Forschungsinstitut für anwendungsorientierte Wissensverarbeitung/n
Lehrstuhl für Informatik an der Universität Ulm

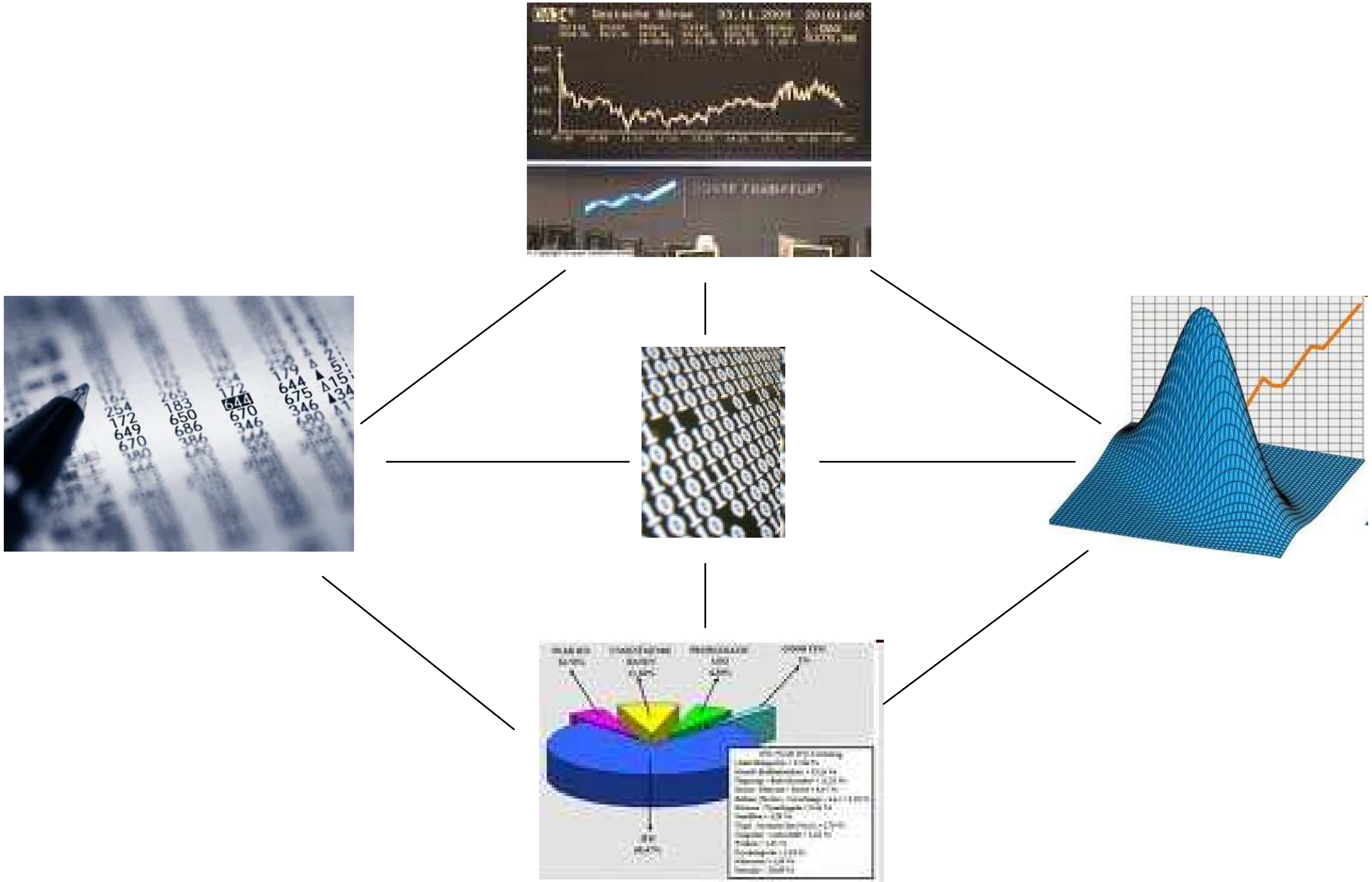
ulm university universität ulm

Zahlensysteme

Dipl.-Ing. Halit Ünver | 27.10.2010 |
Datenbanken/Künstliche Intelligenz | FAW/n

Inhalt

Teil I: Informatik und Zahlen für WiWi‘s



Teil II : Zahlensysteme

- Modell → Wirklichkeit
- Die „einfachste" Form der Darstellung einer Zahl

„7" ≈ | | | | | | |

„15" ≈ | | | | | | | | | | | | | | |

- Basis-Zahlendarstellung

$$zahl = \sum_{i=0}^{n} a_i \cdot b^i$$

Die „0" wird unbedingt benötigt!

- die Basis $b \geq 2$ ist aus den natürlichen Zahlen
- die Ziffer a_i ist aus den natürlichen Zahlen $0 \leq a_i \leq b-1$
- die Darstellung ist eindeutig

- **Schreibweise: zahl = $(a_n \ldots a_0)_b$**

  Beispiel: $(1024)_{10} = 1 \cdot 10^3 + 0 \cdot 10^2 + 2 \cdot 10^1 + 4 \cdot 10^0$

Teil II : Zahlensysteme

- Operationen: Summe, Differenz, Produkt, Quotient, Potenz etc.

  Vorteil: Algorithmus trivial für „Strichdarstellung"
  Man hängt die einen Striche an die anderen Striche.
  (oder z.B. Wegstreichen)

  Nachteil: Hoher Aufwand,
  Ergebnis schwer nutzbar.

Teil II : Zahlensysteme (Übung - Zählen)

- Zählen in verschiedenen Zahlensystemen

2-er System:	3-er System:	4-er System:	7-er System:	10-er System:
0	0	...	...	0
1	1			1
10	2			2
11	10			3
100	11			4
101	12			5
110	20			6
111	21			7
1000	22			8
1001	100			9
1010	101			10
1011	102			11
1100	110			12
1101	111			13
1110	112			14
1111	120			15
...	...			....

Teil II: Zahlensysteme

- Addition

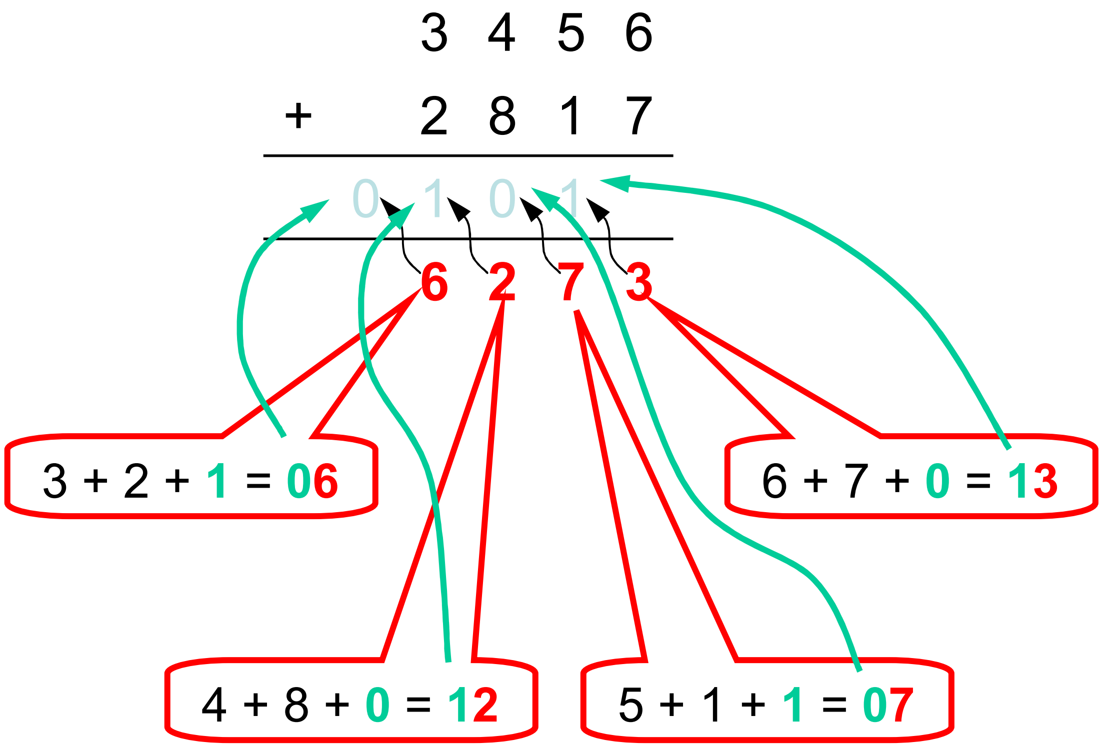
$$\begin{array}{cccccc|c|cc}
a_n b^n & + & a_{n-1} b^{n-1} & + & \dots & + & a_1 b^1 & + & a_0 b^0 \\
a'_n b^n & + & a'_{n-1} b^{n-1} & + & \dots & + & a'_1 b^1 & + & a'_0 b^0 \\
\hline
 & & \dots & & & & \ddot{u}_0 & & \\
\hline
 & & \dots & & & & s_1 \cdot b^1 & + & s_0 \cdot b^0
\end{array}$$

$$\ddot{u}_0 = \begin{cases} 0, \text{ falls } a_0 + a'_0 < b \\ 1, \text{ falls } a_0 + a'_0 \geq b \end{cases}$$

$$s_0 = \begin{cases} a_0 + a'_0, \text{ falls } a_0 + a'_0 < b \\ (a_0 + a'_0) - b, \text{ falls } a_0 + a'_0 \geq b \end{cases}$$

Teil II : Zahlensysteme

- Addition



Teil II : Zahlensysteme

- Addition

```
     3 4 9 7 1 6 5
+      5 4 8 2 1 3
     1 1 1 0 0 0
     4 0 4 5 3 7 8
```

Übertrag ist immer 0 oder 1

Addieren von 2 Zahlen = 3-er Addition, von denen eine nur aus 0-en und 1-en besteht

- Additionstabelle
  5-er System

+	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	**10**
2	2	3	4	**10**	**11**
3	3	4	**10**	**11**	**12**
4	4	**10**	**11**	**12**	**13**

Teil II : Zahlensysteme

- Additionstabelle 12-er System

+	0	1	2	3	4	5	6	7	8	9	A	B
0	0	1	2	3	4	5	6	7	8	9	A	B
1	1	2	3	4	5	6	7	8	9	A	B	**10**
2	2	3	4	5	6	7	8	9	A	B	**10**	**11**
3	3	4	5	6	7	8	9	A	B	**10**	**11**	**12**
4	4	5	6	7	8	9	A	B	**10**	**11**	**12**	**13**
5	5	6	7	8	9	A	B	**10**	**11**	**12**	**13**	**14**
6	6	7	8	9	A	B	**10**	**11**	**12**	**13**	**14**	**15**
7	7	8	9	A	B	**10**	**11**	**12**	**13**	**14**	**15**	**16**
8	8	9	A	B	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**
9	9	A	B	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**
A	A	B	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**
B	B	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**1A**

Teil II : Zahlensysteme

- Multiplikation

1	2	3	·	4	5	6
		4				
			8			
			1	2		
		4	9	2	0	0
			5			
			1	0		
				1	5	
			6	1	5	0
				6		
				1	2	
					1	8
+				7	3	8
		1	1	0	0	
		5	6	0	8	8

Teil II : Zahlensysteme

- Multiplikationstabelle 10-er System

·	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	8	9
2	0	2	4	6	8	10	12	14	16	18
3	0	3	6	9	12	15	18	21	24	27
4	0	4	8	12	16	20	24	28	32	36
5	0	5	10	15	20	25	30	35	40	45
6	0	6	12	18	24	30	36	42	48	54
7	0	7	14	21	28	35	42	49	56	63
8	0	8	16	24	32	40	48	56	64	72
9	0	9	18	27	36	45	54	63	72	81

Teil II: Zahlensysteme

- Multiplikationstabelle 5-er System

·	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	11	13
3	0	3	11	14	22
4	0	4	13	22	31

Kleiner, einfacher als Tabelle im 10-er System

Teil II: Zahlensysteme (Übung: Addieren)

- $(34)_7 + (41)_7$
- $(1011)_3 + (1101)_3$
- $(123)_6 + (345)_6$
- $(123)_4 + (3121)_4$
- $(A5A5)_{16} + (F0F0)_{16}$

Teil II : Zahlensysteme (Übung: Multiplikation)

- $(230)_4 \times (132)_4$
- $(1011)_3 \times (1101)_3$
- $(123)_6 \times (345)_6$
- $(123)_4 \times (3121)_4$
- $(555)_7 \times (666)_7$

Teil III : Zahlendarstellung im Rechner

- Für Informatik besonders wichtig: 2-er System

Vorteil: nur noch 0 und 1;
2 gut identifizierbare und technisch realisierbare
Zustände (Strom fließt vs. Strom fließt nicht, positive

Frage:

$$\overbrace{|\;|\;|\;|\;|\;|\;|\;|\;|\;|\;|\;|\;|}^{13} = (?)_2$$

$13 \quad = \quad 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 \quad \rightarrow \quad 1101$

$5 \quad \rightarrow \quad 101$

Teil III : Zahlendarstellung im Rechner

- Additionstabelle 2-er System

+	0	1
0	0	1
1	1	**10**

+ (+1)	0	1
0	1	**10**
1	**10**	**11**

Die grün gekenn-zeichneten Fälle erzeugen einen Übertrag

Teil III : Zahlendarstellung im Rechner

- Konvertierung

$(377)_{10} \quad = \quad (?)_2$

377	:	2	=	188	Rest	1
188	:	2	=	94	Rest	0
94	:	2	=	47	Rest	0
47	:	2	=	23	Rest	1
23	:	2	=	11	Rest	1
11	:	2	=	5	Rest	1
5	:	2	=	2	Rest	1
2	:	2	=	1	Rest	0
1	:	2	=	0	Rest	1

Stop bei 0

Ergebnis: Rückwärtsanordnung der Reste, d.h.

$(1\ 0\ 1\ 1\ 1\ 1\ 0\ 0\ 1)_2 \quad = \quad (377)_{10}$

Probe, d.h. Konvertierung vom 2-er System in 10-er System

1	0	1	1	1	1	0	0	1		
256 +		64 +	32 +	16 +	8 +			1	=	377
2^8		2^6	2^5	2^4	2^3			2^0		

Teil III : Zahlendarstellung im Rechner

- Digitalrechner sind **endlich** !
- Häufig werden 32 bit, d.h. 32 Dualstellen bereitgestellt
- Kleinste Zahl $(0 \ldots 0)_2 = (0)_{10}$
- Größte Zahl $(1 \ldots 1)_2 = (4\,294\,967\,295)_{10}$
- Größere Zahlenbereiche als der 32 bit sind möglich z.B. 48 bit oder 64 bit
- Summer, Produkt darstellbarer Zahlen ist u.U. nicht mehr darstellbar („Overflow“)
- Bsp.: nur 4 bits

$$
\begin{array}{c|cccc}
 & 0 & 1 & 1 & 1 \\
+ & 1 & 0 & 0 & 1 \\
\hline
1 & 1 & 1 & 1 & \\
\hline
1 & 0 & 0 & 0 & 0
\end{array}
$$

?

=> Summe nicht ausführbar!

Teil I : Zahlensysteme (Übung: Konvertierung)

- $(57)_{10} = (?)_2$
- $(16)_{10} = (?)_2$
- $(124)_{10} = (?)_8$
- $(1101111010)_2 = (?)_{10}$
- $(14F5B)_{16} = (?)_{10}$
- $(14876)_{10} = (?)_{16}$

Teil III : Zahlendarstellung im Rechner

- Negative Zahlen und Subtraktion

1. (signed integer)

Ein Bit wird als Vorzeichen reserviert.
Dann sind nun weniger Bits zur eigentlichen Zahlendarstellung (Betragsdarstellung) verfügbar.
Typischerweise wird das Bit ganz links als Vorzeichen verwendet.

- Vorzeichenbit = 0 $\leftrightarrow$ +
- Vorzeichenbit = 1 $\leftrightarrow$ –

Beispiel:

$$\left.\begin{array}{l} (\mathbf{0}\ 1\ 1\ 1\ 0\ 0\ 1\ 0)_2 = 114 \\ \mathbf{+}\ 64 + 32 + 16 + 8 + 2 \\ \\ (\mathbf{1}\ 1\ 1\ 1\ 0\ 0\ 1\ 0)_2 = -114 \\ \mathbf{-}\ 64 + 32 + 16 + 8 + 2 \end{array}\right\}$$

Darstellbarkeit negativer Zahlen erscheint in Ordnung

Teil III : Zahlendarstellung im Rechner

- Negative Zahlen und Subtraktion

a.) *(Signed Integer)*

Problem:

	0	0001001	(= 9)
+	1	0000100	(= – 4)
	1	0001101	(= – 13)

Problem:

$$(0 \dots 0)_2 \quad = \quad 0 \quad = \quad (1\ 0 \dots 0)_2$$

„+ 0“ „– 0“

=> Die „0“ hat zwei Darstellungen

Teil III : Zahlendarstellung im Rechner

- Negative Zahlen und Subtraktion

2-er Komplement (Bedeutet Ergänzung)

Idee: Löse das Problem rückwärts!
Finde zu irgendeiner Binärzahl, z.B. 00001001_2 eine Binärzahl, so dass deren Summe 0 ist. (Beschränkung auf 8 Bit)

$$\begin{array}{cc|ccccccc} & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ + & & & & & ? & & & \\ \hline & & & & & & & & \\ \hline & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

Teil III : Zahlendarstellung im Rechner

- Negative Zahlen und Subtraktion

b.) ***2-er Komplement***

	0	0	0	0	1	0	0	1
+	**1**	**1**	**1**	**1**	**0**	**1**	**1**	**1**
1	1	1	1	1	1	1	1	
	0	0	0	0	0	0	0	0

1. Beobachtung:

Es entsteht eine neue Stelle.
→ ignorieren

2. Beobachtung:

Die zu addierende Zahl ist das Komplement der ursprünglichen Zahl – bis auf die letzte Stelle.

Komplement: 0 → 1
1 → 0

Teil III : Zahlendarstellung im Rechner

- Negative Zahlen und Subtraktion

b.) ***2-er Komplement***

		00001111
Reine Komplementbildung		11110000
Addition von 1	+	00000001
		11110001

		00001110
Reine Komplementbildung		11110001
Addition von 1	+	00000001
		11110010

Teil III : Zahlendarstellung im Rechner

- Negative Zahlen und Subtraktion

b.) ***2-er Komplement Beispiel:*** $(13 - 9)_{10}$

1. Schritt: **Bildung** des 2-er Komplements von 9

	0	0	0	0	1	0	0	1
	1	1	1	1	0	1	1	0
+	0	0	0	0	0	0	0	1
	1	1	1	1	0	1	1	1

2-er Komplement von $(9)_{10}$

Teil III : Zahlendarstellung im Rechner

- Negative Zahlen und Subtraktion

b.) ***2-er Komplement***

2. Schritt: **Addition** von 13 und dem 2-er Komplement von 9

	0	0	0	0	1	1	0	1	=	$(13)_{10}$
+	1	1	1	1	0	1	1	1	=	2-er Komplement von $(9)_{10}$
1	1	1	1	1	1	1	1			
	0	0	0	0	0	1	0	0	=	$(4)_{10}$

ignorieren des Übertrags!

Teil III : Zahlendarstellung im Rechner

- Negative Zahlen und Subtraktion

b.) ***2-er Komplement von 8-bit Zahlen***

- Subtraktion = Addition mit 2-er Komplement
- 2-er Komplement entspricht Addition ohne Überträge
- Reines Komplement = Ziffernweise/Bitweise Negation

```
    00001001
+   11111111
------------
    11110110
```

Diese beiden Zahlen sind die reinen Komplemente voneinander.

Überträge ignorieren

Es wird stets diese Zahl zum Binärmuster addiert.

Teil III : Zahlendarstellung im Rechner

- Negative Zahlen und Subtraktion

Multiplikationen beim 2-er Komplement werden in dieser Vorlesung (Übungen, Tutorien, Klausur) **auch nicht** behandelt.

Teil III : Zahlendarstellung im Rechner (Übung)

- $(-100)_{10} = (?)_2$
- $(119)_{10} = (?)_2$
- $(-76)_{10} = (?)_2$

Teil IV: Gleitkommazahlen

- Beispiel:

$$
\begin{array}{rrlcccccc}
 & 11{,}375 & < & 16 & \rightarrow & 0 & \cdot & 2^{4} & = 0 \\
- & 0 & & & & & & & \\ \hline
 & 11{,}375 & \geq & 8 & \rightarrow & 1 & \cdot & 2^{3} & = 8 \\
- & 8 & & & & & & & \\ \hline
 & 3{,}375 & < & 4 & \rightarrow & 0 & \cdot & 2^{2} & = 0 \\
- & 0 & & & & & & & \\ \hline
 & 3{,}375 & \geq & 2 & \rightarrow & 1 & \cdot & 2^{1} & = 2 \\
- & 2 & & & & & & & \\ \hline
 & 1{,}375 & \geq & 1 & \rightarrow & 1 & \cdot & 2^{0} & = 1 \\
- & 1 & & & & & & & \\ \hline
 & 0{,}375 & < & 0{,}5 & \rightarrow & 0 & \cdot & 2^{-1} & = 0 \\
- & 0 & & & & & & & \\ \hline
 & 0{,}375 & \geq & 0{,}25 & \rightarrow & 1 & \cdot & 2^{-2} & = \frac{1}{4} \\
- & 0{,}25 & & & & & & & \\ \hline
 & 0{,}125 & \geq & 0{,}125 & \rightarrow & 1 & \cdot & 2^{-3} & = \frac{1}{8} \\
- & 0{,}125 & & & & & & & \\ \hline
 & 0{,}0 & & & & & & &
\end{array}
$$

Ergebnis: $(1\ 0\ 1\ 1,\ 0\ 1\ 1)_2$

$1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2} + 1 \cdot 2^{-3}$

Teil IV: Gleitkommazahlen

- Bisher: $1\,2\,3 = 1 \cdot 10^2 + 2 \cdot 10^1 + 3 \cdot 10^0$
- Jetzt:

7 : 4	=	**1**	Rest	3
3 **0** : 4	=	**7**	Rest	2
2 **0** : 4	=	**5**	Rest	0

„geliehen“

„geliehen“

= 1 . 75

Man muss die Stelle markieren, an der man begonnen hat, sich „Stellen auszuleihen“.
⇒ Einführung des „Kommas“

„Basiszahlendarstellung“:

$$1 \cdot \underbrace{1}_{10^0} + 7 \cdot \frac{1}{\underbrace{10}_{10^{-1}}} + 5 \cdot \frac{1}{\underbrace{100}_{10^{-2}}}$$

Teil IV: Gleitkommazahlen

- Normierung:

$$345 \cdot 10^{9} = 345 \cdot 1\,000\,000\,000$$
$$= 345\,000\,000\,000$$
$$= 3{,}45 \cdot 10^{11} = 0{,}345 \cdot 10^{12}$$

$$123 \cdot 10^{-5} = 123 \cdot 0{,}0001$$
$$= 0{,}00123$$
$$= 1{,}23 \cdot 10^{-3} = 0{,}123 \cdot 10^{-2}$$

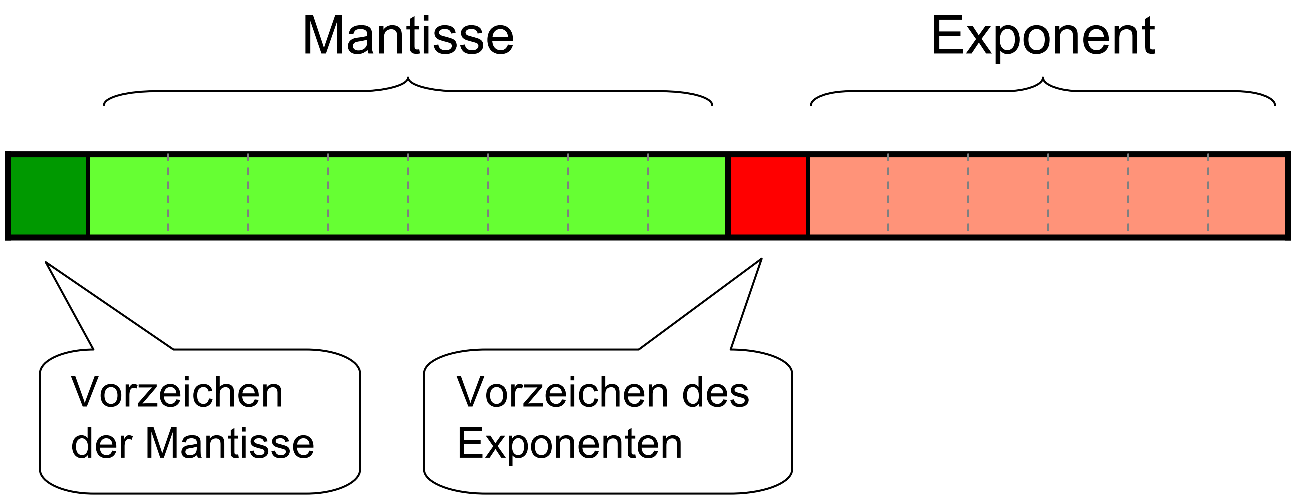
Ob auf x,xxx oder 0,xxxx normiert wird ist letztendlich eine willkürliche Festlegung

standardmäßig auftretender Fall: 1,xxxxx oder 0,1xxxxx

Teil IV: Gleitkommazahlen

- Darstellung: Wieder basierend auf dem 2er System.

  Aufteilung eines Bereiches für die „Nachkommazahlen“ (bestimmte Anzahl von Stellen / Bits als Genauigkeit) und eines Bereiches für den Exponenten.



Teil IV: Gleitkommazahlen

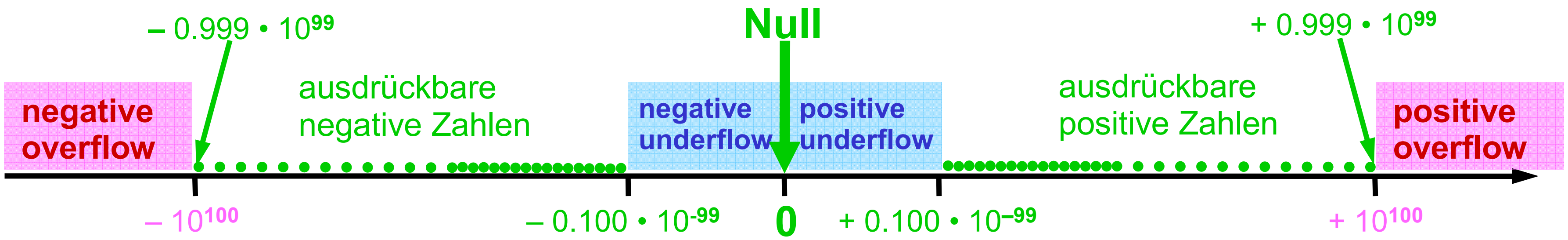
- „Hidden Bit“:

  Um Platz zu sparen, hat man sich daher entschieden, das erste Bit einzusparen und einfach anzunehmen, dass es auf 1 gesetzt ist. (so genanntes „hidden Bit“)

  Die restliche Mantisse hat daher 1 Bit mehr zur Verfügung.

Teil IV: Gleitkommazahlen

- Konsequenzen:



•jede Gleitkommazahl **repräsentiert ein Intervall** der reellen Zahlen; als Intervall wächst mit zunehmendem Betrag der Zahl, d.h. die **Dichte der Repräsentation** nimmt mit zunehmendem Betrag der Zahl ab

•Eine Abschätzung des Einflusses der Ungleichverteilung der Repräsentanten auf die Rechenoperationen ist nicht trivial

•Behandlung von **overflow** / **underflow**, **Null**, „undefiniert“?

Teil III : Gleitkommazahlen (Übung)

- $(2.6)_{10} = (?)_2$
- $\pi = (?)_2$, mit $\pi = (3.14159265)_{10}$

Umrechnung von Zahlensystemen

Link:

http://www.arndt-bruenner.de/mathe/scripts/Zahlensysteme.htm