

# Algorithmen und Datenstrukturen (für ET/IT)

Wintersemester 2012/13

Dr. Tobias Lasser

Computer Aided Medical Procedures  
Technische Universität München



# Programm heute

## ① Einführung

Einordnung in das Computer-Schema

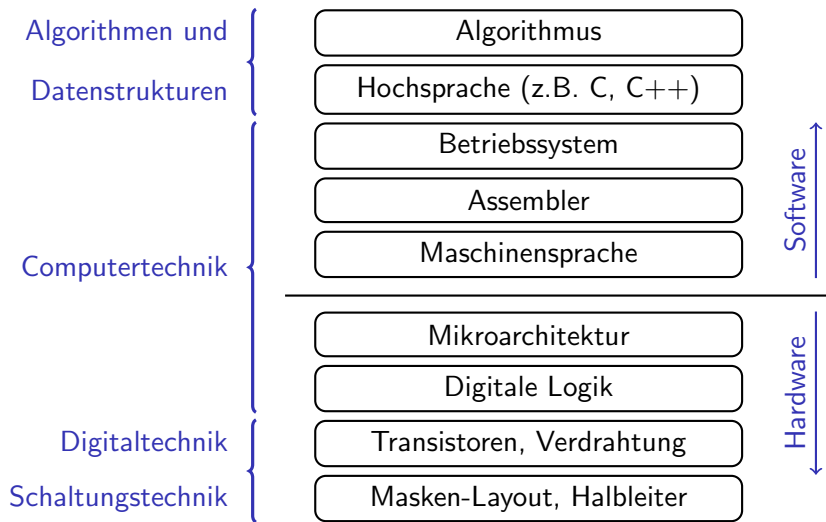
## ② Mathematische Grundlagen

Mengen

Abbildungen

Zahldarstellung

# Wie funktioniert ein Computer?



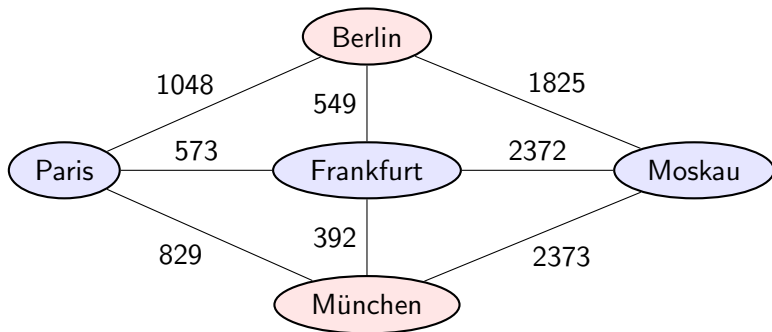
# Einordnung Algorithmen und Datenstrukturen

Beispiel-Problem Navigationssystem Auto:

Finde kürzesten Weg von Berlin nach München.



- **Datenstruktur:** gewichteter Graph ( $\rightarrow$  Kapitel 7)



- **Algorithmus:** kürzester Pfad ( $\rightarrow$  Kapitel 9)

# Programm heute

## ① Einführung

Einordnung in das Computer-Schema

## ② Mathematische Grundlagen

Mengen

Abbildungen

Zahldarstellung

# Definition Menge

## Menge nach Georg Cantor (1883)

Eine Menge ist eine Zusammenfassung  $M$  von bestimmten, wohlunterschiedenen Objekten  $m$  unserer Anschauung oder unseres Denkens zu einem Ganzen.

- die  $m$  heißen Elemente der Menge  $M$
- in Zeichen:  $m \in M$

# Mengen Beispiele

- $\emptyset = \{\}$  die leere Menge
- $\mathbb{N} = \{1, 2, 3, \dots\}$  die natürlichen Zahlen
- $\mathbb{N}_0 = \{0, 1, 2, 3, \dots\}$
- $\mathbb{Z} = \{0, 1, -1, 2, -2, \dots\}$  die ganzen Zahlen
- $\mathbb{Q} = \left\{ \frac{m}{n} : m \in \mathbb{Z}, n \in \mathbb{N} \right\}$  die rationalen Zahlen
- $\mathbb{R} = \left\{ \lim_{n \rightarrow \infty} a_n : (a_n)_{n \geq 1} \subset \mathbb{Q} \text{ konvergent} \right\}$   
die reellen Zahlen
- $M = \{\emptyset, \mathbb{N}, \mathbb{N}_0, \mathbb{Z}, \mathbb{Q}, \mathbb{R}\}$   
Mengen können auch Elemente von Mengen sein

# Russellsche Antinomie

Die Eigenschaft  $E(x)$  bedeute daß  $x \notin x$ .

Definiere

$$M := \{x : E(x)\} = \{x : x \notin x\}.$$

**Frage:** Ist  $M \in M$  oder  $M \notin M$ ?

Angenommen  $M \notin M$ . Dann hat  $M$  Eigenschaft  $E$ . Also  $M \in M$ .  
Widerspruch!

Angenommen  $M \in M$ . Dann hat  $M$  nicht Eigenschaft  $E$ . Also  
 $M \notin M$ . Widerspruch!

**Lösung:** moderne Mengenlehre nach Zermelo-Fraenkel mit  
Aussonderungssaxiom,  $M$  ist keine Menge.

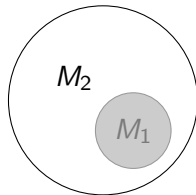


# Teilmengen

## Definition Teilmenge

Seien  $M_1, M_2$  Mengen.

- $M_1$  ist eine **Teilmenge** von  $M_2$  (in Zeichen:  $M_1 \subset M_2$ )
- genau dann, wenn
- aus  $m \in M_1$  folgt  $m \in M_2$



$M_1$  und  $M_2$  sind gleich ( $M_1 = M_2$ ), genau dann wenn  $M_1 \subset M_2$  und  $M_2 \subset M_1$ .

# Kardinalität

## Kardinalität

Die Anzahl der Elemente einer Menge  $M$  heißt **Kardinalität** und wird mit  $|M|$  bezeichnet.

Beispiele:

- $|\{0, 8, 15\}| = 3$
- $|\emptyset| = 0$
- $|\mathbb{N}| = \infty$ , d.h.  $\mathbb{N}$  hat unendlich viele Elemente

# Operationen auf Mengen

Seien  $M, N$  Mengen.

- Vereinigung:  $M \cup N := \{x : x \in M \text{ oder } x \in N\}$
  
- Durchschnitt:  $M \cap N := \{x : x \in M \text{ und } x \in N\}$
  
- Differenz:  $M \setminus N := \{x : x \in M \text{ und } x \notin N\}$

# Operationen auf Mengen: Rechenregeln

Seien  $A, B, C$  Mengen.

$$A \cup B = B \cup A$$

kommutativ

$$A \cap B = B \cap A$$

$$A \cup (B \cup C) = (A \cup B) \cup C$$

assoziativ

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

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

distributiv

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

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

de Morgan Gesetz

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

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

# Potenzmenge

## Potenzmenge

Sei  $M$  Menge. Die Menge

$$\mathcal{P}(M) := \{N : N \subset M\}$$

heißt **Potenzmenge** von  $M$ .

**Eigenschaft:** Ist  $|M| = n$ , so ist  $|\mathcal{P}(M)| = 2^n$ .

**Beispiel:**

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

# Kartesisches Produkt von Mengen

Statt Mengen  $\{x, y\}$  kann man auch **geordnete Paare**  $(x, y)$  betrachten.

## Kartesisches Produkt

Seien  $X, Y$  Mengen. Die Menge

$$X \times Y := \{(x, y) : x \in X, y \in Y\}$$

heißt **kartesisches Produkt** von  $X$  und  $Y$ .

**Eigenschaft:**  $|X \times Y| = |X| \cdot |Y|$ .

**Beispiele:**

- $\{a, b, c\} \times \{1, 2\} = \{(a, 1), (a, 2), (b, 1), (b, 2), (c, 1), (c, 2)\}$
- $\mathbb{R} \times \mathbb{R} = \mathbb{R}^2 = \{(x, y) : x, y \in \mathbb{R}\}$  zwei-dimensionaler Raum
- $x$ -Achse =  $\{(x, 0) : x \in \mathbb{R}\}$ ,  $y$ -Achse =  $\{(0, y) : y \in \mathbb{R}\}$

# Programm heute

## ① Einführung

Einordnung in das Computer-Schema

## ② Mathematische Grundlagen

Mengen

**Abbildungen**

Zahldarstellung

# Definition Abbildung

## Abbildung / Funktion

Seien  $X, Y$  Mengen. Eine Vorschrift, die jedem  $x \in X$  genau ein  $y \in Y$  zuordnet, heißt **Abbildung** oder **Funktion**.

Das dem  $x \in X$  zugeordnete  $y \in Y$  wird bezeichnet mit

$$y = f(x)$$

und heißt *das Bild* von  $x$  unter  $f$ .  $x$  heißt *ein Urbild* von  $y$  unter  $f$ .

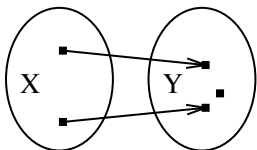
$X$  heißt **Definitionsbereich** von  $f$ ,  $Y$  heißt **Bildbereich** von  $f$ .

**Kurzschreibweise:**  $f : X \rightarrow Y$  oder  $f : x \mapsto y$ .

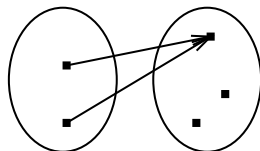
*Achtung:* Es muss nicht jedes  $y \in Y$  als Bild eines  $x \in X$  auftreten.



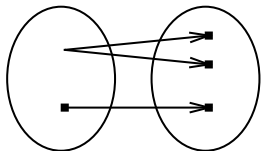
# Abbildung Beispiele



Abbildung

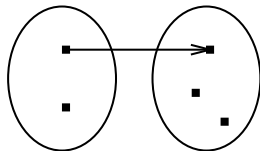


Abbildung



keine Abbildung

(Zwei  $y$  haben dasselbe Urbild)



keine Abbildung

(Ein  $x$  hat kein Bild)

# Abbildung Beispiele

Beispiel-Problem Navigationssystem Auto:



$X$  = Menge der Verbindungen zwischen den Städten  
(z.B. München-Frankfurt)

$Y = \mathbb{R}$  reelle Zahlen

$f : X \rightarrow Y$  weist jeder Verbindung die Distanz in Kilometer zu

z.B.  $f(\text{München-Frankfurt}) = 392$

$f(\text{Frankfurt-Berlin}) = 549$

# Abbildungseigenschaften

## Injektiv, surjektiv, bijektiv

Sei  $f : X \rightarrow Y$  Abbildung.

- $f$  heißt **injektiv** genau dann, wenn zu jedem  $y \in Y$  **höchstens ein**  $x \in X$  existiert mit  $f(x) = y$ .
- $f$  heißt **surjektiv** genau dann, wenn zu jedem  $y \in Y$  **mindestens ein**  $x \in X$  existiert mit  $f(x) = y$ .
- $f$  heißt **bijektiv** genau dann, wenn zu jedem  $y \in Y$  **genau ein**  $x \in X$  existiert mit  $f(x) = y$ .

**Eigenschaft:**  $f$  ist bijektiv genau dann, wenn es injektiv und surjektiv ist.

**Beispiel:**  $f : \mathbb{N} \rightarrow \mathbb{N}$ ,  $f(n) = n + 1$  ist injektiv, aber nicht surjektiv (da  $1 \notin f(\mathbb{N})$ ) und damit nicht bijektiv.

# Folgen

## Folge

Sei  $M$  Menge. Eine **Folge** ist eine Abbildung  $a : \mathbb{N} \rightarrow M$  mit

$$i \mapsto a_i \quad \text{bzw.} \quad a(i) = a_i.$$

Eine **endliche Folge** ist eine Abbildung  $a : \{1, \dots, n\} \rightarrow M$  für  $n \in \mathbb{N}$ .  $n$  ist die **Länge** der Folge.

**Kurzschreibweise:**  $(a_i)_{i \in \mathbb{N}}$  bzw.  $(a_i)_{i=1, \dots, n}$

**Beispiele:**

- Folge der Primzahlen:  $(2, 3, 5, 7, \dots)$
- Folge der ersten acht Zweierpotenzen:  
 $(2, 4, 8, 16, 32, 64, 128, 256)$

# Polynome

## Polynome

Sei  $n \in \mathbb{N}$ . Eine Funktion  $p : \mathbb{R} \rightarrow \mathbb{R}$  der Form

$$p(x) = c_n x^n + c_{n-1} x^{n-1} + \dots + c_1 x^1 + c_0$$

heißt **Polynom** in  $x$  vom **Grad**  $n$ . Hierbei heißen die  $c_i \in \mathbb{R}$ ,  $i = 1, \dots, n$ , **Koeffizienten** von  $p$ ,  $c_n \neq 0$ .

**Beispiel:** Dezimaldarstellung von Zahlen

$$123 = 1 \cdot 10^2 + 2 \cdot 10^1 + 3 \cdot 10^0$$

# Programm heute

## ① Einführung

Einordnung in das Computer-Schema

## ② Mathematische Grundlagen

Mengen

Abbildungen

Zahldarstellung

# Zahldarstellung

- Dezimalsystem:

- Basis  $x = 10$
- Koeffizienten  $c_n \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
- Beispiel:  $123_{10} = 1 \cdot 10^2 + 2 \cdot 10^1 + 3 \cdot 10^0$

- Binärsystem:

- Basis  $x = 2$
- Koeffizienten  $c_n \in \{0, 1\}$
- Beispiel:  $1101_2 = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 13_{10}$

# Zahldarstellung

- Oktalsystem:

- Basis  $x = 8 (= 2^3)$
- Koeffizienten  $c_n \in \{0, 1, 2, 3, 4, 5, 6, 7\}$
- Beispiel:  $173_8 = 1 \cdot 8^2 + 7 \cdot 8^1 + 3 \cdot 8^0 = 123_{10}$

- Hexadezimalsystem:

- Basis  $x = 16 (= 2^4)$
- Koeffizienten  $c_n \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$
- Beispiel:  $7B_{16} = 7 \cdot 16^1 + B \cdot 16^0 = 123_{10}$



# Wie viele Ziffern pro Zahl?

## Problem

Gegeben Zahl  $z \in \mathbb{N}$ , wie viele Ziffern  $m$  werden bezüglich Basis  $x$  benötigt?

## Lösung

$$m = \lfloor \log_x(z) \rfloor + 1$$

Erläuterung: ( $a \in \mathbb{R}$ )

- $\lfloor a \rfloor = \text{floor}(a) =$  größte ganze Zahl kleiner gleich  $a$
- $\lceil a \rceil = \text{ceil}(a) =$  kleinste ganze Zahl größer gleich  $a$

$$a - 1 < \lfloor a \rfloor \leq a \leq \lceil a \rceil < a + 1$$

- $\log_x(z) = \frac{\ln(z)}{\ln(x)}$ , wobei „ln“ der natürliche Logarithmus ist

# Wie viele Ziffern pro Zahl?

## Lösung

$$m = \lfloor \log_x(z) \rfloor + 1$$

Beispiele:  $z = 123$

- Basis  $x = 10$ :

$$m = \lfloor \log_{10}(123) \rfloor + 1 = \lfloor 2.0899 \dots \rfloor + 1 = 3$$

- Basis  $x = 2$ :

$$m = \lfloor \log_2(123) \rfloor + 1 = \lfloor 6.9425 \dots \rfloor + 1 = 7$$

- Basis  $x = 8$ :

$$m = \lfloor \log_8(123) \rfloor + 1 = \lfloor 2.3141 \dots \rfloor + 1 = 3$$

- Basis  $x = 16$ :

$$m = \lfloor \log_{16}(123) \rfloor + 1 = \lfloor 1.7356 \dots \rfloor + 1 = 2$$

# Größte Zahl pro Anzahl Ziffern?

## Problem

Gegeben Basis  $x$  und  $m$  Ziffern, was ist die größte darstellbare Zahl?

## Lösung

$$z_{\max} = x^m - 1$$

## Beispiele:

- $x = 2, m = 4$ :

$$z_{\max} = 2^4 - 1 = 15 = 1111_2$$

- $x = 2, m = 8$ :

$$z_{\max} = 2^8 - 1 = 255 = 11111111_2$$

- $x = 16, m = 2$ :

$$z_{\max} = 16^2 - 1 = 255 = FF_{16}$$

# Primitive Datentypen

## Primitive Datentypen

In Computern verwendet man **Binärdarstellung** mit einer fixen Anzahl Ziffern (genannt **Bits**).

Die **primitiven Datentypen** sind

- **8 Bits** (auch genannt **Byte**), darstellbare Zahlen:  $\{0, \dots, 255\}$   
in C, C++: `unsigned char`
- **16 Bits**, darstellbare Zahlen:  $\{0, \dots, 65535\}$   
in C, C++: `unsigned short int` oder nur `unsigned short`
- **32 Bits**, darstellbare Zahlen:  $\{0, \dots, 4294967295\}$   
in C, C++: `long int` oder nur `unsigned long`, 32 Bit System: `unsigned int`
- **64 Bits**, darstellbare Zahlen:  $\{0, \dots, 2^{64} - 1\}$   
in C, C++: `unsigned long long`, 64 Bit System: `unsigned int`

1001110010001  
0101001001000100010001  
001000101010100100100010001  
1110010001010101001001000100011  
10010001 010101 00100100  
01001110 00101 00010011  
001001110 10011 011100101  
10010001010 0100110 00111010111  
010011100001001110000100111011101100110  
110100 1001110010100011010001110 101001  
11010 01001110010001110101100 01011  
10011 000100111010010100111 10100  
010100 01001110100101100 001001  
10101110 010010100 10011101  
100101010 010011101  
001001110100110110010  
0010011101011

# Zusammenfassung

## ① Einführung

Einordnung in das Computer-Schema

## ② Mathematische Grundlagen

Mengen

Abbildungen

Zahldarstellung