Einführung in die Informatik

Zahlen und Logik

Meik Teßmer

Inhalte

Lernziele

- Zahlen
- Logik (Boolesche Algebra)

Exkurs: Was ist *Logik*?

- Was ist darunter zu verstehen?
- Wozu braucht man das?

Exkurs: Was ist *Logik*?

- altgriechisch für „denkende Kunst“, „Folgerichtigkeit“
- beschäftigt sich mit dem *vernünftigen Schlussfolgern*
- Bestandteile: Argumente und ihre Gültigkeit, Wahrheitswerte
- es gibt mehrere Logiken:

  klassische/formale Logiken, z.B.

  - Aussagenlogik
  - Prädikatenlogik

  nichtklassische Logiken, bspw.

  - philosophische Logiken (Modallogik, Temporale Logik)
  - intuitionistische Logik
  - Relevanzlogik
  - Mehrwertige Logik

Rechenmaschinen

Funktionsweise einer Rechenmaschine

Manuelles Rechnen

Wie kann man prinzipiell rechnen?

Funktionsweise einer Rechenmaschine

Automatisiertes Rechnen
Wie könnte man das auf ein Gerät übertragen?

Erste Geräte

- 1623: Rechenmaschine von Wilhelm Schickard
- 1645: Blaise Pascal: Pascaline
- 1673: Leibniz: Staffelwalzen-Maschine

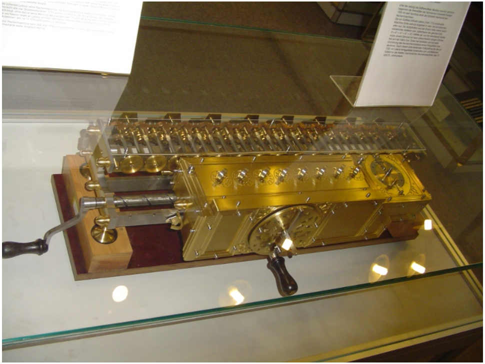

Abbildung 1: Staffelwalzenmaschine.

Der Weg zur Moderne

- 1820-1822: Charles Babbage: Erste Gedanken zur Differenzmaschine
- ab 1822: Entwurf der *Analytical Machine*
- 1837: erste Veröffentlichtung, wurde aber nie gebaut → aus heutiger Sicht wäre sie funktionsfähig gewesen!

Abbildung 2: Differenzmaschine No. 2

Die Analytische Maschine

- Eigenschaften:
  - Antrieb: Dampfmaschine
  - Größe: über 30 Meter lang und 10 Meter breit
  - Eingabe über Lochkarten
  - Ausgabe: Drucker, Kurvenplotter, Glocke
  - Speicher für 1000 *Wörter* mit je 50 Dezimalstellen
  - 4 Grundrechenarten
- Programmierung: Assembler-ähnliche Sprache mit Schleifen und Verzweigungen
  - 3 Arten von Lochkarten: arithmetische Operationen, numerische Konstanten, Lade-/Speicheroperationen

→ damit wäre sie das erste universell programmierbare System gewesen

Probleme mechanischer Rechenmaschinen

- begrenzter Zahlenraum
- Mechanik:
  - Schwergängigkeit bei größeren Zahlen
  - Geschwindigkeit beschränkt durch die Belastbarkeit der Bauteile
- Beschränkung der Rechenoperationen
- Fehleranfälligkeit durch Verschleiß, unpräzise Fertigung

$\rightarrow$ Lösungsvorschläge?

Moderne Rechenmaschinen

Grundlage moderner Rechenmaschinen

- Grundprinzip: Dualsystem (auch Binärsystem)
- Zahldarstellung durch 0 und 1
- Stellenwertsystem zur Basis 2

  $\rightarrow$ 4 Stellen $= 2^4 =$ 16 Zahlen (0 bis 15)
- Umsetzung:
  - Strom (Relais oder Transistor)
  - Licht (Fototransistor)

Umrechnen ins Dualsystem

- Modulo-Methode: durch Basis (2) dividieren, Reste sind die gesuchte binäre Zahl

$$\begin{array}{rcccl}
41 & :2 & = & 20 & \text{Rest } \mathbf{1} \\
20 & :2 & = & 10 & \text{Rest } \mathbf{0} \\
10 & :2 & = & 5 & \text{Rest } \mathbf{0} \\
5 & :2 & = & 2 & \text{Rest } \mathbf{1} \\
2 & :2 & = & 1 & \text{Rest } \mathbf{0} \\
1 & :2 & = & 0 & \text{Rest } \mathbf{1}
\end{array}$$

Abbildung 3: Modulo-Rechnung

$\rightarrow 41_{(10)} = 101001_{(2)}$

Umrechnen ins Dezimalsystem

- Stellenwertsystem: Basis$^{\text{Stelle}}$ * Wert an dieser Stelle
- Beispiel:

  $1010_{(2)} = 1 * 2^3 + 0 * 2^2 + 1 * 2^1 + 0 * 2^0$

  $= 8 + 2 = 10_{(10)}$
- tabellarischer Ansatz:

	Stellenwert							
	32	16	8	4	2	1		
Dualzahl	0	0	0	1	0	1	5	Dezimalzahl
	1	0	0	0	1	1	35	
	0	0	1	0	1	0	10	

Abbildung 4: Umwandlung mit Tabelle

Rechnen im Dualsystem

Addition

```
0 + 0 =  0
0 + 1 =  1
1 + 0 =  1
1 + 1 = 10
```

Beispiel:

$$\begin{array}{rr} & 1011_2 \\ + & 11_2 \\ \hline & 1110_2 \end{array}$$

Abbildung 5: Addition

Rechnen im Dualsystem

Subtraktion

```
0 - 0 =  0
0 - 1 = -1
1 - 0 =  1
1 - 1 =  0
```

Beispiel:

$$\begin{array}{lr} & 1011_2 \\ - & 111_2 \\ \hline & 100_2 \end{array}$$

Abbildung 6: Subtraktion

Subtraktion: Übertrag

```
      ↓                                  ↓
 1101110                           1101110      1-1=0, -1 vom
-  10111                          -  10111      Übertrag → 1 und
       1    Übertrag                     1      weiterer Übertrag
--------                          --------
       1                                11
```

```
 1101110
-  10111    Gesamtrechnung
--------
 1010111
```

Rechnen im Dualsystem

Multiplikation

```
0 * 0 = 0
0 * 1 = 0
1 * 0 = 0
1 * 1 = 1
```

$$\begin{array}{r} 1010_2 \\ \cdot \quad 11_2 \\ \hline 11110_2 \end{array}$$

Abbildung 7: Multiplikation

Schriftliche Multiplikation

```
1100 · 1101
───────────
    1100
   + 1100
   +  0000
   +   1100
───────────
   10011100 (156)
```

Rechnen im Dualsystem

Division

```
0 / 0 = nicht definiert
0 / 1 = 0
1 / 0 = nicht definiert
1 / 1 = 1
```

$$\begin{array}{rr} & 1010_2 \\ / & 10_2 \\ \hline & 101_2 \end{array}$$

Abbildung 8: Division

Schriftliche Division

```
  1000010 / 11 = 010110 Rest 0 (=22 im Dezimalsystem)
- 011
_____
  00100
  - 011
   ____
    0011
   - 011
   _____
       0
```

Zahldarstellung

Darstellung negativer Zahlen

- Varianten:
  - ein Bit als *Marker* für negative Zahlen
  - Darstellung im Zweierkomplement
- erste Variante: linkes Bit als Marker
  - 00000001 ist 1
  - 10000001 ist -1
  - Problem: erfordert aber zusätzliche Bausteine bei der physikalischen Umsetzung
- Zweierkomplement:
  - keine Unterscheidung notwendig, aber Umrechnung
  - Stellen negieren und 1 addieren
  - darstellbare Zahlen (8 Bit): -128 bis 127

Darstellung negativer Zahlen

8-Bit Zweierkomplement

Binärwert	Hex-Wert	Interpretation als Zweierkomplement	Interpretation als vorzeichenlose Zahl
00000000	00	0	0
00000001	01	1	1
...	...	...	...
01111110	7E	126	126
01111111	7F	127	127
10000000	80	−128	128
10000001	81	−127	129
10000010	82	−126	130
...	...	...	...
11111110	FE	−2	254
11111111	FF	−1	255

Abbildung 9: Zweierkomplement

Zweierkomplement

Umwandlung vom Dezimal- ins Dualsystem

- Verfahren:
  - Vorzeichen ignorieren und ins Binärsystem umrechnen: 4(10) = 00000100(2)
  - Invertieren: Not[00000100] = 11111011
  - Eins addieren: 11111011 + 00000001 = 11111100 (=-4)
- Rückumwandlung:
  - ist linkes Bit 1: negative Zahl; positive Zahlen können direkt umgewandelt werden
  - 1 subtrahieren und Ziffern negieren:

```
Zahl:           11111101   (= -3)
1 subtrahieren: 11111100
invertieren:    00000011
```

00000011 entspricht Dezimalsystem der Zahl 3.

Rechnen mit dem Zweierkomplement

- Subtraktion und Addition müssen nicht mehr unterschieden werden → Subtraktion wird auf Addition zurückgeführt
- Beispiel: $3 - 4$:

  ist dasselbe wie $-4 + 3 = -1$:

```
  11111100
+ 00000011
= 11111111
```

- Beispiel: $4 + (-4) = 0$:

```
   00000100
+  11111100
= 100000000
```

Die vorderste 9. Stelle wird verworfen (haben nur 8 Bit).

Logik

Boolesche Algebra

- benannt nach George Boole (1847: Logikkalkül)
- verwandt mit der *Aussagenlogik*
- Regeln:

Konjunktion

∧	**0**	**1**
0	0	0
1	0	1

Disjunktion

∨	**0**	**1**
0	0	1
1	1	1

Negation

	¬
0	1
1	0

Abbildung 10: Boolesche Algebra

Gesetze

- Kommutativgesetze:

$$a \wedge b = b \wedge a$$
$$a \vee b = b \vee a$$

- Assoziativgesetze:

$$(a \wedge b) \wedge c = a \wedge (b \wedge c)$$
$$(a \vee b) \vee c = a \vee (b \vee c)$$

Gesetze

- Distributivgesetze:

$$a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$$
$$a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$$

- doppelte Negation: $\neg(\neg \mathsf{a}) = \mathsf{a}$

Gesetze

- De Morgansche Gesetze:

$$\neg(a \wedge b) = \neg a \vee \neg b$$

$$\neg(a \vee b) = \neg a \wedge \neg b$$

- Darstellung und Berechnung einer *Formel* mit einer *Wahrheitstafel*:

```
a   b   ¬a   ¬a v b
--------------------
1   1    0      1
1   0    0      0
0   1    1      1
0   0    1      1
```

Bauelemente

Übertragung auf Bauelemente

- Logikgatter: Bauelement für die Umsetzung einer Booleschen Funktion
- mögliche Bauelemente: Dioden, Transistoren, Relais, Mechanik, Optik
- Liste von Gattern und zugehörigen Symbolen
- alle Gatter lassen sich auf die Grundbausteine AND, OR und NOT
- OR und NOT können mit NAND-Gattern aufgebaut werden
- oder: AND und NOT aus NOR-Gattern

  $\rightarrow$ sämtliche Schaltungen können aus NAND bzw. NOR aufgebaut werden

Ende