in \{1,\dots,n\}$ werden Zeile 3 und 4 genau einmal ausgeführt. Es gibt $n \cdot n = n^2$ viele solcher (i,j)-Paare, also werden Zeilen 3 und 4 insgesamt genau n^2 ausgeführt und haben somit Laufzeit $2n^2 \cdot c$. Zeile 2 wird n-mal ausgeführt und hat somit Laufzeit cn Insgesamt ist die Laufzeit also $2c + cn + 2cn^2 \in O(n^2)$. ■

Aufgabe 7.5 Beweisen Sie: Für alle $n,k,m \in \mathbb{N}_0$ gilt:

$$\binom{n}{m} \cdot \binom{m}{k} = \binom{n}{k} \cdot \binom{n-k}{m-k}.$$

Beweis. Auf der linken Seite wählen wir zunächst eine Teilmenge der Größe m und dann aus dieser eine Teilmenge der Größe k. Nun fragen wir uns, wie häufig jede Teilmenge der Größe k dabei produziert werden kann. Eine feste Teilmenge K der Größe k können wir zu einer Teilmenge M der Größe m erweitern, indem wir aus den $n-k$ Elementen $m-k$ Elemente auswählen. Da wir dies für jede k-elementige Teilmengen tun können, folgt die Behauptung. ■

Aufgabe 7.6 Beweisen Sie: Für alle $n,k \in \mathbb{N}_0$ gilt:

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

Hinweis: Benutzen Sie den binomischen Lehrsatz.

Beweis. Der binomische Lehrsatz sagt uns, dass für alle $x,y \in \mathbb{R}$ und alle $n \in \mathbb{N}_0$ gilt, dass

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

Setze nun $x = -1$ und $y = 1$ und wir erhalten auf der linken Seite eine 0. Auf der rechten Seite steht

$$\sum_{k=0}^{n} \binom{n}{k} x^k y^{n-k} = \sum_{k=0}^{n} \binom{n}{k} (-1)^k \underbrace{1^{n-k}}_{=1} = \sum_{k=0}^{n} (-1)^k.$$

■

Aufgabe 7.7 Geben Sie eine möglichst gute obere Schranke für die Laufzeit des folgenden Algorithmus A, der eine natürliche Zahl n als Eingabe erhält:

```
A(n)
1 :  i := n;
2 :  s := 0;
3 :  while i > 0 :
4 :      s := s + 1;
5 :      i := ⌊i/2⌋;
6 :  return s
```

Beweis. Sei wieder c die maximale Ausführungszeit einer Zeile. Die Zeilen 1, 2 und 6 werden jeweils einmal ausgeführt und verursachen Kosten $3c$. Wir haben bereits in der dritten Woche gesehen, dass man die Zahl n genau $\lfloor \log_2(n) \rfloor - 1$-mal durch 2 teilen kann, bis man bei der 1 angekommen ist. Also werden die Zeilen 3, 4 und 5 maximal $\lfloor \log_2(n) \rfloor - 1$-mal ausgeführt und wir haben eine Gesamtlaufzeit von $3c + 3c \cdot [\lfloor \log_2(n) \rfloor - 1] \in O(\log n)$. ■

Aufgabe 7.8 Beweisen Sie, dass für alle $n \in \mathbb{N}_0$ gilt:

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

Beweis. Auf der rechten Seite zählen wir die Anzahl der n-elementigen Teilmengen der $2n$-elementigen Menge $\{1,\ldots,2n\}$. All diese Teilmengen können wir auch eindeutig so bekommen, dass wir k Elemente aus $\{1,\ldots,n\}$ und $n-k$ Elemente aus $\{n+1,\ldots,n\}$ wählen. Dafür gibt es genau $\binom{n}{k} \cdot \binom{n}{n-k}$ Möglichkeiten. Da $\binom{n}{n-k} = \binom{n}{k}$, folgt die Aussage. ■

Aufgabe 7.9 Beweisen Sie, dass für alle $n \in \mathbb{N}_0$ gilt:

$$\sum_{k=0}^{n} \binom{n-k}{k} = F_{n+1}.$$

Hierbei ist F_n die $n$te Fibonacci-Zahl.

Beweis. Für $n = 0$ gilt zunächst einmal $\sum_{k=0}^{n} \binom{n-k}{k} = \sum_{k=0}^{0} \binom{n-k}{k} = \binom{0}{0}$. Die 0-elementige Menge $\emptyset$ hat genau eine Teilmenge, nämlich $\emptyset$! Also gilt $\binom{0}{0} = 1 = F_1$.

Für $n = 1$ gilt $\sum_{k=0}^{n} \binom{n-k}{k} = \sum_{k=0}^{1} \binom{n-k}{k} = \binom{1-0}{0} + \binom{1-1}{1} = 1 + 1 = 2 = F_2$.

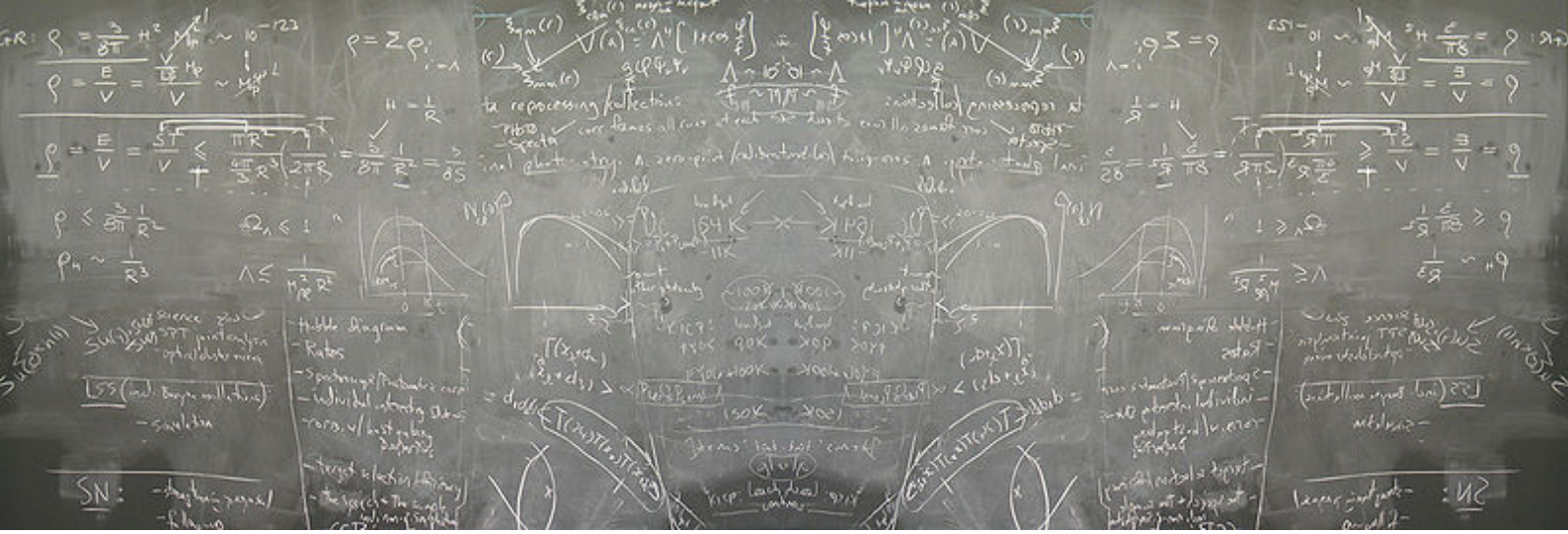
Für $n \geq 2$ gilt nach dem Buch, dass

$$\sum_{k=0}^{n} \binom{n+1-k}{k} = [\sum_{k=0}^{n} \binom{n-k}{k} + \binom{n-k}{k-1}] =$$
$$[\sum_{k=0}^{n} \binom{n-k}{k}] + [\sum_{k=0}^{n} \binom{n-k}{k-1}] =$$
$$[\sum_{k=0}^{n} \binom{n-k}{k}] + [\sum_{k=0}^{n} \binom{n-1-(k-1)}{k-1}] =$$
$$[\sum_{k=0}^{n} \binom{n-k}{k}] + [\sum_{k=0}^{n-1} \binom{n-1-k}{k}]$$

Sei $a_n = \sum_{k=0}^{n} \binom{n-k}{k}$. Dann gilt $a_0 = F_1$, $a_1 = F_2$ und $a_n = a_{n-1} + a_{n-2}$. Da $F_n = F_{n-1} + F_{n-2}$, folgt sofort $a_n = F_{n+1}$ für alle $n \in \mathbb{N}_0$. ■

Lernziele:

- Die Begriffe Binomialkoeffizient und Fakultät kennen
- Grundlegende Identitäten des Binomialkoeffizient kennen
- Einfache kombinatorische Beweise mittels Abzählargumenten entwickeln können
- Den Binomischen Lehrsatz anwenden können
- Die Begriffe Teilmenge und geordnetes Tupel unterscheiden können
- Erste Laufzeitabschätzungen tätigen können



Literaturverzeichnis

[Bel18] Sarah-Marie Belcastro. *Discrete mathematics with ducks*. CRC Press, 2018. ISBN: 9781466504998.

[Beu99] Albrecht Beutelspacher. “Das ist o. B. d. A. trivial!” In: *Tips und Tricks zur Formulierung mathematischer Gedanken* 5 (1999).

[LLM18] Eric Lehman, F Thomson Leighton und Albert R Meyer. *Mathematics for Computer Science*. 2018.

[Wol17] Karsten Wolf. *Präzises Denken für Informatiker*. Springer-Verlag, 2017. ISBN: 3662549727.