

Vorkurs für Informatikstudierende

Dr. Julien Klaus

Skript zur Vorlesung

Inhaltsverzeichnis

1 Mengenlehre	5
1.1 Einleitung	5
1.2 Eigenschaften von Mengen	5
1.3 Operationen auf Mengen	6
1.3.1 Kurzschreibweisen für Mengen	8
2 Logik	9
2.1 Aussagen und Wahrheitswerte	9
2.2 Operationen auf Aussagen	9
2.3 Prädikatenlogik	11
2.4 Aussagen über Mengen	13
3 Analysis	15
3.1 Elementare Funktionen	15
3.1.1 Polynome und rationale Funktionen	15
3.1.2 Potenz-, Exponential- und Logarithmusfunktionen	17
3.1.3 Trigonometrische Funktionen	18
3.1.4 Minimum, Maximum und Betrag	19
3.1.5 Eigenschaften von Funktionen	20
3.2 Folgen und Reihen	21
3.2.1 Grenzwert	22
3.2.2 Vergleich von Funktionen	22
3.2.3 Vollständige Induktion	23
3.3 Differenzialrechnung	24
3.4 Integralrechnung	29
4 Einführung Informatik	33
5 Algorithmik	35
5.1 Motivation	35
5.2 Eigenschaften von Algorithmen	35
5.3 Flussdiagramme und Pseudocode	37
5.3.1 Flussdiagramme	37
5.3.2 Start und Stop	37
5.3.3 Anweisungen	38
5.3.4 Verzweigungen	38
5.3.5 Ein- und Ausgabe	38
5.3.6 Programmaufrufe	38

5.3.7	Anwendung von Flussdiagrammen	39
5.3.8	Andere Darstellungen	39
6	Python	41
6.1	Geschichte	41
6.2	Installation und Start	41
6.3	Syntax	41
6.3.1	Anweisungen	41
6.3.2	Verzweigungen	42
6.3.3	Schleifen	42
6.3.4	Ein- und Ausgabe	43
6.3.5	Listen	43
6.3.6	Funktionen	43
6.3.7	Rekursion	44
6.4	CodinGame	45
7	Kommandozeile	47
7.1	Bewegen in Verzeichnissen	47
7.2	Hilfe	47
7.3	Dateioperationen	47
7.4	Programme	48
7.5	Weitere Befehle	48

1 Mengenlehre

1.1 Einleitung

Eine **Menge** ist eine beliebige Zusammenfassung von bestimmten wohlunterschiedenen Objekten unserer Anschauung oder unseres Denkens zu einem Ganzen. - *Georg Cantor (1845 – 1918)*

Mengen begegnen uns in der Mathematik ständig. Selbst in der Schule haben wir uns schon mit Mengen befasst ohne dies zu wissen. Ein Beispiel für eine Menge sind die natürlichen Zahlen \mathbb{N} . So ist 4 beispielsweise ein **Element** der Menge \mathbb{N} . Wir schreiben dafür kurz $4 \in \mathbb{N}$. Allerdings ist die Menge der natürlichen Zahlen nicht allumfassend. Die gebrochenrationalen Zahlen gehören nicht dazu. Wir können also schreiben $\frac{1}{2} \notin \mathbb{N}$.

1.2 Eigenschaften von Mengen

Zwei Mengen sind **gleich**, wenn sie die gleichen Elemente haben. Nehmen wir als Beispiel die zwei Mengen $M = \{1, 2, 3\}$ und $N = \{2, 3, 1, 2\}$. Hier sind die Elemente innerhalb der Mengen gleich. Es gilt also $M = N$. Dies bringt uns auch gleich zwei Eigenschaften von Mengen. Sie sind **nicht sortiert** und **doppelte Elemente** zählen nur einfach. Weiter werden Mengen häufig mit großen Buchstaben bezeichnet.

Die Mengen in unseren Beispielen wurden bisher nur durch Angabe ihrer Elemente beschreiben. Mengen können auch durch die Angabe von Eigenschaften angegeben werden. Die natürlichen Zahlen könnten alternativ auch als $\mathbb{N} := \{x \mid x \text{ ist eine natürliche Zahl}\} = \{1, 2, 3, \dots\}$ definiert werden. Wollen wir die null als Element der natürlichen Zahlen haben, schreiben wir $\mathbb{N}_0 = \{0, 1, 2, \dots\}$.

Um einen weiteren wichtigen Begriff in der Mengenlehre einzuführen, müssen wir uns nun überlegen wieviele Elemente die Menge $M = \{x \mid x \in \mathbb{N} \text{ und } x < 1 \text{ und } x > 1\}$ hat. Diese Menge besitzt keine Elemente und wir bezeichnen solch eine Menge als **leere Menge**. Diese wird als \emptyset gekennzeichnet.

Definition 1 (Teilmenge). *Eine Menge A wird als **Teilmenge** einer Menge B bezeichnet, falls für alle Elemente $a \in A$ gilt: $a \in B$. Man schreibt in diesem Fall $A \subseteq B$. Falls B Elemente enthält, die A nicht enthält schreibt man $A \subset B$.*

1 Mengenlehre

Die Menge aller Teilmengen einer Menge wird als Potenzmenge bezeichnet. Ein Beispiel für Teilmengen sind die natürlichen und die reellen Zahlen. Es gilt also $\mathbb{N} \subset \mathbb{R}$. Durch die Definition der Teilmenge können wir nun folgenden Satz beweisen.

Satz 1 (Gleichheit von Mengen). *Es seien A und B Mengen. Dann sind die folgenden zwei Aussagen äquivalent*

1. $A = B$,
2. $A \subseteq B$ und $B \subseteq A$.

Beweis. Beweis durch die Teilmengendefinition. □

Definition 2 (Kardinalität). *Die Größe n einer Menge A wird als **Kardinalität** bezeichnet und durch die Schreibweise $|A| = n$ angegeben.*

1.3 Operationen auf Mengen

Definition 3 (Komplement). *Das **Komplement** einer Menge $A \subseteq M$ wird als $\bar{A} = \{x \in M \mid x \notin A\}$ definiert.*

Wenn wir also das Komplement einer Menge angeben möchten, brauchen wir eine Übermenge, die unsere Menge enthält. Beispielsweise ist das Komplement der Menge $\{0\} \in \mathbb{N}_0 \subseteq \mathbb{N}_0$ gleich der Menge \mathbb{N} .

Definition 4 (Durchschnitt und Vereinigung). *Als der **Durchschnitt** $A \cap B$ zweier Mengen A und B werden alle Elemente bezeichnet, die sowohl in A als auch in B vorkommen.*

*Die **Vereinigung** zweier Mengen A und B schreiben wir als $A \cup B$. Diese enthält alle Elemente, die entweder in A oder in B oder in beiden vorhanden sind.*

Beispiel 1 (Durchschnitt und Vereinigung). *Sei die Menge $A = \{1, 2, 3\}$ und die Menge $B = \{2, 3, 4\}$. Die Vereinigung dieser Mengen $A \cup B$ ist gleich $\{1, 2, 3, 4\}$ und der Durchschnitt $A \cap B$ ist gleich $\{2, 3\}$.*

Definition 5 (Differenz). *In der **Differenz** $A \setminus B$ zweier Mengen A und B sind die Elemente aus A , die nicht in B vorkommen.*

Satz 2 (Komplement von B in A).

$$A \setminus B \text{ ist äquivalent zu } A \cap \bar{B}.$$

Beweis. In der Differenz von $A \setminus B$ sind alle Element von A ohne die von B . Die Menge, die alle Element enthält, die nicht in B sind ist das Komplement \bar{B} . Der Durchschnitt von $A \cap \bar{B}$ enthält nun also alle Elemente von A ohne die aus B . □

Definition 6 (Symmetrische Differenz). *Die **Symmetrische Differenz** zweier Mengen A und B ist definiert als:*

$$A \triangle B = (A \setminus B) \cup (B \setminus A).$$

Satz 3 (Eigenschaften von Mengenoperationen). *Für die zweistelligen Operationen auf Mengen gelten folgende Eigenschaften. Seien hierzu A, B, C Mengen.*

- $A \cap B = B \cap A$ (Kommutativität)
- $(A \cap B) \cap C = A \cap (B \cap C)$ (Assoziativität)
- $A \subseteq B$ und $B \subseteq C$ folgt $A \subseteq C$
- $A \subseteq B$ folgt $A \cup B = B$
- $A \subseteq B$ folgt $A \cap B = A$
- $(A \cup B) \cap C = (A \cap C) \cup (B \cap C)$
- $\overline{(A \cup B)} = \bar{A} \cap \bar{B}$

Beweis. Der Beweis kann über Mengendiagramme durchgeführt werden. Dies wird in der Übung gezeigt. \square

Wie wir bereits gelernt haben sind Mengen nicht sortiert. Falls wir eine Sortierung brauchen, können wir dies mit dem Konstrukt aus der nächsten Definition lösen.

Definition 7 (n -Tupel). *Seien $x_1, \dots, x_n \in M, n \in \mathbb{N}$ n Elemente einer Grundmenge M . Ein **n -Tupel** ist eine Zusammenfassung dieser Elementen, dargestellt als*

$$(x_1, \dots, x_n),$$

wobei die Reihenfolge dieser Elemente von Bedeutung ist. An der i -ten Stelle steht dabei das Element x_i .

Bemerkung 1 (Geordnete Paar). *Eine Sonderform des n -Tupel ist das **geordnete Paar**. Dies ist ein Tupel, welches nur zwei Elemente enthält*

Definition 8 (Kartesisches Produkt). *Seien A, B beliebige Mengen. Die Menge aller geordneten Paare zwischen A und B wird als **kartesisches Produkt** $A \times B$ bezeichnet. Diese ist definiert als:*

$$A \times B = \{(a, b) \mid a \in A \text{ und } b \in B\}.$$

Falls $A = B$ verabschieden wir die Schreibweise A^2 .

Beispiel 2. *Bei der Anwendung des Kreuzproduktes erhalten wieder eine Menge.*

$$(A \times C) \cup (B \times C) = \{(x, y) \mid x \in A \text{ und } y \in C \text{ oder } x \in B \text{ und } y \in C\} = (A \cup B) \times C$$

Natürlich können wir diese Definition auch beliebig auf n -Tupel erweitern. Für die nächste Schreibweise, die wir definieren möchte, benötigen wir in dem einfachen Fall erst einmal nur geordnete Paare.

Definition 9 (Abbildung). *Seien \mathcal{D}, \mathcal{W} Mengen. Eine **Abbildung** (Funktion) $f : \mathcal{D} \rightarrow \mathcal{W}$ ist dann die Menge aller geordneten Paare*

$$\{(x, f(x)) \mid x \in \mathcal{D}, f(x) \in \mathcal{W}\}.$$

Man bezeichnet \mathcal{D} als den Definitionsbereich und \mathcal{W} als den Wertebereich.

1.3.1 Kurzschreibweisen für Mengen

Einige Mengen werden in der Mathematik häufig benötigt. Wir haben bereits einige gesehen, wie \mathbb{R} oder \mathbb{N} . In diesem kurzen Kapitel wollen wir uns noch Intervall- und Indextmengen definieren.

Definition 10 (Intervallmengen). *Sei $x, y \in \mathbb{R}, x \leq y$. Dann definieren wir die folgenden Mengen*

$$\begin{aligned}(x, y) &= \{z \mid z > x \text{ und } z < y\}, \\(x, y] &= \{z \mid z > x \text{ und } z \leq y\}, \\[x, y) &= \{z \mid z \geq x \text{ und } z < y\}, \\[x, y] &= \{z \mid z \geq x \text{ und } z \leq y\}.\end{aligned}$$

Definition 11 (Indexmenge). *Sei $n \in \mathbb{N}$. Dann definieren wir die folgende Menge*

$$[n] = \{i \mid i \in \{1, 2, \dots, n\}\}.$$

2 Logik

2.1 Aussagen und Wahrheitswerte

Für Informatiker ist die Logik ist eines der wichtigsten Gebiete der Mathematik. Die Logik basiert dabei auf Aussagen denen ein Wahrheitswert zugeordnet wird.

Definition 12 (Aussage). Eine **Aussage** ist eine Zeichenfolge, der genau einer der beiden Wahrheitswerte **wahr** ($1, w$) oder **falsch** ($0, f$) zugeordnet werden kann.

Wenn p eine Aussage ist, dann bezeichnen wir mit $\alpha(p) \in \{w, f\}$ den Wahrheitswert von p .

Aussagen können die verschiedensten Formen annehmen. Beispiele für Aussagen sind unter anderen:

- $4 \in \mathbb{N}$,
- 3 ist eine Primzahl,
- Hello ist das englische Wort für Hallo,
- Montag ist ein Tag des Wochenendes,
- $\frac{1}{2} = 0.87$.

Wir können für diese Beispiele natürlich die Wahrheitswerte angeben. Wie für Mengen existieren auch für Aussagen Operationen.

2.2 Operationen auf Aussagen

Die einfachste Operation auf Aussagen ist die der Negation. Diese werden wir auch als einzige einstellige Operation einführen. Operationen werden durch die Anwendung einer Wahrheitswerttabelle dargestellt. Diese gibt für Belegungen einer Aussage, die Wahrheitswerte die durch die Operation entstehen an.

Definition 13 (Negation). Sei p eine Aussage. Die **Negation** $\neg p$ von p wird durch die folgende Wahrheitswerttabelle definiert.

$\alpha(p)$	$\alpha(\neg p)$
w	f
f	w

2 Logik

Oftmals gibt es für die Negation von Aussagen andere Schreibweisen. Ein Beispiel hierfür könnte die Aussage $4 \in \mathbb{N}$ sein. Deren Negation kann man als $\neg(4 \in \mathbb{N})$ oder kurz als $4 \notin \mathbb{N}$ schreiben.

Folgerung 1 (Zweifache Verneinung). *Es sei p eine Aussage. Dann gilt*

$$\alpha(\neg(\neg(p))) = \alpha(p)$$

Beweis. Beweis in der Übung. □

Definition 14 (Konjugation, Disjunktion, Implikation, Äquivalenz). *Seien p und q Aussagen. Die wichtigsten Operationen der Logik (neben der Negation) sind die*

- **Konjugation** $p \wedge q$ (und),
- **Disjunktion** $p \vee q$ (oder),
- **Implikation** $p \rightarrow q$ (wenn, dann) und
- **Äquivalenz** $p \leftrightarrow q$ (genau dann, wenn).

Diese sind in der folgenden Wahrheitswerttabelle definiert.

$\alpha(p)$	$\alpha(q)$	$\alpha(p \wedge q)$	$\alpha(p \vee q)$	$\alpha(p \rightarrow q)$	$\alpha(p \leftrightarrow q)$
w	w	w	w	w	w
w	f	f	w	f	f
f	w	f	w	w	f
f	f	f	f	w	w

Wie man in der Definition sieht, ist es für die logische Oder-Operation nicht ausschließlich. Umgangssprachlich ist dies oft als 'und/oder' formuliert. Das ausschließende Oder ist eine andere Verknüpfung die als $p \oplus q$ zwischen den Aussagen p und q dargestellt wird. Diese ist wie folgt definiert:

$$p \oplus q \leftrightarrow (\neg p \wedge q) \vee (p \wedge \neg q).$$

Wichtig ist hier, dass wir die beiden Aussagen mit einer Äquivalenzoperation verknüpfen (die linke Aussage ist genau dann wahr, wenn die rechte Aussage wahr ist). Diese Aussage ist also immer wahr. Man bezeichnet Aussagen, die immer wahr sind als **Tautologie**. Wichtige Tautologien sind im folgenden Satz zusammengefasst.

Satz 4 (Tautologien). *Seien p, q, r Aussagen. Dann sind die folgenden Aussagen Tau-*

tologien.

$$p \leftrightarrow \neg(\neg p) \quad (2.1)$$

$$(p \wedge p) \leftrightarrow p \quad (2.2)$$

$$(p \wedge q) \leftrightarrow (q \wedge p) \quad (2.3)$$

$$((p \wedge q) \wedge r) \leftrightarrow (p \wedge (q \wedge r)) \quad (2.4)$$

$$(p \vee p) \leftrightarrow p \quad (2.5)$$

$$(p \vee q) \leftrightarrow (q \vee p) \quad (2.6)$$

$$((p \vee q) \vee r) \leftrightarrow (p \vee (q \vee r)) \quad (2.7)$$

$$((p \wedge q) \vee r) \leftrightarrow ((p \vee r) \wedge (q \vee r)) \quad (2.8)$$

$$((p \vee q) \wedge r) \leftrightarrow ((p \wedge r) \vee (q \wedge r)) \quad (2.9)$$

$$((p \vee q) \wedge q) \leftrightarrow q \quad (2.10)$$

$$((p \wedge q) \vee q) \leftrightarrow q \quad (2.11)$$

$$(\neg(p \wedge q)) \leftrightarrow ((\neg p) \vee (\neg q)) \quad (2.12)$$

$$(\neg(p \vee q)) \leftrightarrow ((\neg p) \wedge (\neg q)) \quad (2.13)$$

$$(p \rightarrow q) \leftrightarrow (\neg p \vee q) \quad (2.14)$$

$$(p \rightarrow q) \leftrightarrow (\neg q \rightarrow \neg p) \quad (2.15)$$

$$(p \leftrightarrow q) \leftrightarrow ((p \rightarrow q) \wedge (q \rightarrow p)) \quad (2.16)$$

$$((p \rightarrow q) \wedge (q \rightarrow r) \wedge (r \rightarrow p)) \leftrightarrow ((p \leftrightarrow q) \wedge (q \leftrightarrow r) \wedge (r \leftrightarrow p)) \quad (2.17)$$

Beweis. Alle diese Aussagen können über eine Wahrheitswerttabelle bewiesen werden. Ein alternativer Weg, diese Aussagen zu beweisen ist über geschicktes Umformen.

Wir zeigen dies für Aussage (15), $(p \rightarrow q) \leftrightarrow (\neg q \rightarrow \neg p)$. Hierzu nehmen wir an, dass die Tautologien 1, 6 und 14 bereits bewiesen sind.

$$\begin{aligned} & (p \rightarrow q) \\ \Leftrightarrow & \neg p \vee q & (2.14) \\ \Leftrightarrow & \neg p \vee \neg(\neg q) & (2.1) \\ \Leftrightarrow & \neg(\neg q) \vee \neg p & (2.6) \\ \Leftrightarrow & \neg q \rightarrow \neg p & (2.14) \end{aligned}$$

In der Übungen werden die weiteren und angenommenen Tautologien bewiesen. \square

Bemerkung 2 (Schreibweise). *Das Symbol \rightarrow kann nur zwischen Aussagen stehen. Beispielsweise ist es korrent für $x \in \mathbb{N}$ zu schreiben $(x - 1)^2 = 0 \rightarrow x = 1$. Allerdings wäre es falsch zu schreiben $(x - 1)^2 \rightarrow x^2 - 2x + 1$. Die richtige Schreibweise ist hier $(x - 1)^2 = x^2 - 2x + 1$, was eine wahre Aussage ist.*

2.3 Prädikatenlogik

Der Begriff Prädikatenlogik soll hier nicht abschreckend wirken. Wir benötigen ihn um umfassendere Aussagen zu beschreiben. In Aussagen müssen alle Variablen durch eine

2 Logik

Belegung gebunden sein. Beispielsweise ist $x^2 = 4$ keine Aussage, da man ihr keinen eindeutigen Wahrheitswert zuweisen kann. Erst durch die Information, dass es mindestens ein x gibt, so dass gilt $x^2 = 4$ oder durch das Setzen von x auf einen festen Wert wird es zu einer Aussage.

Definition 15 (All- und Existenzquantor). *Es sei I eine nicht leere Menge und für alle $i \in I$ sei p_i eine Aussage. Dann kann man diese Aussagen mithilfe des Allquantor \forall oder des Existenzquantor \exists verknüpfen. Dabei steht $\forall i \in I : p_i$ dafür, dass für alle $i \in I$ die Aussage p_i gilt. Die Aussage $\exists i \in I : p_i$, dass es mindestens ein $i \in I$ gibt, für das p_i gilt.*

Die Wahrheitswerte der Verknüpfungen ist intuitiv definiert als:

$$\alpha(\forall i \in I : p_i) = w \leftrightarrow \alpha(p_i) = w \text{ für alle } i \in I$$

und

$$\alpha(\exists i \in I : p_i) = w \leftrightarrow \alpha(p_i) = w \text{ für mindestens ein } i \in I.$$

Satz 5 (Negation von Quantoren). *Die folgenden Aussagen sind Tautologien über der nicht leeren Menge I und den Aussagen $p_i, i \in I$.*

$$\neg(\forall i \in I : p_i) \leftrightarrow \exists i \in I : \neg p_i \quad (2.18)$$

$$\neg(\exists i \in I : p_i) \leftrightarrow \forall i \in I : \neg p_i \quad (2.19)$$

. Von Formel 2.18]

$$\begin{aligned} & \neg(\forall i \in I : p_i) \\ \leftrightarrow & \neg \bigwedge_{i \in I} p_i \\ \leftrightarrow & \neg(p_1 \wedge p_2 \wedge \dots \wedge p_n) \\ \leftrightarrow & \neg p_1 \vee \neg p_2 \vee \dots \vee \neg p_n \\ \leftrightarrow & \bigvee_{i \in I} \neg p_i \\ \leftrightarrow & \exists i \in I : \neg p_i \end{aligned}$$

Formel 2.19 verläuft synonym. □

Bemerkung 3 (Verknüpfung von Quantoren). *Quantoren können beliebig verknüpft werden. Ein Beispiel hierfür ist die Definition der Kommutativität der Multiplikation.*

$$\forall a \in \mathbb{R} \forall b \in \mathbb{R} : a \cdot b = b \cdot a$$

2.4 Aussagen über Mengen

Nachdem wir Mengenlehre- und Logikkenntnisse erworben haben können wir jetzt komplexere Aussagen über Mengen beweisen. Wir wollen dies hier an einem Beispiel zeigen.

Beispiel 3. Seien $A, B, C \neq \emptyset$ nichtleere Mengen. Wir wollen zeigen, dass gilt:

$$(A \cap B) \times C = (A \times C) \cap (B \times C).$$

Beweis. Wir müssen für alle (x, y) aus der Menge $D = (A \cap B) \times C$ zeigen, dass dies auch in $(A \times C) \cap (B \times C)$ liegen:

$$\begin{aligned} \forall (x, y) \in D : (x, y) \in (A \cap B) \times C &\leftrightarrow \forall (x, y) \in D : x \in (A \cap B) \wedge y \in C \\ &\leftrightarrow \forall (x, y) \in D : (x \in A \wedge x \in B) \wedge y \in C \\ &\leftrightarrow \forall (x, y) \in D : x \in A \wedge y \in C \wedge x \in B \wedge y \in C \\ &\leftrightarrow \forall (x, y) \in D : (x \in A \wedge y \in C) \wedge (x \in B \wedge y \in C) \\ &\leftrightarrow \forall (x, y) \in D : (x, y) \in (A \times C) \wedge (x, y) \in (B \times C) \\ &\leftrightarrow \forall (x, y) \in D : (x, y) \in (A \times C) \cap (B \times C). \end{aligned}$$

□

3 Analysis

3.1 Elementare Funktionen

In der Analysis dreht sich alles um Funktionen und deren Eigenschaften. Wir wollen uns in diesem Kapitel deshalb erst einmal einige elementare Funktionen anschauen. Beginnen wir mit polynomen und rationalen Funktionen.

3.1.1 Polynome und rationale Funktionen

Bevor wir zu der Definition von Polynomfunktionen kommen, möchten wir hier das Summenzeichen einführen.

Definition 16 (Summenzeichen). Seien $a_0, a_1, \dots, a_n \in \mathbb{R}, n \in \mathbb{N}$. Die Summe dieser Zahlen kann als

$$\sum_{i=0}^n a_i = a_0 + a_1 + \dots + a_n$$

geschrieben werden. Hierbei wird \sum als **Summenzeichen**, i als **Laufindex**, 0 als **Startwert** und n als **Endwert** bezeichnet. a_i ist die Funktion bezüglich der Laufvariable.

Mit dem Summenzeichen können nun komplizierte Additionen verkürzt dargestellt werden. Jede Summe kann so umgeschrieben werden, dass ihr Startwert bei 0 beginnt.

Bemerkung 4 (Indexverschiebung). Sei $k, n \in \mathbb{R}, k \neq 0, n > k$ und seien $a_k, \dots, a_n \in \mathbb{R}$. Dann gilt:

$$\sum_{i=k}^n a_i = \sum_{i=0}^{n-k} a_{i+k}.$$

Bemerkung 5 (Produktzeichen). Synonym zu dem Summenzeichen kann man das **Produktzeichen** einführen. Seien hierzu $a_0, a_1, \dots, a_n \in \mathbb{R}, n \in \mathbb{N}$. Das Produkt dieser Zahlen dann geschrieben werden als:

$$\prod_{i=0}^n a_i = a_0 \cdot a_1 \cdot \dots \cdot a_n.$$

Wir werden die Benutzung des Summenzeichens und des Produktzeichenes in der Übung ausführlich trainieren um ein Gefühl dafür zu bekommen. Nachdem wir nun das Summenzeichen eingeführt haben können wir nun einfach Polynome definieren.

Definition 17 (Polynom). Sei f eine Funktion der Form $f : \mathbb{R} \rightarrow \mathbb{R}$. Dann heißt

$$f(x) = \sum_{i=0}^n a_i \cdot x^i = a_0 \cdot x^0 + a_1 \cdot x^1 + \cdots + a_n \cdot x^n, n \in \mathbb{N}$$

reelles Polynom mit dem Grad $\deg(f) = n$ ($a_n \neq 0$). Die Zahlen $a_0, \dots, a_n \in \mathbb{R}$ werden als Koeffizienten bezeichnet.

Wir können die Polynome aus Definition 17 beliebig addieren und multiplizieren ohne dass die Polynom Eigenschaft verschwindet. Bei der Division allerdings könnte das Nennerpolynom gleich null sein, was uns zur Definition der rationalen Funktionen bringt.

Definition 18 (Rationale Funktion). Seien $p(x)$ und $q(x)$ zwei Polynome. Der Quotient

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

dieser Polynome heißt **rationale Funktion**. Diese ist nur für diejenigen x definiert, für die $q(x) \neq 0$ gilt.

Natürlich kann bei der Verknüpfung von Polynomen das resultierende Polynom vereinfacht werden. Bei rationalen Funktion spricht man hier von der sogenannten Polynomdivision

Bemerkung 6 (Polynomdivision). Seien $p(x), q(x)$ Polynome und der Grad von q kleiner gleich dem Grad von p , dann gibt es Polynome $s(x)$ und $r(x)$, so dass gilt

$$p(x) = s(x) \cdot q(x) + r(x).$$

Falls x_0 eine Nullstelle des Polynoms $p(x)$ ist, dann ist $r(x) = 0$ falls $q(x) = (x_0 - x)$.

Eine Polynomdivision auszuführen ist häufig ungewohnt. Wir wollen deshalb ein einfaches Beispiel angeben und mehr Beispiele in der Übung bearbeiten.

Beispiel 4 (Polynomdivision). Sei $p(x) = x^3 - 8x^2 + 19x - 12$ und $q(x) = (x^2 + 1)$. Zu Beginn schreiben wir die Polynome wie bei der schriftlichen Division.

$$\begin{array}{rcccccl}
 (x^3 & -8x^2 & +19x & -12) & : & (x^2 + 1) & = & x - 8 \\
 -(x^3 & & & & & & & \\
 \hline
 & -8x^2 & +18x & -12 & & & & \\
 & -(-8x^2 & & & & & & 8) \\
 \hline
 & & +18x & -20 & & & &
 \end{array}$$

Im ersten Schritt überlegen wir uns was wir zu x^2 multiplizieren müssen um x^3 zu erhalten. Die Antwort lautet natürlich x und wir schreiben dies als erstes hinter das Gleichheitszeichen. Nun rechnen wir $x^2 * x$ und $1 * x$ und schreiben dieses unter die Ausdrücke

des mit dem gleichen Grad. Danach ziehen wir diese beiden Zeilen voneinander ab. Das resultierende Polynom $-8x^2 + 18x - 12$ wird nun Synonym behandelt. Wir überlegen uns wieder was wir mit x^2 multiplizieren müssen um $-8x^2$ zu erhalten und schreiben das Ergebnis -8 hinter das Gleichheitszeichen. Danach rechnen wir wieder das Restpolynom aus. Diese kann nun nicht mehr durch x^2 geteilt werden und wir beenden die Polynomdivision. Das Ergebnis der Polynomdivision lautet also

$$\frac{x^3 - 8x^2 + 19x - 12}{x^2 + 1} = (x - 8) + \frac{18x - 20}{(x^2 + 1)},$$

wobei $s(x) = (x - 8)$ und $r(x) = 18x - 20$ ist.

3.1.2 Potenz-, Exponential- und Logarithmusfunktionen

Bevor wir die verschiedenen Funktionen definieren. Möchten wir hier noch einmal Potenzregeln und Logarithmenregeln wiederholen.

Bemerkung 7 (Potenzregeln). Seien $a, b > 0$ und $x, y \in \mathbb{R}$. Dann gelten folgende Regeln:

$$a^x \cdot a^y = a^{x+y} \quad (3.1)$$

$$a^x \cdot b^x = (a \cdot b)^x \quad (3.2)$$

$$(a^x)^y = a^{(x \cdot y)} \quad (3.3)$$

$$a^{-x} = \frac{1}{a^x} \quad (3.4)$$

$$a^0 = 1 \quad (3.5)$$

$$a^{\frac{x}{y}} = \sqrt[y]{a^x} \quad (3.6)$$

$$\forall x \in \mathbb{R} : a^x > 0 \quad (3.7)$$

$$(3.8)$$

Bemerkung 8 (Logarithmenregeln). Seien $a, b > 0$ und $x, y \in \mathbb{R}$. Dann gelten die folgenden Logarithmusregeln:

$$\log_x(a \cdot b) = \log_x(a) + \log_x(b) \quad (3.9)$$

$$\log_x\left(\frac{a}{b}\right) = \log_x(a) - \log_x(b) \quad (3.10)$$

$$\log_x(a^b) = b \cdot \log_x(a) \quad (3.11)$$

$$\log_x(a) = \frac{\log_y(a)}{\log_y(x)} \quad (3.12)$$

$$\log_x(1) = 0 \quad (3.13)$$

$$\log_b(b^x) = x \quad (3.14)$$

$$b^{\log_b(x)} = x \quad (3.15)$$

3 Analysis

Diese Regeln aus Bemerkung 7 und 8 sind sehr wichtig und sollten möglichst viel geübt werden. Wir werden uns dem in der Übung widmen. Nachdem wir uns nun mit den Potenz- und Logarithmengesetzen auseinandergesetzt haben, können wir nun Potenz- und Exponentialfunktionen definieren.

Definition 19 (Potenz- und Exponential- und Logarithmusfunktion). Sei $a, x \in \mathbb{R}, x > 0$. Dann heißt die Funktion

$$f(x) = x^a$$

Potenzfunktion. Sei nun $b, y \in \mathbb{R}, b > 0$. Dann heißt die Funktion

$$g(y) = b^y$$

Exponentialfunktion. Sei zuletzt $c, z \in \mathbb{R}, c > 0, c \neq 1, z > 0$. Dann heißt die Funktion

$$h(z) = \log_c(z)$$

Logarithmusfunktion zur Basis c .

Bemerkung 9 (Umkehrfunktion). Die Logarithmusfunktion ist die Umkehrfunktion der Exponentialfunktion. Seien $f(x) = a^x$ und $f^{-1}(x) = \log_a(x)$, wobei $a, x \in \mathbb{R}$ und $x > 0$. Hierbei wird f^{-1} als die **Umkehrfunktion** der Funktion f bezeichnet und es gilt:

$$f(f^{-1}(x)) = x.$$

3.1.3 Trigonometrische Funktionen

Definition 20 (Sinus und Kosinus). Sei die Menge $E = \{(x, y) \mid x^2 + y^2 = 1\}$ die Menge aller Punkte auf dem Einheitskreis. Sei nun θ ein beliebiger Winkel und $e = (s, c) \in E$ der dazugehörige Punkt auf dem Einheitskreis. Dann sind **Sinus** und **Kosinus** wie folgt definiert:

$$\sin(\theta) = s \text{ und } \cos(\theta) = c.$$

Bemerkung 10 (Einheitskreis). Die Definition des Sinus und Kosinus ist mit einer kleinen Skizze besser zu verstehen.

Bemerkung 11 (Zusammenhang zwischen Sinus und Kosinus). Seien $\alpha, \beta \in \{x \mid x \in [0, 2\pi]\}$ Winkel im Bogenmaß ($2\pi \text{ rad} = 360^\circ$). Dann gilt:

$$\sin(\alpha) = -\cos\left(\alpha + \frac{\pi}{2}\right) = \cos\left(\alpha - \frac{\pi}{2}\right)$$

und

$$\sin^2(\alpha) + \cos^2(\alpha) = 1.$$

Weiterhin gelten folgende Regeln:

$$\begin{aligned} \sin(\alpha \pm \beta) &= \sin(\alpha) \cos(\beta) \pm \cos(\alpha) \sin(\beta) \\ \cos(\alpha \pm \beta) &= \cos(\alpha) \cos(\beta) \mp \sin(\alpha) \sin(\beta) \end{aligned}$$

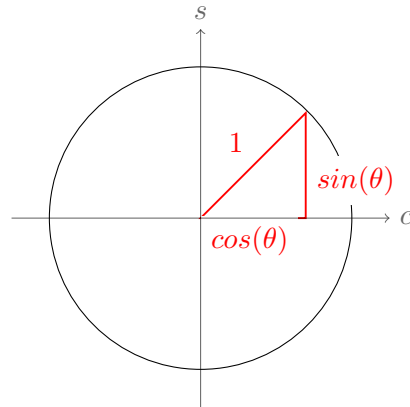


Abbildung 3.1: Definition des Sinus und Kosinus am Einheitskreis.

Definition 21 (Tangens und Kotangens). *Als letzten beiden trigonometrischen Funktionen wollen wir den **Tangens** \tan und **Kotangens** \cot definieren. Sei hierzu α ein Winkel im Bogenmaß und für den Tangens $\cos(\alpha) \neq 0$, sowie für den Kotangens $\sin(\alpha) \neq 0$. Dann sind diese Funktionen wie folgt definiert*

$$\begin{aligned}\tan(\alpha) &= \frac{\sin(\alpha)}{\cos(\alpha)}, \\ \cot(\alpha) &= \frac{1}{\tan(\alpha)}.\end{aligned}$$

Natürlich gibt es noch weitere trigonometrischen Funktionen. Wir wollen die Auswahl allerdings hier auf die gegebenen begrenzen.

3.1.4 Minimum, Maximum und Betrag

Definition 22 (Minimum und Maximum). *Seien $x, y \in \mathbb{R}$. Dann definieren wir folgende Funktionen:*

$$\min(x, y) = \begin{cases} x & , x \leq y \\ y & , x > y \end{cases}$$

und

$$\max(x, y) = \begin{cases} x & , x \geq y \\ y & , x < y. \end{cases}$$

Definition 23 (Betrag). *Sei $x \in \mathbb{R}$. Der Betrag $|\cdot|$ ist dann wie folgt definiert:*

$$|x| = \begin{cases} x & , x \geq 0 \\ -x & , x < 0. \end{cases}$$

Beispiel 5. *Mit der Hilfe des Betrages können wir nun Ungleichungen der Form*

$$|x + 1| \leq 3$$

3 Analysis

lösen. Dazu müssen unterschieden werden, wann der Betrag positiv und negativ wird. In diesem Fall gilt:

$$|x + 1| \geq 0 \leftrightarrow x \geq -1.$$

Wir müssen nun zwei Fälle betrachten.

1. $x \geq -1$, der Betrag ist in diesem Fall positiv und kann aufgelöst werden

$$|x + 1| \geq 3 \leftrightarrow x + 1 \geq 3 \leftrightarrow x \geq 2$$

2. $x < -1$

$$|x + 1| \geq 3 \leftrightarrow -(x + 1) \geq 3 \leftrightarrow -x - 1 \geq 3 \leftrightarrow x \leq -4$$

Aus den beiden Lösungen liefern uns die Lösungsmenge $\{x | x \geq 2 \wedge x \leq -4\}$.

3.1.5 Eigenschaften von Funktionen

In diesem Kapitel wollen wir uns einige charakteristische Eigenschaften von Funktionen anschauen.

Definition 24 (Nullstelle). Sei f eine Funktion. Die Stelle x_0 , an der $f(x_0) = 0$ gilt, wird als **Nullstelle** der Funktion f bezeichnet.

Definition 25 (Polstelle). Sei f eine Funktion. Eine **Polstelle** der Funktion f ist ein Punkt x_0 , in dem f nicht definiert ist und die Punkte in der Umgebung von x_0 beliebig groß oder klein werden.

Definition 26 (Gerade und ungerade Funktion). Sei f eine Funktion. Wir bezeichnen f als **gerade**, falls gilt

$$f(-x) = f(x)$$

und **ungerade**, falls gilt

$$f(-x) = -f(x).$$

Definition 27 (Periodische Funktion). Eine Funktion f heißt **periodisch**, wenn es ein $\delta > 0$ gibt, so dass

$$f(x + \delta) = f(x).$$

Beispiel 6. Nachdem wir nun die Definitionen angegeben haben, möchten wir diese auch mit einem Beispiel hinterlegen. Sei hierzu $f(x) = x^2 - 4$, $g(x) = \frac{x^2}{x-1}$ und $h(x) = \sin(x)$. Die Funktion f besitzt bei $x_0 = 2$ und bei $x_0 = -2$ eine Nullstelle und ist gerade. Die Funktion g besitzt eine Nullstelle bei $x_0 = 0$ und eine Polstelle bei $x_1 = 1$. Die Funktion h besitzt unendlich viele Nullstellen aus der Menge $\{i \cdot \pi \mid i \in \mathbb{Z}\}$. Weiterhin ist diese Funktion periodisch und ungerade.

3.2 Folgen und Reihen

In diesem Kapitel wollen wir kurz wiederholen, was Folgen und Reihen sind und den Grenzwert einführen. Weiter möchten wir euch eine Intuition geben für das Verhalten von Funktionen nahe des Grenzwertes.

Definition 28 (Folgen). Sei M eine Menge. Eine **Folge** (a_n) ist eine Abbildung von den natürlichen Zahlen auf eine reelle Zahl, die jeder natürlichen Zahl eine reelle Zahl eindeutig zuordnet.

Eine Folge bei der für alle $i \in \mathbb{N}$ gilt

$$a_{i+1} - a_i = d$$

wird als **arithmetische Folge** bezeichnet. Falls für alle $i \in \mathbb{N}$ gilt

$$\frac{a_{i+1}}{a_i} = q$$

sprechen wir von einer **geometrischen Folge**.

Beispiel 7 (Folge). Eine Folge wäre beispielsweise

$$(a_n) = (n^2)_{n \in \mathbb{N}} = 1, 4, 9, \dots$$

die Folge der Quadratzahlen.

Definition 29 (Eigenschaften von Folgen). Sei (a_n) eine Folge. Wir sagen (a_n) ist **monoton zunehmend**, falls gilt

$$\forall n \in \mathbb{N} : a_{n+1} \geq a_n,$$

ist **monoton abnehmend**, falls gilt

$$\forall n \in \mathbb{N} : a_{n+1} \leq a_n,$$

und **konstant**, falls gilt

$$\forall n \in \mathbb{N} : a_{n+1} = a_n.$$

Seien nun $a, b \in \mathbb{R}$. Wir sagen (a_n) ist **beschränkt**, falls gilt

$$\forall n \in \mathbb{N} : a \leq a_n \leq b.$$

Wir bezeichnen a als die untere Schranke und b als die obere Schranke.

Definition 30. Sei (a_n) eine Folge. Wir definieren die n -te Partialsumme einer Folge als

$$s_n = \sum_{i=1}^n a_i.$$

Die Folge (s_n) der n -ten Partialsummen wird als **Reihe** bezeichnet.

3.2.1 Grenzwert

Sei (a_n) eine Folge und (s_n) eine Reihe. Wie wir bereits gelernt haben läuft das n bis unendlich. Wir können uns in diesem Fall fragen, was der Wert der Folge oder Reihe in der Nähe von unendlich ist. Man nennt dies den Grenzwert.

Definition 31 (Grenzwert). Sei (a_n) eine Folge und $\epsilon > 0$ eine kleine Zahl. Man sagt, die Folge hat einen **Grenzwert** a , falls es eine Stelle n_0 gibt, so dass für alle $i \geq n_0$ gilt:

$$|a - a_i| < \epsilon.$$

Wir schreiben den Grenzwert einer Folge gegen den Punkt i_0 als

$$\lim_{i \rightarrow i_0} (a_n) = a.$$

Bemerkung 12 (Wichtige Grenzwerte). Grenzwerte treten ständig auf und es ist wichtig einige der häufigsten zu kennen.

$$\begin{aligned} \lim_{n \rightarrow \infty} \frac{a}{n} &= 0, & a \in \mathbb{R} \\ \lim_{n \rightarrow \infty} a^n &= 0, & |a| < 1 \\ \lim_{n \rightarrow \infty} \sqrt[n]{a} &= 1, & a \in \mathbb{R}^+ \\ \lim_{n \rightarrow \infty} \sqrt[n]{n} &= 1 \\ \lim_{n \rightarrow \infty} \left(1 + \frac{1}{n}\right)^n &= e \\ \lim_{n \rightarrow \infty} \left(1 - \frac{1}{n}\right)^n &= \frac{1}{e} \\ \lim_{n \rightarrow \infty} \left(1 + \frac{k}{n}\right)^n &= e^k \end{aligned}$$

Satz 6 (Rechenregeln für Grenzwerte). Seien f, g Funktionen mit den Grenzwerten $\lim_{x \rightarrow x_0} f(x) = a$ und $\lim_{x \rightarrow x_0} g(x) = b$, dann gilt

$$\begin{aligned} \lim_{x \rightarrow x_0} c \cdot f(x) &= a, & c \in \mathbb{R} \\ \lim_{x \rightarrow x_0} f(x) \pm g(x) &= a \pm b \\ \lim_{x \rightarrow x_0} f(x) \cdot g(x) &= a \cdot b \\ \lim_{x \rightarrow x_0} \frac{f(x)}{g(x)} &= \frac{a}{b}, & b \neq 0 \end{aligned}$$

3.2.2 Vergleich von Funktionen

Oft ist es interessant zu wissen, ob eine Funktion schneller steigt als eine andere. Man bezeichnet dies als asymptotische Analyse. Diese klassifiziert das Grenzverhalten einer

Funktion, indem man diese auf ihren wesentlichen Eigenschaften reduziert. Dies wird häufig verwendet, um die Laufzeit von Algorithmen anzugeben. Wir definieren hierfür, wann zwei Funktionen in etwa gleich schnell sind.

Definition 32 (Gleiches asymptotisches Verhalten). *Zwei Funktionen f, g haben ein gleiches asymptotisches Verhalten, wenn gilt*

$$\lim_{i \rightarrow \infty} \left| \frac{f(i)}{g(i)} \right| < \infty.$$

Beispiel 8 (Asymptotisches Verhalten). *Sei $f(x) = 3x^2$ und $g(x) = x^2 + 2$. Um zu zeigen, dass diese beiden Funktionen asymptotisch gleich laufen, müssen wir nur zeigen, dass der Grenzwertbetrag ihres Quotienten kleiner als unendlich ist.*

$$\begin{aligned} \lim_{x \rightarrow \infty} \left| \frac{f(x)}{g(x)} \right| &= \lim_{x \rightarrow \infty} \left| \frac{3x^2}{x^2 + 2} \right| \\ &= \lim_{x \rightarrow \infty} \left| \frac{x^2(3)}{x^2 \left(1 + \frac{2}{x^2}\right)} \right| \\ &= \lim_{x \rightarrow \infty} \left| \frac{3}{1 + \frac{2}{x^2}} \right| \\ &= \left| \frac{3}{1 + 0} \right| = 3 < \infty \end{aligned}$$

Wir können also für eine beliebige Funktion eine Menge von Funktionen definieren, die gleich schnell sind.

3.2.3 Vollständige Induktion

Die Darstellung von Reihen ist meist einfacher möglich, als mit dem Summenzeichen. Angenommen wir haben eine Vermutung, wie die n -te Partialsumme einer Reihe berechnet wird. Wie beweisen wir diese Vermutung? Für diese Art von Beweisen verwendet man das Beweisprinzip der vollständigen Induktion.

Sei $A(n)$ eine beliebige Aussage über $n \in \mathbb{N}$. Die vollständige Induktion beginnt bei dem Beweis mit dem sogenannten **Induktionsanfang** (IA). Dieser testet ob die Aussage für $A(1)$ gilt.

Danach folgt der **Induktionsbeweis**. Dieser beginnt mit einer **Induktionsvoraussetzung** (IV), welche die Richtigkeit von $A(n)$ für ein **beliebiges, festes** $n \in \mathbb{N}$ annimmt und damit die Richtigkeit der **Induktionsbehauptung** (IB) $A(n+1)$ folgert.

Wenn diese Folgerung gelingt, ist die Aussage für alle $n \in \mathbb{N}$ korrekt.

Beispiel 9. *Damit dieses wichtige Prinzip etwas besser verständlich wird, wollen wir es hier einmal an einem sehr prominentem Beispiel durchspielen.*

3 Analysis

Sei hierzu folgende Aussage gegeben

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

Man bezeichnet dies als Gaußsche Summenformel.

Beginnen wir mit dem Induktionsanfang. Für $n = 1$ gilt

$$\sum_{i=1}^1 i = 1 = \frac{2}{2} = \frac{1 \cdot 2}{2}.$$

Die Aussage stimmt und wir können unsere Induktionsvoraussetzung formulieren. Falls für ein beliebiges aber festes $n \in \mathbb{N}, n > 1$

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

gilt, dann gilt auch

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

Der nun folgende Beweis wird auch Induktionsschritt genannt. Wir beginnen mit der linken Seite der Induktionsbehauptung und versuchen aus dieser mithilfe der Induktionsvoraussetzung die rechte Seite der Induktionsbehauptung zu erhalten.

$$\begin{aligned} \sum_{i=1}^{n+1} i &= \sum_{i=1}^n i + (n+1) \\ &= \frac{n(n+1)}{2} + (n+1) \\ &= \frac{n(n+1)}{2} + \frac{2(n+1)}{2} \\ &= \frac{n(n+1) + 2(n+1)}{2} = \frac{(n+1)(n+2)}{2} \end{aligned}$$

Wie wir in dem Induktionsschritt sehen, erhalten wir nach einigem Umformen die Induktionsbehauptung. Unsere Aussage stimmt also für alle $n \in \mathbb{N}$.

Wir werden auch hierzu noch weitere Beispiele in den Übungsaufgaben sehen.

3.3 Differenzialrechnung

Nachdem wir uns nun schon mit Grenzwerten beschäftigt haben brauchen wir für die Differenzialrechnung nun nur noch den Begriff der Stetigkeit.

Definition 33 (Stetigkeit). Die Funktion f ist an der Stelle x_0 **stetig**, falls zu jedem $\epsilon > 0$ ein $\delta > 0$ existiert, so dass für alle $x \in \mathcal{D}$ mit $|x - x_0| < \delta$ gilt,

$$|f(x) - f(x_0)| < \epsilon.$$

Bemerkung 13 (Stetigkeit). Die Stetigkeit kann auch über den Grenzwert definieren. Sei hierzu f eine Funktion. Man sagt, f ist an der Stelle x_0 stetig, wenn der Grenzwert $\lim_{x \rightarrow x_0} f(x)$ existiert und gleich $f(x_0)$ ist.

Eine Funktion f heißt stetig, wenn sie auf jeder Stelle ihres Definitionsbereiches \mathcal{D} stetig ist.

Satz 7 (Zwischenwertsatz). Falls die Funktion f im Intervall $[a, b]$ stetig ist, dann nimmt sie jeden Wert zwischen $f(a)$ und $f(b)$ an.

Beweis. Intuitiv können wir uns dies mit einer Skizze klar machen. □

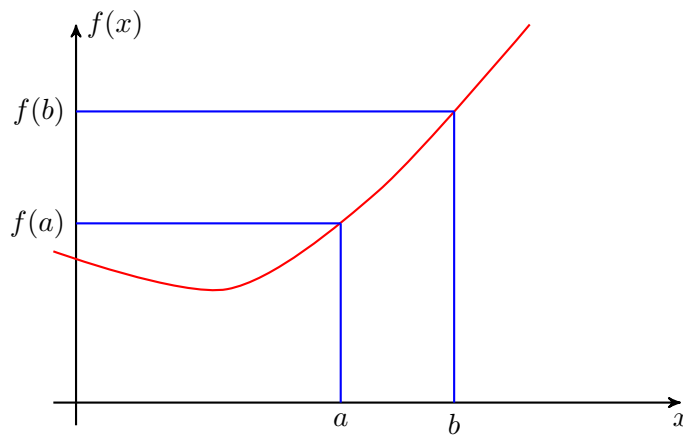


Abbildung 3.2: Wie wir in der Skizze sehen, wird in dem Intervall $[a, b]$ jeder Wert in dem Intervall $[f(a), f(b)]$ angenommen.

Definition 34 (Differenzierbar). Die Funktion f ist an der Stelle $x_0 \in \mathcal{D}$ **differenzierbar**, falls der Anstieg der Tangente an der Funktion an der Stelle x_0 als

$$\lim_{x \rightarrow x_0} \frac{f(x) - f(x_0)}{x - x_0}$$

existiert. Wir nennen die Steigung dieser Tangente am Punkt x_0 die Ableitung oder das Differenzial.

Definition 35 (Ableitung). Wenn die Funktion f in jedem Punkt ihres Definitionsbereiches differenzierbar ist, dann sagen wir die Funktion f ist differenzierbar. Wir definieren dann eine Funktion f' als die **Ableitung** von f , die jedem Punkt x die Ableitung von $f(x)$ zuordnet. Wir schreiben dies auch als

$$f'(x_0) = \frac{df}{dx}(x_0).$$

3 Analysis

Satz 8 (Mittelwertsatz). *Sei die Funktion f auf $[a, b]$ stetig und auf (a, b) differenzierbar. Dann gibt es einen Punkt $x_0 \in (a, b)$ für den gilt*

$$f'(x_0) = \frac{f(b) - f(a)}{b - a}.$$

Beweis. Beweisintuition durch Skizze in Abbildung 3.2. Wir sehen, dass die Funktion in (a, b) so gut wie linear ist. Wie berechnet man die Steigung dieser linearen Funktion und was bedeutet die Ableitung an einer Stelle dieser Funktion? Diese Fragen verdeutlichen die Beweisidee. \square

Wir haben nun viel über Stetigkeit und Differenzierbarkeit gehört und einige Wichtige Begriffe und Schreibweisen geklärt. Nun wollen wir aber einmal Ableitungen berechnen, wozu wir folgende Regeln brauchen.

Satz 9 (Ableitungsregeln). *Seien f, g Funktionen und an der Stelle x differenzierbar. Dann können wir folgende Ableitungen berechnen.*

$$\begin{aligned} (f + g)'(x) &= f'(x) + g'(x), \\ (c \cdot f)'(x) &= c \cdot f'(x) \quad , c \in \mathbb{R} \\ (f \cdot g)'(x) &= f'(x) \cdot g(x) + f(x) \cdot g'(x) \quad , \textbf{Produktregel} \\ \left(\frac{f}{g}\right)'(x) &= \frac{f'(x) \cdot g(x) - f(x) \cdot g'(x)}{g(x)^2} \quad , g(x) \neq 0, \textbf{Quotientenregel} \\ (f \circ g)'(x) &= f'(g(x)) \cdot g'(x) \quad , \textbf{Kettenregel} \end{aligned}$$

Weiter können wir folgende Ableitungen von elementaren Funktionen angeben.

$$\begin{aligned}
 \frac{d}{dx} c &= 0, c \in \mathbb{R} \\
 \frac{d}{dx} x^n &= n \cdot x^{n-1}, n \in \mathbb{Z} \\
 \frac{d}{dx} x^a &= a \cdot x^{a-1}, x > 0, a \in \mathbb{R} \\
 \frac{d}{dx} a^x &= a^x \ln(a), a > 0 \\
 \frac{d}{dx} \log_a(x) &= \frac{1}{x \ln(a)}, a \neq 1, a > 1 \\
 \frac{d}{dx} \ln(x) &= \frac{1}{x} \\
 \frac{d}{dx} e^x &= e^x \\
 \frac{d}{dx} \sin(x) &= \cos(x) \\
 \frac{d}{dx} \cos(x) &= -\sin(x) \\
 \frac{d}{dx} \tan(x) &= \frac{1}{\cos^2(x)} \\
 \frac{d}{dx} \cot(x) &= \frac{-1}{\sin^2(x)}
 \end{aligned}$$

Diese Liste ist natürlich nicht komplett, aber sie umfasst einige der am häufigsten genutzten Ableitungen.

Nachdem wir nun den Grenzwert und die Differenzierbarkeit von Funktionen kennen können wir einen wichtigen Satz verstehen, der die Berechnung von Grenzwerten vereinfacht.

Satz 10 (Regel von l'Hospital). *Seien f und g Funktionen, die differenzierbar sind. Wenn der Grenzwert der Funktionen gegen eine Stelle x_0 gleich 0 oder der Grenzwert des Betrages der Funktionen gegen ∞ gleich ∞ ist, so gilt:*

$$\lim_{x \rightarrow x_0} \frac{f(x)}{g(x)} = \lim_{x \rightarrow x_0} \frac{f'(x)}{g'(x)} \text{ mit } g'(x) \neq 0.$$

Bevor wir mit dem Kapitel über Differenzialrechnung enden möchten wir noch einen kurzen Einblick in die Bedeutung und Eigenschaften der Differenzierbarkeit.

Satz 11 (Eigenschaften von differenzierbaren Funktionen). *Sei f eine Funktion und auf*

3 Analysis

einem Intervall I zweimal differenzierbar. Dann gelten folgende Aussagen

$$\begin{aligned} \forall x \in I : f'(x) &\geq 0 \leftrightarrow & f \text{ ist } \mathbf{monoton\ wachsend} \text{ auf } I, \\ \forall x \in I : f'(x) &\leq 0 \leftrightarrow & f \text{ ist } \mathbf{monoton\ fallend} \text{ auf } I, \\ \forall x \in I : f''(x) &\geq 0 \leftrightarrow & f \text{ ist } \mathbf{konvex} \text{ auf } I, \\ \forall x \in I : f''(x) &\leq 0 \leftrightarrow & f \text{ ist } \mathbf{konkav} \text{ auf } I, \end{aligned}$$

Definition 36 (Minimum, Maximum und Wendepunkt). Sei $f : \mathcal{D} \rightarrow \mathcal{R}$ eine Funktion und sei weiterhin $U_\epsilon(x_0) = (x_0 - \epsilon, x_0 + \epsilon)$, $\epsilon \in \mathbb{R}^+$ eine Umgebung um die Stelle $x_0 \in \mathcal{D}$.

Wir bezeichnen ein x_0 als **lokales Minimum**, falls gilt

$$\forall x \in U_\epsilon(x_0) \cap \mathcal{D} : f(x) \geq f(x_0).$$

Die Stelle x_0 ist ein **globales Minimum**, falls die Aussage für alle $x \in \mathcal{D}$ gilt.

Ein x_0 für welches gilt:

$$\forall x \in U_\epsilon(x_0) \cap \mathcal{D} : f(x) \leq f(x_0).$$

bezeichnen wir als **lokales Maximum**. Auch hier reden wir von einem **globalen Maximum**, falls die Aussage für alle $x \in \mathcal{D}$ gilt.

Die Stelle x_0 wird als **Wendepunkt** bezeichnet, falls die Krümmung der Funktion am Punkt $(x_0, f(x_0))$ von konkav zu konvex oder umgekehrt übergeht.

Satz 12 (Kriterium für Minimum, Maximum und Wendepunkt). Sei f eine dreimal differenzierbare Funktion auf einem Intervall (a, b) . Für einen Punkt $x_0 \in (a, b)$ gilt dann:

- $(x_0, f(x_0))$ ist ein lokales Minimum von f , wenn $f'(x_0) = 0$ und $f''(x_0) > 0$,
- $(x_0, f(x_0))$ ist ein lokales Maximum von f , wenn $f'(x_0) = 0$ und $f''(x_0) < 0$ und
- $(x_0, f(x_0))$ ist ein Wendepunkt von f , wenn $f''(x_0) = 0$ und $f'''(x_0) \neq 0$. Man bezeichnet diesen Punkt als **Sattelpunkt**, wenn zusätzlich $f'(x_0) = 0$ ist.

Wir können dies zusammenfassen und verallgemeinern. Sei dazu f auf dem Intervall (a, b) n -mal differenzierbar. Sei weiter $x_0 \in (a, b)$ und $f'(x_0) = 0$ und $f^{(n)}(x_0) \neq 0$. Dann können wir folgende Fallunterscheidung machen:

n ist gerade $(x_0, f(x_0))$ ist ein Minimum oder Maximum von f ,

n ist ungerade $(x_0, f(x_0))$ ist ein Sattelpunkt.

Im nächsten Kapitel wollen wir uns nun abschließend der Analysis mit der Umkehrung der Differenzialrechnung beschäftigen.

3.4 Integralrechnung

Definition 37 (Stammfunktion). Sei $f : \mathcal{D} \rightarrow \mathbb{R}$ eine beliebige Funktion auf dem Intervall $\mathcal{D} \subseteq \mathbb{R}$. Die Stammfunktion $F : \mathcal{D} \rightarrow \mathbb{R}$ von f ist eine Funktion, für die gilt

$$\forall x \in \mathcal{D} : F'(x) = f(x).$$

Oftmals kann man die Stammfunktion einer Funktion erraten. So wissen wir sofort, dass eine Stammfunktion von $6x^2$ gleich $2x^3$ ist. Wir schreiben hier eine, da es beliebig viele Stammfunktionen von $6x^2$ gibt. Eine alternative Stammfunktion könnte $2x^3 + 9$ sein.

Definition 38 (Unbestimmte Integral). Wir bezeichnen die Menge aller Stammfunktionen von f als **unbestimmtes Integral** von f und vereinbaren folgende Schreibweise.

$$\int f(x)dx = F(x) + c,$$

wobei $c \in \mathbb{R}$ die **Integrationskonstante**, x die **Integrationsvariable** und f den **Integranden** bezeichnet.

Satz 13 (Integrationsregeln). Wie auch für die Differenziation gibt es für die Integration einige Regeln, die uns helfen das unbestimmte Integral auszurechnen. Seien hierzu die Funktionen $f, g : \mathcal{D} \rightarrow \mathbb{R}$. Dann gelten folgende Regeln:

$$\begin{aligned} \int (f(x) + g(x))dx &= \int f(x)dx + \int g(x)dx \\ \int (c \cdot f(x))dx &= c \cdot \int f(x)dx, \quad c \in \mathbb{R} \\ \int f(x)g'(x)dx &= f(x)g(x) - \int f'(x)g(x)dx, \quad \textbf{Partielle Integration} \\ \int f(g(x))g'(x)dx &= \int f(u)du, \quad \textbf{Integration durch Substitution,} \end{aligned}$$

wobei bei der Integration durch Substitution $g(x)$ durch u substituiert wurde.

Auch für die Integration gibt es einige elementare Funktionen die wir hier mit ihren

3 Analysis

Stammfunktion angeben möchte. Hierbei sei $k \in \mathbb{R}$ die Integrationskonstante.

$$\begin{aligned}\int c dx &= c \cdot x + k && , c \in \mathbb{R} \\ \int x^a dx &= \frac{x^{a+1}}{a+1} + k && , a \in \mathbb{R}, a \neq -1 \\ \int a^x dx &= \frac{a^x}{\ln(a)} + k && , a \in \mathbb{R}, a > 0 \\ \int x^{-1} dx &= \ln|x| + k \\ \int e^x dx &= e^x + k \\ \int \sin(x) dx &= -\cos(x) + k \\ \int \cos(x) dx &= \sin(x) + k\end{aligned}$$

Integration wird oftmals als schwieriger empfunden, als die Differenziation. Aus diesem Grund wollen wir hier einige Beispiele dazu angeben. Mehr Beispiele werden in der Übung angesprochen.

Beispiel 10 (Unbestimmte Integrale). Sei $f(x) = 3x^2 + \sin(x) + \frac{2}{x}$. Wir berechnen ein unbestimmtes Integral als

$$\begin{aligned}\int (3x^2 + \sin(x) + \frac{2}{x}) dx &= \int 3x^2 dx + \int \sin(x) dx + \int \frac{2}{x} dx \\ &= 3 \cdot \int x^2 dx + \int \sin(x) dx + 2 \cdot \int \frac{1}{x} dx \\ &= 3 \cdot \frac{1}{3} x^3 - \cos(x) + 2 \cdot \ln|x| + c.\end{aligned}$$

Die Integrationskonstante sei hierbei $c \in \mathbb{R}$.

Sei nun $h(x) = x^2 \cdot \sin(x)$. Wir benutzen nun die partielle Integration mit $f(x) = x^2$ und $g(x) = -\cos(x)$.

$$\begin{aligned}\int x^2 \cdot \sin(x) dx &= x^2 \cdot (-\cos(x)) - \int -\cos(x) x dx \\ &= -x^2 \cdot \cos(x) + 2x \cdot \sin(x) - \int 2 \cdot \sin(x) dx \\ &= -x^2 \cdot \cos(x) + 2x \cdot \sin(x) + 2 \cos(x) + c.\end{aligned}$$

Auch hier ist Integrationskonstante $c \in \mathbb{R}$. Wir müssen hier zweimal partiell integrieren.

Als letztes wollen wir ein Beispiel für eine Integration durch Substitution durchrechnen. Schauen wir uns hierzu die Funktion $i(x) = \cos(x^2) \cdot 2x$ an. Wir setzen $u = x^2$ und

erhalten dadurch $\frac{du}{dx} = 2x \leftrightarrow dx = \frac{du}{2x}$. Durch die Substitution können wir das Integral wie folgt berechnen:

$$\begin{aligned} \int \cos(x^2) \cdot 2x dx & \stackrel{x^2=u, dx=\frac{du}{2x}}{=} \int \cos(u) du \\ & \stackrel{=}{=} \sin(u) + c \\ & \stackrel{u=x^2}{=} \sin(x^2) + c. \end{aligned}$$

Mit der Integrationskonstanten $c \in \mathbb{R}$.

Satz 14 (Bestimmtes Integral und Hauptsatz der Differential- und Integralrechnung). Das Integral kann dazu benutzt werden die Fläche unter einer Funktion $f : \mathcal{D} \rightarrow \mathbb{R}$ im Intervall $I = [a, b]$ zu berechnen. Wir bezeichnen a und b dabei als **untere- und obere Intervallgrenze** und schreiben das ganze als:

$$\int_a^b f(x) dx$$

Dieses bezeichnen wir als **bestimmtes Integral**. Wenn f auf $[a, b]$ stetig ist können wir dieses als

$$\int_a^b f(x) dx = F(b) - F(a)$$

berechnen. Dabei ist F eine beliebige Stammfunktion von f .

4 Einführung Informatik

Wissenschaft von der systematischen Verarbeitung von Informationen, besonders der automatischen Verarbeitung mithilfe von Computern. - Gesellschaft für Informatik

Informatik ist die Wissenschaft von der systematischen Darstellung, Speicherung, Verarbeitung und Übertragung von Informationen, besonders der automatischen Verarbeitung mithilfe von Digitalrechnern - Wikipedia

Angewandte Informatik beschäftigt sich mit konkreten Anwendungsgebieten der Informatik und ihren spezifischen Anforderungen (Beispiele: Bioinformatik, Wirtschaftsinformatik, medizinische Informatik).

Praktische Informatik behandelt Konzepte der Programmierung und der Entwicklung von Programmen und komplexen, informationsverarbeitenden Systemen (Beispiele: Softwaretechnik, Datenbanksysteme).

Technische Informatik beschäftigt sich mit Computer-Hardware, deren Entwurf und maschinennahen Aspekten (Beispiele: Rechnerarchitektur, Signalanalyse).

Theoretische Informatik bietet theoretische Grundlagen für alle Teilbereiche der Informatik (Beispiele: Komplexitätstheorie, Algorithmik).



Abbildung 4.1: Grundlegender Ablauf in der Informatik

Die Informatik ist im Grunde die ganze Zeit damit beschäftigt Probleme zu lösen. Den grundlegenden Ablauf vom Problem bis zu Lösung oder Ausführung ist in Abbildung 4.1 dargestellt.

5 Algorithmmik

Wir kommen nun von der Formalisierung unseres Problems zur Entwicklung und Analyse von Algorithmen.

5.1 Motivation

Algorithmen werden in vielen Bereichen des täglichen Lebens verwendet. Ob es ein Kochrezept ist, die IKEA Anleitung oder einfach die Reihenfolge in der wir uns am Morgen anziehen oder unsere Kleidung auswählen. Algorithmen sind allgegenwärtig. Selbst in diesem Vorkurs habt ihr bereits einige Algorithmen verwendet. Beispielsweise zur Umrechnung von dezimalen Zahlen in binär Zahlen.

Versuchen wir einmal einen eigenen Algorithmus für ein Problem zu entwickeln.

Wer ist der jüngste Studierenden hier im Raum? Entwickeln wir einen Algorithmus:

Problem Wir wollen den jüngsten Studierenden in der Vorlesung finden.

Formalisierung Wir könnten das Alter oder das Geburtsdatum benutzen. Weiter müssen wir uns überlegen ob wir nur die Alter der gerade anwesenden oder die Liste mit Anmeldungen benutzen wollen. Ist die Lösung eindeutig?

Algorithmenentwurf Wir durchlaufen die Liste und merken uns den kleinsten Wert. Weiter können wir auch eine Divide and Conquer Variante nehmen oder eine geeignete Datenstruktur nutzen.

5.2 Eigenschaften von Algorithmen

Jede Abfolge von Aktionen ist nicht gleich ein Algorithmus. Für Algorithmen gelten bestimmte Eigenschaften. So müssen Algorithmen Eindeutig, Ausführbar, Endlich, Terminiert, Deterministisch und Terminiert sein.

Definition 39 (Eindeutig). *An jeder Stelle des Algorithmus muss **eindeutig** festgelegt sein, was als nächstes zu tun ist. Die Anweisung muss also unmissverständlich formuliert sein.*

Definition 40 (Endlichkeit). *Ein Algorithmus darf nur eine **endliche** Länge besitzen. Zudem dürfen seine Daten nur endlich viel Platz belegen.*

Definition 41 (Deterministisch). *Für jede Anweisung gibt es genau eine folge Anweisung. Wenn man sich aussuchen kann wie man weiter macht spricht man von **Nicht-deterministisch**.*

Definition 42 (Determiniertheit). Für die gleichen Eingabedaten muss der Algorithmus das gleiche Resultat liefern.

Definition 43 (Terminierung). Ein Algorithmus muss nach einer **endlichen** Anzahl von Anweisungen beendet sein oder anhalten.

Beispiel 11 (Sortieralgorithmus). Wir können uns die Eigenschaften von Algorithmen an einem einfachen Sortieralgorithmus verdeutlichen.

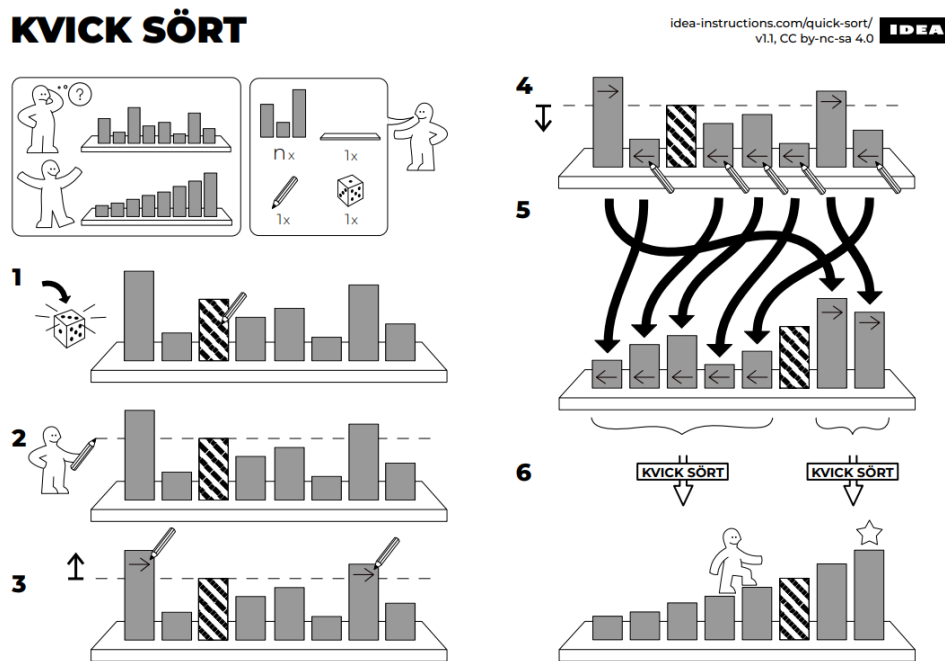


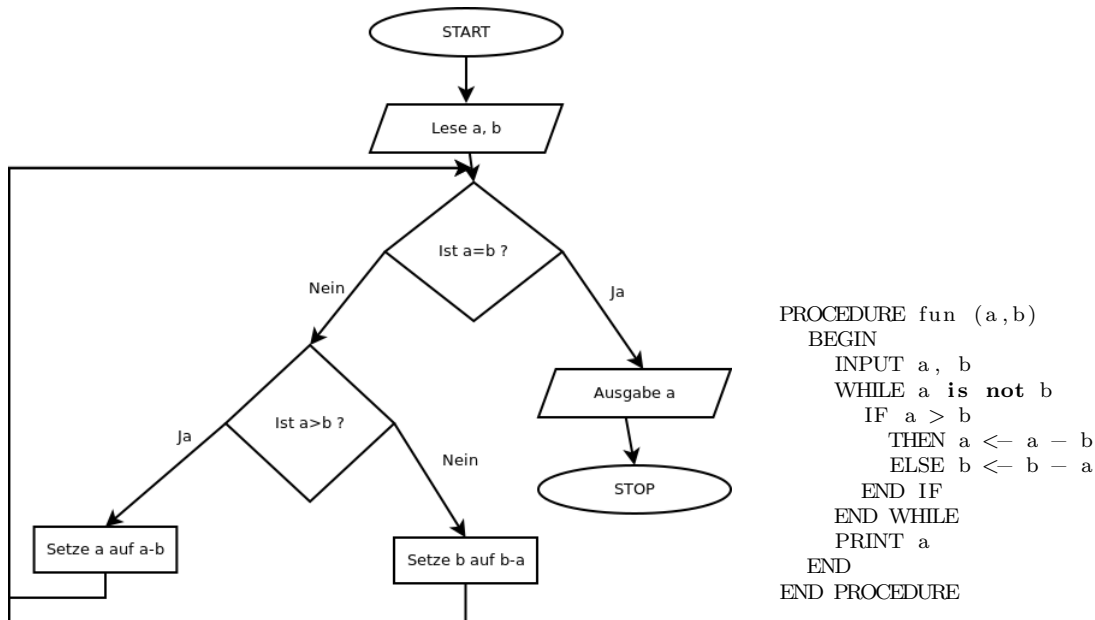
Abbildung 5.1: Erklärung des Quicksort Algorithmus mithilfe einer IKEA Anleitung.

Der dargestellte Sortieralgorithmus arbeitet, indem in jedem Schritt ein Element zur Teilung (in der Abbildung 5.1 schraffiert) ausgewürfelt wird, die Elemente bezüglich dieses Elementes sortiert und anschließend die Elemente kleiner und größer jeweils wieder mit dem Quicksort Algorithmus sortiert werden.

Die Schritte für diesen Algorithmus sind an jeder Stelle **eindeutig** und durch die Nummerierung auch **deterministisch**. Weiter werden die Elemente in jedem Schritt geteilt und die Listen links und rechts werden kleiner (das Element zur Teilung wird nie wieder betrachtet). Dies führt dazu, dass der Algorithmus nach einer **endlichen** Dauer endet (**Terminierung**) und jede Liste von Elementen am Ende sortiert ausgibt (**Determiniertheit**).

5.3 Flussdiagramme und Pseudocode

Wenn man Algorithmen entwickelt muss man sie irgendwie darstellen. Viele tun dies mit Flussdiagrammen, noch mehr mit Pseudocode und sehr wenige mit Struktogrammen¹



Pseudocode verwendet dafür bereits definierte Schlüsselworte. Vermutlich werden Sie recht schnell eine eigene Art entwickeln das Verhalten von Programmen zu beschreiben.

5.3.1 Flussdiagramme

Flussdiagramme sind eine grafische Darstellung von Algorithmen. Sie bestehen im Grunde aus Pfeilen und Symbolen. Wie diese Pfeile und Symbole auszusehen haben und welche Funktion sie haben sind in der DIN 66001 genormt.

5.3.2 Start und Stop

Um den Start und das Ende von Ablaufplänen zu zeigen gibt es spezielle Symbole, welche START und STOP beinhalten.



Abbildung 5.3: Beginn und Ende des Flussdiagrammes.

¹Vergessen Sie gleich wieder.

5.3.3 Anweisungen

Für einfache Anweisungen wie $a = b + c$ verwendet man Rechtecke.

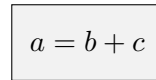


Abbildung 5.4: Eine einfache Anweisung.

5.3.4 Verzweigungen

Verzweigungen werden genutzt um zwischen Alternativen zu unterscheiden. Beispielsweise kann man das Maximum zweier Zahlen a, b bestimmen indem man fragt ob $a < b$ ist. Die Ausgehenden Pfeile werden meist links und rechts angebracht und mit JA und NEIN oder True und False oder $+$ und $-$ bezeichnet.

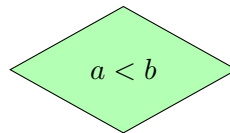


Abbildung 5.5: Eine Verzweigung.

5.3.5 Ein- und Ausgabe

Natürlich möchte man in seinem Algorithmus auch Daten eingeben oder Dinge ausgeben. Hierzu werden Ein- und Ausgabeblöcke verwendet.



Abbildung 5.6: Eine Ausgabe und eine Eingabe.

5.3.6 Programmaufrufe

Oftmals möchte man wiederkehrende Funktionen nicht mehrfach aufschreiben und verwendet deshalb Programmaufrufe oder Unterprogramme.

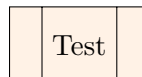
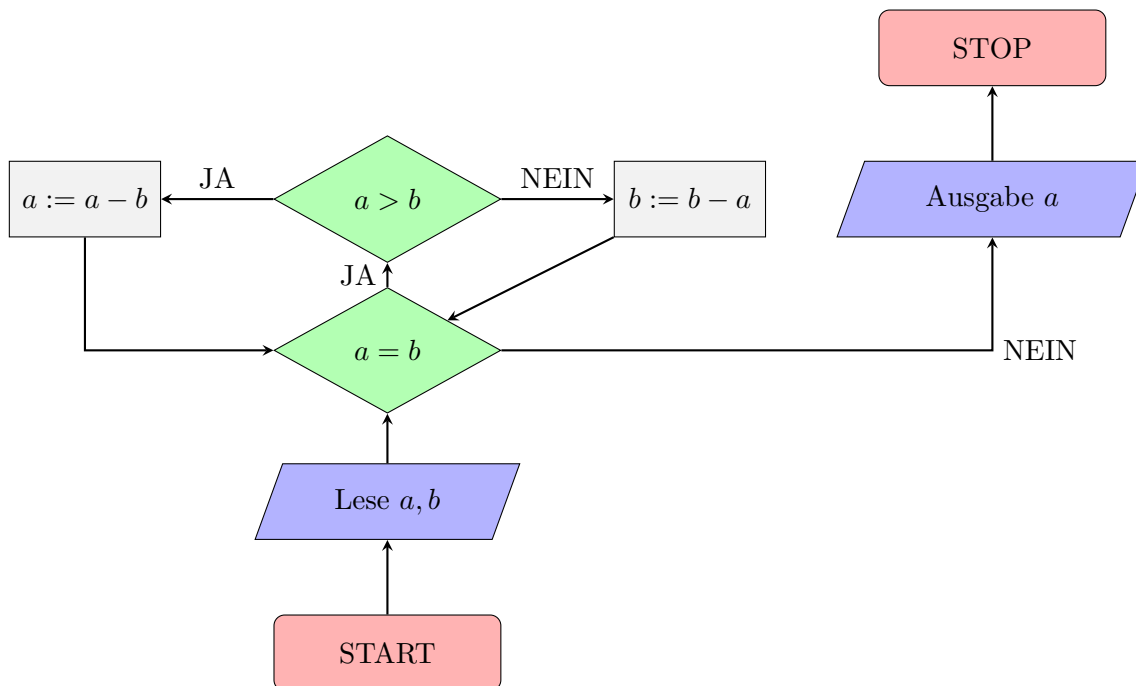


Abbildung 5.7: Ein Aufruf eines Unterprogrammes.

5.3.7 Anwendung von Flussdiagrammen

Hier nun nochmal das Beispiel von oben mit schönen bunten Bildern.



5.3.8 Andere Darstellungen

Pseudocode

Hier soll das oben dargestellte Beispiel noch einmal kurz im Pseudocode angegeben werden.

Algorithm 1 Beispield pseudocode

```

1: Lese  $a, b$ 
2: while  $a \neq b$  do
3:   if  $a > b$  then
4:      $a := a - b$ 
5:   else
6:      $b := b - a$ 
7:   end if
8: end while
9: Ausgabe  $a$ 

```

Wie bereits erwähnt benutzt Pseudocode Schlüsselworte um die verschiedenen Kontrollfunktionen aus Flussdiagrammen zu ersetzen. Verzweigungen werden oftmals mit **if**,

else und Schleifen mit **while**, **for** eingeleitet. Weiter gibt es diverse Möglichkeiten die Blöcke zu begrenzen (in unserem Beispiel **do**, **end if**, **end while**).

Struktogramme

Der Vollständigkeit halber möchte ich zusätzlich ein Struktogramm zeigen. Dieses ist in Abbildung 5.8 abgebildet.

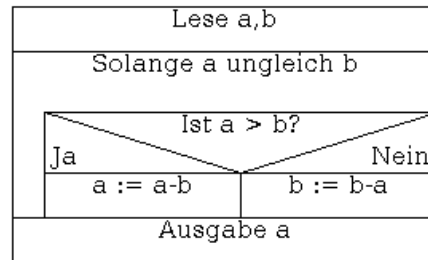


Abbildung 5.8: Ein Struktogramm unseres Beispieles.

6 Python

6.1 Geschichte

Python wurde 1991 von Guido van Rossum als Weiterentwicklung von ABC entwickelt. Das Ziel war eine leicht zu erlernende Programmiersprache zu entwickeln, die der Englischen Sprache angelehnt ist. Die erste Vollversion wurde 1994 veröffentlicht. Danach wurde 2000 Python 2.0, 2008 Python 3.0, 2015 Python 3.5 und 2016 Python 3.6 veröffentlicht. Der Name geht nicht wie viele Denken auf die Schlange zurück, sondern auf die Komikertruppe MontyPython. Trotzdem gibt es viele Anspielungen auf Schlangen in weiteren Paketen wie Boa oder in dem Tool Anacondo.

6.2 Installation und Start

Python kann sehr einfach über die Website www.python.org installiert werden. Wir verwenden in der Vorlesung Python 3. Die interaktive Konsole kann unter Linux mithilfe des Befehles `$python3` gestartet werden. Danach können hinter `>>>` die Python Befehle geschrieben werden.

6.3 Syntax

Unter Syntax versteht man allgemein ein Regelsystem zur Kombination elementarer Zeichen zu zusammengesetzten Zeichen in natürlichen oder künstlichen Zeichensystemen. Die Zusammenfügungsregeln der Syntax stehen hierbei den Interpretationsregeln der Semantik gegenüber. - Wikipedia

Im folgenden besprechen wir einige Syntaxregeln der Sprache Python 3.0. Um mit Python zu programmieren können wir die interaktive Umgebung nutzen. Wir starten sie über die Konsole Mithilfe des Befehls `python3`.

```
$ python3
>>>
```

Listing 6.1: Start der interaktiven Konsole von Python3

6.3.1 Anweisungen

Anweisungen werden einfach eingegeben und mit ENTER bestätigt. Zuweisungen können mit `=` vorgenommen werden. In dem Beispiel wird `a` der Wert 3 zugewiesen und in der nächsten Zeile `2 + a` berechnet. Das Ergebnis 5 wird sofort ausgegeben.

6 Python

```
>>> i = 0
>>> n = 10
>>> while i < n:
    print(i)
    i = i+1
0
1
..
9
>>> for i in range(10):
    # range produziert d
    print(i)
0
1
..
9
```

Listing 6.4: Beispiele zu Schleifen

```
>>> a = 3
>>> 2 + a
5
```

Listing 6.2: Einfache Anweisungen

6.3.2 Verzweigungen

Verzweigungen werden mithilfe des Schlüsselwortes **if** eingeleitet. Nach der Bedingung folgt ein **:**. In der nächsten Zeile werden nun mit einem Einschub die Anweisungen der **wahr** Fälle eingegeben. Folgend kann optional **elif** oder **else** angegeben werden. **elif** erwartet wieder eine Bedingung. Beide werden mit einem **:** beendet und in der nächsten Zeile mit Einschub die Anweisungen angegeben.

```
>>> if a > 2:
    print(a)
elif a == 2:
    print("a-ist-2")
else:
    print("Fehler")
a ist 2
```

Listing 6.3: Verzweigung über a

6.3.3 Schleifen

Schleifen werden entweder mit **while** oder **for** eingeleitet. Der Unterschied besteht in der Verwendung. Die **while**-Schleife erwartet eine Bedingung und wird so lange ausgeführt, wie die Bedingung zu **wahr** ausgewertet wird. Die **for**-Schleife im Gegensatz kann dazu verwendet werden Elemente einer Liste oder eines Bereiches (**range**) zu durchlaufen.

```

>>> a = list()
>>> a.append(1)
>>> a.append(4)
>>> a.append(3)
>>> a.sort()
>>> print(len(a))
3
>>> for i in range(len(a)):
>>>     print(i)
0
1
2
>>> for i in a:
>>>     print(i)
1
3
4
>>> a.pop(1)
3
>>> print(a)
[1, 4]

```

Listing 6.6: Verzweigung über a

6.3.4 Ein- und Ausgabe

Ausgaben werden einfach mit dem `print` Befehl realisiert. Mithilfe des Befehls `input` können Inhalte eingelesen werden. Diese werden als Zeichenkette eingelesen und können mittels `int` oder `float` umgewandelt werden.

```

>>> a = int(input("Zahl: -"))
Zahl: 3
>>> print(a)
3

```

Listing 6.5: Ein- und Ausgabe

6.3.5 Listen

Listen sind ein elementarer Datentyp von Python. Eine Liste kann einfach als `[]` oder `list()` angelegt werden. Mithilfe von `append` können Elemente hinzugefügt werden und `pop` entfernt ein Element. Die Länge der Liste liefert der Befehl `len`. Mithilfe von `sort` können Listen sortiert werden.

6.3.6 Funktionen

Wiederkehrende Aufgaben möchte man häufig nicht jedes mal neu schreiben sondern wiederbenutzen. Hierzu eignet sich die Definition einer Funktion. Eine Beispielfunktion ist die Fakultät. Wir wollen nicht jedes mal neu beschreiben, wie man die Fakultät ausrechnen sondern lieber einfach die Zahl n angeben und das Ergebnis erhalten. Dies könnte die Funktion *fak* sein. n nennt man hierbei den Parameter.

6 Python

```
def fak(n):  
    m = 1  
    for i in range(1, n+1):  
        m *= i  
    return m
```

Abbildung 6.1: Beispiel einer Fakultätsfunktion mit Parameter n .

Beispiel 12 (Binomialkoeffizient). *Der Binomialkoeffizient ist eine Funktion die Anzahl der Möglichkeiten aus einer Menge mit n Objekten k Objekte auszuwählen zu berechnen. Angenommen wir schauen uns sämtliche Binomialkoeffizienten von $n \leq 100$ und $k < n$ an. Wie viele dieser sind größer als 10^6 ?*

Um dieses Problem zu lösen, brauchen wir zuerst zwei Funktionen die uns die Fakultät und den Binomialkoeffizienten ausrechnet.

```
def fak(n):  
    m = 1  
    for i in range(1, n+1):  
        m *= i  
    return m  
  
def binom(n, k):  
    return fak(n) / (fak(k) * fak(n-k))
```

Abbildung 6.2: Die Funktionen Fakultät und Binomialkoeffizient

Nun können wir alle Möglichkeiten durchlaufen und schauen ob der Binomialkoeffizient größer als 10^6 ist. Wenn dies der Fall ist zählen wir eine Variable l eins hoch.

```
l = 0  
for n in range(1, 101):  
    for k in range(1, n+1):  
        if binom(n, k) > 10**6:  
            l += 1  
  
print(l)
```

Abbildung 6.3: Durchtesten aller Möglichkeiten

Das `print` an der letzten Stelle liefert uns die Antwort 4075.

6.3.7 Rekursion

Wenn eine Funktionen einen Funktionsaufruf von sich selbst enthält reden wir von einer Rekursion. So können wir die Fakultätsberechnung aus dem vorherigen Beispiel auch rekursiv ausrechnen.

Der Aufruf `fak(3)` würde in diesem Fall als `fak(3) = 3*fak(2) = 3*2*fak(1) = 3*2*1 = 6` ausgewertet werden. Weitere prominente Beispiele sind die Berechnung der

Fibonacci Zahlen $f_n = f_{n-1} + f_{n-2}, f_1 = f_2 = 1, n \geq 3$ oder die Ackermann Funktion

$$\begin{aligned} a(0, m) &= m + 1, \\ a(n + 1, 0) &= a(n, 1), \\ a(n + 1, m + 1) &= a(n, a(n + 1, m)). \end{aligned}$$

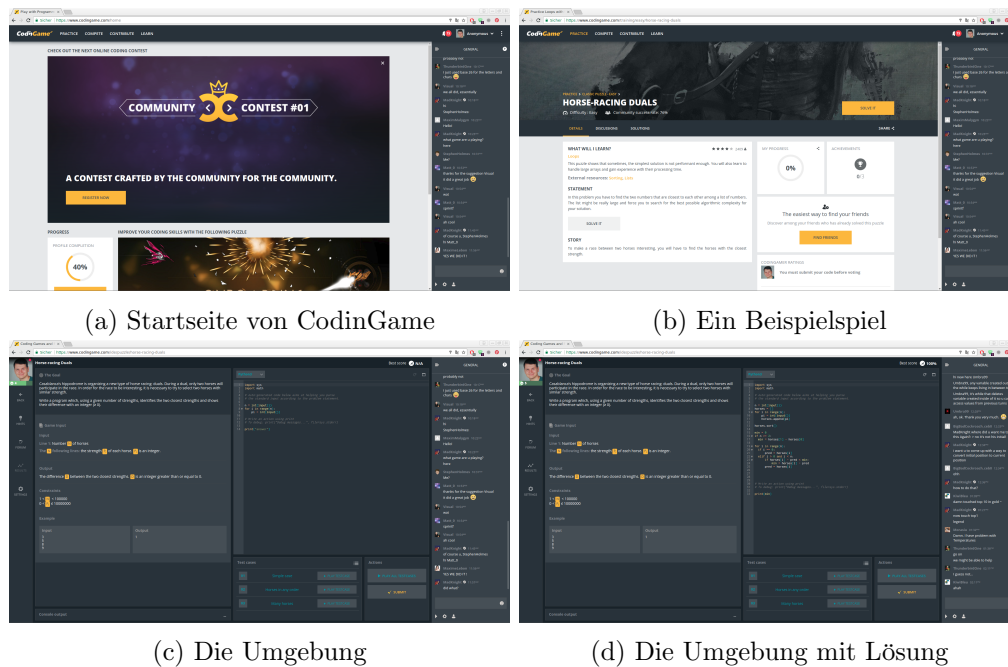
Sie können diese zur Übung selbstständig in Python Code übersetzen.

```
def fak(n):
    if n <= 1:
        return 1
    else:
        return n*fak(n-1)
```

Abbildung 6.4: Rekursive Fakultätsberechnung

6.4 CodinGame

Für die Übung verwenden wir die Website CodinGame www.codingame.com. Die folgenden Bilder enthalten eine Übersicht über das Lösen eines Spieles. Wir verwenden als Beispiel das Spiel **Horse-racing Duals**.



7 Kommandozeile

Das folgende Kapitel soll den Umgang mit der Konsole an einigen Beispielen demonstrieren. Hierzu werden Befehle angegeben. Die Befehle haben die Form **Befehl** *Parameter*.

7.1 Bewegen in Verzeichnissen

Am meisten muss man in der Konsole von einem Verzeichnis in ein anderes Verzeichnis mit den benötigten Dateien wechseln. Hierzu helfen die folgenden Befehle.

cd **Verzeichnis** wechselt in das Verzeichnis. Mit **cd ..** wird ein Verzeichnis nach oben gegangen.

pwd gibt den aktuellen Verzeichnispfad aus.

ls listet alle Ordner und Dateien des Verzeichnisses auf.

7.2 Hilfe

Sollte man einmal nicht weiter wissen kann man mit **man** *Befehl* das Manual zu dem Befehl öffnen oder mit **info** *Thema*, Befehle zu einem Thema finden.

7.3 Dateioperationen

Wenn man nun einmal in dem Verzeichnis gelandet ist, welches man benötigt kann man dort Dateien erzeugen, betrachten und löschen. Hierzu helfen folgende Befehle.

cp **Quelle** **Ziel** kopiert die Datei von der Quelle zum Ziel.

mv **Quelle** **Ziel** verschiebt die Datei von der Quelle zum Ziel.

mkdir **Verzeichnis** legt das Verzeichnis an. **rmdir** kann zum Löschen verwendet werden.

touch **Datei** erzeugt die Datei. **rm** kann zum Löschen verwendet werden.

cat **Datei** zeigt den Inhalt der Datei in der Konsole an. Ein alternativer Befehl ist **less**. Wenn man nur den Anfang oder das Ende der Datei anzeigen möchte kann man **head** oder **tail** benutzen.

nano **Datei** öffnet die Datei in einen Texteditor in der Konsole. Wenn die Datei noch nicht existiert, wird sie erzeugt.

gedit **Datei** öffnet die Datei in einem externen Texteditor.

7.4 Programme

Oft möchte man Programme in der Konsole starten. Diese werden einfach als **Befehl** gestartet. In unserem Kurs haben wir beispielsweise Java als `java -jar petri.jar` gestartet oder aber die interaktive Umgebung von Python3 als `python3`.

7.5 Weitere Befehle

Es stehen im Terminal noch weitere wichtige Befehle zur Verfügung, die uns die Arbeit im Linux Betriebssystem vereinfacht.

`clear` fügt genügend Leerzeilen ein, damit die aktuelle Befehlszeile wieder am oberen Anfang des Terminals steht.

`whatis Programm` liefert eine kurze Beschreibung eines übergebenen Programmes.

`which Programm` gibt den Installationspfad zum Programm an.

`df` zeigt die Auslastung des Dateisystems (disk free).

`ps -e` gibt eine Übersicht über die aktuell laufenden Prozesse aus. Jedem Prozess wird dabei eine PID zugewiesen.

`sudo killall -9 PID` beendet den Prozess mit der übergebenen PID.

`shutdown -h delay` fährt den Rechner nach delay Minuten herunter. Falls kein delay angegeben wird fährt der Rechner sofort herunter.