- **Quantenmechanik:** auf Ebene feinster Körnigkeit sind physikalische Prozesse diskret und unscharf

- **Abtasttheorem:** jede periodische Funktion der Form

$$f(t) = \frac{a_0}{2} + \sum_{k=1}^{\cancel{\infty}\; n} (a_k \cos k\omega_0 t + b_k \sin k\omega_0 t)$$

Fourier-Reihe

mit Grenzfrequenz $\omega_g = n\omega_0$ lässt sich aus Meßwerten rekonstruieren, wenn Abtastfrequenz $\geq 2\omega_g$

Beispiel:

Meßgeräte schneiden hohe Frequenzen ab

$\Rightarrow$ Voraussetzung des Abtasttheorems praktisch immer gegeben

1.1 Codierung ganzer Zahlen

binär codierte Dezimalzahlen: codieren einzelne Dezimalziffern

- rechnen im Dezimalsystem

  z.B. Darstellung von z = 295 z = OOLO LOOL OLOL
  (BCD: *binary coded decimal*)

  2 9 5

Binärdarstellung ganzer Zahlen: $z = \sum_{i=0}^{n-1} z_i \, 2^i$

$37 = 1\,2^5 + 0\,2^4 + 0\,2^3 + 1\,2^2 + 0\,2^1 + 1\,2^0$

- rechnen im Binärsystem
- zusammenfassen oft im Oktalsystem (Basis 8) oder Sedezimalsystem (Basis 16)

> Darstellung der Zahlen 10-15 im Sedezimalsystem:
>
> A ≙ 10, B ≙ 11, C ≙ 12, D ≙ 13, E ≙ 14, F ≙ 15

- in Rechnern und Programmiersprachen ist n fest

  Langzahlarithmetik: n unbeschränkt

  $n = 37_{10} = 100101_2$ 2 5 $_{16}$ 4 5 $_8$

- keine negativen Zahlen

alle$_{16}$ $0 \le n \le 2^{31} - 1$

1.1 Codierung ganzer Zahlen: negative Zahlen

Einerkomplement: invertiert jedes Bit, d.h. $-z \triangleq 2^n - 1 - z = \sum_{i=0}^{n-1} (1-z_i)2^i$

$$+0 = \sum_{i=0}^{n-1} 0 * 2^i \qquad -0 = \sum_{i=0}^{n-1} 1 * 2^i$$

Beispiel: Binärzahl: 001100001110

Einerkomplement: 110011110001

0 . pos

1 .. neg

Zweierkomplement: $-z \triangleq 2^n - z = 1 + \sum_{i=0}^{n-1} (1-z_i)\, 2^i$

- darstellbarer Zahlenbereich mit n+1 Bits: $-2^n \leq z \leq 2^n - 1$
- n und daher darstellbarer Zahlenbereich beliebig

0 0 ...0

1 . 1

Vorzeichen und Betrag: $-z \triangleq (L, |z|)$

Beispiel: Binärzahl: 0100110 (dezimal : 38)

1 0 . 0

0 0 0

Einerkomplement: 1011001

$2^{31} - 10$ 0

\+ 1

$1 - 2^{31}$ 1-10 .0

Zweierkomplement: 1011010 (dezimal : -38)

Vorzeichen und Betrag: 1100110 (dezimal : -38)

	Quellsystem	Zielsystem ← Rechn
Q = 10	12 : 2 = 6 R 0 6 : 2 = 3 R 0 3 : 2 = 1 R 1 1 : 2 = 0 R 1	12 $\times 1010_2$ 1 1010 + 10 = 1100
Q = 2	1100 : 1010 = 12 1010 10_2 1010	1100 2 1 2 + 1 = 3 2 6 + 0 = 6 2 12 + 0 = 12

← Rechner.

Konvertierung ganzer Zahlen:

- rechnen im **Zielsystem**: (Zielsystem ist Dezimalsystem)

Beispiel: 1100011_2 dezimal darstellen:

Quellziffer	1	1	0	0	0	1	1
zu addieren	-	2	6	12	24	48	98
Quellbasis	*2	*2	*2	*2	*2	*2	
Stellensumme	1	3	6	12	24	49	99

(nach Hornerschema)

Hornerschema: neues Ergebnis := altes Ergebnis·Basis + Ziffer

hier: (((((**1**· 2 +**1**) ·2+**0**)·2+**0**)·2+**0**)·2+**1**)·2+**1**

die Lösung ist 99_{10}

$Z = \sum z_i \cdot 2^i$

Darstellung von Brüchen

- Nicht ⇒ Ganzzahlen
- Festkomma

```
  1101,1011
+ 0000,0001
-----------
  1101,1100
```

```
2,1 * 1,1
    21
     21
  ------
   2,31
```

- Fließkomma

Festkommadarstellung: $z = \sum_{i=-c}^{d-1} z_i * 2^i$

- komplette Menge der reellen Zahlen kann nicht durch endliche Zeichenreihen über $\mathbb{B}$ dargestellt werden
- Anwendung bei Prozeßsteuerungen zur Geschwindigkeitssteigerung

Gleitkommadarstellung: $z = m * b^e$, b, e ganzzahlig, $b \geq 2$, $0 \leq m < b$.

- m heißt **Mantisse**, e heißt **Exponent**, b heißt **Basis**
- 3.14159, 0.314159E1, 0.0314159E2 (typisch für Programmiersprachen) $\hat{=}$ $0.314159 \times 10^1 =$ 3.14159×10^0 0.000314×10^4
- im Rechner: m, e sind binär codiert, b = 2

$0 \leq m \leq 1$

normalisierte Gleitkommadarstellung: $1 \leq m < b$ oder m=0, falls z=0

$\Rightarrow$ in Binärdarstellung ist erstes Bit von m gleich L

Gleitkommadarstellung 1937 von Konrad Zuse erfunden

gemischte Logarithmische Darstellung

1.1 Gleitkommazahlen nach IEEE 754-1985

einfache Länge: float 32 **doppelte Länge:** double 64

1	8	23
S	E	M
Vorzeichen	Exponent (verschoben)	Mantisse (gekappt)

1	11	52
S	E	M
Vorzeichen	Exponent (verschoben)	Mantisse (gekappt)

- **Exponent**: e = E - 127 bzw. E – 1023

  $e = 0_{10} \Rightarrow E = 1111111_2$

- $1 \le m < 2 \Rightarrow$ erstes Bit in **Mantisse** ist implizit und immer 1
- **Null:** e = 0, m = 0

  $\Rightarrow$ positive und negative Null

  Schlecht: Test auf 0 häufig

  Gut $1/\infty = +0$

  $-1/\infty = -0$

- **Unendlich:** e = 255 und m = 0 sonst keine Zahl mit s = 255

  $\Rightarrow$ nicht alle Bitfolgen entsprechen Zahlen: NaN (*Not a Number*)

1	8	23
S	E	M
Vorzeichen	Exponent (verschoben)	Mantisse (gekappt)

1	11	52
S	E	M
Vorzeichen	Exponent (verschoben)	Mantisse (gekappt)

- 1.0 ≙ 0 $\underbrace{01111111}_{=127}$ $\underbrace{00000000000000000000000}_{\hat{=}1,0}$ + $2^{127-127} \cdot 1,0 = 1$
  0.5 ≙ 0 01111110 00000000000000000000000 $2^{126-127} \cdot 1,0 =$
  254 ≙ 1111110 $2^{-1} \cdot 1,0 = 1/2$
- Rechnen: $z_1 = m_1 * 2^{e_1}$, $z_2 = m_2 * 2^{e_2}$
  - $z_1 \pm z_2$ = normalisiere $(m_1 \pm m_2 * 2^{e_2 - e_1}) 2^{e_1}$
  - $z_1 * z_2$ = normalisiere $(m_1 * m_2) * 2^{e_1 + e_2}$
  - z_1 / z_2 = normalisiere $(m_1 / m_2) * 2^{e_1 - e_2}$

$\overbrace{2,3}^{2} \times 10^0$ +

$\overbrace{3,4}^{2} \times 10^1$

Überlauf: zu großer Exponent

Unterlauf: zu kleiner Exponent

Rundungsfehler: Fehler durch Runden der Mantisse

2,3
34
$\underbrace{36,3}_{3}$

$\boxed{\text{Bit}}$ $\quad b \in \mathbb{B} = \{L, O\}$

$A = \{x, y\}$

$f: A \to \mathbb{B}^*$, $x \to LL$, $y \to OO$

$n \in A^*$, $|n| = k$

$|f(n)| =$ 2k Bit (Binärsymbole)

$\underbrace{\text{k bit (Information)}}_{\text{Nachrichtentechnik, Kryptographie}}$

~~A~~ Mathematical Theory of Communication

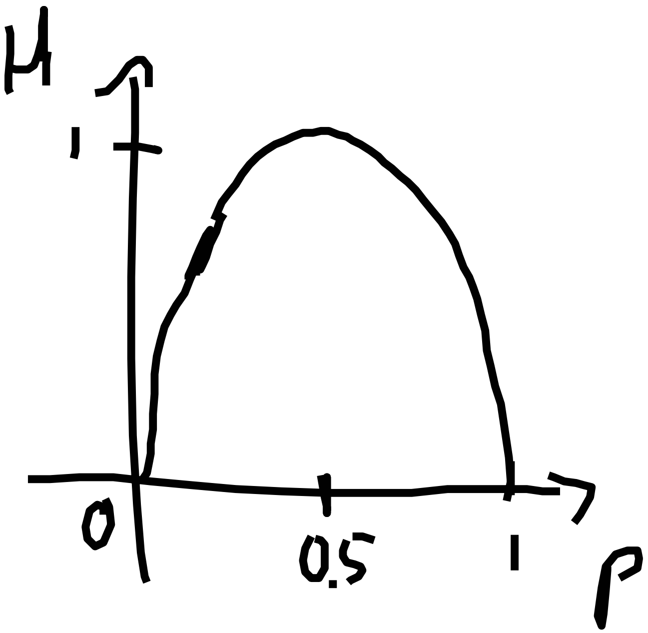
The

Shannon, 1948
1949

Univ. Illinois Press ~ 1996

$$A = \{ \underset{p}{x}, \underset{1-p}{y} \}$$

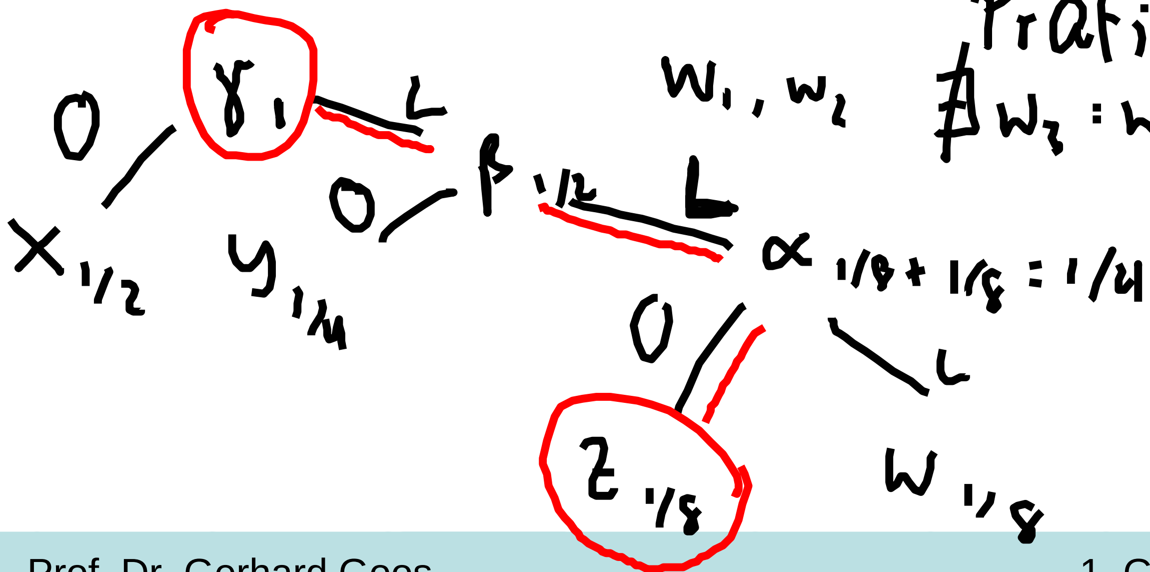
$$H = -\left(p \log p + (1-p) * \log (1-p) \right)$$



$A = \{x, y, z, w\}$

$w \in A^* = xyzxxwyx$, $|w| = 8$

x : IIII, $= 4$	$p(x) = 4/8 = 1/2$	x 0
y : II, $= 2$	$p(y) = 2/8 = 1/4$	y L0
z, w : I, $= 1$	$p(z), p(w) = 1/8$	z LL0
		w LLL



Präfix-Eigensch.

$\nexists w_3 : w_1 w_3 = w_2$

x y z w x x y x

0 10 110 111 0 0 10 0

x y z . . . . .

x	0
y	L0
z	LL0
w	LLL