

Lernzettel

Pascal Diller

October 24, 2024

Contents

Logik	4
Mengen	4
Relationen	6
Äquivalenzrelationen	7
Abbildungen	8
Zahlensysteme	9
Binärsystem	9
Carry-Flag	9
Zweierkomplement	9
Hexadezimalsystem	9
Oktalsystem	10
Festkommazahlen	10
Gleitkommazahlen: IEEE 754	10
Aufbau	10
Dezimal zu IEEE 754	11
IEEE 754 zu Dezimal	11
Fehlererkennung	12
Redundanzen	12
Hamming-Distanz	12
Parität	12
Zweidimensionale Parität	13
Hamming-Code	13
Summenzeichen und Produktzeichen	14
Summenzeichen	14
Produktzeichen	14
Rechenregeln	15
Bruchregeln	15
Potenzgesetze	15
Wurzelgesetze	15
Logarithmengesetze	15

Trigonometrie	16
Bogenmaß	16
Differentialrechnung	16
Ableitungsregeln	16
Natürliche Potenzen	16
Summenregel	16
Produktregel	16
Kettenregel	16
Ebenen	17
Parameterform	17
Normalenform	17
Koordinatenform	17

Logik

- " \wedge ": **Und**
- " \vee ": **Oder**
- " \neg ": **Nicht** (Verneinung)
- $A \implies B$: A **impliziert** B
- $A \impliedby B$: A wird durch B **impliziert**
- $A \iff B$: A ist äquivalent zu B
Es gilt: $A \implies B$ und $A \impliedby B$
- \forall : **Für alle**
- \exists : **Es existiert (mindestens) ein**

Mengen

Eine **Menge** ist eine Zusammenfassung von (mathematischen) Objekten.
Die Objekte in einer Menge werden als **Elemente** bezeichnet.

- $x \in M$: x in/Element M
- $x \notin M$: x nicht in/Element M

Defintion einer Menge:

- Aufzählung:

$$M_1 = \{0, 1, 2, 3, 5, 8, -1\}; \quad M_2 = \{1, 2, 3, 4, 5, \dots\}$$

Es kommt **nicht** auf die **Reihenfolge** und **nicht** auf **Verdopplungen** an: $\{1, 3, 2, 3\} = \{3, 2, 1\} = \{1, 2, 3\}$

- Beschreibung:

$$M_3 = \{x \in \mathbb{R} : x \geq -1 \wedge x \leq 1\} = [-1, 1]$$

Menge B ist eine **Teilmenge** von Menge A , wenn für jedes $x \in A$ auch $x \in B$ gilt.

- $A \subset B$ ("A ist eine Teilmenge von B")
- $A \supset B$ ("B ist eine Teilmenge von A")

Mengenoperationen:

- **Vereinigung** der Mengen A und B

$$A \cup B = \{x : x \in A \vee x \in B\} \text{ ("A vereinigt B")}$$

- **Durchschnitt** der Mengen A und B

$$A \cap B = \{x : x \in A \wedge x \in B\} \text{ ("A geschnitten B")}$$

- **Differenzmenge** der Mengen A und B

$$A \setminus B = \{x : x \in A \wedge x \notin B\} \text{ ("A ohne B")}$$

Kartesisches Produkt:

sei $n \in \mathbb{N}$ und seien X_1, \dots, X_n Mengen, dann ist

$$X_1 \times \dots \times X_n = \{(x_1, \dots, x_n) : x_i \in X_i, \text{ für } i = 1, \dots, n\}$$

die Menge der **n-Tupel** mit i -ter Koordinate x_i in X_i für $i = 1, \dots, n$.

Potenzmenge:

Die Menge aller Teilmengen einer Menge X heißt Potenzmenge von X und wird mit $\mathcal{P}(X)$ bezeichnet:

$$\mathcal{P}(X) = \{Y : Y \subset X\}$$

Es gilt immer: $\emptyset \in \mathcal{P}(X)$ und $X \in \mathcal{P}(X)$.

Beispiel: Sei $X = \{1, 2, 3\}$. Dann ist

$$\mathcal{P}(X) = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}\}$$

Sei P eine Menge bestehend aus Mengen. Dann steht

$$\bigcup_{Y \in P} Y = \{y : \text{es gibt } Y \in P \text{ so dass } y \in Y\}$$

für die (möglicherweise unendliche) Vereinigung aller Mengen in P .

Partitionen:

Sei X eine Menge. Eine Partition von X ist eine Teilmenge $P \in \mathcal{P}(X) \setminus \{\emptyset\}$ sodass

- für alle $Y, Z \in P$ mit $Y \neq Z, Y \cap Z = \emptyset$ (Y und Z sind disjunkt).
- $\bigcup_{Y \in P} Y = X$.

Definierte Mengen:

- Leere Menge: $\emptyset = \{\}$
- Natürliche Zahlen: $\mathbb{N} = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, \dots\}$ ($0 \notin \mathbb{N}$)
- Ganze Zahlen: $\mathbb{Z} = \{0, 1, -1, 2, -2, 3, -3, 4, -4, \dots\}$
- Rationale Zahlen: $\mathbb{Q} = \left\{ \frac{p}{q} : p, q \in \mathbb{Z}, q \neq 0 \right\}$
- Reelle Zahlen: \mathbb{R} , Menge aller **reellen Zahlen**, die man **nicht abzählen** kann

Es gilt: $\mathbb{N} \in \mathbb{Z} \in \mathbb{Q} \in \mathbb{R}$

Relationen

Eine (**binäre**) **Relation** zwischen zwei Mengen X und Y ist eine Teilmenge

$$R \in X \times Y$$

Im Falle $X = Y$ sprechen wir von einer Relation auf X .

$x \in X$ steht in Relation zu $y \in Y$ genau dann wenn $(x, y) \in R$.

Auch geschrieben: $x R y$ oder $x \sim_R y$ für $(x, y) \in R$ und $x \not R y$ oder $x \not\sim_R y$ für $(x, y) \notin R$.

Seien X, Y und Z Mengen und $R \subset X \times Y, S \subset Y \times X$ Relationen.

- Die zu R **inverse Relation** ist

$$R^{-1} = \{(y, x) \in Y \times X : (x, y) \in R\}$$

- Die Verkettung von R und S ist

$$S \circ R = \{(x, z) \in X \times Z : \text{es gibt } y \in Y \text{ mit } (x, y) \in R \text{ und } (y, z) \in S\}$$

Eine binäre Relation R auf der Menge X heißt:

- **reflexiv**, wenn $x R x$ für alle $x \in X$.
- **symmetrisch**, wenn für alle $x, y \in X$ aus $x R y$ stets $y R x$ folgt.
- **antisymmetrisch**, wenn für alle $x, y \in X$ aus $x R y$ und $y R x$ stets $x = y$ folgt.
- **asymmetrisch**, wenn für alle $x, y \in X$ aus $x R y$ stets $y \not R x$ folgt.
- **transitiv**, wenn für alle $x, y, z \in X$ aus $x R y$ und $y R z$ stets $x R z$ folgt.

Äquivalenzrelationen

Sei X eine nicht leere Menge. Eine Relation R auf X die reflexiv, symmetrisch und transitiv ist, heißt **Äquivalenzrelationen**. Für $x \in X$ nennt man die Menge

$$[x] \sim_R = \{y \in X : x R y\}$$

die **Äquivalenzklasse** von x . Man nennt x und jedes andere Element aus $[x] \sim_R$ einen **Vertreter** oder **Repräsentanten** dieser Äquivalenzklasse.

Abbildungen

Eine **Abbildung** $f : X \rightarrow Y$ besteht aus:

- einer Menge X , der **Definitionsbereich** von f ;
- einer Menge Y , der **Wertebereich** von f ;
- einer **Vorschrift**, die jedem $x \in X$ eindeutig ein $y \in Y$ zuordnet.

Notation: $f : X \rightarrow Y, x \mapsto f(x)$

Seien X, Y Mengen, $f : X \rightarrow Y$ eine Abbildung und $x \in X, y \in Y$ sodass $f(x) = y$.

Dann ist y das **Bild** von x und x ein **Urbild** von y .

Für eine Teilmenge $X_0 \subset X$ ist

$$f(X_0) := \{y \in Y : \text{es gibt } x \in X_0, \text{ sodass } f(x) = y\} \subset Y$$

das **Bild** von X_0 und für eine Teilmenge $Y_0 \subset Y$ ist

$$f^{-1}(Y_0) := \{x \in X : f(x) \in Y_0\} \subset X$$

das **Urbild** von Y_0 .

Seien X und Y Mengen und $f : X \rightarrow Y$ eine Abbildung.

f ist **injektiv** falls aus $x_1, x_2 \in X$ mit $f(x_1) = f(x_2)$ stets $x_1 = x_2$ folgt.

f ist **surjektiv** falls es für jedes $y \in Y$, ein $x \in X$ existiert so dass $f(x) = y$.

f ist **bijektiv** falls f injektiv und surjektiv ist.

Seien X, Y, Z Mengen und $f : X \rightarrow Y$ und $g : Y \rightarrow Z$ Abbildungen.

Die **Komposition** oder **Verknüpfung** von f und g ist die

Abbildung $g \circ f : X \rightarrow Z$, definiert durch $(g \circ f)(x) = g(f(x))$.

Zahlensysteme

Binärsystem

Eine Binärzahl b mit $n + 1$ Stellen hat die Form $b_n \dots b_1 b_0$ mit $b_i \in \{0, 1\}$.

Sie entspricht der Dezimalzahl d mit $d = b_n \cdot 2^n + \dots + b_1 \cdot 2^1 + b_0 \cdot 2^0$

Beispiel: $1101_2 = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 13_{10}$

Carry-Flag

Wenn bei einer Addition oder Subtraktion ein **Übertrag in der höchsten Stelle** auftritt, wird die Carry-Flag gesetzt. Dieser kann von nachfolgenden Befehlen aufgerufen werden.

Zweierkomplement

Um **negative Zahlen** darzustellen wird der entsprechende Wert des **höchsten Bits negiert**.

Beispiel bei 4 Bit: $1011_{2c} = 1 \cdot (-2^3) + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = -5$

Um von einer positiven ganzen Zahl zur negativen Zahl (oder umgekehrt) gleichen Betrags zu gelangen werden **alle Bits invertiert** und **1 zum Ergebnis addiert**.

Hexadezimalsystem

Eine Hexadezimalzahl h mit $n + 1$ Stellen hat die Form $h_n \dots h_1 h_0$ mit $h_i \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A(\hat{=10}), B(\hat{=11}), C(\hat{=12}), D(\hat{=13}), E(\hat{=14}), F(\hat{=15})\}$.

Sie entspricht der Dezimalzahl d mit $d = h_n \cdot 16^n + h_1 \cdot 16^1 + h_0 \cdot 16^0$.

Beispiel: $5F_{16} = 5 \cdot 16^1 + 15 \cdot 16^0 = 95_{10}$

4 Binärziffern lassen sich zu einer Hexadezimalzahl zusammenfassen:

$$\underbrace{1101}_{13_{10}=D_{16}} \underbrace{0011_2}_{3_{16}} = D3_{16}$$

Oktalsystem

Eine Oktalzahl o mit $n + 1$ Stellen hat die Form $o_n \dots o_1 o_0$ mit $o_i \in \{0, 1, 2, 3, 4, 5, 6, 7\}$

Sie entspricht der Dezimalzahl d mit $d = o_n \cdot 8^n + \dots + o_1 \cdot 8^1 + o_0 \cdot 8^0$.

Beispiel: $36_8 = 3 \cdot 8^1 + 6 \cdot 8^0 = 30_{10}$

3 Binärziffern lassen sich zu einer Oktalzahl zusammenfassen:

$$\underbrace{11}_{3_8} \underbrace{010}_{2_8} \underbrace{011}_{{}_2} = 323_8$$

Festkommazahlen

Eine Festkommazahl besteht aus einer **festen Anzahl von Ziffern vor und nach dem Komma**.

2^3	2^2	2^1	2^0		2^{-1}	2^{-2}	2^{-3}	2^{-4}
1	1	0	1	.	0	1	0	1

Gleitkommazahlen: IEEE 754

3 Formate:

- Single Precision: 32 Bit
- Double Precision: 64 Bit
- Extended Precision: 80 Bit

Basiert auf der wissenschaftlichen Notation.

Aufbau

Single Precision:	1 Bit Vorzeichen	8 Bit Exponent	23 Bit normalisierte Mantisse
Double Precision:	1 Bit Vorzeichen	11 Bit Exponent	52 Bit normalisierte Mantisse

Vorzeichen: 0 = +; 1 = −

Exponent: wird gespeichert, indem man den festen Biaswert (127:SP, 1023:DP) addiert.

Die Mantisse beginnt mit einem "Hidden Bit" (immer 1).

Dezimal zu IEEE 754

Beispiel: -62.058

1. Vorzeichen Bit bestimmen

Vorzeichen Bit = 1

2. Zu pur Binär umwandeln

$$62.058_{10} = 111110.10010100_2$$

3. Normalisieren für Mantisse und Exponent (ohne Bias)

$$111110.10010100_2 = 1.1111010010100_2 \cdot 2^5$$

4. Exponent mit Bias bestimmen

$$5 + 127 = 132_{10} = 10000100_2$$

5. Führende 1 der Mantisse abschneiden

$$1.1111010010100_2 \rightarrow 1111010010100_2$$

6. Zusammenfügen

$$-62.058_{10} = \underbrace{1}_{\substack{\text{Vorzeichen} \\ \text{Bit}}} \underbrace{10000100}_{\text{Exponent}} \underbrace{1111010010100}_{\text{Mantisse}}$$

IEEE 754 zu Dezimal

Beispiel: 01000010011010100000000000000000

1. Vorzeichen bestimmen

Vorzeichen: +

2. Exponent bestimmen (Bias muss abgezogen werden)

$$10000100_2 - 127_{10} = 132_{10} - 127_{10} = 5_{10}$$

3. Mantisse bestimmen

$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{32}$	$\frac{1}{64}$
1	1	0	1	0	1

$$\frac{1}{2} + \frac{1}{4} + \frac{1}{16} + \frac{1}{64} = \frac{53}{64} = 0.828125$$

4. 1 zur Mantisse addieren (Hidden Bit) und Vorzeichen einrechnen

$$1.828125$$

5. Ergebnis berechnen

$$1.828125 \cdot 2^5 = 58.5_{10}$$

Fehlererkennung

Redundanzen

Eine Einheit von n Datenbits und k Redundanzbits nennt man **Codewort**.

Die **Länge** eines Codeworts ist insgesamt $n + k$.

Die Menge aller gültigen Codewörter nennt man **Code**.

Hamming-Distanz

Die Hamming Distanz zweier Codewörter ist gegeben als die Anzahl der Bitpositionen, in denen sie sich unterscheiden.

Beispiel: 11110000 und 11001100 \implies Hamming-Distanz beträgt 4

Die Hamming Distanz eines Codes ist die kleinste Hamming-Distanz zweier Codewörter

Beispiel: $\{1100, 0011, 1111\} \implies$ Hamming-Distanz beträgt 2

c -Bit Fehler können erkannt werden, wenn die Hamming-Distanz $c+1$ beträgt.
 c -Bit Fehler können korrigiert werden, wenn die Hamming-Distanz $2c + 1$ beträgt.

Parität

Durch Hinzufügen eines **Paritätsbits** wird ein Code mit Hamming-Distanz 2 erzeugt.

gerade Parität: Gesamtzahl der 1en ist gerade (inklusive Paritätsbit)

$$\underbrace{00100101}_{\text{Datenbits}} \quad \underbrace{1}_{\text{Paritätsbit}}$$

ungerade Parität: Gesamtzahl der 1en ist ungerade (inklusive Paritätsbit)

$\underbrace{00100101}_{\text{Datenbits}} \underbrace{0}_{\text{Paritätsbit}}$

Zweidimensionale Parität

Die zweidimensionale Parität konstruiert einen Code mit Hamming-Distanz 4.

Dabei werden n Wörter zu je n Bits in einer $n \times n$ -Matrix untereinander geschrieben und über jede Zeile und jede Spalte je ein Paritätsbit berechnet.

Bei einem 1-Bit-Fehler stimmen die Paritätsbits genau einer Zeile und Spalte nicht.

Dann ist die Position des Fehlers klar und er kann korrigiert werden.

fehlerfrei:	0	1	1	0	0	1-Bit-Fehler:	0	1	1	0	0
	1	0	0	0	1		1	0	1	0	1
	0	1	0	0	1		0	1	0	0	1
	0	1	1	1	1		0	1	1	1	1
	1	1	0	1	1		1	1	0	1	1

Hamming-Code

Ein Hamming-Code mit n **Redundanzbits** hat maximal $2^n - 1$ Bits und maximal $2^n - 1 - n$ Datenbits (mit $n \in \mathbb{N}$)

Die Bits des Codewortes werden, beginnend bei 1, durchnummeriert.

Das i -te **Prüfbit** (auch Redundanzbit) steht im Codewort an Position 2^i ($\Rightarrow 1, 2, 4, 8, \dots$)

	Position	Bits des Codewortes		
Beispiel:	1_{10}	0	Prüfbit	Gespeichertes Datenwort: 0101
	2_{10}	1	Prüfbit	
	3_{10}	0		
	4_{10}	0	Prüfbit	
	5_{10}	1		
	6_{10}	0		
	7_{10}	1		

Jedes Prüfbit ist ein Paritätsbit über eine eindeutige Menge von Bits.

Summenzeichen und Produktzeichen

Summenzeichen

Seien $m, n \in \mathbb{Z}$ mit $m \leq n$. Die Summen der Zahlen a_m, a_{m+1}, \dots, a_n wird folgendermaßen bezeichnet:

$$\sum_{i=m}^n a_i = a_m + a_{m+1} + \dots + a_n$$

Dabei gilt: $i \hat{=}$ **Summationsindex**; $m/n \hat{=}$ **untere/obere Summationsgrenze**.
Rechenregeln:

$$\begin{aligned} \sum_{i=m}^n c \cdot a_i &= c \cdot \sum_{i=m}^n a_i \\ \sum_{i=m}^n (a_i + b_i) &= \sum_{i=m}^n a_i + \sum_{i=m}^n b_i \end{aligned}$$

Produktzeichen

Seien $m, n \in \mathbb{Z}$ mit $m \leq n$. Das Produkt der Zahlen a_m, a_{m+1}, \dots, a_n wird folgendermaßen bezeichnet:

$$\prod_{i=m}^n a_i = a_m \cdot a_{m+1} \cdot \dots \cdot a_n$$

Dabei gilt: $i \hat{=}$ **Laufindex**; $m/n \hat{=}$ **untere/obere Grenze**.

Rechenregeln

Bruchregeln

$$\begin{aligned}\frac{a}{b} &= \frac{a \cdot c}{b \cdot c} & \frac{a}{b} + \frac{c}{b} &= \frac{a+c}{b} \\ \frac{a}{b} \cdot \frac{c}{d} &= \frac{ac}{bd} & \frac{a}{b} : \frac{c}{d} &= \frac{a}{b} \cdot \frac{d}{c} = \frac{ad}{bc}\end{aligned}$$

Potenzgesetze

$$\begin{aligned}a^n \cdot a^m &= a^{n+m} & a^n \cdot b^n &= (a \cdot b)^n \\ (a^n)^m &= (a^m)^n = a^{n \cdot m} & a^{-n} &= \frac{1}{a^n}, a \neq 0 \\ a^0 &= 1, a \in \mathbb{R}\end{aligned}$$

Wurzelgesetze

$$\begin{aligned}\sqrt[n]{a^n} &= a & (\sqrt[n]{a})^n &= a \\ \sqrt[n]{a \cdot b} &= \sqrt[n]{a} \cdot \sqrt[n]{b} & \sqrt[n]{\frac{a}{b}} &= \frac{\sqrt[n]{a}}{\sqrt[n]{b}}, b \neq 0 \\ a^{\frac{1}{n}} &= \sqrt[n]{a} & a^{-\frac{1}{n}} &= \frac{1}{\sqrt[n]{a}}, a > 0\end{aligned}$$

Logarithmengesetze

$$\begin{aligned}\log 1 &= 0 & \log e &= 1 \\ a^x = b &\Leftrightarrow x = \log_a(b) & \log(a^x) &= x \log a \\ \log(x \cdot y) &= \log x + \log y & \log\left(\frac{x}{y}\right) &= \log x - \log y\end{aligned}$$

Trigonometrie

Bogenmaß

Der Bogenmaß ist die Länge des Kreisbogens des Einheitskreises und gibt den Betrag des Winkels an. Der Umfang des Einheitskreises beträgt 2π .

Bogenmaß	0	$\frac{\pi}{6}$	$\frac{\pi}{4}$	$\frac{\pi}{3}$	$\frac{\pi}{2}$	π	$\frac{3\pi}{2}$	2π
Gradmaß	0°	30°	45°	60°	90°	180°	270°	360°

Umwandlung von Winkel α von Gradmaß zu Bogenmaß: Bogenmaß = $\alpha \frac{\pi}{180^\circ}$

Umwandlung von Winkel α von Bogenmaß zu Gradmaß: Gradmaß = $\alpha \frac{180^\circ}{\pi}$

Differentialrechnung

Ableitungsregeln

Natürliche Potenzen

$$\frac{d}{dx}x^n = nx^{n-1}$$

Summenregel

$$\frac{d}{dx}(u(x) + v(x)) = \frac{d}{dx}u(x) + \frac{d}{dx}v(x)$$

Produktregel

$$\frac{d}{dx}(u(x) \cdot v(x)) = \frac{d}{dx}u(x) \cdot v(x) + u(x) \cdot \frac{d}{dx}v(x)$$

Kettenregel

$$\frac{d}{dx}u(v(x)) = \frac{d}{dv}u(v) \cdot \frac{d}{dx}v(x)$$

Ebenen

Parameterform

$$E : \vec{x} = \vec{p} + r \cdot \vec{u} + s \cdot \vec{v}$$

$$\begin{aligned}\vec{p} &\hat{=} \text{Stützvektor} \\ \vec{u}, \vec{v} &\hat{=} \text{Spannvektoren}\end{aligned}$$

Normalenform

$$E : (\vec{x} - \vec{p}) \cdot \vec{n} = 0$$

$$\begin{aligned}\vec{x} &\hat{=} \text{Ortsvektor} \\ \vec{p} &\hat{=} \text{Stützvektor} \\ \vec{n} &\hat{=} \text{Normalenvektor, orthogonal zu Spannvektoren}\end{aligned}$$

Koordinatenform

$$ax_1 + bx_2 + cx_3 = d = \vec{n} \cdot \vec{p}$$

Umwandlung von Koordinatenform in Normalenform: