



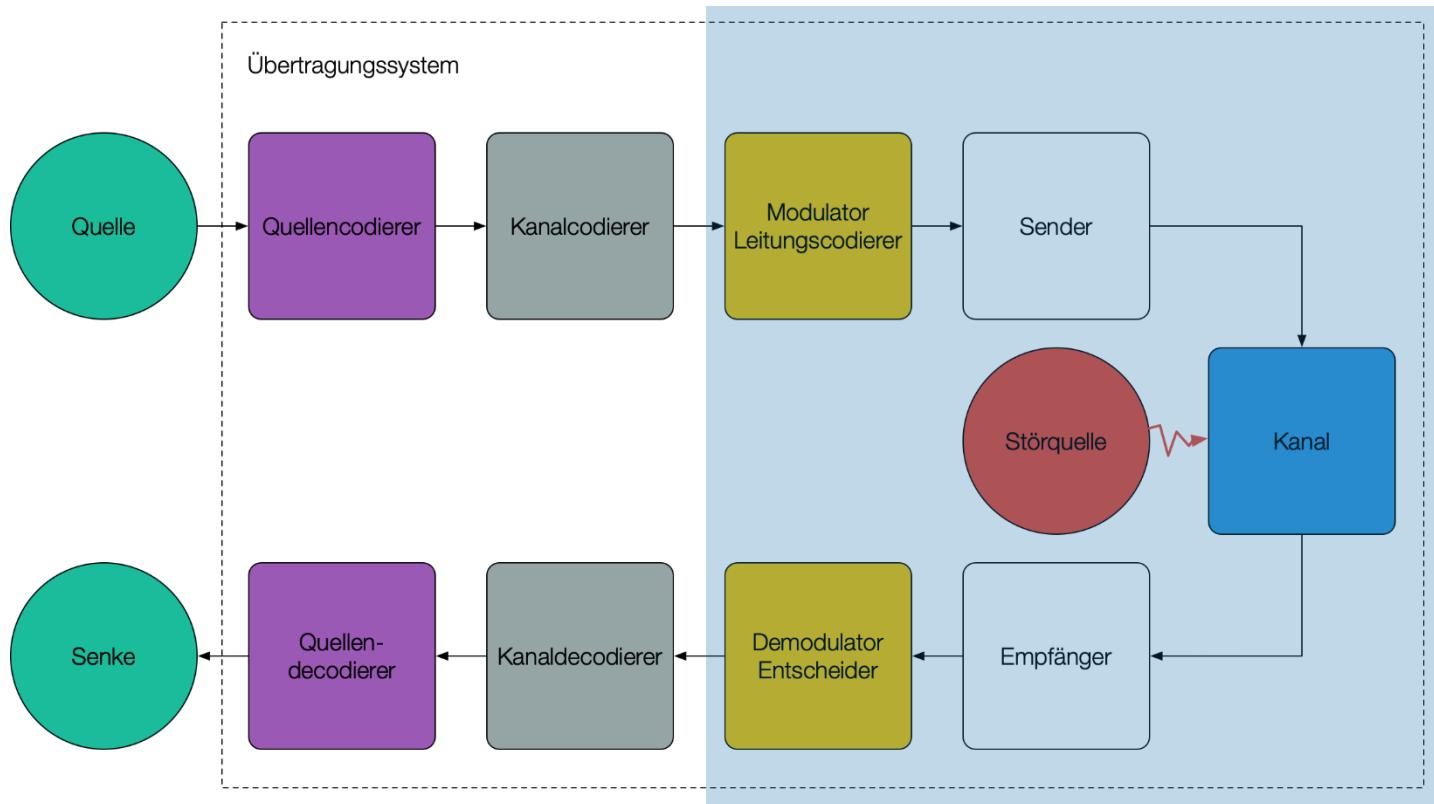
**OST**  
Ostschweizer  
Fachhochschule

# ***Informations- und Codierungstheorie***

## ***Teil 1: Einführung***

- Prof. Dr.-Ing. Andreas Rinkel  
[andreas.rinkel@hsr.ch](mailto:andreas.rinkel@hsr.ch)
- Tel.: +41 (0) 55 2224461
- Mobil: +41 (0) 79 3320562
- Teams

# Modell der Informationsverarbeitung



## Ziel der Wahrscheinlichkeitsrechnung:

- **Modellierung & Vorhersage von zufälligen Vorgängen, wie z.B.**
  - Auftreten eines Bitfehlers
  - Auftreten eines Zeichens in einem Code
  - Definition des Informationsgehaltes eines Zeichens



Alles was lediglich  
wahrscheinlich ist,  
ist wahrscheinlich falsch.

**René Descartes**

# Zufallsvorgänge & Ereignisse

## **Definition: (Zufallsvorgang, Zufallsexperiment)**

**Unter einem *Zufallsvorgang* verstehen wir einen Vorgang, bei dem**

- im Voraus feststeht, welche möglichen Ausgänge dieser theoretisch haben kann (z.B. Ein Bit wird gedreht, 0 oder 1, oder ein lesbares Zeichen wird in ein anderes lesbares Zeichen überführt).
- der sich einstellende, tatsächliche Ausgang im Voraus jedoch unbekannt ist (Tritt ein Bitfehler bei der Datenübertragung auf oder nicht? Unsicherheit bei einem Folgezeichen).

**Zufallsvorgänge, die geplant sind und kontrolliert ablaufen, heißen  
Zufallsexperimente.**

# Zufallsvorgänge Beispiele

- **Ziehung der Lottozahlen**
- **Roulette, Münzwurf, Würfelwurf**
- **Messen der Bitfehlerrate**
- **Ermitteln des Informationsgehalts eines Zeichens**
- **Ermittlung der Kanalmatrix**

# Ergebnis & Ergebnismenge

## Definition: (*Ergebnismenge*)

Die **Menge aller möglichen Ausgänge (Ergebnisse)** eines Zufallsvorgangs heisst **Ergebnismenge** und wird mit  $\Omega$  bezeichnet.

Ein **einzelnes Element**  $\omega \in \Omega$  heisst **Ergebnis**. Wir notieren die Anzahl aller Elemente von  $\Omega$ , d.h. die Anzahl aller Ergebnisse mit  $|\Omega|$ .

# Ereignismenge

## Definition: (Ereignismenge)

Die Menge aller möglichen Teilmengen (Ereignisse), die aus der Ergebnismenge eines Zufallsvorgangs gebildet werden heisst  
*Ereignismenge.*

Die Anzahl aller möglichen Ereignisse ist durch die Potenzmenge  $\mathcal{P}$  der Ergebnismenge definiert.

Die Mächtigkeit für endliche Mengen  $X$  ist gegeben durch:  $|\mathcal{P}| = 2^{|X|}$

Siehe auch Ex&Ev, Das System der Ereignisse

Beispiele:

- $\mathcal{P}(\emptyset) = \{\emptyset\}$
- $\mathcal{P}(\{a\}) = \{\emptyset, \{a\}\}$
- $\mathcal{P}\{a, b\} = \{\emptyset, \{a\}, \{b\}, \{a, b\}\}$
- $\mathcal{P}\{a, b, c\} = \{\emptyset, \{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}$

# Ergebnismengen Beispiele

**Zufallsvorgang: 'Werfen eines Würfels'**

$$\Omega = \{1, 2, 3, 4, 5, 6\}$$

**Zufallsvorgang: 'Werfen einer Münze solange, bis Kopf erscheint':**

$$\Omega = \{K, ZK, ZZK, ZZZK, ZZZZK, \dots\}$$

# Eigenschaften von Wahrscheinlichkeiten

## **Wahrscheinlichkeit des Komplementärereignisses:**

- $P(\bar{A}) = 1 - P(A)$

## **Wahrscheinlichkeit des unmöglichen Ereignisses:**

- $P(\emptyset) = 0$

## **Wertebereich der Wahrscheinlichkeit:**

- $0 \leq P(A) \leq 1$

## Additionssatz für Wahrscheinlichkeiten:

- $P(A \cup B) = P(A) + P(B) - P(A \cap B)$

## Additionssatz für 3 Ereignisse (Wahrscheinlichkeit, dass A, B oder C eintritt):

- $P(A \cup B \cup C) = P(A) + P(B) + P(C) - P(A \cap B) - P(A \cap C) - P(C \cap B) + P(A \cap B \cap C)$

# Wahrscheinlichkeitsdefinition nach Laplace

**Pierre-Simon Marquis de Laplace, 1982:**

**Wenn ein Experiment eine Anzahl verschiedener und gleich möglicher Ausgänge hervorbringen kann und einige davon als günstig anzusehen sind, dann ist die Wahrscheinlichkeit eines günstigen Ausgangs gleich dem Verhältnis der Anzahl der günstigen zur Anzahl der möglichen Ausgänge.**

$$P(A) = \frac{\text{Anzahl der guenstigen Ergebnisse } A}{\text{Anzahl aller Ergebnisse } \Omega} = \frac{|A|}{|\Omega|} = \frac{|A|}{n}$$

# Beispiel fairer Würfel

**Es ist:**

- $\Omega = \{\omega_i\}$  mit  $i = 1 \dots 6 = \{1, 2, 3, 4, 5, 6\}$

**Laplace-Wahrscheinlichkeit für das Ereignis A:**

- **Würfeln einer beliebigen Zahl  $\omega_i$  aus der Menge  $\Omega$ .**
- $P(\{\omega_i\}) = \frac{\text{Anzahl der guenstigen Ergebnisse}}{\text{Anzahl aller Ergebnisse } \Omega} = \frac{|\omega_i|}{|\Omega|} = \frac{1}{6}$
- **Wie gross ist die Laplace-Wahrscheinlichkeit eine gerade Zahl zu würfeln?**

**Laplace-Wahrscheinlichkeit erfordert Berechnung von Anzahlen**

**Mathematische Technik hierfür: *Kombinatorik***

**Einige grundsätzliche Fragen der Kombinatorik:**

- Wie viele Möglichkeiten gibt es, bestimmte Objekte anzugeordnen?
- Wie viele Möglichkeiten gibt es, bestimmte Objekte aus einer Menge auszuwählen?
- Hier betrachten wir nur soweit nötig die geordnete und die ungeordnete Probe.

## Geordnete Proben

Die Anzahl der **k-Tupel** aus einer **n-Menge** mit Wiederholung ist  **$n^k$** .

- **Beispiel**
- Bei einem Zifternschloss muss man eine 5-stellige Zahl einstellen, die aus den Ziffern 0,1,...,9 gebildet wird.
- Wie viele Kombinationen gibt es?

$$10^5 = 100.000$$

## Geordnete Proben

- Die Anzahl der **k-Tupel aus einer n-Menge ohne Wiederholung** ist
- **Anzahl =  $n \cdot (n-1) \cdot (n - 2) \cdot \dots \cdot (n - k + 1)$**
- Formel: **Anzahl =  $\frac{n!}{(n-k)!}$**
- Oder besser:  
**Anzahl =  $\prod_n^{n-k+1} n$**

## Geordnete Proben

### Beispiel

Beim Pferde-Toto “3 aus 18” muss man von 18 Pferden 3 gemäß der Reihenfolge ihres Zieleinlaufs tippen.

Wie viele Tippmöglichkeiten gibt es?

■ Anzahl =  $\prod_n^{n-k+1} n = \prod_{n=18}^{n=16} n = 18 \cdot 17 \cdot 16 = 4896$

## Geordnete Proben

Die Zahl der Permutationen einer n-Menge ist  $n!$

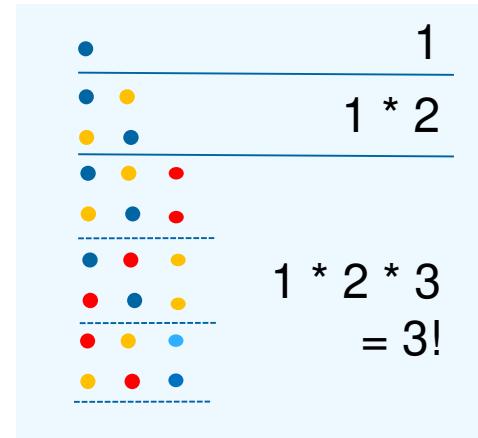
- **Bemerkung:**

- Dies ist ein Spezialfall eines **k-Tupels** aus einer **n-Menge** ohne Wiederholung für  $n = k$ .

### Beispiel

Zur Festlegung einer Sitzordnung bei einer Feier mit 10 Personen gibt es

$10! = 3.628.800$  Möglichkeiten.



## Ungeordnete Proben

- Die Anzahl der **k-elementigen Teilmengen aus einer n-elementigen Menge ist:**

$$\binom{n}{k} = \frac{\prod_{n-k+1}^n n}{k!} = \frac{n!}{k! (n-k)!}$$

Die Anzahl aller möglichen Kombinationen k aus n unter Berücksichtigung der Reihenfolgen.

Kompakte aber nicht praktikable Formel.  
Warum?

Da die Reihenfolgen keine Rolle spielt, muss noch durch die Anzahl aller möglichen Kombinationen von k als k! geteilt werden!

- **Beispiel**

Eine Schulklasse mit 25 Schülern möchte ein Schachturnier austragen, bei dem jeder Schüler **einmal** gegen jeden anderen Schüler spielt.  
Wie viele Spiele werden ausgetragen?

$$\binom{25}{2} = \frac{25 \cdot 24}{2!} = 300$$



**OST**  
Ostschweizer  
Fachhochschule

# ***Informations- und Codierungstheorie***

## ***Teil 2. Einführung und Übersicht***

- Prof. Dr.-Ing. Andreas Rinkel  
[andreas.rinkel@hsr.ch](mailto:andreas.rinkel@hsr.ch)
- Tel.: +41 (0) 55 2224461
- Mobil: +41 (0) 79 3320562
- Teams

## **Wahrscheinlichkeit des sicheren Ereignisses**

- $P(A) = 1$

## **Wahrscheinlichkeit des Komplementärereignisses:**

- $P(\bar{A}) = 1 - P(A)$

## **Wahrscheinlichkeit des unmöglichen Ereignisses:**

- $P(\emptyset) = 0$

## **Wertebereich der Wahrscheinlichkeit:**

- $0 \leq P(A) \leq 1$

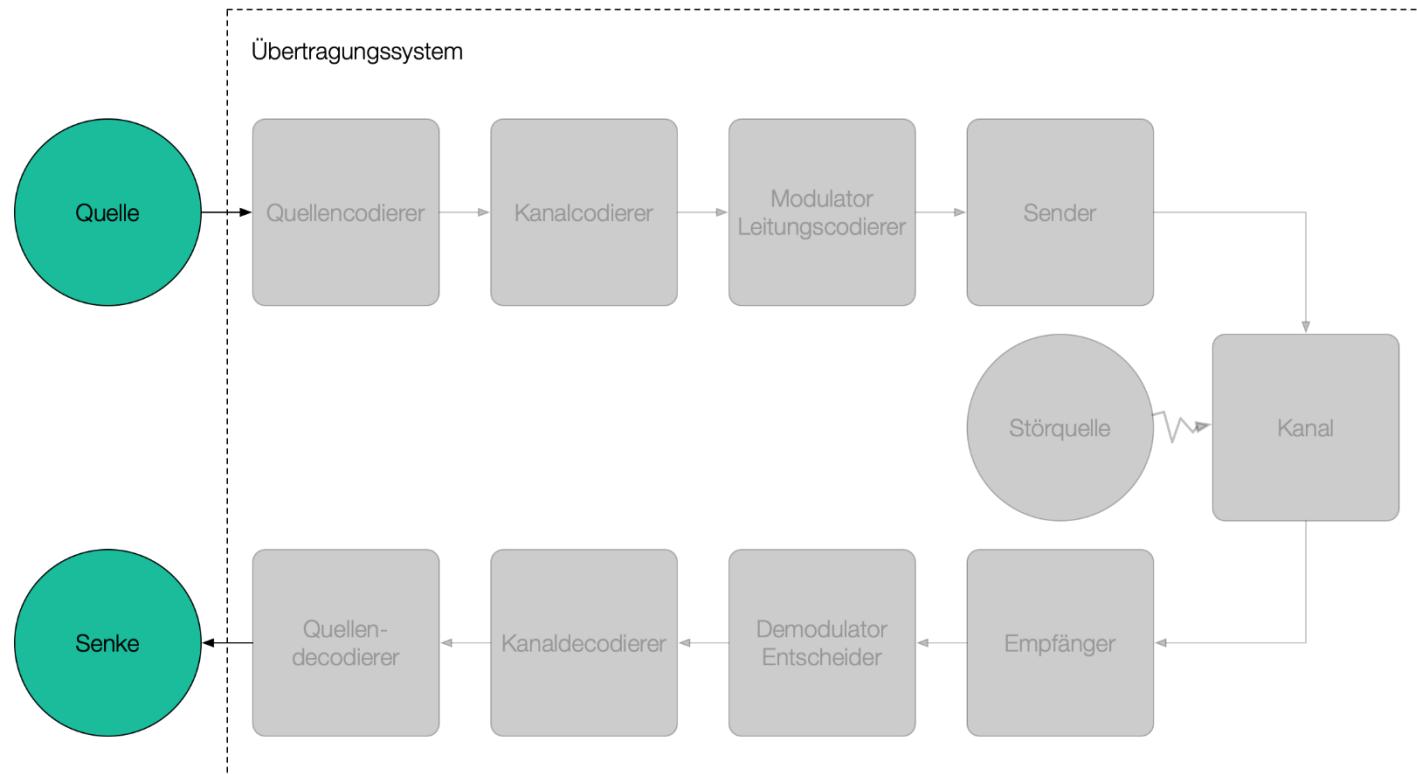
## **Additionssatz für Wahrscheinlichkeiten:**

- $P(A \cup B) = P(A) + P(B) - P(A \cap B)$

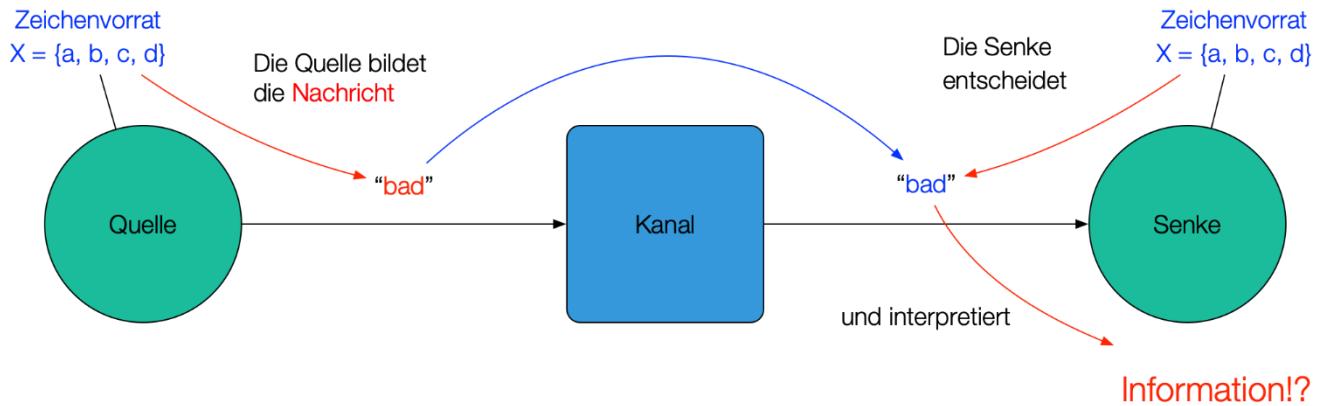
## **Additionssatz für 3 Ereignisse (Wahrscheinlichkeit, dass A, B oder C eintritt):**

- $P(A \cup B \cup C) = P(A) + P(B) + P(C) - P(A \cap B) - P(A \cap C) - P(C \cap B) + P(A \cap B \cap C)$

# Modell der Informationsverarbeitung



# Information

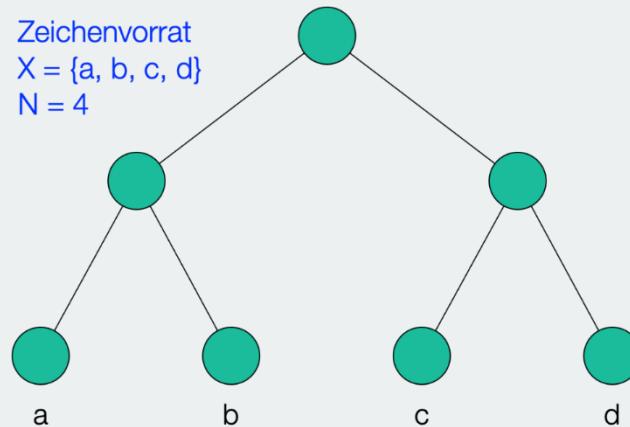


Nachricht (Darstellung & Bedeutung)	redundant	nicht-redundant
irrelevant	Zeichenvorrat bei Quelle und Senke verschieden	
relevant	vorhersagbar	Information

## Definition: (Entscheidungsgehalt)

**Mass für den Aufwand, der zur Bildung einer Nachricht bzw. für die Entscheidung einer Nachricht notwendig ist, ist der Entscheidungsgehalt**

$$H_0 = \log_2(N) \text{ [bit]}$$



**Definition:** (*Entscheidungsfluss*)

Der *Entscheidungsfluss* ist definiert als

$$H_0^* = \frac{\log_2(N)}{\tau} \left[ \frac{\text{bit}}{\text{s}} \right]$$

wobei  $\tau$  die Zeit ist, die zur Übertragung eines Quellzeichens benötigt wird.

# Informationsgehalt

## Definition: (*Informationsgehalt*)

Der *Informationsgehalt* eines Zeichens sagt aus, wie viele Elementarentscheidungen zur Bestimmung dieses Zeichens zu treffen sind.

$$I(x_k) = \log_2 \left( \frac{1}{p(x_k)} \right) [\text{bit}]$$

## Definition: (Informationsgehalt)

Der *Informationsgehalt* eines Zeichens sagt aus, wie viele Elementarentscheidungen zur Bestimmung dieses Zeichens zu treffen sind.

$$I(x_k) = \log_2 \left( \frac{1}{p(x_k)} \right) [\text{bit}]$$

# Entropie

## Definition: (Entropie)

Die **Entropie** bezeichnet den **mittleren Informationsgehalt** der Quelle. Sie zeigt also auf, wie viele Elementarentscheidungen die Quelle/Senke im Mittel pro Zeichen treffen muss

$$H(X) = \sum_{k=1}^N p(x_k) * I(x_k) = \sum_{k=1}^N p(x_k) * \log_2\left(\frac{1}{p(x_k)}\right) \text{ [bit/Zeichen]}$$

**Wann wird der mittlere Informationsgehalt, die Entropie, einer Quelle/Senke maximal? (Ansatz: Binäre Quelle!)**

$$X = \{x_1, x_2\}$$

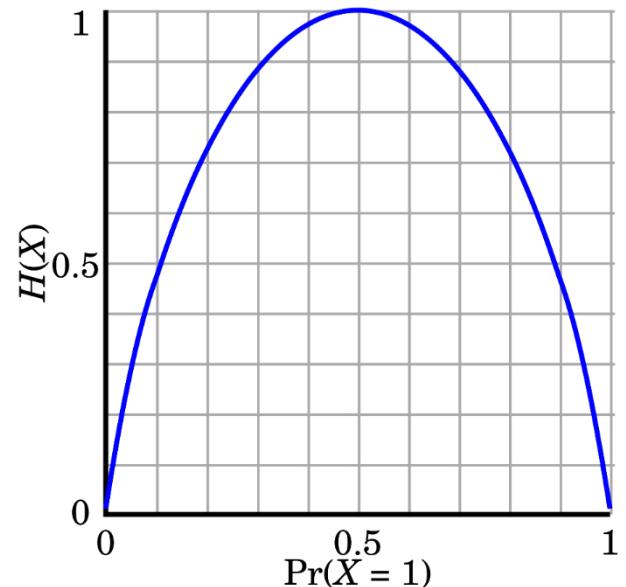
$$p(x_1) = p$$

$$p(x_2) = 1 - p$$

$$\Rightarrow H(X) = p * \log_2 \left( \frac{1}{p} \right) + (1 - p) * \log_2 \left( \frac{1}{1 - p} \right)$$

**Redundanz der Quelle:**

$$R_Q = H_0 - H(X) \text{ [bit/Zeichen]}$$



# Zusammenhang Informationsgehalt und Codewortlänge

**Es gilt: Informationsgehalt eines Zeichens ist:**

$$I(x_k) = \log_2 \left( \frac{1}{p(x_k)} \right) \text{ [bit]}, \text{ kann eine reelle Zahl sein!}$$

**Damit folgt für die Entropie als Mittelwert des Informationsgehalt:**

$$H(X) = \sum_{k=1}^N p(x_k) * I(x_k) = \sum_{k=1}^N p(x_k) * \log_2 \left( \frac{1}{p(x_k)} \right) \left[ \frac{\text{bit}}{\text{Zeichen}} \right], \text{ kann eine reelle Zahl sein!}$$

**Die tatsächliche Codewortlänge L:**

$$L(x_k) = \text{Aufgerundet} \left( \log_2 \left( \frac{1}{p(x_k)} \right) \text{ [bit]} \right), \text{ muss ein Integer sein!}$$

**Damit folgt für die Entropie des Codes als Mittelwert der Codewortlänge:**

$$H_c(X) = \sum_{k=1}^N p(x_k) * L(x_k) \left[ \frac{\text{bit}}{\text{Zeichen}} \right], \text{ kann eine reelle Zahl sein!}$$

## Definition: (Mittlere Codewortlänge)

Die **Mittlere Codewortlänge** berechnet sich wie folgt

$$L = \sum_{i=1}^N p(x_i) \cdot L(x_i) \text{ [bit/Zeichen]}$$

- Bei der Quellencodierung werden die diskreten Zeichen der Quelle auf binäre CW abgebildet.
- Günstig ist, wenn die mittlere Codewortlänge  $L$  möglichst klein ist.
- Beispiele für CW:
  - *ASCII*: Blockcode mit fester Wortgrösse  $L = 8$  [bit] (7 bit ASCII + 0)
  - *Morsecode*: variable Wortgrösse  $L = 1 - 4$  [bit] (*Berücksichtigung der Auftrittswahrscheinlichkeit der einzelnen Zeichen*)

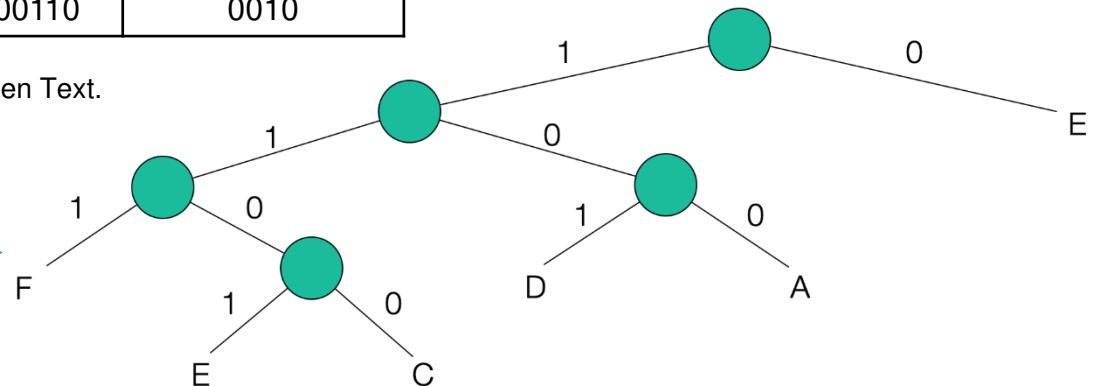
# Binärcodierung: Präfixeigenschaft

Zeichen	Wahrscheinlichkeit*	ASCII Code	Morse Code
A	0.0642	(0)1000001	01
B	0.0127	(0)1000010	1000
C	0.0218	(0)1000011	1010
D	0.0317	(0)1000100	100
E	0.1031	(0)1000101	0
F	0.0208	(0)1000110	0010

Problem?

\*Annahme: Wahrscheinlichkeitsangabe für englischen Text.

Besitzt die Präfixeigenschaft  
bzw.  
ist ein kommafreier Code.



# Shannon'sches Codierungstheorem

1. Für jede beliebige zugehörige Binärcodierung mit Präfixeigenschaft ist die mittlere Codewortlänge nicht kleiner als die Entropie  $H(X)$ :

$$H(X) \leq L$$

2. Für jede beliebige Quelle kann eine Binärcodierung gefunden werden, so dass die folgende Ungleichung gilt:

$$H(X) \leq L \leq H(X) + 1$$

Der Begriff der Redundanz der Quelle wird erweitert um den Begriff der *Redundanz des Codes*:

- *Redundanz der Quelle:*  $R_Q = H_0 - H(X)$  [bit/Zeichen]
- *Redundanz des Codes:*  $R_C = L - H(X)$  [bit/Zeichen]

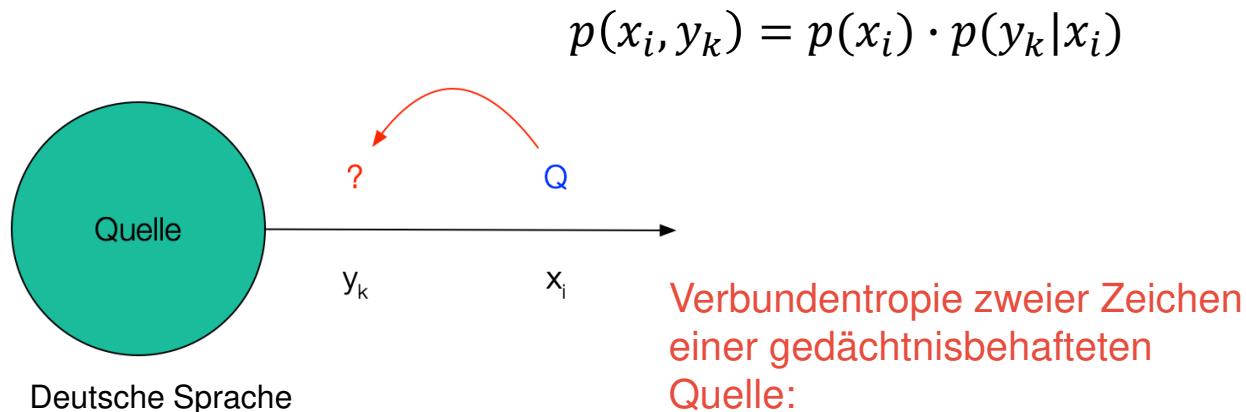
# Diskrete Quellen mit und ohne Gedächtnis

## Bisher: Quelle ohne Gedächtnis

- Die Auftrittswahrscheinlichkeit eines Zeichens ist unabhängig von dem zuvor emittierten Zeichen bzw. der zuvor emittierten Zeichenfolge, d.h. die Verbundwahrscheinlichkeit für die beiden Zeichen  $x_i$  und  $y_k$  ist:

$$p(x_i, y_k) = p(x_i) \cdot p(y_k)$$

- Allgemein kann nicht von einer gedächtnislosen Quelle ausgegangen werden!



$$H(X, Y) = H(X) + H(Y|X)$$

## Diskrete Quellen mit und ohne Gedächtnis

**Es gilt:**  $H(X) = \sum_{i=1}^N p(x_i) \cdot \log_2\left(\frac{1}{p(x_i)}\right)$

**Um  $H(X, Y)$  zu bestimmen**

*setzen wir statt  $p(x_i)$ ,  $p(x_i, y_k) = p(x_i) \cdot p(y_k|x_i)$  in die obige Gleichung ein, es folgt:*

$$H(X, Y) = \sum_{i=1}^N \sum_{k=1}^N p(x_i, y_k) \cdot \log_2\left(\frac{1}{p(x_i, y_k)}\right) =$$

$$\sum_{i=1}^N \sum_{k=1}^N p(x_i) \cdot p(y_k|x_i) \cdot \log_2\left(\frac{1}{p(x_i) \cdot p(y_k|x_i)}\right)$$

## Diskrete Quellen mit und ohne Gedächtnis

$$\sum_{i=1}^N \sum_{k=1}^N p(x_i) \cdot p(y_k|x_i) \cdot \log_2 \left( \frac{1}{p(x_i) \cdot p(y_k|x_i)} \right)$$

$$H(X, Y) = \left[ \sum_{i=1}^N \sum_{k=1}^N p(x_i) \cdot p(y_k|x_i) \cdot \log_2 \left( \frac{1}{p(x_i)} \right) \right] + \left[ \sum_{i=1}^N \sum_{k=1}^N p(x_i) \cdot p(y_k|x_i) \cdot \log_2 \left( \frac{1}{p(y_k|x_i)} \right) \right]$$
$$= H(X) + H(Y|X)$$

Es kann gezeigt werden, dass gilt:

$$H_0 \geq H(Y) \geq H(Y|X)$$

Für die Redundanz galt:

$$R_Q = H_0 - H_{oG}(X) \leq H_0 - H_{mG}(X) = H_0 - (H(Y|X))$$

## Interpretation:

- Die mittlere Entropie einer Quelle ohne Gedächtnis ist stets grösser oder gleich der Entropie einer Quelle mit Gedächtnis.

$$R_Q = H_0 - H_{oG}(X) \leq H_0 - H_{mG}(X)$$

- In der Quellencodierung sind daher nicht Einzelzeichen zu codieren, sondern stets Zeichenketten.
- Beispiel der Redundanz der deutschen Sprache:

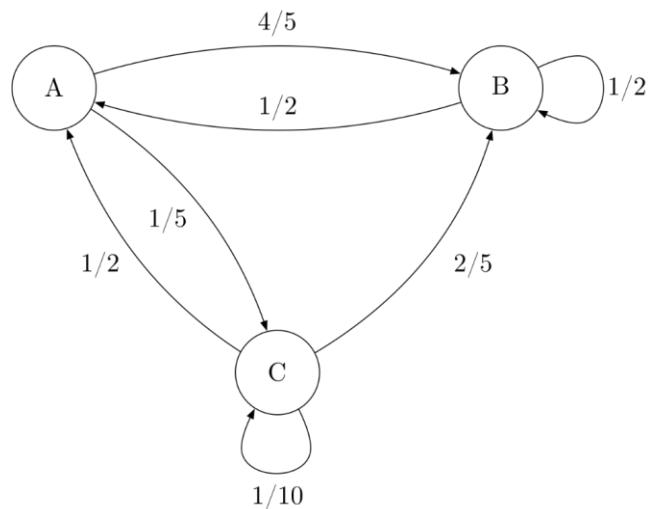
- $H_0 = 4.7 \text{ [bit/Zeichen]}$

- Entropie der Einzelzeichen  $H = 4.097 \text{ [bit/Zeichen]}$   $\rightarrow R = 0.6 \text{ [bit/Zeichen]}$

- Entropie bei Ausnutzung aller Abhängigkeiten  $H = 1.6 \text{ [bit/Zeichen]}$   $\rightarrow R = 3.1 \text{ [bit/Zeichen]}$

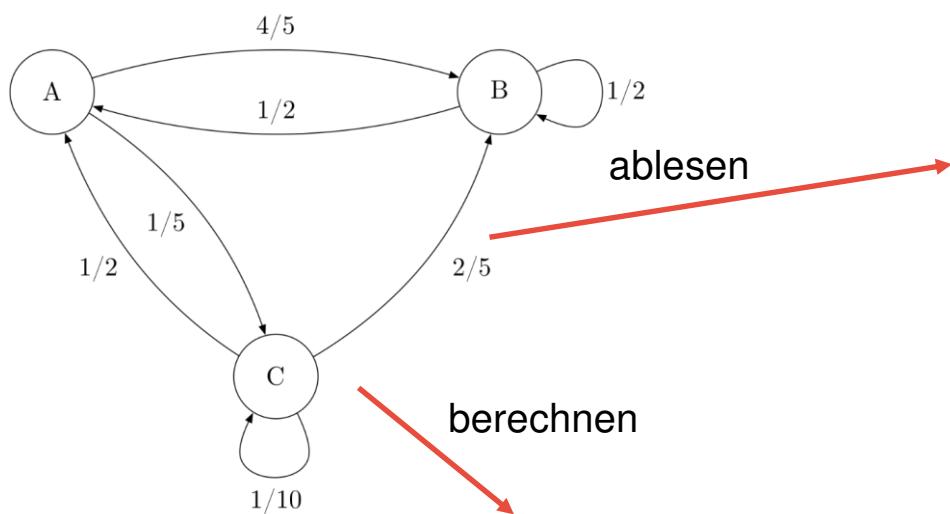
## Beispiel: Diskrete Quelle mit Gedächtnis

**Gegeben sei eine Quelle mit dem Alphabet  $A, B, C$ . Die Abhängigkeiten werden durch ein Markov-Diagramm 1. Ordnung beschrieben.**



**Zur Berechnung der Entropien  $H$  mit  $H(X, Y)$  werden die Wahrscheinlichkeiten  $p(x)$ ,  $p(y|x)$  und  $p(x, y)$  benötigt.**

# Beispiel: Diskrete Quelle mit Gedächtnis



$p(y x)$		$y =$		
		A	B	C
$x =$	A	0	$\frac{4}{5}$	$\frac{1}{5}$
	B	$\frac{1}{2}$	$\frac{1}{2}$	0
	C	$\frac{1}{2}$	$\frac{2}{5}$	$\frac{1}{10}$

berechnen

$x_i =$	$p(x)$
A	$\frac{9}{27}$
B	$\frac{16}{27}$
C	$\frac{2}{27}$

berechnen

$p(x,y)$		$y =$		
		A	B	C
$x =$	A	0	$\frac{4}{15}$	$\frac{1}{15}$
	B	$\frac{8}{27}$	$\frac{8}{27}$	0
	C	$\frac{1}{27}$	$\frac{4}{135}$	$\frac{1}{135}$

berechnen

$$H(X, Y) = \sum_{i=1}^N \sum_{k=1}^N p(x_i, y_k) \cdot \log_2 \left( \frac{1}{p(x_i, y_k)} \right)$$



**OST**  
Ostschweizer  
Fachhochschule

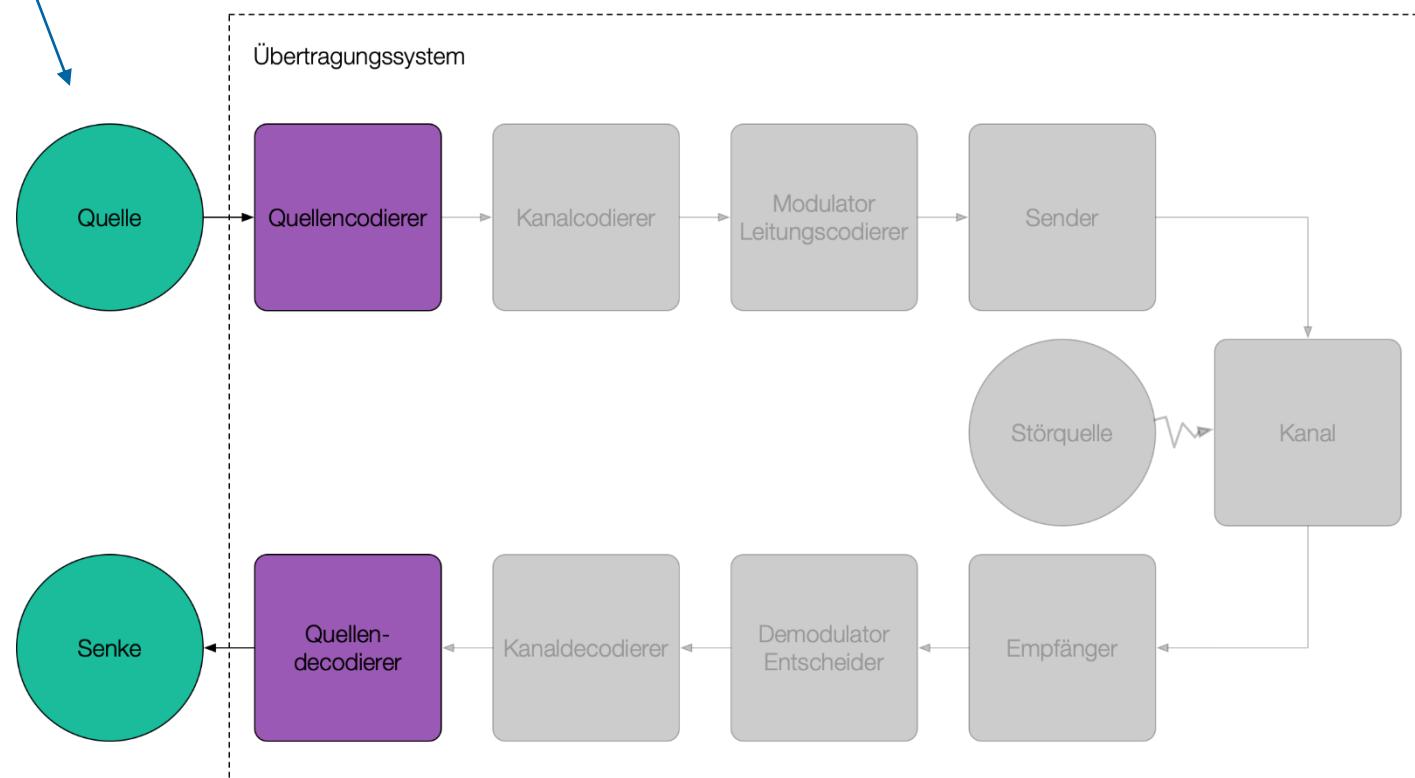
# ***Informations- und Codierungstheorie***

## ***Teil 2. Quellencodierung und Komprimierung***

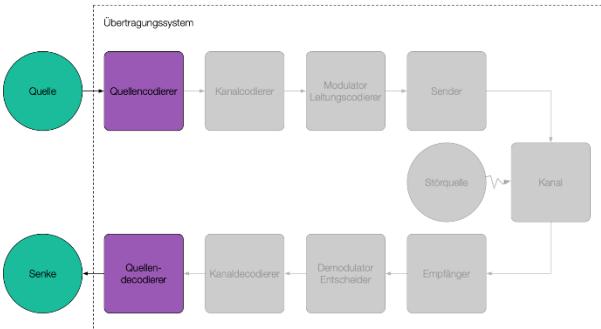
- Prof. Dr.-Ing. Andreas Rinkel  
[andreas.rinkel@hsr.ch](mailto:andreas.rinkel@hsr.ch)
- Tel.: +41 (0) 55 2224461
- Mobil: +41 (0) 79 3320562
- Teams

# Modell der Informationsverarbeitung

- Hat Eigenschaften
- Das sind:
- Informationsgehalt → erhalten
  - Entropie → maximieren
  - Redundanz → entfernen
  - Präfixeigenschaft



# Beispiele zur Quellencodierung



## Quellencodierung

### Datenkomprimierung

- Verlustfrei
- Verlustbehaftet

- ### Verschlüsselung
- Symmetrisch
  - Asymmetrisch

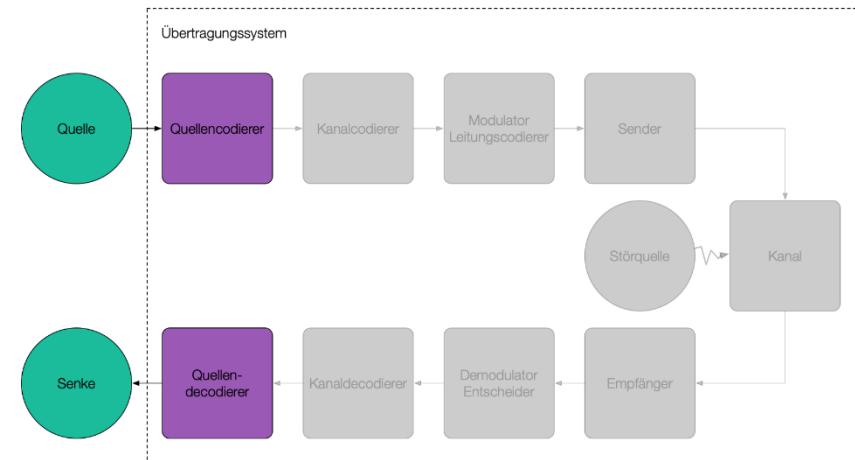
Zusätzlich Literatur Verschlüsselungsverfahren:  
Einführung unter:  
<http://andreas-romecke.de/Projekt1/ffbr/web.html>

# Beispiele zur Quellencodierung

## Verlustfreie Datenkomprimierung

Anforderungen:

- hohe Komprimierungsrate  
für alle Typen von Daten (idealerweise ohne Kenntnis über die Eigenart der Daten )
- hohe Encode- und Decode-Geschwindigkeit
- geringe Ansprüche an die Hardware



**Das Ziel der Datenkomprimierung ist, den Aufwand der Datenspeicherung und Datenübertragung zu reduzieren.**

**D.h. Entfernen von Redundanz und Irrelevanz**

## Verfahren zur Datenkomprimierung

**Statische Verfahren**  
z.B. Huffmann-Codierung für die deutsche Sprache

**Adaptive Verfahren**  
z.B. Huffmann-Codierung mit gemessener Häufigkeitsverteilung

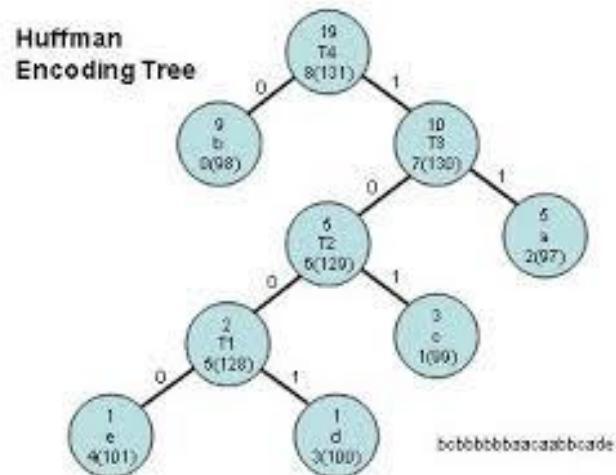
**Dynamische Verfahren**  
z.B. ITU Standard V42.bis basiert auf LZ77 (Lempel, Ziv)

Eigenart der Daten werden berücksichtigt

Eigenart der Daten werden **nicht** berücksichtigt

# Datenkomprimierung nach Huffman

- Entfernen der Redundanz der Quelle



### Verfahren zur Entwicklung eines kommafreien Codes mit minimaler mittlerer Codewortlänge

**Rekursives Verfahren, d.h. der Binärbaum wird nicht von der Wurzel, sondern von den Blättern aus entwickelt**

**Verfahren:**

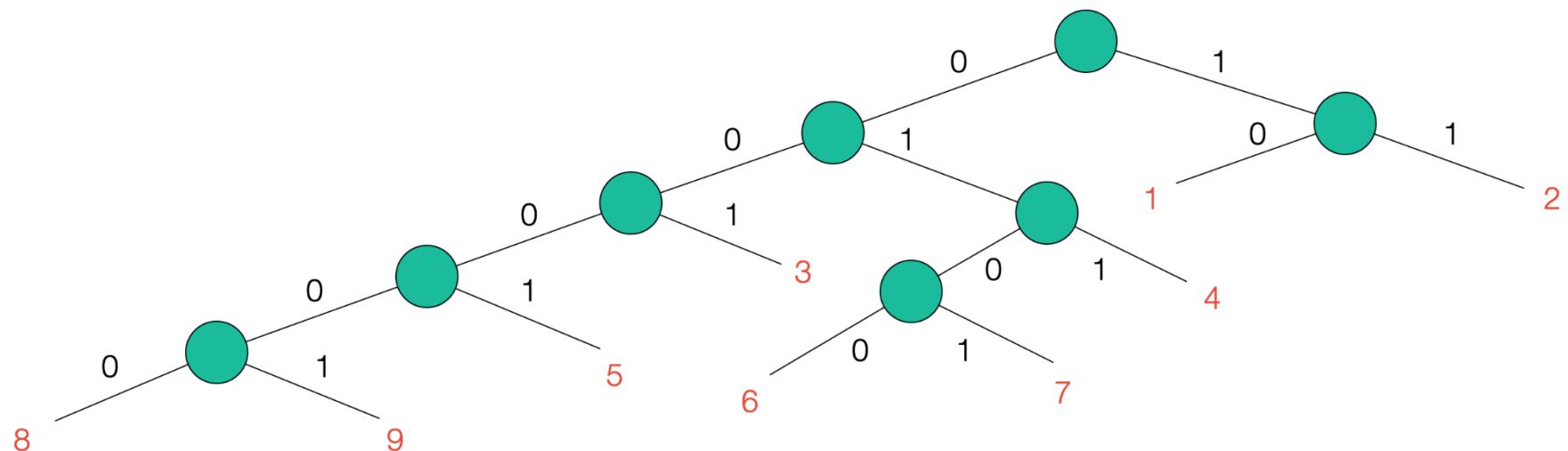
- Ordne die Zeichen gemäss ihrer Auftrittswahrscheinlichkeit
- Die beiden Zeichen mit der kleinsten Auftrittswahrscheinlichkeit haben die gleiche CW-Länge  $L_N$
- Sei  $L_N$  die mittlere CW-Länge für eine Quelle mit  $N$  Zeichen und  $L_{N-1}$  die mittlere CW-Länge für den Fall, dass die beiden letzten zu einem einzigen Zeichen zusammengefasst werden, dann gilt:

$$L_N - (p(x_{N-1}) + p(x_N)) \cdot L(x_N) = L_{N-1} - (p(x_{N-1}) + p(x_N)) \cdot (L(x_N) - 1)$$

$$\Rightarrow L_N = L_{N-1} + p(x_{N-1}) + p(x_N)$$

## Beispiel: Huffman-Codierung

$x_i$	1	2	3	4	5	6	7	8	9
$p(x_i)$	0.22	0.19	0.15	0.12	0.08	0.07	0.07	0.06	0.04



## Huffman-Code: Animation

$x_i$	1	2	3	4	5	6	7	8	9
$p(x_i)$	0.22	0.19	0.15	0.12	0.08	0.07	0.07	0.06	0.04

$x_i$	1	2	3	4	8	9	5	6	7
Code					0	1			
$p(x_i)$	0.22	0.19	0.15	0.12	0.1		0.08	0.07	0.07

$x_i$	1	2	3	6	7	4	8	9	5
Code				0	1		0	1	
$p(x_i)$	0.22	0.19	0.15	0.14		0.12	0.1		0.08

## Huffman-Code: Animation

$x_i$	1	2	8	9	5	3	6	7	4
Code			00	01	1		0	1	
$p(x_i)$	0.22	0.19		0.18		0.15		0.14	0.12

$x_i$	6	7	4	1	2	8	9	5	3
Code	00	01	1			00	01	1	
$p(x_i)$		0.26		0.22	0.19		0.18		0.15

$x_i$	8	9	5	3	6	7	4	1	2
Code	000	001	01	1	00	01	1		
$p(x_i)$		0.33				0.26		0.22	0.19

## Huffman-Code: Animation

$x_i$	1	2	8	9	5	3	6	7	4
Code	0	1	000	001	01	1	00	01	1
$p(x_i)$	0.41			0.33			0.26		

$x_i$	8	9	5	3	6	7	4	1	2
Code	0000	0001	001	01	100	101	11	0	1
$p(x_i)$				0.59				0.41	

$x_i$	8	9	5	3	6	7	4	1	2
Code	00000	00001	0001	001	0100	0101	011	10	11
$p(x_i)$					1.0				

# Einfache Lauflängenkomprimierung

## ■ Erkennen von Wiederholungen

- RLE (*Run Length Encoding*), oder
  - RLC genannt (*Run Length Coding*),
  - wird bei vielen Bildformaten benutzt zum Beispiel [BMP](#), PCX und TIFF).
  - Diese Komprimierungsmethode beruht auf der Verkürzung von Wiederholungen aufeinanderfolgender Elemente.
- 
- Quelltext w: Agggbbehfffgggg =>  $|w|=15$
  - Codiert  $w_c$  : A3g2beh3f4g =>  $|w_c|=11$

# Lauflängenkomprimierung: Bit-Folgen

- Bei der Kodierung von Bitfolgen existieren nur zwei Möglichkeiten:
  - Eine Folge von Nullen oder
  - eine Folge von Einsen.
- Auf jede Sequenz von Nullen folgt garantiert mindestens eine Eins – und umgekehrt ebenfalls.
- Ausnahme ist, wenn das Ende der Nachricht erreicht ist. eof
- Der Kodierer einigt sich nun mit dem Dekodierer darauf, mit welchem Bit begonnen wird, "0" ODER "1"
  - Das kann entweder durch Konvention sein oder
  - bspw. durch ein zusätzliches Bit zu Beginn.
- Anschließend werden abwechselnd die Längen der Null- und Eins-Folgen übertragen.
- Der Dekodierer muss anschließend nichts anderes tun, als zu jedem empfangenen Wert entsprechend viele Null- oder Eins-Bits auszugeben

## Lauflängenkomprimierung: Bitfolgen

- Beispiel:
- Orginalnachricht:  $w = 1111\ 1110\ 0000\ 1000\ 0001\ 1111 \Rightarrow |w| = 24 \text{ bit}$
- Start mit einer "1"

Code: 7 5 1 6 5 :

um jede Stelle von 0 .. 7 zu codieren reichen 3 Bit je Stelle aus

Das Codewort wird zu:  $w_c = 111\ 101\ 001\ 110\ 101 \Rightarrow |w_c| = 15$

# Datenkomprimierung nach Lempel Ziv

## ■ Erkennen von wiederkehrenden Mustern

*Die Ziv-Lempel Verfahren sind, in der Gruppe der verlustfrei packenden Algorithmen, die, die diesem Ideal noch am nächsten kommen. Die ersten Verfahren von Jacob Ziv und Abraham Lempel sollen in dieser Arbeit vorgestellt werden. Die Grundidee all dieser Verfahren ist es eine Zeichenkette, die schon einmal gesendet/gespeichert wurde und nun wieder auftaucht, durch einen Zeiger auf die Stelle wo sie eher auftauchte oder durch einen Zeiger auf ein Wörterbuch zu ersetzen. Damit soll die Länge der zu codierenden Zeichenkette beträchtlich gekürzt werden.*

André Lichei, TU-Chemnitz

# Datenkomprimierung nach Lempel Ziv

## ■ Erkennen von wiederkehrenden Mustern

LZ77 ist die Abkürzung für das Komprimierungsverfahren, welches Jacob Ziv und Abraham Lempel 1977 in dem Artikel "A Universal Algorithm for Sequential Data Compression" in "IEEE Transactions on Information Theory" vorgestellt haben

Der LZ77 Algorithmus war das erste vorgestellte tabellengesteuerte Kompressionsverfahren.

Es komprimierte dadurch, dass zuvor eingelesener Text als Tabelle genutzt wird.

Phrasen aus dem Eingabetext werden durch Zeiger in der Tabelle ersetzt. Dadurch wird die Komprimierung erreicht.

Der Grad der Komprimierung hängt von der Länge der Phrasen, Fenstergröße und Entropie des Ursprungstextes (bezüglich LZ77) ab. Bei gleichen nah beieinander liegenden Textsequenzen kommt es sehr schnell zur Kompression

# Lempel-Ziv Encoder: Grundüberlegung

- **der zu komprimierende Code hat wiederkehrende Muster oder Phrasen**
- **anstatt den Code vollständig zu übertragen, werden „nur“ die codierten Phrasen übertragen**
- **Dazu müssen die Phrasen**
  - zur Laufzeit erfasst und
  - **in einem Phrasenspeicher oder Wörterbuch gespeichert und codiert werden**
  - **die Grösse des Wörterbuchs und des „look ahead buffers“ muss bestimmt werden**
- **LZ77 wurde 1977 von Jacob Ziv und Abraham Lempel entwickelt**
- **Problem: „Effiziente Umsetzung des Phrasenspeichers?“**

## Lempel-Ziv Encoder: Umsetzung

- **Während des Durchlaufens der Daten wird ein ständig wachsender Baum erzeugt.**
- **Der Baum dient als Wörterbuch und zeigt Regularitäten auf.**
- **Die Knoten dienen als Referenzen.**
- **Werden gleiche Subdaten wiederholt geparsst, so kann auf den entsprechenden Knoten des Wörterbuches referenziert werden.**
- **LZ77 wurde 1977 von Jacob Ziv und Abraham Lempel entwickelt.**

# Lempel-Ziv Encoder: Umsetzung

- **Die Datenstruktur besteht aus**
- **einem Textfenster, dem search buffer**
  - hier stehen die schon kodierten Symbole
- **einem nach vorn gerichteten Puffer look-ahead buffer ,**
  - Der Puffer zeigt auf die als nächstes zu kodierenden Symbole.
- **Sollte eine Sequenz von Symbolen in dem Puffer und dem Textfenster übereinstimmen, so wird ein Kode bestehend aus Position und Länge im Textfenster gebildet und abgespeichert.**
- **Ansonsten wird der Kode so gespeichert. wie er vorlag.**
- **Anschließend schieben sich beide Puffer eine Position nach vorn. deshalb wird diese Methode auch die Methode der gleitenden Fenster genannt.**

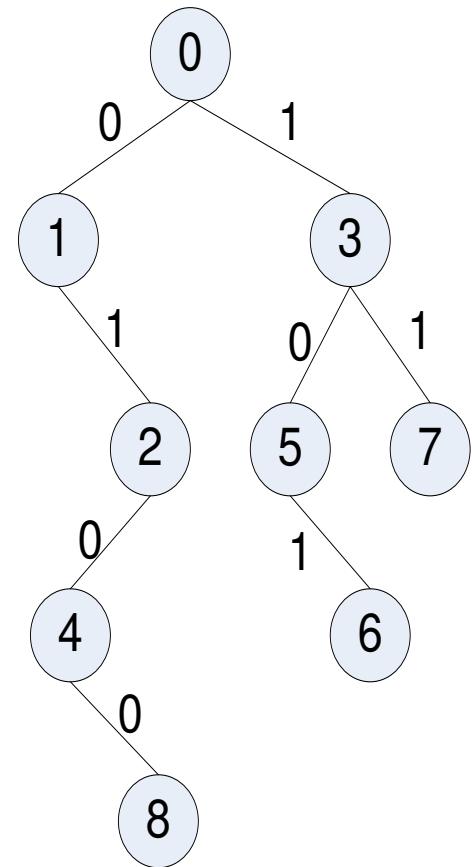
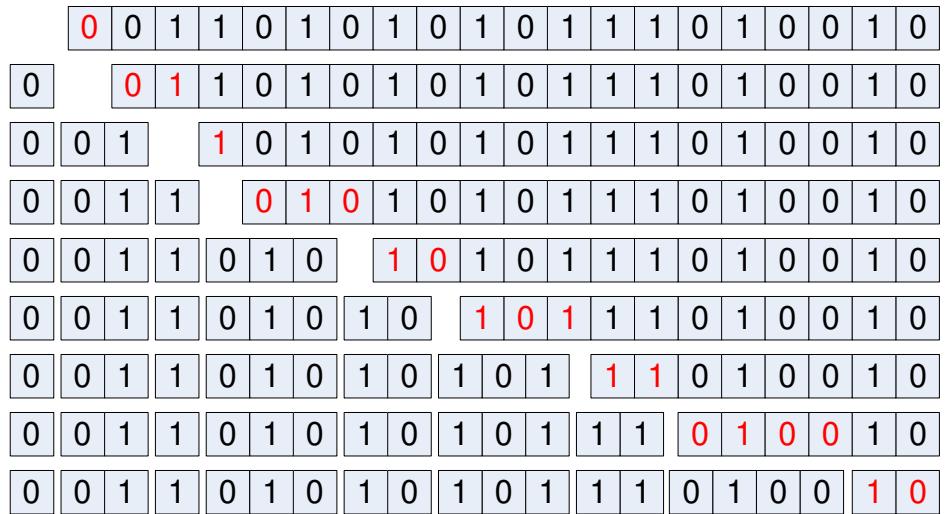
# Lempel-Ziv Encoder: Beispiel

search buffer	Look-ahead buffer	coding
	sir_sid_eastman	(0,0,"s")
s	ir_sid_eastman	(0,0,"i")
si	r_sid_eastman	(0,0,"r")
sir	_sid_eastman	(0,0,"_")
sir_	sid_eastman	(4,2,"d")
sir_sid	_eastman	(4,1, "e)

Codierung:

- suche in der Tabelle eine möglichst lange Zeichenfolge, die mit den nächsten n zu codierenden Zeichen übereinstimmt
- bilde ein Token und speichere es
- verschiebe das Fenster um (n+1) Zeichen
- wiederhole, bis alle Zeichen codiert sind

# Beispiel Lempel-Ziv Encoder I



"0" "01" "1" "010" "10" "101" "11" "0100"

(0,0) (1,1) (0,1) (2,0) (3,0) (5,1) (3,1) (4,0) (5, eof)

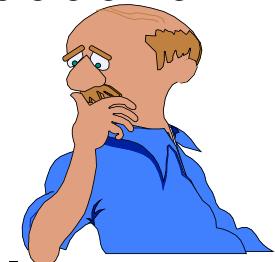
## Beispiel Lempel-Ziv Encoder II

(0,0) (1,1) (0,1) (2,0) (3,0) (5,1) (3,1) (4,0) 5

System: Erste Komponente binär + Zweite Komponente binär  
oder fehlt = eof

## Komprimierung?

- **Komprimierter Text:** 0110010100110101101110000101
- **Original:** 00110101010111010010
- **Voraussetzung: Text muss Regelmässigkeit beinhalten.**
- **Am effizientesten, wenn sich lange Pfade entwickeln, z.B.  
{00000}**





**OST**  
Ostschweizer  
Fachhochschule

# ***Informations- und Codierungstheorie***

## ***Teil 2. Quellencodierung und Verschlüsselung***

- Prof. Dr.-Ing. Andreas Rinkel  
[andreas.rinkel@hsr.ch](mailto:andreas.rinkel@hsr.ch)
- Tel.: +41 (0) 55 2224461
- Mobil: +41 (0) 79 3320562
- Teams

- Altgriechisch: krypto für verborgen, geheim und grafie für schreiben, Schrift
- Die heutige Veranstaltung gibt nur einen qualitativen und sehr unvollständigen Überblick gängiger Verfahren
- Ein bisschen genauer werden wir die Grundlagen des RSA-Verfahrens anschauen
- Einen guten Überblick und mehr finden sie auf der Seite "kryptografie.de", ein Teil der hier vorgestellten Inhalte basieren auf Inhalten dieser Seite
- Eine vertiefte praktische Anwendung der Verfahren erhalten sie in den Security-Modulen der OST

## ▪ **Symmetrische Verfahren: eine grobe Übersicht**

- Ein einfaches Substitutionsverfahren: Der Caesar Chiffre
- Transpositionsverfahren
- Vigenère-Chiffre
- DES

- In allen Fällen benötigt man zum Ver- und Entschlüssel den gleichen Schlüssel.
- Daraus folgt: wollen 100 Mitglieder paarweise geheime Botschaften austauschen, muss für jedes Paar ein eigener Schlüssel erstellt werden.
- Anzahl der erforderlichen Schlüssel ist dann:  $N = \binom{100}{2} = \frac{100*99}{2} = 4950$
- Ein grosses Problem: das Schlüsselmanagement

# Substitutionsverfahren

- Das einfachste und bekannteste Verfahren ist der sogenannte **Caesar Chiffre**.
- Der Schlüssel ist im gegebenen Beispiel gleich 4. Das heist, das Chiffrieralphabet wird um 4 Zeichen verschoben wird.
- Symmetrisches Verfahren



Schlüssel  $k = 4$

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z

Klartext: bald ist weihnachten

Schlüssel  $k = 4$

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z

Chiffretext: feph mwx aimlreglxir

# Substitutionsverfahren

- Auf den ersten Blick sieht das Verfahren ja interessant aus, "feph mwx aimlreglxir" ist schliesslich nicht lesbar!

Aber:

- die statistischen Eigenschaften von Klar- und Chiffriertext sind nach wie vor unverändert.
  - Kennen wir die Sprache und ist unsere Probe gross genug, können wir den Schlüssel leicht ermitteln
  - auch ist die Schlüsselanzahl recht übersichtlich, hier braucht es keinen Computer, um alle auszuprobieren
- Was leiten Sie daraus für die Entwicklung von Chiffrieralgorithmen

# Transpositionsverfahren

Beim **Transpositionverfahren** werden nach gegebenen Regeln die Zeichenfolge des Klartextes «verwürfelt», d.h. es findet keine Ersetzung (Substitution) der Zeichen statt.

Ein einfaches Beispiel wird unten gezeigt:

Klartext:

DIE WORTE HOER ICH WOHL  
ALLEIN MIR FEHLT DER  
GLAUBE

Chiffretext:

DTILNHGIECAMLLEHHLITAW  
OWLRDUOEEOFEBRRHIERE

Erstellen einer Tabelle  
zeilenweise

Auslesen  
spaltenweise

Hier sind  
Permutationen  
der Spalten  
möglich!

D	I	E	W	O	R
T	E	H	O	E	R
I	C	H	W	O	H
L	A	L	L	E	I
N	M	I	R	F	E
H	L	T	D	E	R
G	L	A	U	B	E

- In der ersten Hälfte des 16ten Jahrhunderts entwickeltes polyalphabetisches Verfahren entwickeltes Verfahren
- Das Verfahren konnte erst 300 Jahre später von Charles Babbage geknackt werden
- Der erste der Verfahren zum Angriff auf den Code beschrieb war Friedrich Wilhelm Kasiski
- Weitere Infos auch unter: <https://de.wikipedia.org/wiki/Vigen%C3%A8re-Chiffre>

A	S	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

▪ **Der Algorithmus (Quelle:**

<http://kryptografie.de/kryptografie/chiffre/vigenere.htm>)

Klassischerweise gibt es eine feste Verschlüsselungstabelle, auch Tabula recta genannt, die zum händischen Chiffrieren benutzt wird.

Sie funktioniert so: man sucht den Klartextbuchstaben links in der 1. Spalte und geht in dieser Zeile soweit nach rechts, bis man in der obersten Zeile den Buchstaben des Schlüssels gefunden hat. Nun kann man dort den Chiffre-Buchstaben ablesen.

Praktikablerweise schreibt man das Chiffrat neben die Tabelle, damit man immer einen Überblick hat, in welcher Zeile man ist.

Mathematisch gesehen kann man aber auch mit Offsets rechnen, die sich aus dem Schlüssel ergeben. Dabei hat A den Wert 0, B den Wert 1 usw. Der Offset wird dem Klartextbuchstaben dann hinzugezählt, um den Chiffratbuchstaben zu erhalten. Ist das Alphabet zu Ende, wird wieder bei A begonnen.

Wenn das Schlüsselwort kürzer als der Klartext ist, wird es mehrmals hintereinander geschrieben.

# DES: Data Encryption Standard

## ■ Der DES-Algorithmus

- wurde als offizieller Standard für die US-Regierung im Jahr 1977 bestätigt und wird seither international vielfach eingesetzt

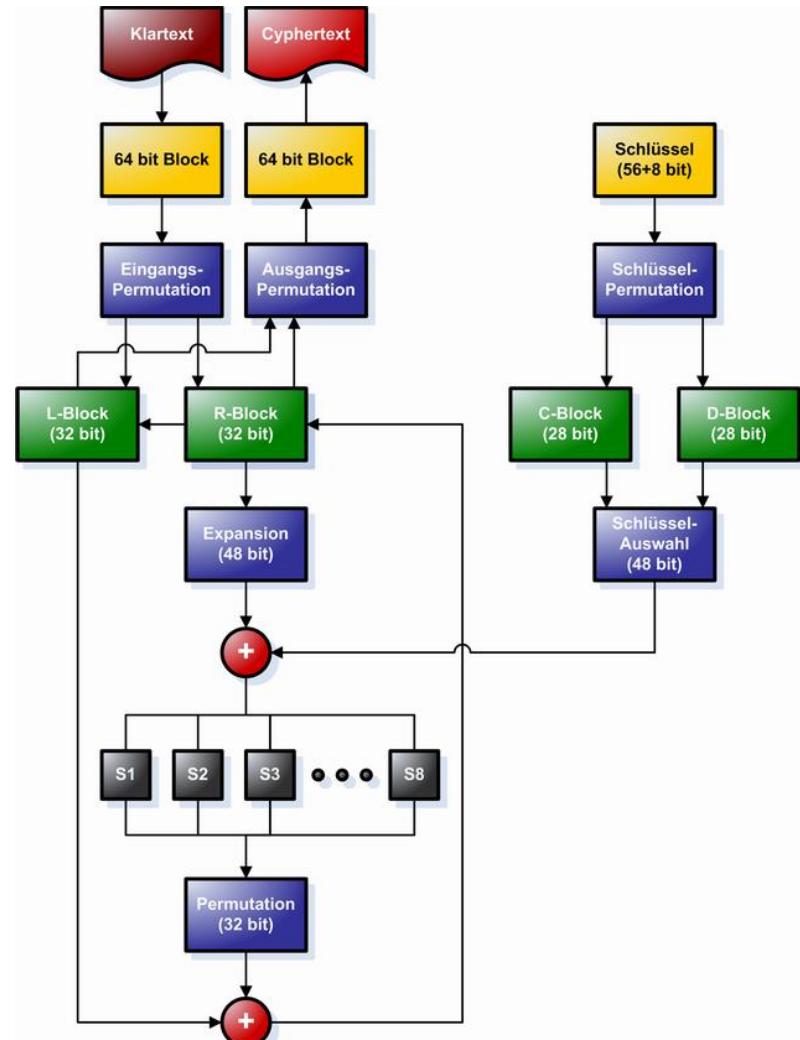
## ■ Seine Entstehungsgeschichte

hat wegen der Beteiligung der [NSA](#) am Design des [Algorithmus](#) immer wieder Anlass zu [Spekulationen](#) über seine Sicherheit gegeben

## ■ Heute

wird DES aufgrund der verwendeten [Schlüssellänge](#) von nur 56 [Bits](#) für viele Anwendungen als nicht ausreichend sicher erachtet.

- Quelle:  
[https://de.wikipedia.org/wiki/Data\\_Encryption\\_Standard](https://de.wikipedia.org/wiki/Data_Encryption_Standard)



## IDEE: Gibt es ein asymmetrisches Kryptoverfahren?

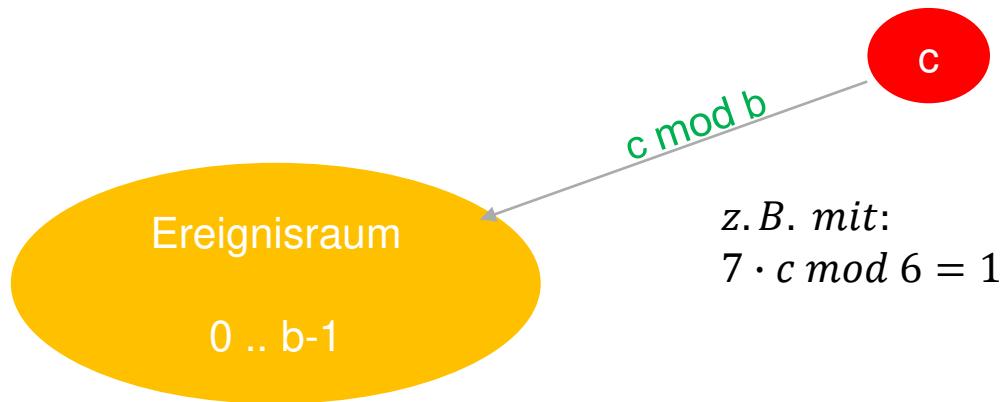
- **Das heisst, lässt sich ein Verfahren finden, dass es erlaubt**
  - einen Schlüssel zur Codierung und
  - einen zweiten Schlüssel zur De-Codierung benutzen?
- **Als Rahmenbedingung ist es weiterhin erforderlich, dass die Kenntnis eines Schlüssels nichts über die Identität des anderen “Partner”-Schlüssels verrät.**

Ursprünglich sind die drei Mathematiker **Rivest, Shamir** und **Adleman (RSA)** angetreten, um zu zeigen, dass das **nicht** möglich ist.

**Aber:** Sie haben gezeigt, dass ein asymmetrisches Kryptoverfahren möglich ist (heute auch bekannt als RSA-Verfahren).

# Inverse Zahlen: Was ist eine inverse Zahl?

- Im Reellen ist  $a^{-1}$  die zu  $a$  inverse Zahl, da die Operation  $a \cdot a^{-1} = a^0 = 1$  ergibt.
- Durch **mod b** wird ein **Ergebnisraum** von  $0 \dots b-1$  möglich, d.h. es lässt sich eine **Zahl c** (ausserhalb des Ergebnisraumes) finden, die in dieser Rechenvorschrift eine inverse Zahl zu  $a$  ist.
- Bemerkenswert ist, dass sich **die Zahl c nicht unmittelbar oder einfach aus der Zahl a ergibt**.
- Ein Ansatz für ein asymmetrisches Verfahren?



Für zwei teilerfremde Zahlen  $a, b$  ( $ggT = 1$ ) existiert eine Zahl  $c$ , so das gilt:

$$a \cdot c \bmod b = 1$$

Beispiel: Sei  $b = 6$  und  $a = 7$  wobei  $a$  und  $b$  teilerfremd sind, dann ergibt die Rechnung

für den Fall, dass  $c = 13$  ist.

$$7 \cdot c \bmod 6 = 1$$

# Eulerfunktion

**Die Eulerfunktion gibt die Anzahl der zu einer Zahl  $n$  teilerfremden Zahlen an, d.h. die Zahlenpaare  $(n, x)$  mit  $x < n$ , die keinen gemeinsamen Teiler haben.**

**$\Phi(n) = \text{Anzahl der relativ primen Zahlen}$**

$$\Phi(18) = 6 \longrightarrow 18 \text{ ist teilerfremd zu } 1, 5, 7, 11, 13, \text{ und } 17$$

**Für Primzahlen  $p$  gilt:**

$$\Phi(p) = p - 1 \longrightarrow \Phi(5) = 4 \quad 5 \text{ ist teilerfremd zu } 1, 2, 3 \text{ und } 4$$

**Für das Produkt der Primzahlen  $p \cdot q$  gilt:**

$$\Phi(p \cdot q) = (p - 1) \cdot (q - 1) \longrightarrow$$

Beispiel:

Mit  $p = 5, \Phi(5) = 4$  und  
 $q = 3, \Phi(3) = 2$  ergibt sich  
 $\Phi(15) = 4 * 2 = 8$

## Satz von Euler

**Satz von Euler:** Für zwei teilerfremde Zahlen  $a$  und  $b$  gilt:  $a^{\Phi(b)} \text{ mod } b = 1$ .

**Beispiel:** Sei  $b = 5$ , dann ist  $a$  für die Werte 1, 2, 3, 4 teilerfremd zu  $b$  oder relativ prim.

- $1^4 \text{ mod } 5 = 1$
  - $2^4 \text{ mod } 5 = 16 \text{ mod } 5 = 1$
  - $3^4 \text{ mod } 5 = 81 \text{ mod } 5 = 1$
  - $4^4 \text{ mod } 5 = 256 \text{ mod } 5 = 1$
  - $5^4 \text{ mod } 5 = 625 \text{ mod } 5 = 0$
  - $6^4 \text{ mod } 5 = 1296 \text{ mod } 5 = 1$
  - ...
  - $10^4 \text{ mod } 5 = 10000 \text{ mod } 5 = 0$
- Solange der Wert  $a$  also kleiner als  $b$  ist, gilt die Aussage immer.

## **Satz von Euler:**

**Für zwei teilerfremde Zahlen  $a$  und  $b$  gilt:  $a^{\Phi(b)} \bmod b = 1$ .**

Für  $b$  können wir jetzt zwei Primzahlen  $p, q$  wählen, dann folgt:

$$a^{\Phi(p \cdot q)} \bmod (p \cdot q) = 1$$

$$a^{(p-1) \cdot (q-1)} \bmod (p \cdot q) = 1$$

Und das gilt in jedem Fall, solange  $a < pq$  gilt!

**Es gilt: nach dem Satz von Euler ist  $b$  das Produkt zweier Primzahlen, dann gilt:**

$$a^{\Phi(b)} \bmod b = 1$$

**Jetzt zeigen wir noch das:**  $a^y \bmod (p \cdot q) = a^y \bmod \Phi(p \cdot q) \bmod (p \cdot q)$

$$\begin{aligned} a^y \bmod (p \cdot q) &\stackrel{!}{=} a^{y \bmod \Phi(p \cdot q)} \bmod (p \cdot q) \\ &= a^{y-n \cdot \Phi(p \cdot q)} \bmod (p \cdot q) \\ &= a^y \cdot a^{-n \cdot \Phi(p \cdot q)} \bmod (p \cdot q) \\ &= [a^y \bmod (p \cdot q)] \cdot \underbrace{[a^{\Phi(p \cdot q)} \bmod (p \cdot q)]^{-n}}_{\text{q.e.d.}} \end{aligned}$$

### Es gilt:

- $a^y \text{ mod } (p \cdot q) = a^{y \text{ mod } \Phi(p \cdot q)} \text{ mod } (p \cdot q)$  (mit  $a$  teilerfremd zu  $p \cdot q$ ).
- Nehmen wir an, wir haben das Codewort  $a$  wobei  $|a| < pq$  ist
- Setzten wir  $y = e^*d$  wobei "e" der encryption- und "d" der decryption Schlüssel sei

### Dann folgt

- $a^{ed} \text{ mod } (p \cdot q) = a^{e^*d \text{ mod } \Phi(p \cdot q)} \text{ mod } (p \cdot q)$
- Wenn wir jetzt noch  $e$  und  $d$  so bestimmen, dass gilt:  $1 = ed \text{ mod } \Phi(p \cdot q)$  gilt, dann
  - Können wir mit  $e$  verschlüsseln  $a > a^e \text{ mod } (pq)$  und später
    - mit  $d$  entschlüsseln  $> a^{ed} \text{ mod } (pq) = a$ , da ja  $e^*d \text{ mod } \Phi(p \cdot q) = 1$  wird!
  - D.h. wir haben jetzt einen **Schlüssel zum Zuschliessen** und
    - einen **Schlüssel zum Aufschliessen**, ein **asymmetrisches Verfahren**

## Beispiel

**Seien die beiden Primzahlen  $p = 47, q = 59$  gegeben.**

**Dann folgt daraus:**  $n = p \cdot q = 2773$  und  $\Phi(n) = 2668$ .

- Wir bestimmen den Wert  $e = 17$ , relativ prim zu  $\Phi(n)$ ,  $1 < e < \Phi(n)$ .
- Wir berechnen den zu  $e$  inversen Wert  $d$  mit 157.

**Encrypt:**

$$\begin{array}{ccc} m & \rightarrow & m^e \\ 12 & & 2218611106\dots \\ & & 336 \end{array} \quad \rightarrow \quad c = m^e \bmod n$$

**Decrypt:**

$$\begin{array}{ccc} c & \rightarrow & c^d \\ 336 & & 4317840049\dots \\ & & 12 \end{array} \quad \rightarrow \quad m = c^d \bmod n$$

**Problem: Wie kann ich den zu  $e$  oder  $d$  inversen Wert berechnen?**

# Euklidischer Algorithmus: Basis zur Berechnug

**Der euklidsche Algorithmus dient zur Bestimmung des grössten gemeinsamen Teilers zweier Zahlen oder Polynome.**

**Sei  $r_0$  und  $r_1$  gegeben, dann ergibt sich der  $ggT(r_n)$  von  $r_0$  und  $r_1$  zu:**

- $r_0 = q_1 \cdot r_1 + r_2$
- $r_1 = q_2 \cdot r_2 + r_3$
- $r_2 = q_3 \cdot r_3 + r_4$
- $r_{n-2} = q_{n-1} \cdot r_{n-1} + r_n$
- $r_{n-1} = q_n \cdot r_n + 0$

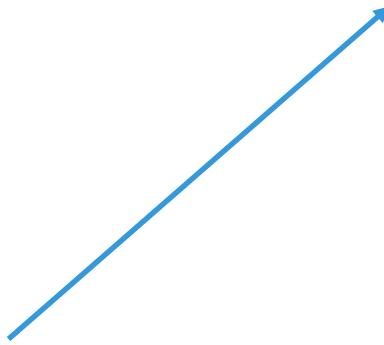
**Beispiele: Wenden Sie das Verfahren an auf die Zahlenpaare (21, 9) und (48, 18) und (19, 13)**

**Gesucht:**  $20 \cdot b \bmod 43 = 1$

$$r_0 = 43 = 2 \cdot 20 + 3$$

$$r_1 = 20 = 6 \cdot 3 + 2$$

$$r_2 = 3 = 1 \cdot 2 + 1$$


$$\begin{aligned} 1 &= 3 - 1 \cdot 2 \\ 1 &= 3 - 1 \cdot (20 - 6 \cdot 3) \\ &\quad = 7 \cdot 3 - 1 \cdot 20 \\ 1 &= 7 \cdot (43 - 2 \cdot 20) - 1 \cdot 20 \\ 1 &= 7 \cdot 43 - 15 \cdot 20 \end{aligned}$$

Gefragt war b das gilt:  $20 \cdot b \bmod 43 = 1$   
also:  $(7 \cdot 43 - 15 \cdot 20) \bmod 43 = 1$

Dabei gilt:

- $(7 \cdot 43 \bmod 43 - 15 \cdot 20 \bmod 43) = 1$
  - $(0 - 15 \cdot 20 \bmod 43) = 1$
- D.h., die zu 20 inverse Zahl heist  $-15 \bmod 43 = 28$ .

# Grosse Zahlen

## Wahrscheinlichkeit

- Für 6 richtige im Lotto:  $7.1 \cdot 10^{-8}$
- Jährlich vom Blitz getroffen zu werden:  $10^{-7}$
- Von einem Meteoriten erschlagen zu werden:  $16 \cdot 10^{-12}$

## Anzahl Atome

- Erde:  $10^{51}$
- Sonne:  $10^{57}$
- Unsere Galaxis:  $10^{67}$
- Im Weltall (ohne dunkle Materie):  $10^{77}$

Ein Computer, der pro Sekunde  $2.000.000.000 = 2 \cdot 10^9$  AES Verschlüsslungen berechnen kann, benötigt zum Durchprobieren aller  $2^{128}$  Schlüssel ca.  $5.3 \cdot 10^{21}$  Jahre!

## Die Zeit bis

- Zur nächsten Eiszeit: 14000 Jahre
- Die Sonne zur Nova wird:  $10^9$  Jahre
- Alter des Universums:  $10^{10}$  Jahre

## Lebensdauer

- Des Weltalls (falls geschlossen)  $10^{15}$  Jahre
- Des Weltalls (falls offen ):  $10^{19}$  Jahre

## Beispiel

Seien die beiden Primzahlen  $p = 11, q = 7$  gegeben.

Dann folgt daraus  $n = p \cdot q = 77$  und  $\Phi(n) = \Phi(10 \cdot 6) = 60$ .

- Wir bestimmen den Wert  $e = 17$  mit  $1 < e < \Phi(n)$ , relativ prim zu  $\Phi(n)$ .

$$60 = 3 \cdot 17 + 9$$

$$17 = 1 \cdot 9 + 8$$

$$9 = 1 \cdot 8 + 1$$

$$1 = 9 - 1 \cdot 8$$

$$1 = 9 - 1 \cdot (17 - 1 \cdot 9)$$

$$1 = 60 - 3 \cdot 17 - 1 \cdot (17 - 1 \cdot (60 - 3 \cdot 17))$$

$$1 = 2 \cdot 60 - 7 \cdot 17$$

Das heisst  $(-7) \bmod 60 = 53$  ist der zu 17 inverse Schlüssel.



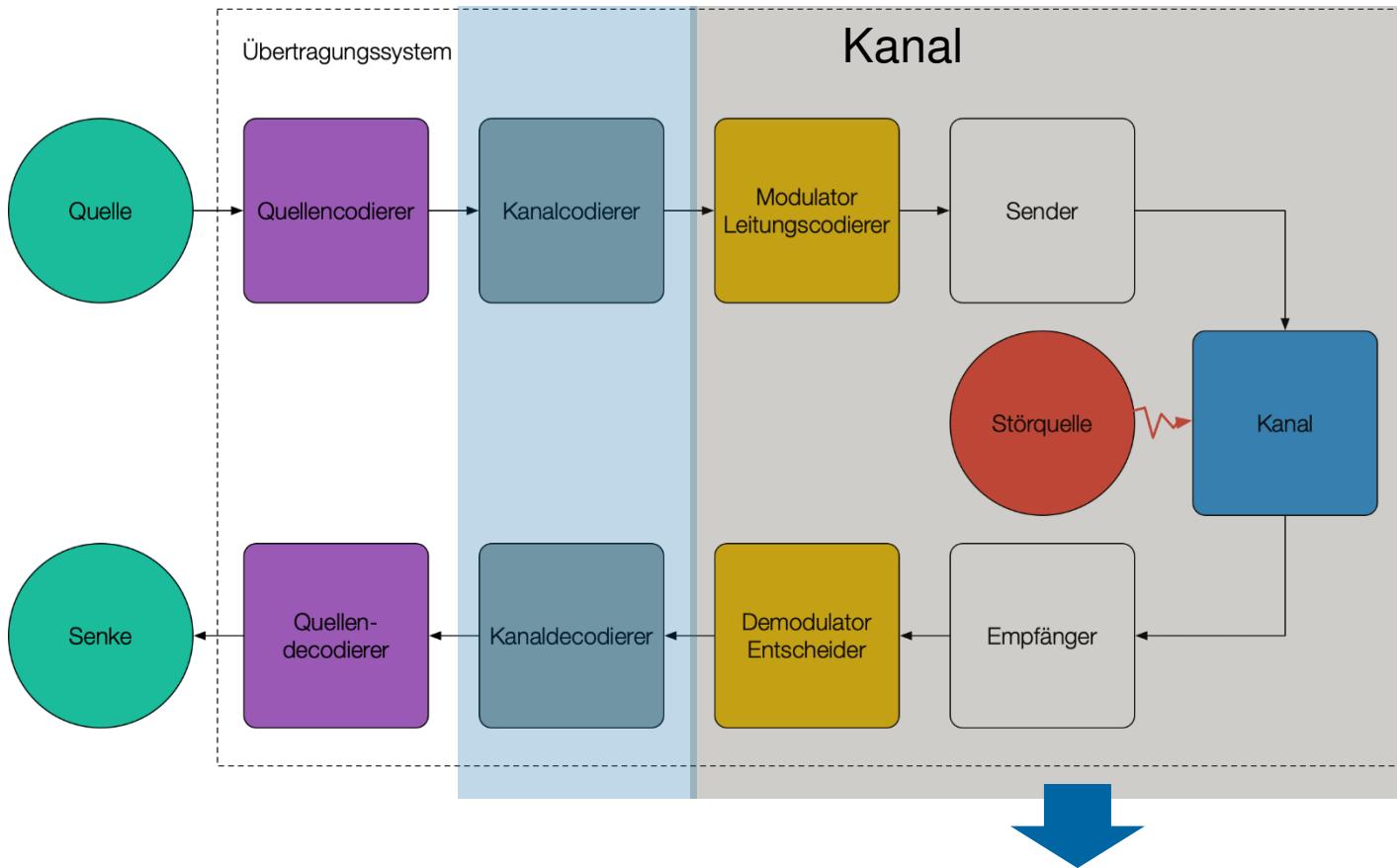
**OST**  
Ostschweizer  
Fachhochschule

# ***Informations- und Codierungstheorie***

## ***Teil 2. Kanalmodell***

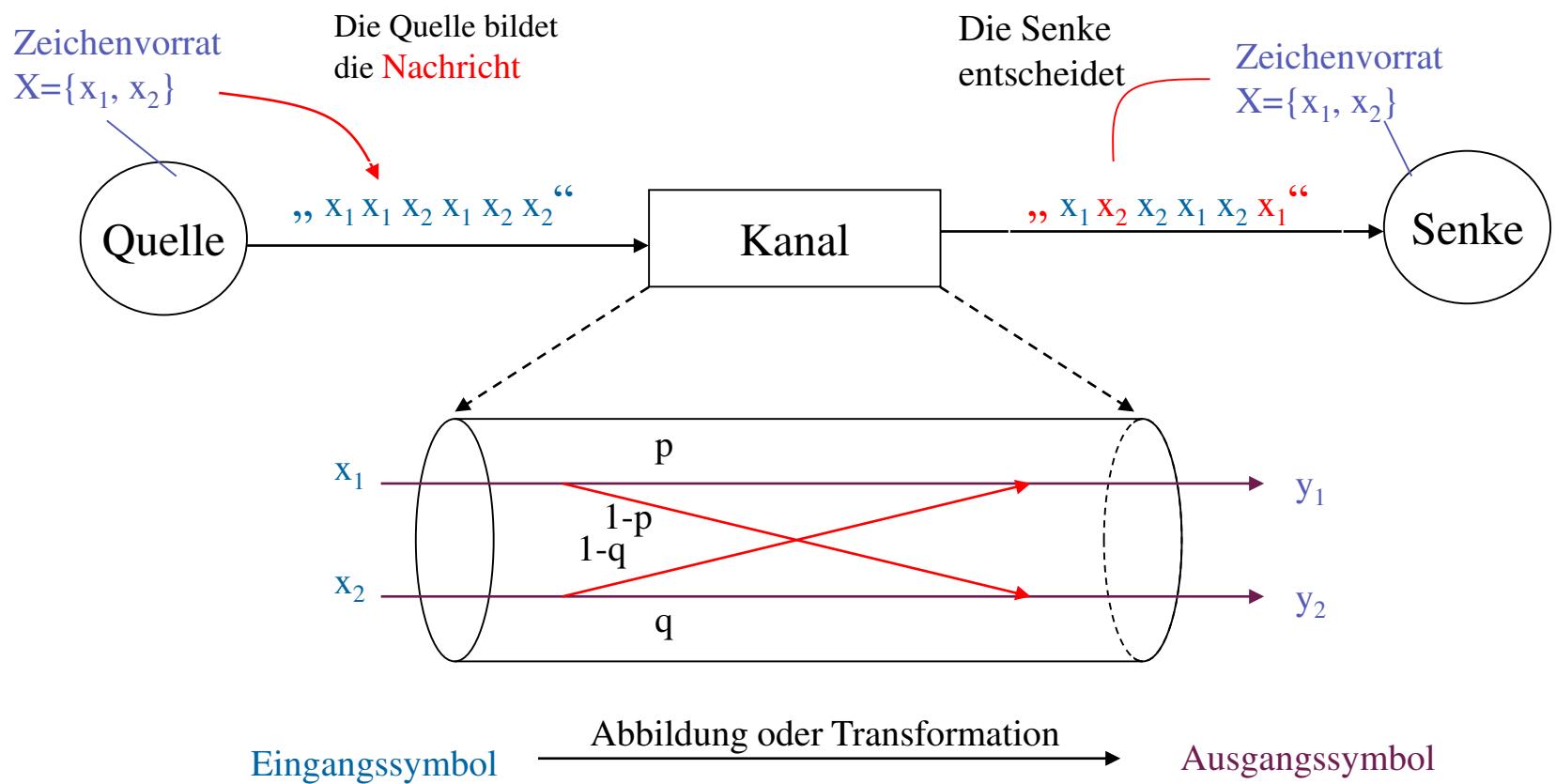
- Prof. Dr.-Ing. Andreas Rinkel  
[andreas.rinkel@hsr.ch](mailto:andreas.rinkel@hsr.ch)
- Tel.: +41 (0) 55 2224461
- Mobil: +41 (0) 79 3320562
- Teams

# Modell der Informationsverarbeitung



Um einen Kanalkodierung zu erstellen, brauchen wir ein **abstraktes Modell** des Kanals!

# Kanalmodell

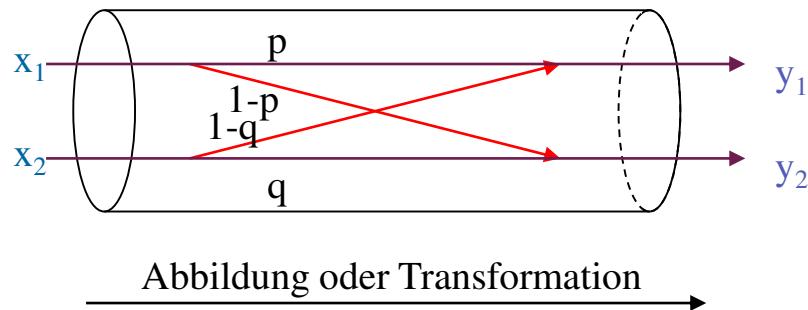


# Kanalmatrix

Eingangssymbol

$$p(x_1)=0.5$$

$$p(x_2)=0.5$$



Ausgangssymbol

$$p(y_1) = p(x_1) \cdot p + p(x_2) \cdot (1-q)$$

$$p(y_2) = p(x_1) \cdot (1-p) + p(x_2) \cdot (q)$$

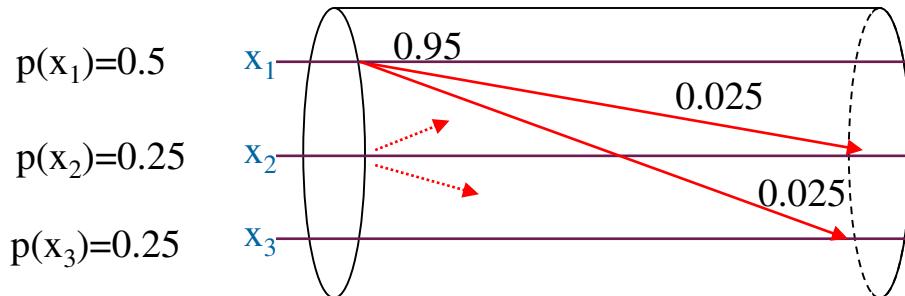
$$p(Y|X) = \begin{bmatrix} p & 1-p \\ 1-q & q \end{bmatrix} \mapsto \begin{bmatrix} \sum = 1 \\ \sum = 1 \end{bmatrix}$$

Kanalmatrix

$$p(Y|X) = \begin{bmatrix} p(y_1|x_1) & p(y_2|x_1) \\ p(y_1|x_2) & p(y_2|x_2) \end{bmatrix}$$

# Beispiel: Kanalmatrix des symmetrischen Kanals

Eingangssymbol



Ausgangssymbol

$$\begin{aligned} y_1 &= p(y_1) \\ y_2 &= p(y_2) \\ y_3 &= p(y_3) \end{aligned}$$

$$\begin{bmatrix} p(y_1) \\ p(y_2) \\ p(y_3) \end{bmatrix} = \begin{bmatrix} p(x_1) \cdot p(y_1|x_1) + p(x_2) \cdot p(y_1|x_2) + p(x_3) \cdot p(y_1|x_3) \\ p(x_1) \cdot p(y_2|x_1) + p(x_2) \cdot p(y_2|x_2) + p(x_3) \cdot p(y_2|x_3) \\ p(x_1) \cdot p(y_3|x_1) + p(x_2) \cdot p(y_3|x_2) + p(x_3) \cdot p(y_3|x_3) \end{bmatrix}$$

$$p(Y|X) = \begin{bmatrix} 0.95 & 0.025 & 0.025 \\ 0.025 & 0.95 & 0.025 \\ 0.025 & 0.025 & 0.95 \end{bmatrix}$$

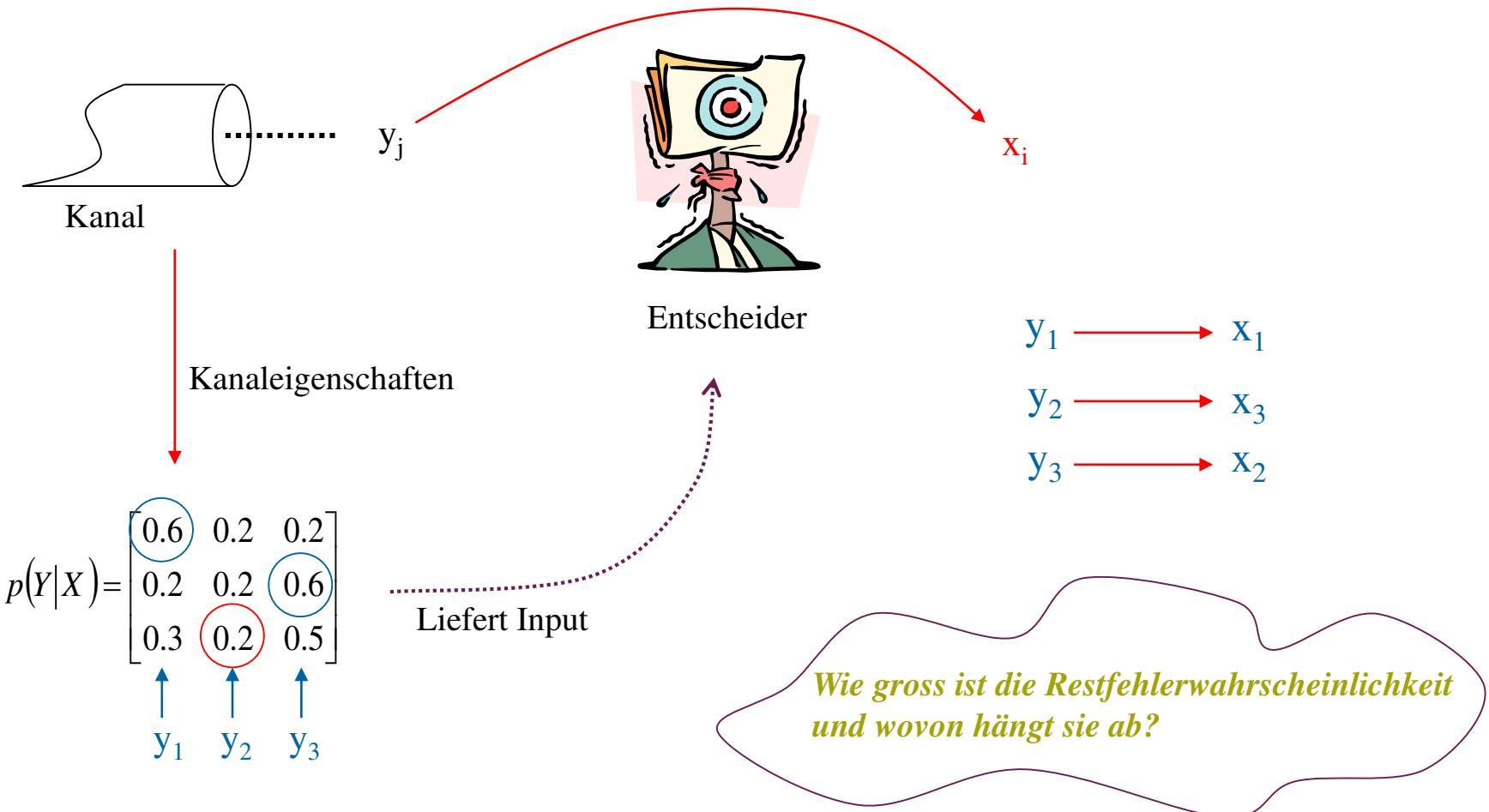
$$\begin{bmatrix} 0.4875 \\ 0.25625 \\ 0.25625 \end{bmatrix} = \begin{bmatrix} 0.5 \cdot 0.95 + 0.25 \cdot 0.025 + 0.25 \cdot 0.025 \\ 0.5 \cdot 0.025 + 0.25 \cdot 0.95 + 0.25 \cdot 0.025 \\ 0.5 \cdot 0.025 + 0.25 \cdot 0.025 + 0.25 \cdot 0.95 \end{bmatrix}$$

Fehlerwahrscheinlichkeit des Kanals.

Diese ist unabhängig von der Auftrittswahrscheinlichkeit der Zeichen der Quelle.

Kanal Symmetrisch

# Maximum-Likelihood-Verfahren,



# Transinformation I

Eingangssymbol

$$p(x_1)=0.5$$

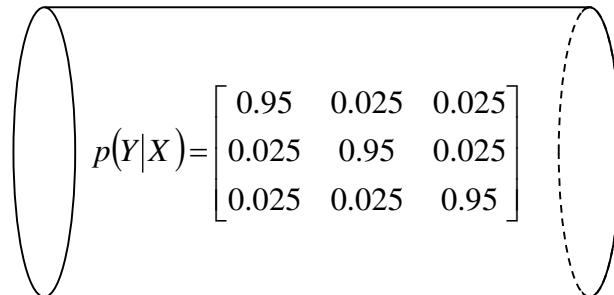
$$p(x_2)=0.25$$

$$p(x_3)=0.25$$

$x_1$

$x_2$

$x_3$



Ausgangssymbol

$y_1$

$y_2$

$y_3$

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 0.4875 \\ 0.25625 \\ 0.25625 \end{bmatrix} = \begin{bmatrix} 0.5 \cdot 0.95 + 0.25 \cdot 0.025 + 0.25 \cdot 0.025 \\ 0.5 \cdot 0.025 + 0.25 \cdot 0.95 + 0.25 \cdot 0.025 \\ 0.5 \cdot 0.025 + 0.25 \cdot 0.025 + 0.25 \cdot 0.95 \end{bmatrix}$$

$$\begin{aligned} H\langle X \rangle &= -\sum_{i=1}^3 p(x_i) \cdot ld(p(x_i)) \\ &= -(0.5 \cdot ld(0.5) + 2 \cdot 0.25 ld(0.25)) \\ &= 0.5 + 1 = 1.5 \end{aligned}$$



$$H\langle X \rangle \neq H\langle Y \rangle$$

$$\begin{aligned} H\langle Y \rangle &= -\sum_{i=1}^3 p(y_i) \cdot ld(p(y_i)) \\ &= -(0.4875 ld(0.4875) + 2 \cdot 0.25625 ld(0.25625)) \\ &= 0.5053 + 1.0067 = 1.512 \end{aligned}$$

?

# Transinformation I

Eingangssymbol

$$\begin{aligned} p(x_1) &= 0.5 \\ p(x_2) &= 0.25 \\ p(x_3) &= 0.25 \end{aligned}$$

Ausgangssymbol

$$p(Y|X) = \begin{bmatrix} 0.95 & 0.025 & 0.025 \\ 0.025 & 0.95 & 0.025 \\ 0.025 & 0.025 & 0.95 \end{bmatrix}$$
$$\begin{aligned} y_1 &= 0.4875 \\ y_2 &= 0.25625 \\ y_3 &= 0.25625 \end{aligned} = \begin{bmatrix} 0.5 \cdot 0.95 + 0.25 \cdot 0.025 + 0.25 \cdot 0.025 \\ 0.5 \cdot 0.025 + 0.25 \cdot 0.95 + 0.25 \cdot 0.025 \\ 0.5 \cdot 0.025 + 0.25 \cdot 0.025 + 0.25 \cdot 0.95 \end{bmatrix}$$

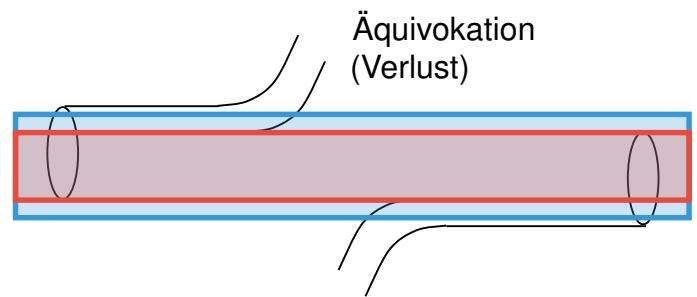
## Verbundentropie:

Das parweise auftreten aller möglichen Kombinationen am Kanalein- und ausgang

$$H(X, Y) = - \sum_i^n \sum_j^n p(x_i, y_j) * \log_2(p(x_i, y_j))$$

# Äquivokation (Verlust)

$$H(X|Y) = - \sum_i^n \sum_j^n p(x_i, y_j) * \log_2(p(x_i|y_j))$$



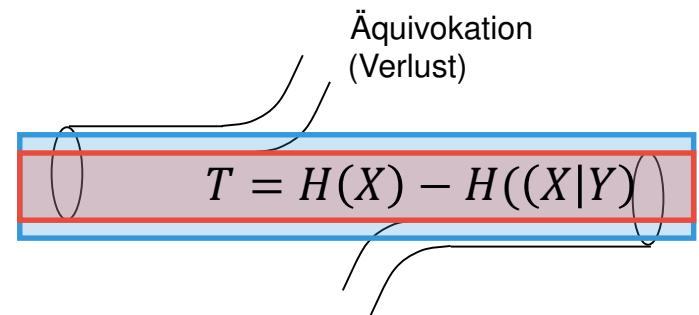
Vergleiche mit der Verbundentropie:

$$H(X, Y) = - \sum_i^n \sum_j^n p(x_i, y_j) * \log_2(p(x_i, y_j))$$

- **Auch Rückschlussentropie genannt**
- **Ungewissheit über das gesendete Zeichen bei bekanntem Empfangszeichen**
- **Merke: Ist der Kanal fehlerfrei, so ist die Äquivokation gleich 0.**

# Äquivokation (Verlust)

$$H(X|Y) = - \sum_i^n \sum_j^n p(x_i, y_j) * \log_2(p(x_i|y_j))$$



Vergleiche mit der Verbundentropie:

$$H(X, Y) = - \sum_i^n \sum_j^n p(x_i, y_j) * \log_2(p(x_i, y_j))$$

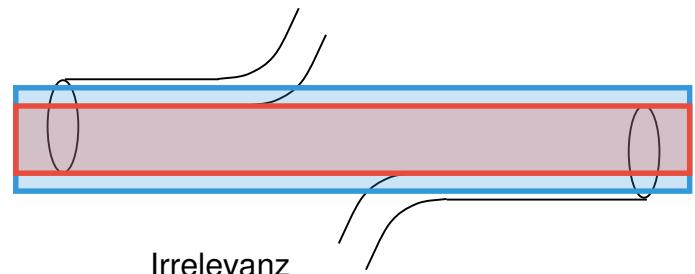
$$H(X|Y) = - \sum_i^n \sum_j^n p(y_i)p(x_j|y_i) * \log_2 p(x_j|y_i)$$

# Irrelevanz (Rauschen)

$$H(Y|X) = - \sum_i^n \sum_j^n p(x_i, y_j) * \log_2(p(y_j|x_i))$$

Vergleiche mit der Verbundentropie:

$$H(X, Y) = - \sum_i^n \sum_j^n p(x_i, y_j) * \log_2(p(x_i, y_j))$$



- **Auch Streuentropie genannt**
- **Ungewissheit der empfangenen Zeichen bei vorgegebenen Sendezeichen**

# Irrelevanz (Rauschen)

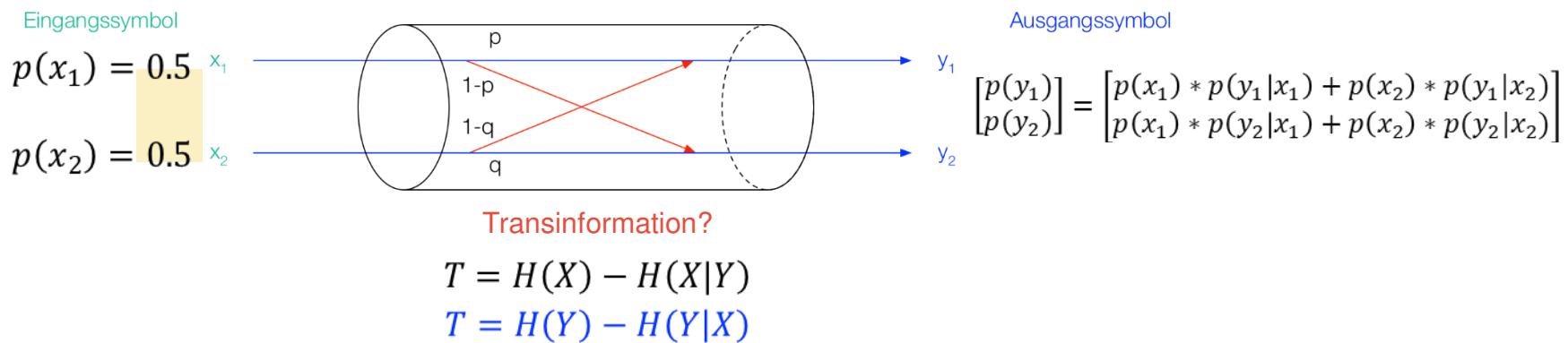
$$H(Y|X) = - \sum_i^n \sum_j^n p(x_i, y_j) * \log_2(p(y_j|x_i))$$

$$T = H(Y) - H((Y|X))$$

Irrelevanz  
(Rauschen)

$$H(Y|X) = - \sum_i^n \sum_j^n p(x_i)p(y_j|x_i) * \log_2 p(y_j|x_i) =$$

# Transinformation Beispiel 1 (4)



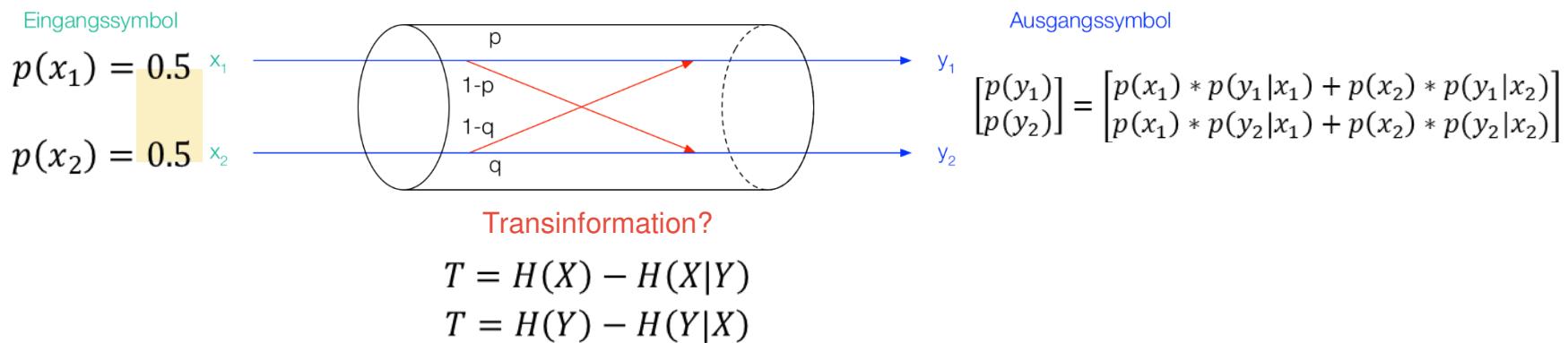
Sei:

$$\begin{aligned} p &= q = 1 \\ \Rightarrow p(y_1) &= p(x_1) = 0.5 \\ p(y_2) &= p(x_2) = 0.5 \\ \Rightarrow H(Y) &= H(X) = 1 \end{aligned}$$

$$\begin{aligned} H(Y|X) &= - \sum_i^n \sum_j^n p(x_i, y_j) * \log_2(p(y_j|x_i)) \\ &= -(p(x_1) * p(y_1|x_1) * \log_2(1) \\ &\quad + p(x_2) * p(y_2|x_2) * \log_2(1)) \\ &= 0 \end{aligned}$$

$$\begin{aligned} T &= H(Y) - H(Y|X) \\ T &= H(Y) = 1 \end{aligned}$$

## Transinformation Beispiel 2 (4)



Sei:

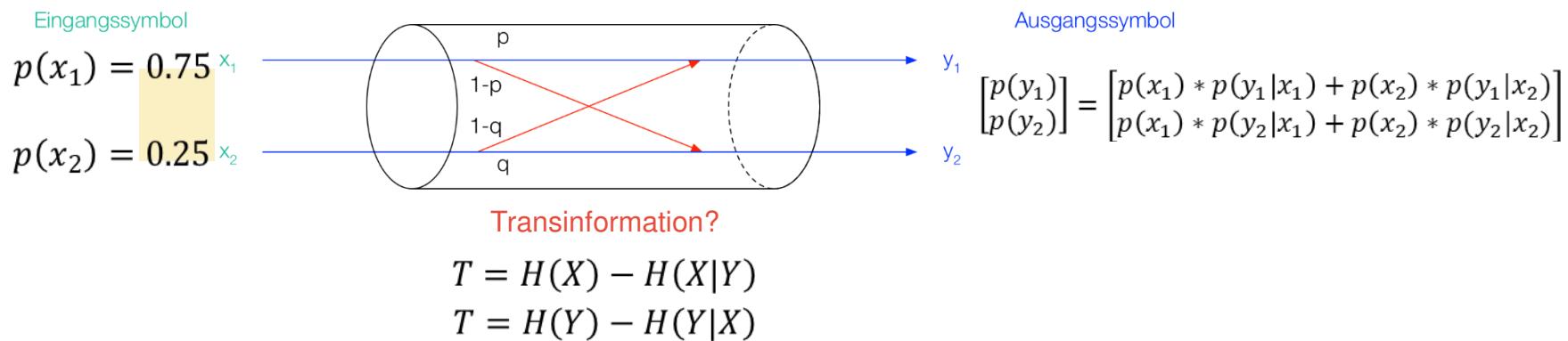
$$\begin{aligned} p &= q = 0.5 \\ \Rightarrow p(y_1) &= 2 * 0.5^2 = 0.5 \\ p(y_2) &= 2 * 0.5^2 = 0.5 \\ \Rightarrow H(Y) &= H(X) = 1 \end{aligned}$$

$$\begin{aligned} H(Y|X) &= - \sum_i^n \sum_j^n p(x_i, y_j) * \log_2(p(y_j|x_i)) \\ &= -(4 * (0.25 * -(1))) \\ &= 1 \end{aligned}$$

$$T = H(Y) - H(Y|X)$$

$$T = 0$$

# Transinformation Beispiel 3 (4)



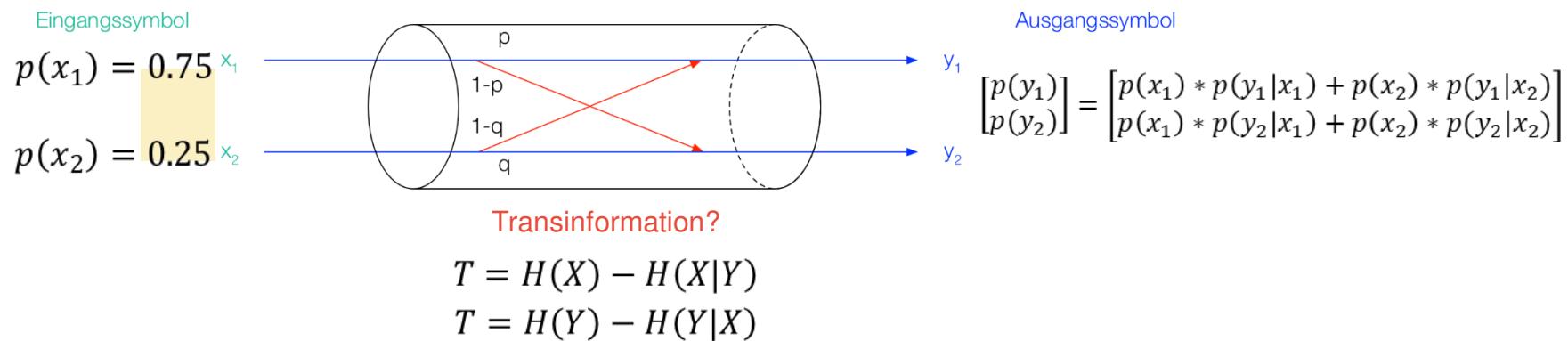
Sei:

$$\begin{aligned} p &= q = 1 \\ \Rightarrow p(y_1) &= p(x_1) = 0.75 \\ p(y_2) &= p(x_2) = 0.25 \\ \Rightarrow H(Y) &= H(X) \approx 0.811 \end{aligned}$$

$$\begin{aligned} H(Y|X) &= - \sum_i^n \sum_j^n p(x_i, y_j) * \log_2(p(y_j|x_i)) \\ &= -(p(x_1) * p(y_1|x_1) * \log_2(1) \\ &\quad + p(x_2) * p(y_2|x_2) * \log_2(1)) \\ &= 0 \end{aligned}$$

$T = H(Y) - H(Y|X)$ 
 $T = H(Y)$

# Transinformation Beispiel 4 (4)



Sei:

$$p = q = 0.5$$

$$\Rightarrow p(y_1) = 0.5 * (0.75 + 0.25) = 0.5$$

$$p(y_2) = 0.5 * (0.25 + 0.75) = 0.5$$

$$\Rightarrow H(Y) = 1; H(X) \approx 0.811$$

$$H(Y|X) = -(2 * (0.75 * 0.5 * (-1)) \\ + 2 * (0.25 * 0.5 * (-1))) \\ = 1$$

$$T = H(Y) - H(Y|X)$$

$$T = 0$$

## Fazit

- Ein nicht gestörter Kanal (Einheitsmatrix) überträgt den mittleren Informationsfluss ohne weiteren Verlust, d.h. die Transinformation wird nur durch die Quelle bestimmt.
- Verändert sich die Entropie der Quelle, so verändert sich auch die Transinformation.
- Nimmt die Fehlerwahrscheinlichkeit zu, so verringert sich die Transinformation.
- Sind alle Positionen der Kanalmatrix gleich besetzt, so wird die Transinformation  $T = 0$ , d.h.  $H(Y) = H(Y|X) = 1$  und zwar unabhängig von der Entropie am Kanaleingang.
- Die Transinformation gibt den maximalen und somit fehlerfreien Informationsfluss über einen gestörten Kanal an.  
Leider haben wir noch keine Aussage, wie dies zu erreichen ist!
- Idee ?



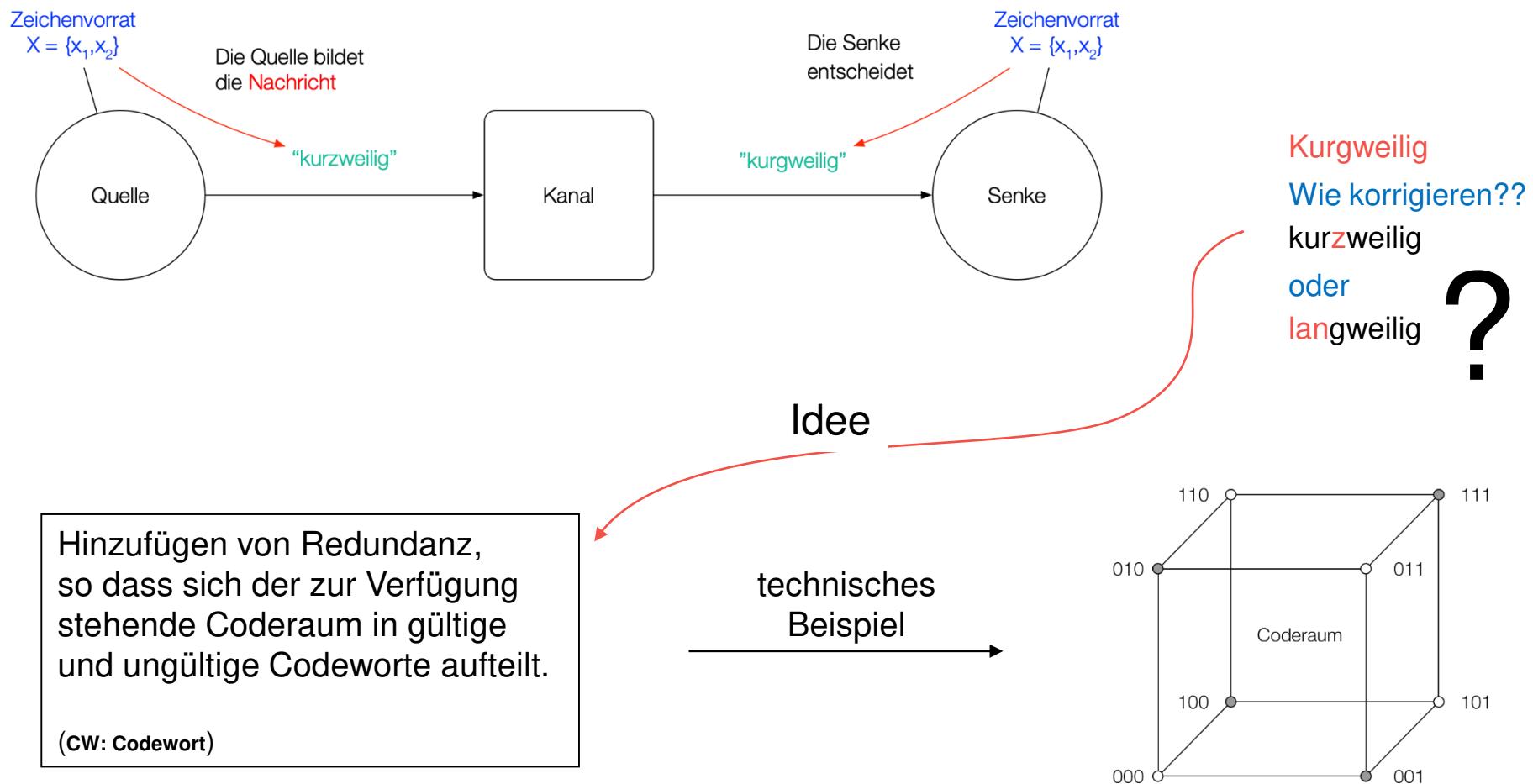
**OST**  
Ostschweizer  
Fachhochschule

# ***Informations- und Codierungstheorie***

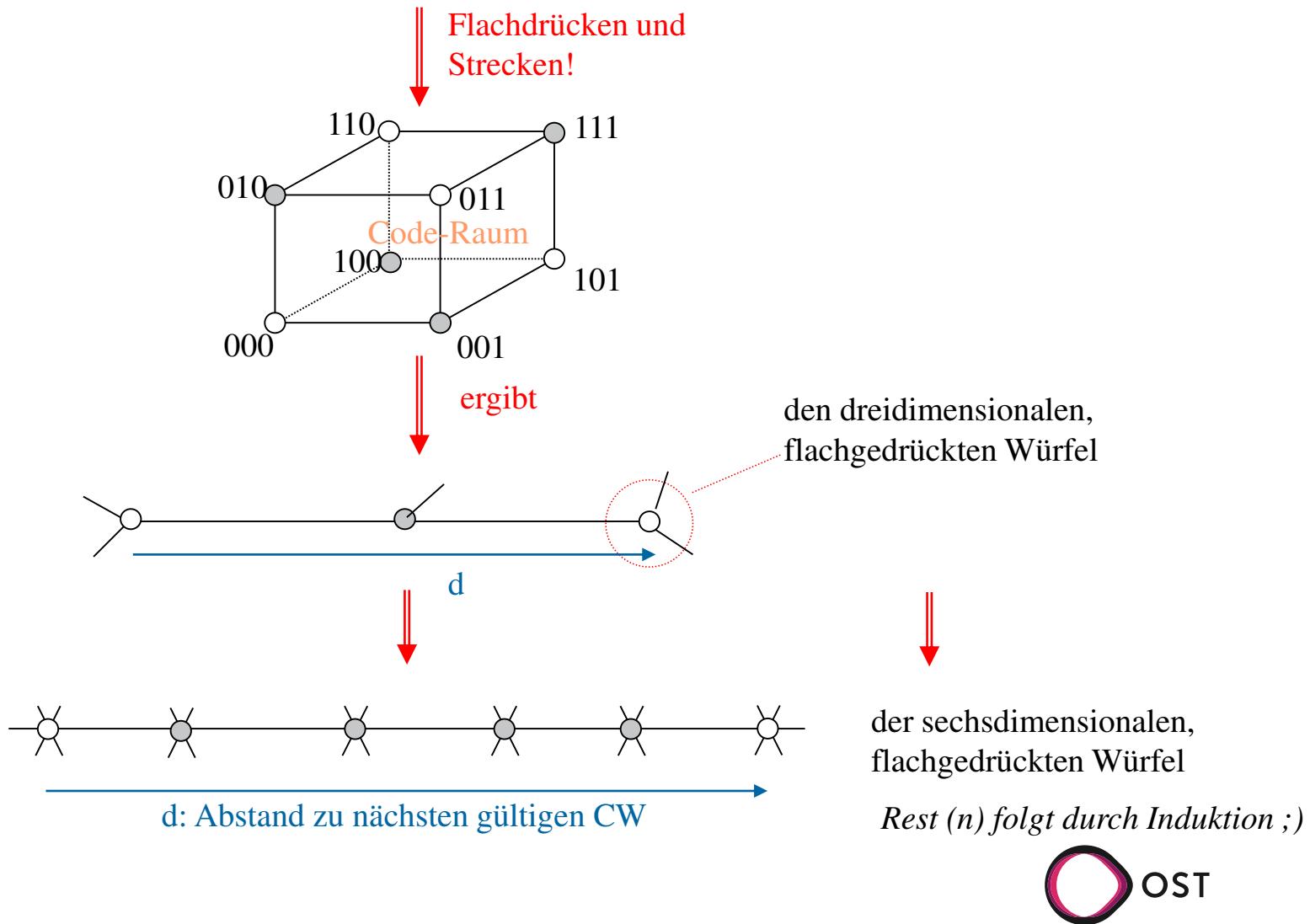
## ***Teil 2. Blockcodes***

- Prof. Dr.-Ing. Andreas Rinkel  
[andreas.rinkel@hsr.ch](mailto:andreas.rinkel@hsr.ch)
- Tel.: +41 (0) 55 2224461
- Mobil: +41 (0) 79 3320562
- Teams

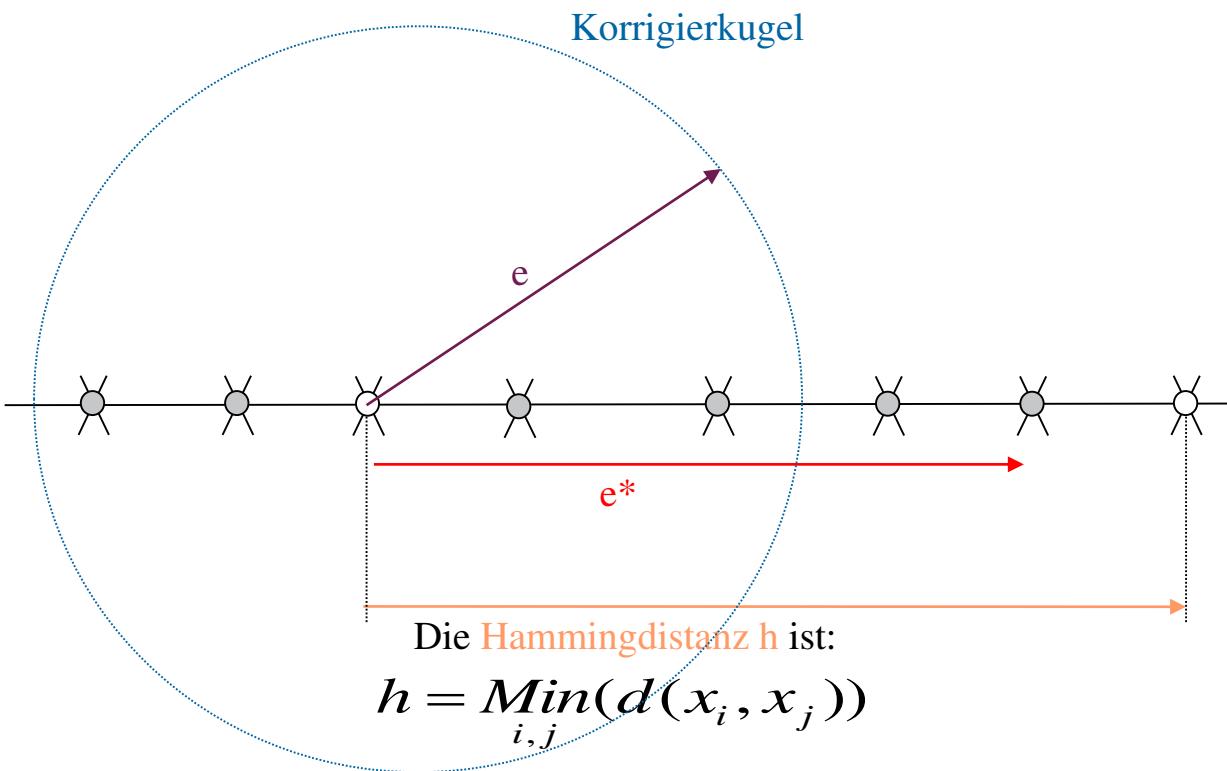
# Kanalcodierung



# Der n-Dimensionale Coderaum



# Coderaum: Definitionen



Anzahl der sicher erkennbaren Fehler

$$e^* = h - 1$$

Anzahl der sicher korrigierbaren Fehler

➤ **h gerade:**

$$h = 2e + 2 \Rightarrow$$

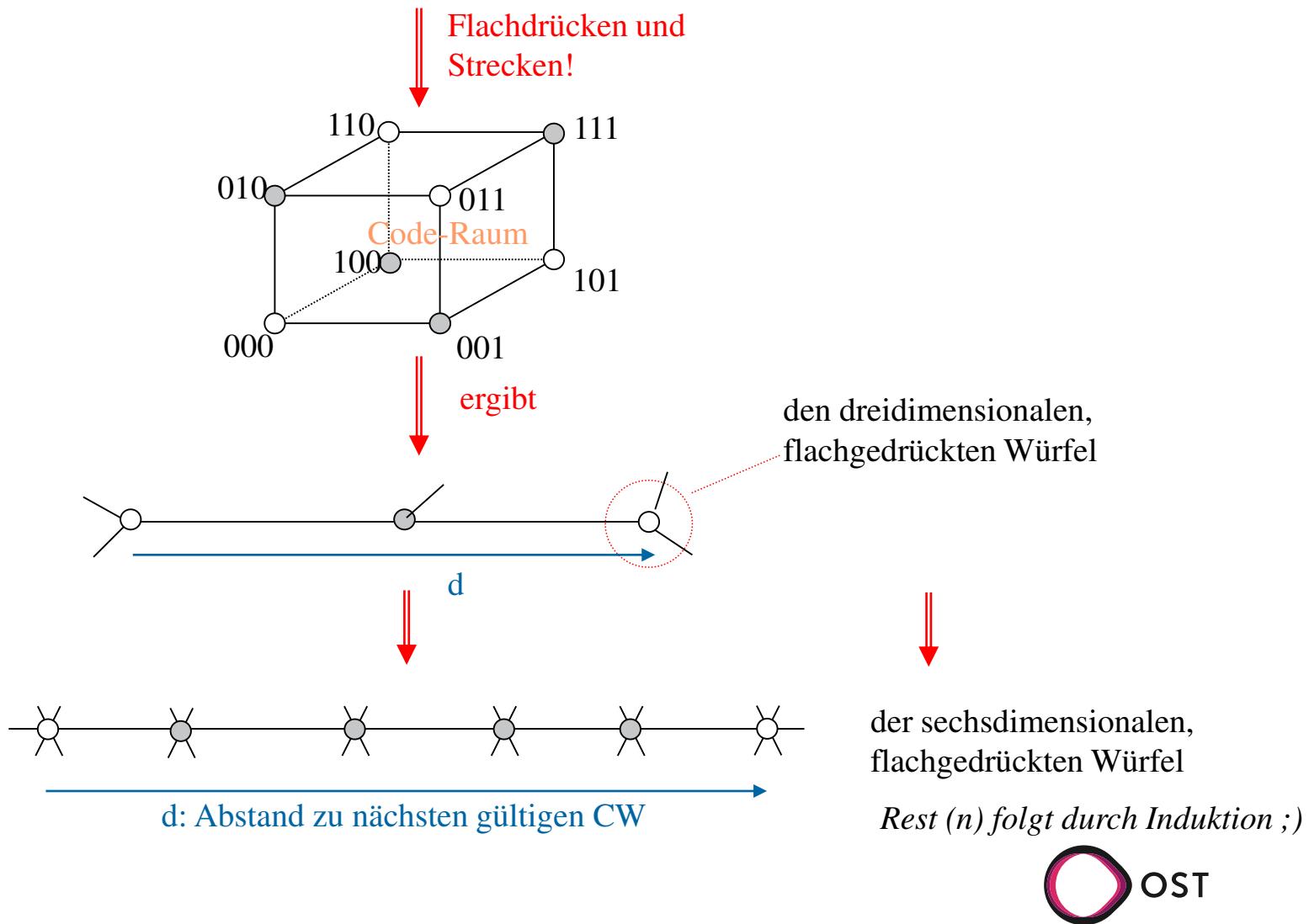
$$e = \frac{h-2}{2}$$

➤ **h ungerade:**

$$h = 2e + 1 \Rightarrow$$

$$e = \frac{h-1}{2}$$

# Der n-Dimensionale Coderaum



## Coderaum: *Dichtgepackt* oder nicht, das ist hier die Frage.

Der Coderaum ist *Dichtgepackt*, wenn sich alle Codewörter (gültige und ungültige) in einer Korrigierkugel befinden.

Sei :

- $n$  die Dimension des Code (Anzahl aller CW =  $2^n$ ),
  - $m$  die Dimension der Nachrichten (Anzahl aller gültigen CW =  $2^m$ )
  - $k$  die Dimension der Kontrollstellen mit  $n = m+k$
- ⇒ So folgt die Codeabschätzung:

$$2^m \cdot \sum_{w=0}^e \binom{n}{w} \leq 2^n$$

Anzahl der CW bzw.  
Korrigierkugeln

Anzahl der CW pro  
Korrigierkugel

Anzahl aller CW

Gilt:

$$2^m \cdot \sum_{w=0}^e \binom{n}{w} = 2^n$$

So ist der Code dichtgepackt!

# Blockcodes: Einführung

Beispiel: Quersummencode

$m=2$        $k=1$

$x_1$	$x_2$	$x_3$
0	0	0
0	1	1
1	0	1
1	1	0
0	0	1
0	1	0
1	0	0
1	1	1

$m$  Nachrichtenstellen



$k$  Kontrollstellen



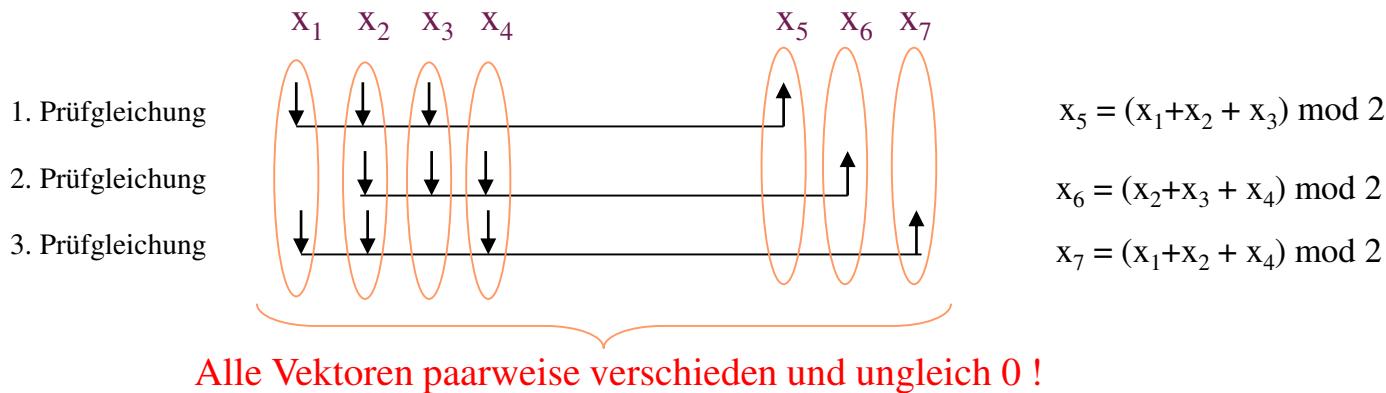
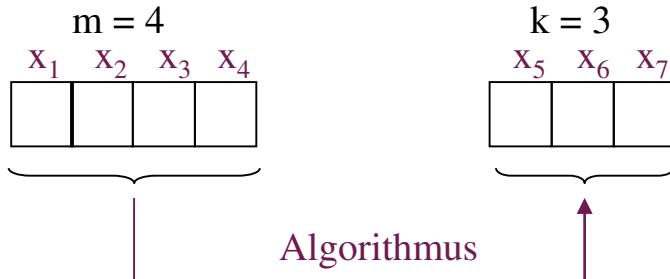
Algorithmus

Gültige Codeworte,  
sie erfüllen den Algorithmus

Ungültige Codeworte,  
sie erfüllen den Algorithmus nicht,  
d.h. sie liefern ein *Fehlermuster*

Algorithmus zur Berechnung  
der Kontrollstellen  
 $x_3 = (x_1 + x_2) \text{ mod } 2$

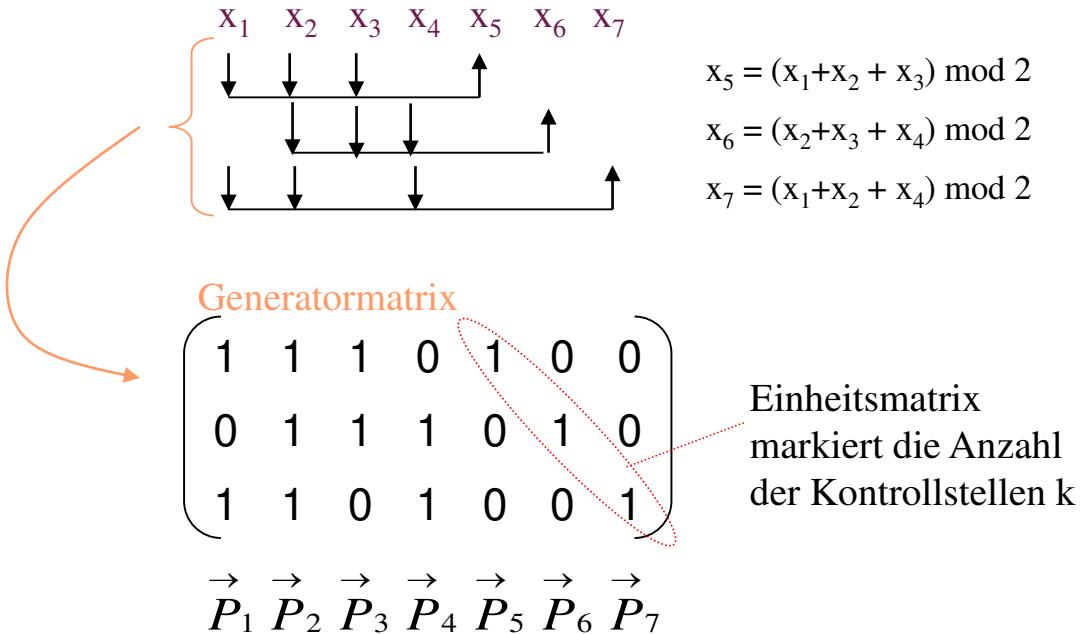
# Blockcodes: Hamming-Code I



**Interpretation:** wird eine Stelle des CW verletzt, so werden jeweils andere Kombinationen von Prüfgleichungen verletzt, d.h. es müsste ein Fehlersyndrom geben, dass es erlaubt, den Fehlerort zu lokalisieren.

**Frage:** wie viele Fehler können nicht mehr erkannt werden?

# Blockcodes: Hamming-Code II



Hieraus folgt  
die Codebedingung:

$$\sum_i x_i \cdot \vec{P}_i \equiv \vec{0} \pmod{2}$$

# Blockcodes: Hamming-Code III

Tabelle der gültigen Codeworte

$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$
0	0	0	0	0	0	0
0	0	0	1	0	1	1
0	0	1	0	1	1	0
0	0	1	1	1	0	1
0	1	0	0	1	1	1
0	1	0	1	1	0	0
0	1	1	0	0	0	1
0	1	1	1	0	1	0
1	0	0	0	1	0	1
1	0	0	1	1	1	0
1	0	1	0	0	1	1
1	0	1	1	0	0	0
1	1	0	0	0	1	0
1	1	0	1	0	0	1
1	1	1	0	1	0	0
1	1	1	1	1	1	1

Nachrichtenstellen

Kontrollstellen

$$\left\{ \begin{array}{ccccccc} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{array} \right\}$$

Die Codebedingung:

$$\sum_i x_i \cdot \vec{P}_i \equiv \vec{0} \bmod 2$$

Wird für alle gültigen  
Codeworte (Tabelle) erfüllt.

Was ergibt die Berechnung der  
Codebedingung bei einem Bitfehler?

$$x_5 = (x_1 + x_2 + x_3) \bmod 2$$

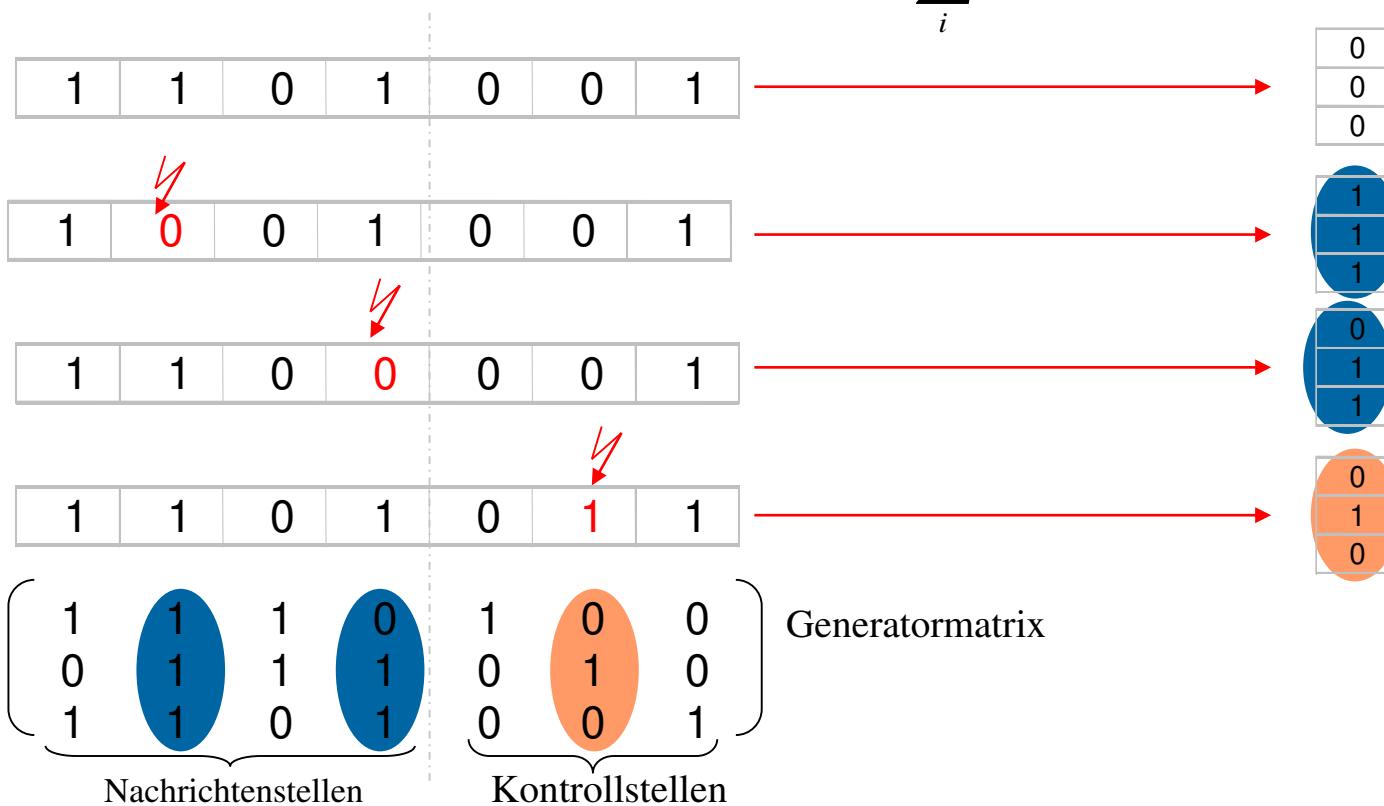
$$x_6 = (x_2 + x_3 + x_4) \bmod 2$$

$$x_7 = (x_1 + x_2 + x_4) \bmod 2$$

# Blockcodes: Hamming-Code IV

Das Syndrom  $Z$ :

$$\vec{Z} = \sum_i x_i \cdot \vec{P}_i \bmod 2$$



## Hamming-Code: Das Fehlersyndrom

Gesendetes  
Codewort

$$X = [x_1, x_2, x_3, \dots, x_n]$$

Überlagert durch  
das Fehlermuster

$$F = [f_1, f_2, f_3, \dots, f_n]$$



Empfangenes Wort

$$X' = X + F$$

$$\begin{aligned} &= [x_1 + f_1, x_2 + f_2, x_3 + f_3, \dots, x_n + f_n] \bmod 2 \\ &= [x'_1, x'_2, x'_3, \dots, x'_n] \end{aligned}$$

Aus der Codebedingung  
folgt das Syndrom

$$\vec{Z} = \sum_i x'_i \cdot \vec{P}_i = \sum_i (x_i + f_i) \cdot \vec{P}_i$$

$$= \sum_i x_i \cdot \vec{P}_i + \sum_i f_i \cdot \vec{P}_i$$

Codebedingung = 0

$$\Rightarrow \vec{Z} = \sum_i f_i \cdot \vec{P}_i$$

Das heisst, bei genau einem Fehler  
markiert die Prüfspalte den Fehlerort.

**Idee:** Generatormatrix kann durch Generatorpolynom beschrieben werden!

**Ziel:** Vereinfachte Berechnung der Kontrollstellen durch rückgekoppelte Schieberegister.

Generatorpolynom  $G(u)$

$$G(u) = \sum_{i=0}^k g_i \cdot u^i$$

Codebedingung

Codewortpolynom  $X(u)$

$$X(u) = \sum_{i=0}^n g_i \cdot u^i$$

$g_i \in \{0,1\}$  mit  $g_0 = g_k = 1$

Das Codewortpolynom ist ohne Rest durch das Generatorpolynom teilbar  
(in mod-2-Rechnung)

$$X(u) \div G(u) \equiv Q(u) \text{ mod } 2$$

$$X(u) \equiv Q(u) \cdot G(u) \text{ mod } 2$$

Grad  $k$  entspricht der Anzahl der Prüfstellen.  
Grad  $n$  entspricht der Anzahl der Codewortstellen.  
Die Zahl der Nachrichtenstellen ist  $m$   
 $\Rightarrow n = m + k$

# Zyklische Codes: Ermittlung der Kontrollstellen durch Polynomdivision

Sei:  $m = 4, k = 3, n = 7$

Nachricht:  $(x_1, x_2, x_3, x_4) = (1 \ 0 \ 0 \ 0)$

Generator:  $G(u) = u^3 + u + 1 \Rightarrow (g_3 \ g_2 \ g_1 \ g_0) = (1 \ 0 \ 1 \ 1)$

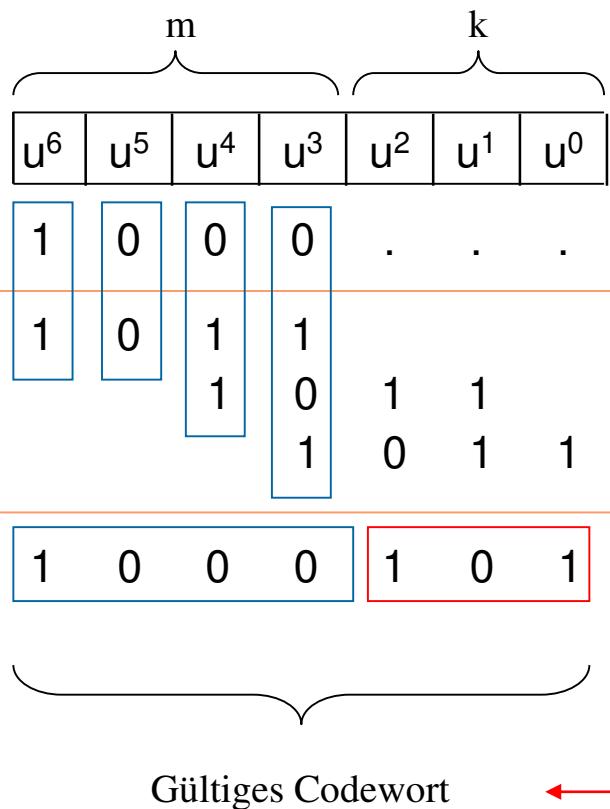
$u^6$	$u^5$	$u^4$	$u^3$	$u^2$	$u^1$	$u^0$			$u^3$	$u^2$	$u^1$	$u^0$			$u^3$	$u^2$	$u^1$	$u^0$
1	0	0	0	1	0	1			1	0	1	1			1	0	1	1
1	0	1	1				:		1	0	1	1			1	0	1	1
%	0	1	1															
	0	0	0	0														
%	1	1	0															
	1	0	1	1														
%	1	1	1															
	1	0	1	1														
%	1	0	1	1														
	1	0	1	1														

**mod 2**

The diagram shows a polynomial division process for generating a cyclic code. The dividend is  $1000 \cdot 1011 = 101011$ . The divisor is  $1011$ . The quotient is  $101$ , which are the sought control digits that satisfy the code condition.

101 sind die gesuchten Kontrollstellen, die die Codebedingung erfüllen.

# Zyklische Codes: Ermittlung der Kontrollstellen durch Mehrfachaddition



Sei:

$$m = 4, k = 3, n = 7$$

$$\text{Nachricht: } (x_1, x_2, x_3, x_4) = (1, 0, 0, 0)$$

$$\text{Generator: } G(u) = u^3 + u + 1 \Rightarrow (g_3, g_2, g_1, g_0) = (1, 0, 1, 1)$$

**Idee:**  $X(u)$  ist durch  $G(u)$  mod2 teilbar, also muss  $X(u)$  durch Addition von  $G(u)$  mod2 erzeugbar sein!

Addition des ersten Terms  $G(u)$  erzeugt die Stellen  $u^6, u^5$  von  $X(u)$

Addition des zweiten Terms  $G(u)$  erzeugt die Stelle  $u^4$  von  $X(u)$

Addition des dritten Terms  $G(u)$  erzeugt die Stelle  $u^3$  von  $X(u)$

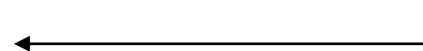
Der Rest muss nach Codebedingung die Kontrollstellen bilden!

# Zyklische Codes: Prüfen der Codebedingung

Empfangenes  
Codewort:

$$\begin{array}{r} 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \\ 1 \ 0 \ 1 \ 1 \\ 1 \ 0 \ 1 \ 1 \\ 1 \ 0 \ 1 \ 1 \\ \hline 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \end{array}$$

Generator: 1 0 1 1



Code- Bedingung erfüllt!

Idee:

Durch die Codebedingung muss die fortgesetzte Addition (mod 2) des Generators zum empfangenen CW Das Nullwort ergeben.

$$\begin{aligned} X(u) \div G(u) &\equiv Q(u) \text{ mod } 2 \\ X(u) &\equiv Q(u) \cdot G(u) \text{ mod } 2 \end{aligned}$$

Empfangenes  
Codewort:

$$\begin{array}{r} 1 \ 0 \ 0 \ 1 \ 1 \ 0 \ 1 \\ 1 \ 0 \ 1 \ 1 \\ 1 \ 0 \ 1 \ 1 \\ \hline 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \end{array}$$

Fehlersyndrom

Code- Bedingung nicht erfüllt!



# Zyklischer Hamming- Code und Generatormatrix ?

Gültiges Codewort: **1 0 0 0 1 0 1**

$$\begin{array}{r} \text{---} \\ \begin{array}{r} 0 0 0 0 1 0 1 \\ \hline 0 0 0 0 1 0 1 \end{array} \end{array}$$

$$\begin{array}{r} \text{---} \\ \begin{array}{r} 1 \textcolor{red}{1} 0 0 1 0 1 \\ 1 0 1 1 \\ 1 0 1 1 \\ \hline 0 0 0 0 1 1 1 \end{array} \end{array}$$

$$\begin{array}{r} \text{---} \\ \begin{array}{r} 1 0 \textcolor{red}{1} 0 1 0 1 \\ 1 0 1 1 \\ 1 0 1 1 \\ \hline 0 0 0 0 1 1 0 \end{array} \end{array}$$

$$\begin{array}{r} \text{---} \\ \begin{array}{r} 1 0 0 \textcolor{red}{1} 1 0 1 \\ 1 0 1 1 \\ 1 0 1 1 \\ \hline 0 0 0 0 0 1 1 \end{array} \end{array}$$

$$\begin{array}{r} \text{---} \\ \begin{array}{r} 1 0 0 0 \textcolor{red}{0} 0 1 \\ 1 0 1 1 \\ 1 0 1 1 \\ 1 0 1 1 \\ \hline 0 0 0 0 1 0 0 \end{array} \end{array}$$

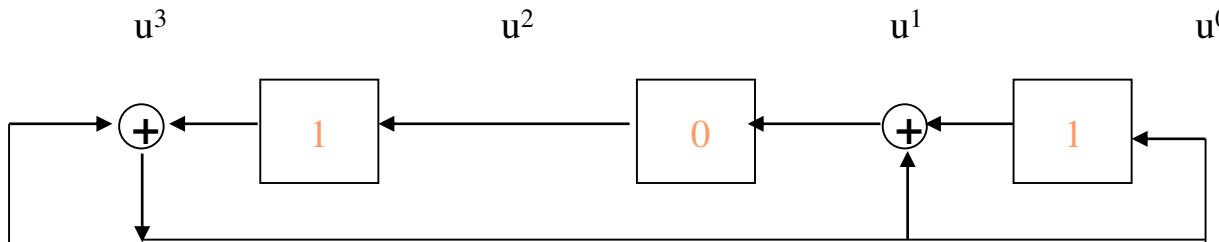
$$\begin{array}{r} \text{---} \\ \begin{array}{r} 1 0 0 0 1 \textcolor{red}{1} 1 \\ 1 0 1 1 \\ 1 0 1 1 \\ 1 0 1 1 \\ \hline 0 0 0 0 0 1 0 \end{array} \end{array}$$

$$\begin{array}{r} \text{---} \\ \begin{array}{r} 1 0 0 0 1 0 \textcolor{red}{0} \\ 1 0 1 1 \\ 1 0 1 1 \\ 1 0 1 1 \\ \hline 0 0 0 0 0 0 1 \end{array} \end{array}$$

Generatormatrix

$$\xrightarrow{\hspace{2cm}} \left( \begin{array}{ccccccc} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{array} \right)$$

# Zyklische Codes: Ermittlung der Kontrollstellen durch rückgekoppeltes Schieberegister



$\Leftarrow G(u)$

Sei:  $m = 4, k = 3, n = 7$   
Nachricht:  $(x_1, x_2, x_3, x_4) = (1 \ 0 \ 0 \ 0)$   
Generator:  $G(u) = u^3 + u + 1$

	0	0	0	
$x_1 = 1$	0	1	1	
$x_2 = 0$	1	1	0	
$x_3 = 0$	1	1	1	
$x_4 = 0$	1	0	1	

Ermittelten Kontrollstellen

Vorbelegung

Nach Übernahme  $x_1$

Nach Übernahme  $x_2$

Nach Übernahme  $x_3$

Nach Übernahme  $x_4$

⊕ Modulo 2 Addierer (XOR)



# Zyklische Codes

## Zyklische Hamming-Codes:

Hammingdistanz  $h=3$

Diese werden gebildet durch sogenannte primitive Polynome  $p(x) = g(x)$ :

$$p(x) = 1+x+x^3$$

$$p(x) = 1+x+x^4$$

$$p(x) = 1+x^2+x^5$$

$$p(x) = 1+x+x^6$$

$$p(x) = 1+x^3+x^7$$

$$p(x) = 1+x^2+x^3 + x^4+x^5+x^6+x^7$$

$$p(x) = 1+x^2+x^3 + x^4+x^5+x^8$$

$$p(x) = 1+x^4+x^9$$

$$p(x) = 1+x^3+x^{10}$$

$$p(x) = 1+x^2+x^{11}$$

$$p(x) = 1+x + x^4+x^6+x^{12}$$

$$p(x) = 1+x+x^3 + x^4+x^{13}$$

$$p(x) = 1+x^2+x^6+x^{10}+x^{14}$$

$$p(x) = 1+x+x^{15}$$

$$p(x) = 1+x^5+x^{23}$$

$$p(x) = 1+x+x^2+x^4+x^5 + x^7+x^8+x^{10}+x^{11}+x^{12}+ \\ x^{16} + x^{22}+x^{23}+x^{26}+x^{32}$$

## Zyklische Abramson-Codes bzw. CRC-Codes:

Hammingdistanz  $h=4$

Diese werden gebildet durch die Multiplikation eines primitiven Polynoms mit dem Term  $(1+x)$

Abramson-Code:  $g(x)=p(x) (1+x)$

Bsp.:

$$g(x) = (1+x+x^3) (1+x)$$

$$g(x) = 1+x^2+x^3+x^4$$

Aus: Martin Werner, Information und Codierung, vieweg 2002



**OST**  
Ostschweizer  
Fachhochschule

# ***Informations- und Codierungstheorie***

## ***Teil 7. Faltungscodes***

- Prof. Dr.-Ing. Andreas Rinkel  
[andreas.rinkel@hsr.ch](mailto:andreas.rinkel@hsr.ch)
- Tel.: +41 (0) 55 2224461
- Mobil: +41 (0) 79 3320562
- Teams

## Die ersten Arbeiten zu Block- und Faltungscodes gehen auf die 50er Jahre zurück.

- **Blockcodes wurden schnell zur Sicherung gegen Übertragungsfehler eingesetzt, nicht zuletzt wegen ihrer einfachen Implementierbarkeit.**
- **Eigenschaften:**
  - Leichte Implementierbarkeit der Encoder und Decoder durch Schieberegister
  - Hohe Fehlererkennungsmächtigkeit (Bündelfehlererkennung bei zyklischen Codes)
  - **Blockbildung der zu codierenden Daten notwendig keine fortlaufende Codierung möglich!**
  - **Eine fortlaufende Codierung ist aber mit Faltungscodes Möglich!**

**Für Faltungscodes wurde erst 1967 ein effizienter Algorithmus zur Dekodierung (Viterbi-Algorithmus) gefunden (Einsatz in der GSM, UMTS Funkübertragung)**

- **Eigenschaften:**

- Faltungscodes erlauben fortlaufende Codierung eines kontinuierlichen Datenstroms (keine Blockbildung erforderlich)
- Decodierung von Faltungscodes benötigt keine Blocksynchronisation
- Gute Faltungscodes werden durch Rechnersimulation gefunden

# Encoderschaltung des (2,1,3) Encoders

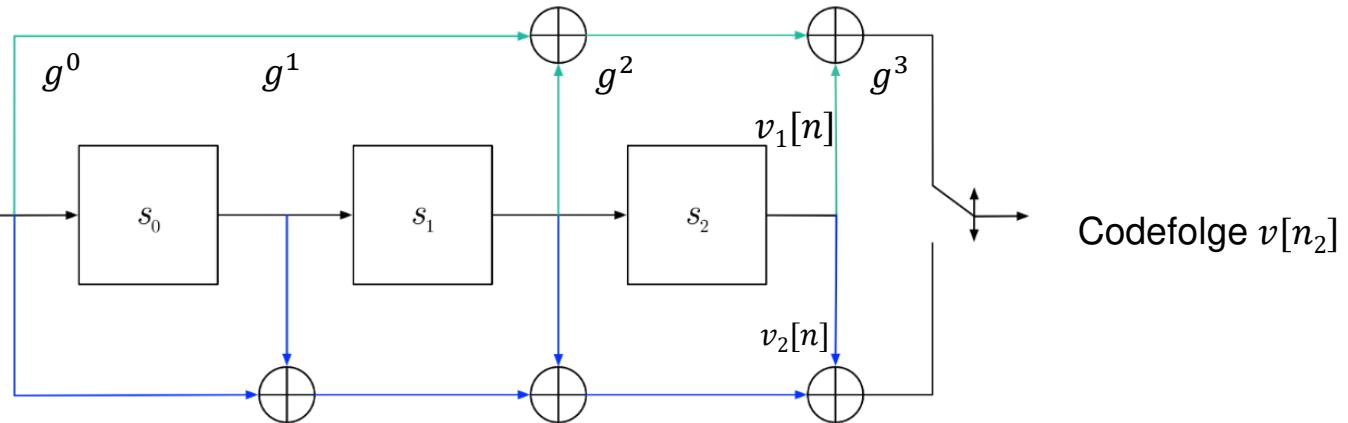
Idee: die Vergangenheit berücksichtigen, ein Beispiel I

$$\{g_1\} = \{1 0 1 1\}$$

oder

$$G_1(x) = 1 + g^2 + g^3$$

Nachrichtenfolge  $u[n]$



$$\{G_2\} = \{1 1 1 1\}$$

oder

$$G_2(x) = 1 + g + g^2 + g^3$$

Beispiel:

Sei  $\{u_n\} = \{1 0 1\}$ ,

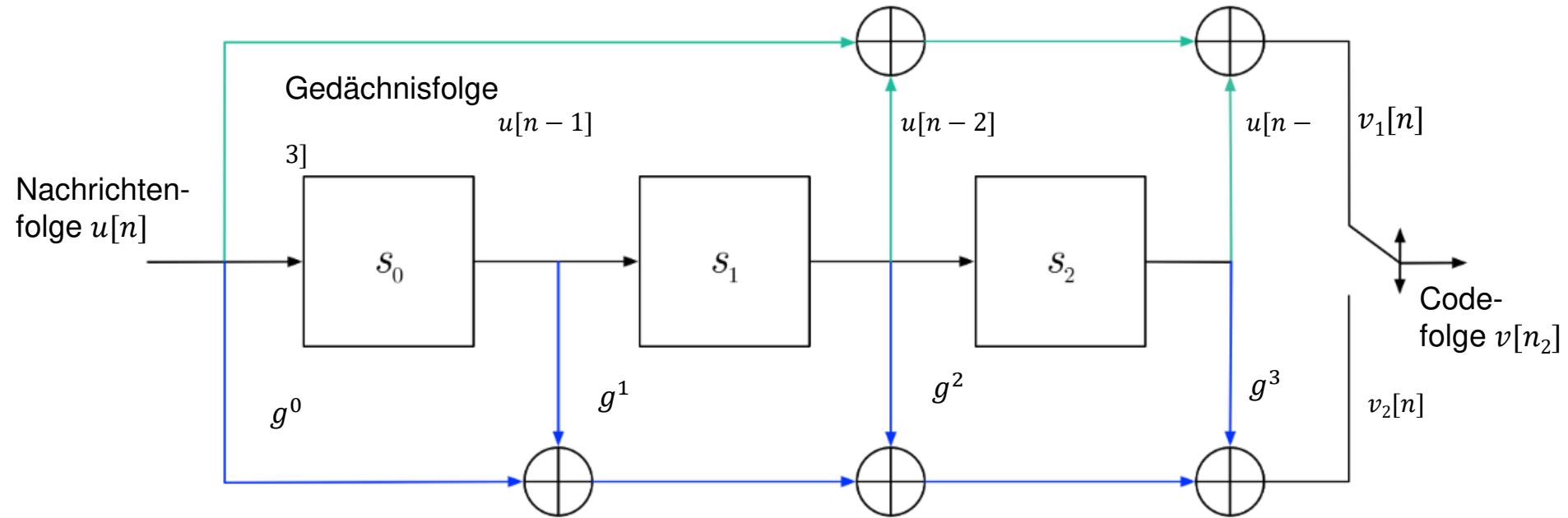
Die Speicherplätze  $s_0, s_1, s_2$  sind mit 0 vorbelegt.

$u[n]$	$s_0$	$s_1$	$s_2$	$v_1[n]$	$v_2[n]$	$v[n_2]$
-	0	0	0	-	-	-
1	0	0	0	1	1	11
0	1	0	0	0	1	01
1	0	1	0	0	0	00
0	1	0	1	1	0	10
0	0	1	0	1	1	11
0	0	0	1	1	1	11
-	0	0	0	-	-	-

Ausgangszustand erreicht

# Encoderschaltung des (2,1,3) Encoders

Idee: die Vergangenheit berücksichtigen, ein Beispiel II

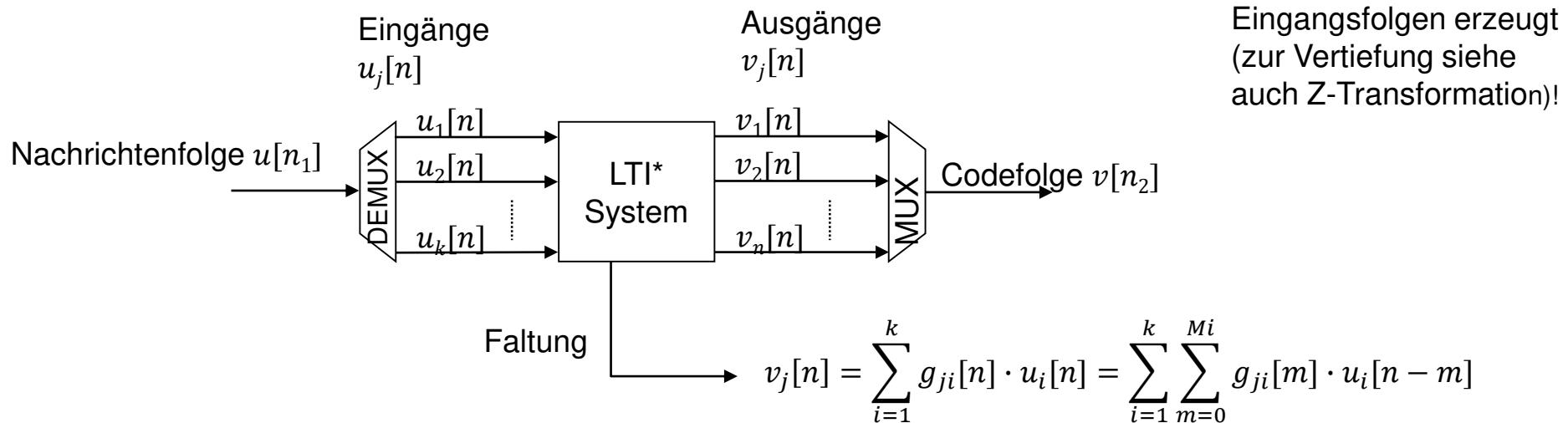


Nachrichtenfolge:  $\{u[n]\} = \{u_1, u_2, \dots, u_n\}$

Ausgangsfolgen:  $\{v[n]\} = \left\{ \left[ g^0 u_{[n]} + g^1 u_{[n-1]} + g^2 u_{[n-2]} + g^3 u_{[n-3]} \right], g^0 u_{[n+1-m]}, \dots \right\}$

$$\{v[n]\} = \sum_{m=0}^M g^m u_{[n-m]}$$

# Binärer Faltungsencoder: Verallgemeinerung

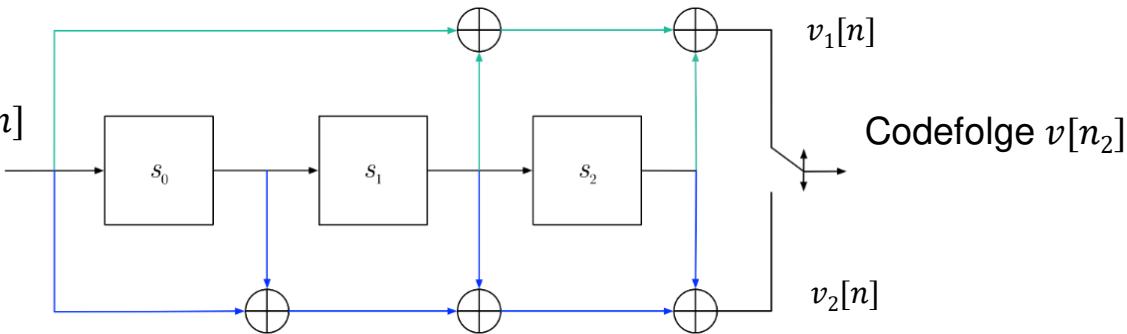


Die Ausgangsfolge wird durch die Faltung der Eingangsfolgen erzeugt (zur Vertiefung siehe auch Z-Transformation)!

\*LTI: Linear Time Invariant

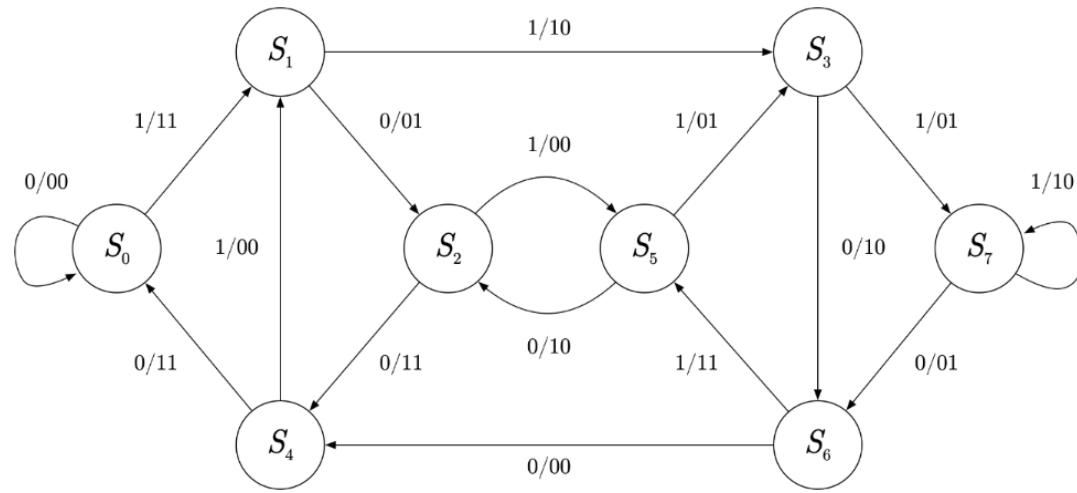
# Zustandsdarstellung des (2,1,3) Encoders

Nachrichtenfolge  $u[n]$



Codefolge  $v[n_2]$

$v_2[n]$



Zustandsgrösse

Zustand  $S_i$

Zustandsgröße			Zustand $S_i$
$s_0$	$s_1$	$s_2$	$(i = s_0 \cdot 2^0 + s_1 \cdot 2^1 + s_2 \cdot 2^2)$
0	0	0	0
1	0	0	1
0	1	0	2
1	1	0	3
0	0	1	4
1	0	1	5
0	1	1	6
1	1	1	7

# Aufgabe

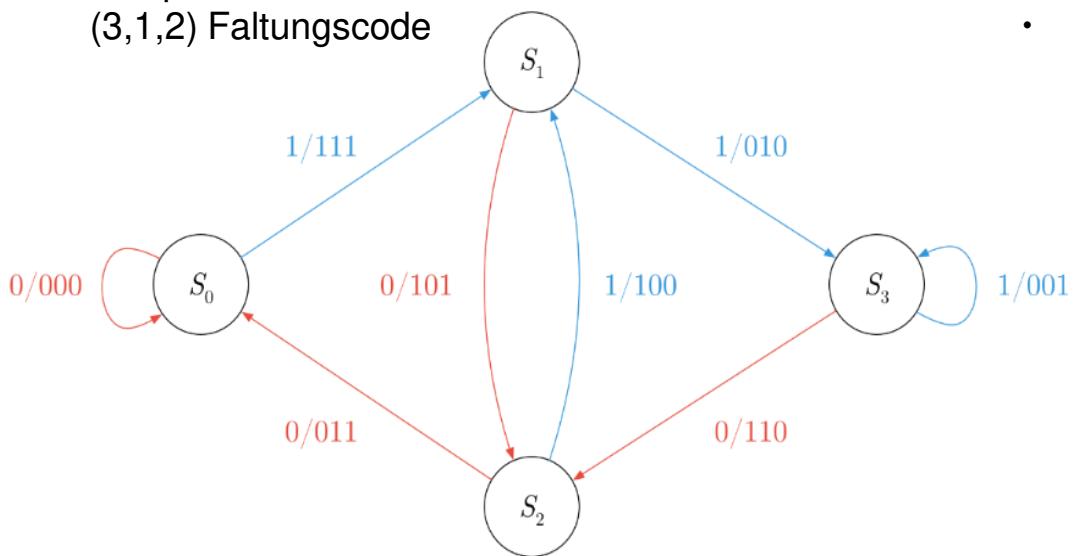
**Gegeben ist ein (3,1,2) Faltungscode mit**

- $g_1(x) = 1 + x$
- $g_2(x) = 1 + x^2$
- $g_3(x) = 1 + x + x^2$

**Ermitteln Sie den Codierer und das Zustandsdiagramm!**

# Struktur: Fundamentalweg und Gewichte

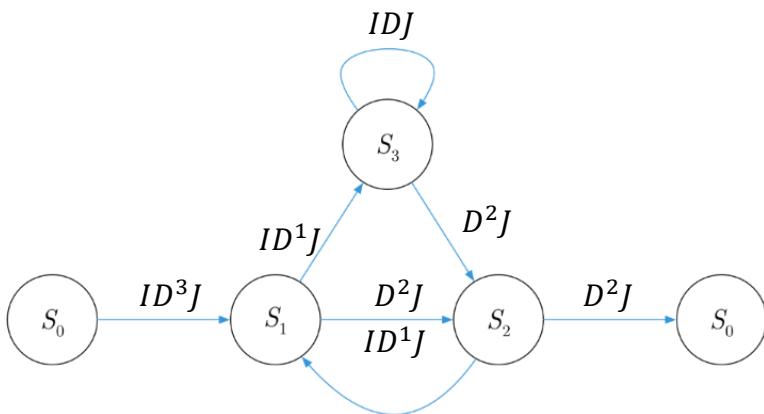
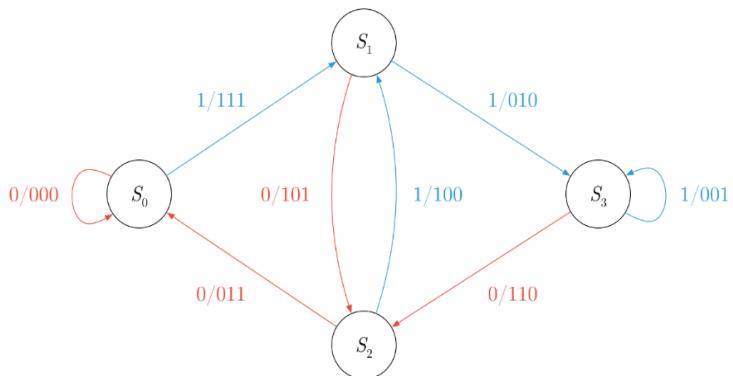
Beispiel:  
(3,1,2) Faltungscode



## Definitionen

- *Gewicht*: des Codes ist die Anzahl von Bitstellen eines Codeworts, die von «0» verschieden sind.
- *Fundamentalweg*: ist der (Teil-) Weg eines Codes, der im Zustand  $S_0$  beginnt und wieder im Zustand  $S_0$  endet. Die Analyse der Fundamentalwege liefert die Struktur des Faltungscodes.

# Struktur: Fundamentalweg und Gewichte



## Definitionen

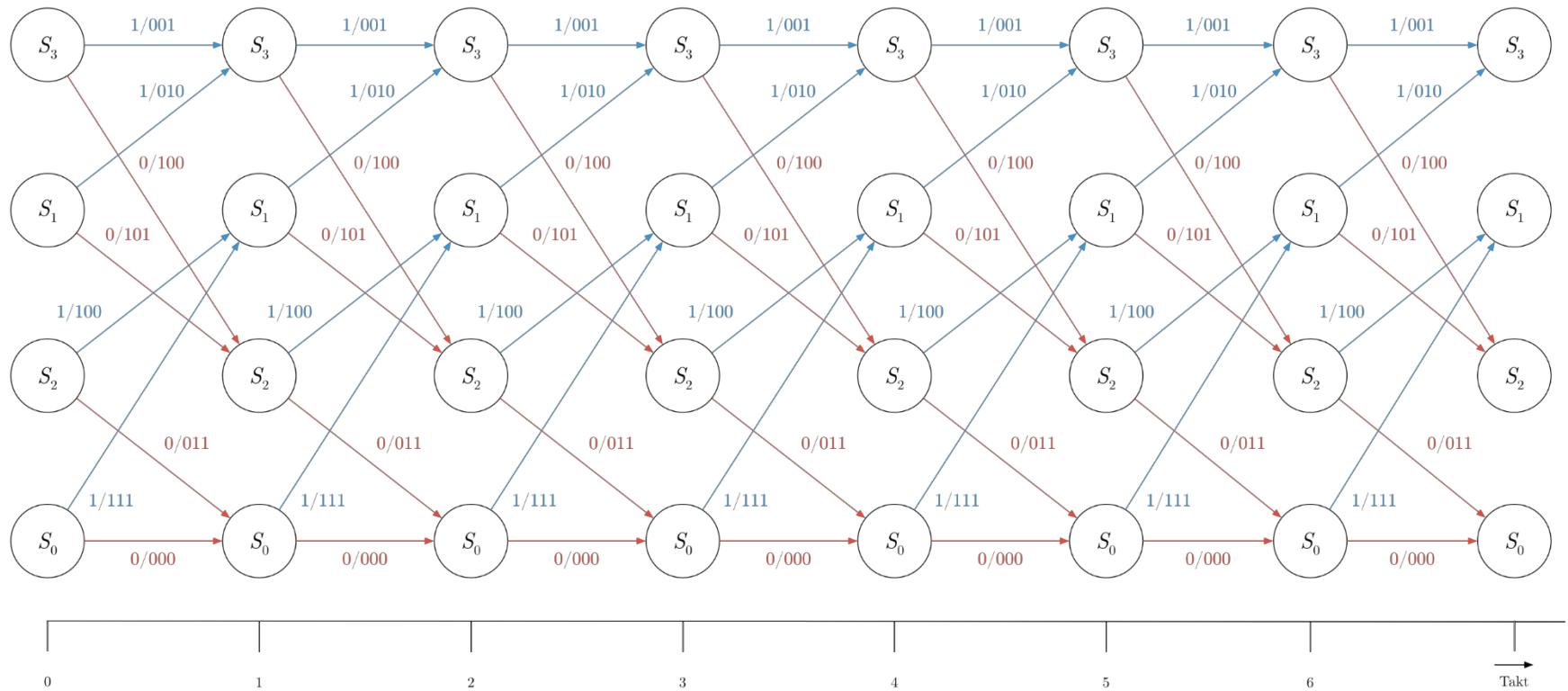
- **Fundamentalweg:** ist der (Teil-) Weg eines Codes, der im Zustand  $S_0$  beginnt und wieder im Zustand  $S_0$  endet. Die Analyse der Fundamentalwege liefert die Struktur des Faltungscodes.
- **Metrik:**
  - $I$ : bezeichne den Zustandsübergang, der durch «1» ausgelöst wird.
  - $D^l$ : bezeichne die Anzahl der durch den Übergang zur Codefolge hinzukommenden «1» Bitstellen (Gewichtszunahme).
  - $J$ : sei eine Zählvariable, die die Anzahl der Übergänge zählt.
  - Jede Kante eines Fundamentalweges lässt sich durch das Triplet  $(I D^l J)$  beschreiben.  
→ *Kantengewicht*

## Beispiel:

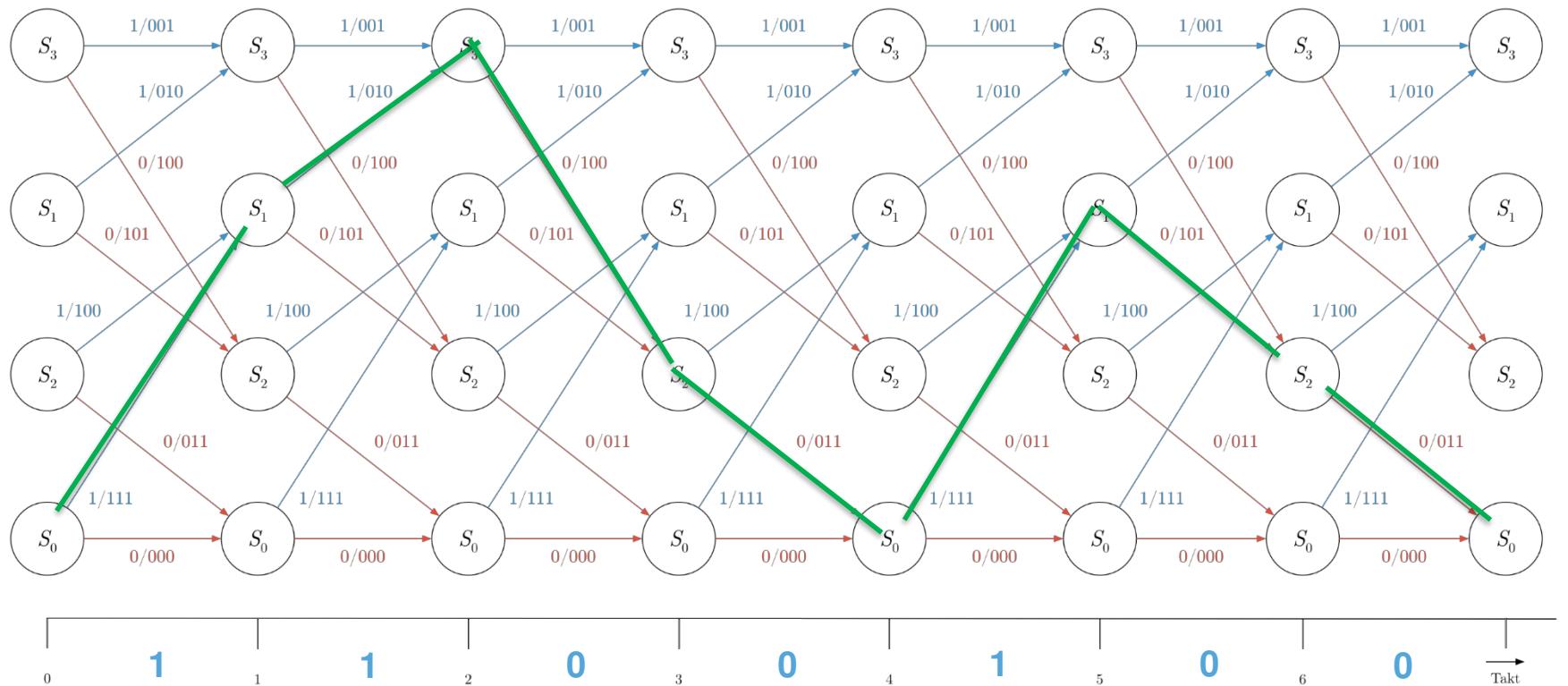
$$\{g[n]\} = \{1 0 0\} \Rightarrow ID^3J \cdot D^2J \cdot D^2J = ID^7J^3$$

: Gesamtgewicht der Folge

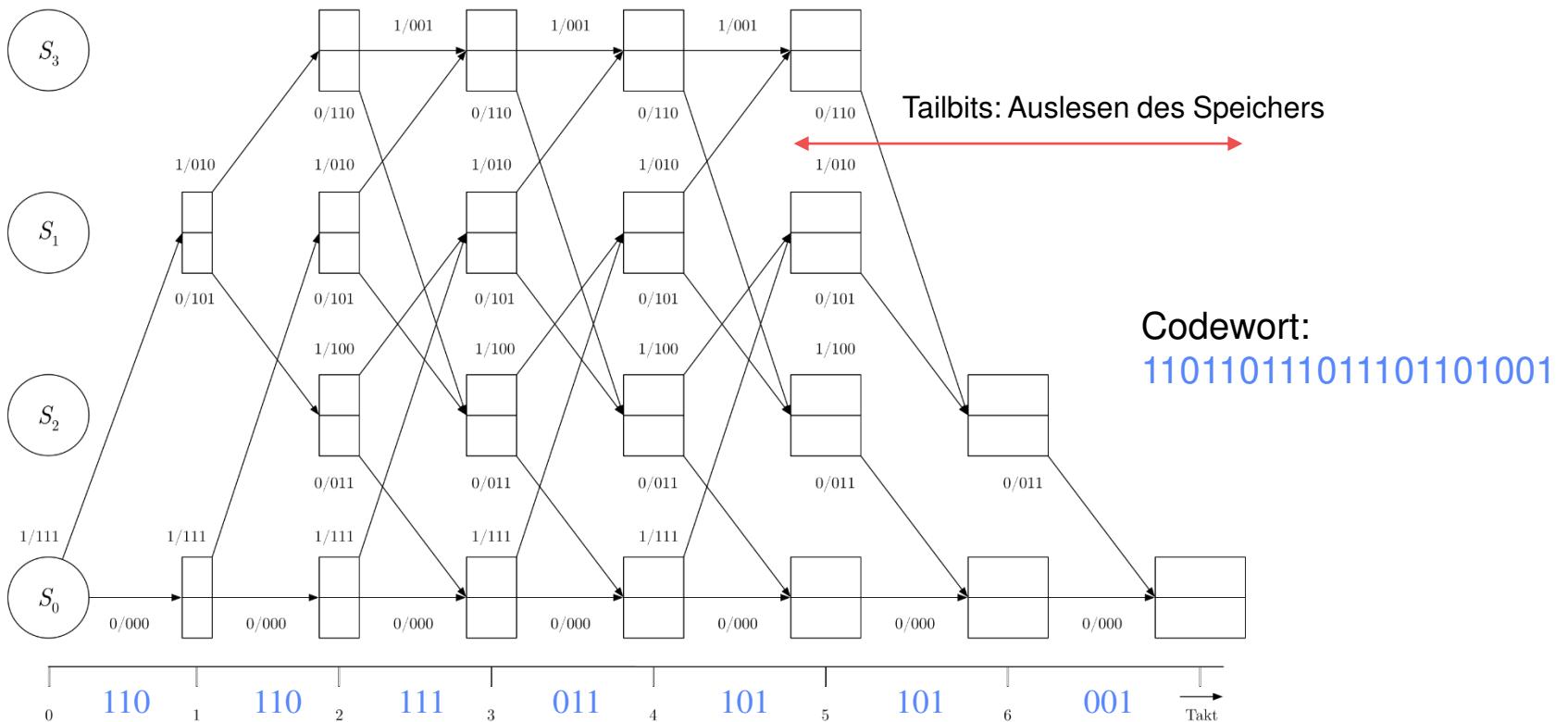
# Netz-/Trellisdiagramm des (3,1,2) Faltungscodes



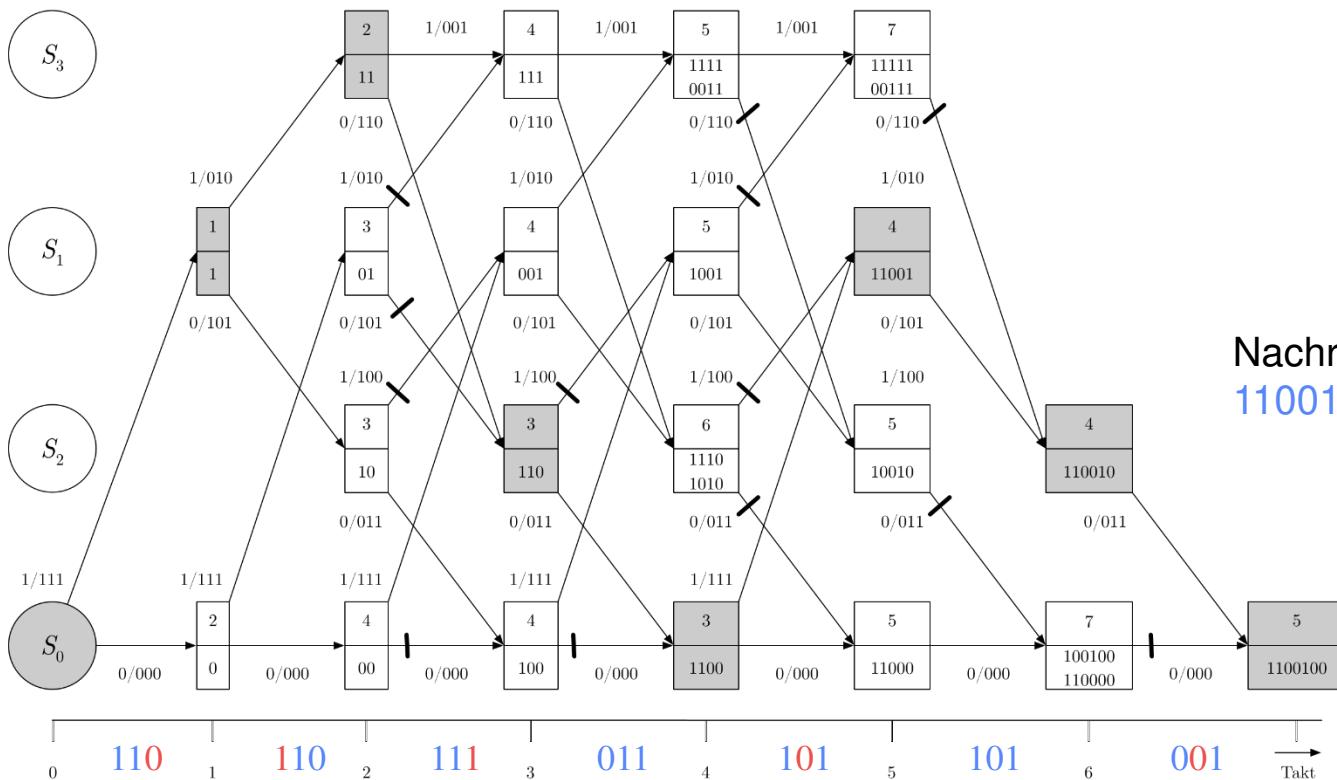
## Codierung des (3,1,2) Faltungscodes



# Decodierung des (3,1,2) Faltungscodes



# Decodierung des (3,1,2) Faltungscodes



# Generatorpolynome für optimale Faltungscodes

m	$R = \frac{1}{2}$			$R = \frac{1}{3}$				$R = \frac{1}{4}$				
	$g_1$	$g_2$	$d_f$	$g_1$	$g_2$	$g_3$	$d_f$	$g_1$	$g_2$	$g_3$	$g_4$	$d_f$
2	5	7	5	5	7	7	8	5	7	7	7	10
3	15	17	6	13	15	17	10	13	15	15	17	15
4	23	35	7	25	33	37	12	25	27	33	37	16
5	53	75	8	47	53	75	13	53	67	71	75	18
6	133	171	10	133	145	175	15	135	135	147	163	20
7	247	371	10	225	331	367	16	235	275	313	357	22
8	561	753	12	557	663	711	18	463	535	733	745	24

$g_i$  in oktaler  
Schreibweise

[Martin Werner, Information und Codierung, vieweg]

# Informations- und Codierungstheorie

## *8. Signale: Träger der Information und einfache Leitungscodes*

Computer Networks and



aF&E

Mobile Communication

Prof. Dr.-Ing. Andreas Rinkel  
[andreas.rinkel@hsr.ch](mailto:andreas.rinkel@hsr.ch)

Sprechstunde: Jeden Montag 16:00 bis 17:00, Raum: 6.110

Tel.: +41 (0) 55 2224928

Mobil: +41 (0) 79 3320562

<http://rinkel.ita.hsr.ch>

# Signal

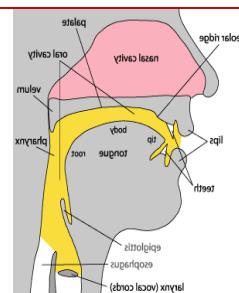
Intrinsische  
Information



Code

Lautfolge „Haus“

Erzeuge Lautfolge



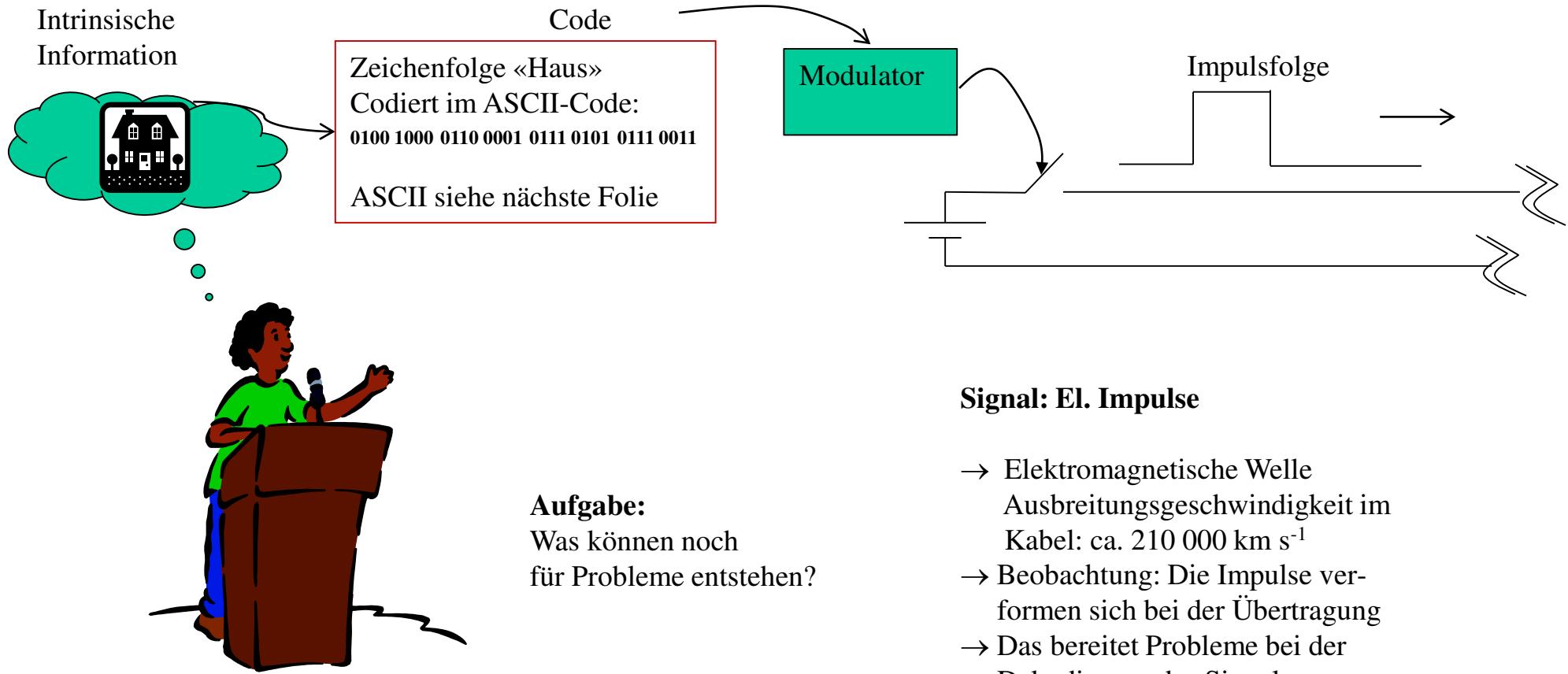
**Aufgabe:**  
Benennen sie andere  
**Code-**  
und  
**Signalarten!**

**Signal: Druckänderung der Luft**

→ Schallwelle Ausbreitungs-  
geschwindigkeit in Luft:  $330 \text{ m s}^{-1}$

- Abnahme der Wellenenergie : Quadratisch zur Entfernung
- Abnahme der Wellenamplitude: linear zur Entfernung

# Signal



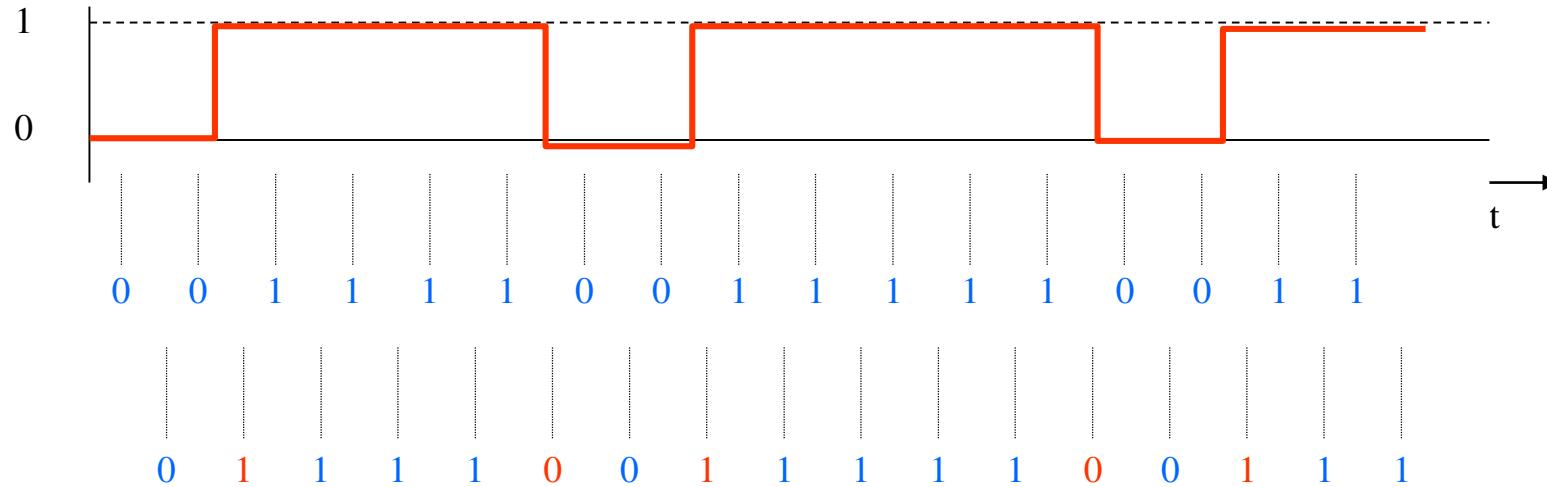
# ASCII-Tabelle zur Codierung

Non-Printing Characters				Printing Characters											
Name	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char			
null	0	0	NUL	32	20	Space	64	40	@	96	60	`			
start of heading	1	1	SOH	33	21	!	65	41	A	97	61	a			
start of text	2	2	STX	34	22	"	66	42	B	98	62	b			
end of text	3	3	ETX	35	23	#	67	43	C	99	63	c			
end of xmit	4	4	EOT	36	24	\$	68	44	D	100	64	d			
enquiry	5	5	ENQ	37	25	%	69	45	E	101	65	e			
acknowledge	6	6	ACK	38	26	&	70	46	F	102	66	f			
bell	7	7	BEL	39	27	'	71	47	G	103	67	g			
backspace	8	8	BS	40	28	(	72	48	H	104	68	h			
horizontal tab	9	9	HT	41	29	)	73	49	I	105	69	i			
line feed	10	0A	LF	42	2A	*	74	4A	J	106	6A	j			
vertical tab	11	0B	VT	43	2B	+	75	4B	K	107	6B	k			
form feed	12	0C	FF	44	2C	,	76	4C	L	108	6C	l			
carriage feed	13	0D	CR	45	2D	-	77	4D	M	109	6D	m			
shift out	14	0E	SO	46	2E	.	78	4E	N	110	6E	n			
shift in	15	0F	SI	47	2F	/	79	4F	O	111	6F	o			
data line escape	16	10	DLE	48	30	0	80	50	P	112	70	p			
device control 1	17	11	DC1	49	31	1	81	51	P	113	71	q			
device control 2	18	12	DC2	50	32	2	82	52	R	114	72	r			
device control 3	19	13	DC3	51	33	3	83	53	S	115	73	s			
device control 4	20	14	DC4	52	34	4	84	54	T	116	74	t			
neg acknowledge	21	15	NAK	53	35	5	85	55	U	117	75	u			
synchronous idle	22	16	SYN	54	36	6	86	56	V	118	76	v			
end of xmit block	23	17	ETB	55	37	7	87	57	W	119	77	w			
cancel	24	18	CAN	56	38	8	88	58	X	120	78	x			
end of medium	25	19	EM	57	39	9	89	59	Y	121	79	y			
substitute	26	1A	SUB	58	3A	:	90	5A	Z	122	7A	z			
escape	27	1B	ESC	59	3B	;	91	5B	[	123	7B	{			
file separator	28	1C	FS	60	3C	<	92	5C	\	124	7C				
group separator	29	1D	GS	61	3D	=	93	5D	]	125	7D	}			
record separator	30	1E	RS	62	3E	>	94	5E	^	126	7E	~			
unit separator	31	1F	US	63	3F	?	95	5F	_	127	7F	DEL			

H	a	u	s
48 <sub>H</sub>	61 <sub>H</sub>	75 <sub>H</sub>	73 <sub>H</sub>
0100 1000	0110 0001	0111 0101	0111 0011

# Leitungscodierung: Signale im Basisband

## Probleme!



Anforderungen:

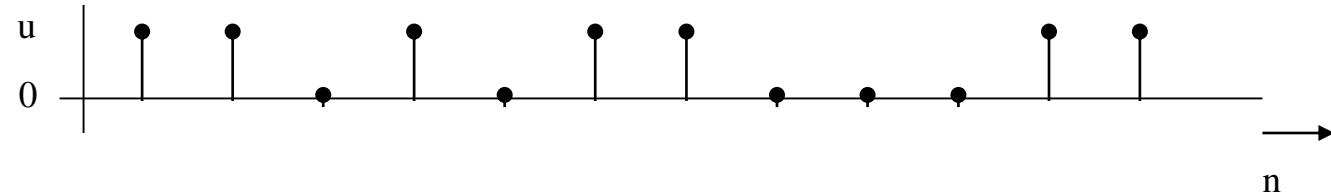
- Unterstützung der Takt- und Phasenrückgewinnung im Empfänger
- Vermeidung von Gleichstromkomponenten  
Übertrager in den Kupferleitungen, Ruheströme, ...
- Optimierung des Bandbreitenbedarfs
- Unempfindlichkeit gegenüber Störungen

Basisband: Signale im Frequenzbereich 0 Hz bis  $f_{og}$

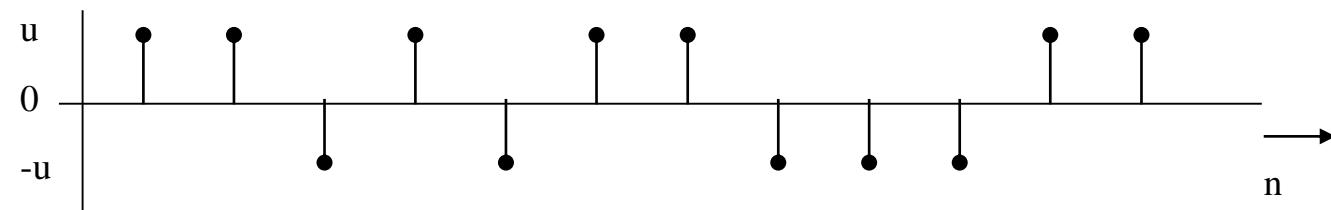
# Klassifizierung binärer Leitungscodes

## 1. Polarität

unipolares Signalv

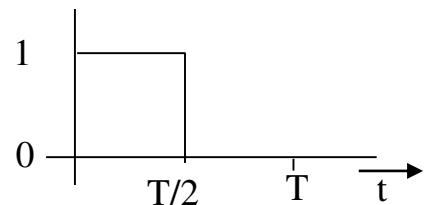


bipolares Signal

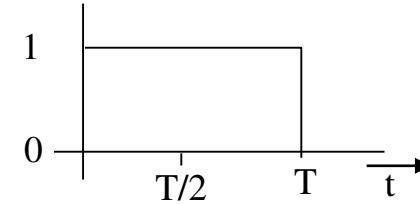


## 2. Impulsform

RZ-Implus (Return to Zero)



NRZ-Implus (no return to zero)

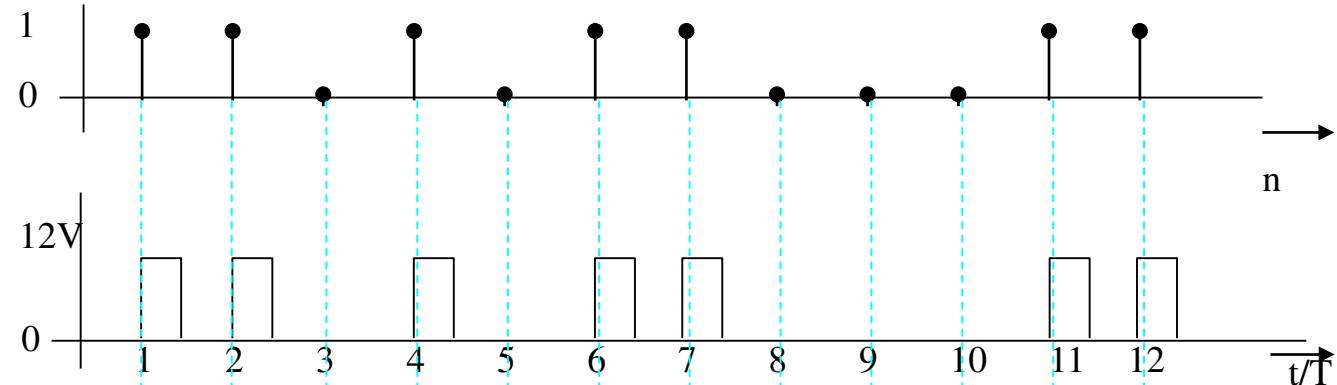


# Beispiele I

1.

Unipolare  
RZ Codierung

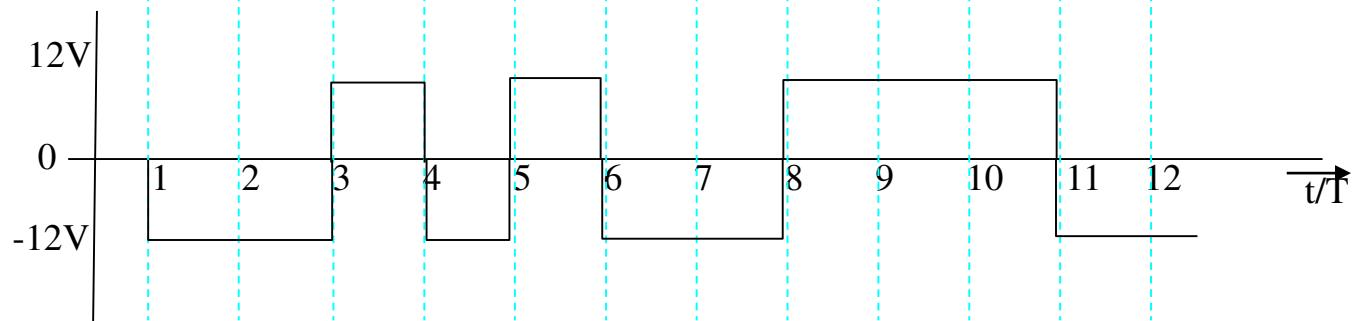
Nachteil:  
hoher Gleichstromanteil



2.

Bipolare  
NRZ Codierung  
z.B. die Schnittstellen:  
CCITT V.24/V.28  
RS 232

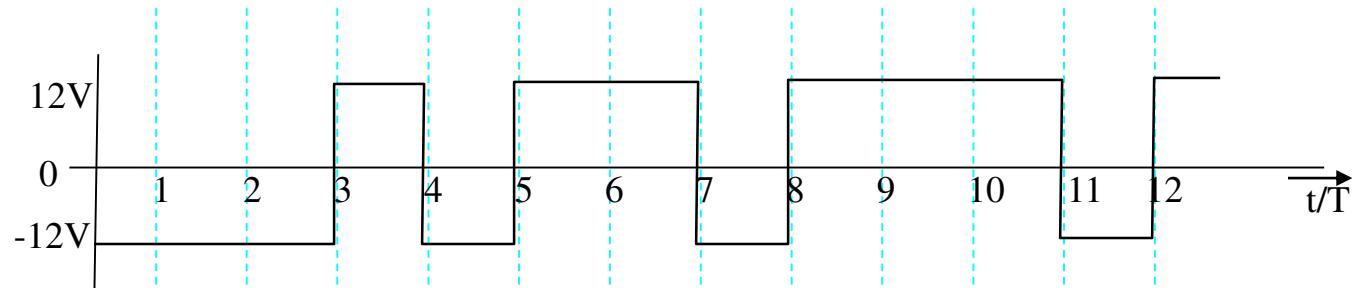
geringer Gleichstromanteil  
Problem: lange „1“ bzw „0“ Folgen



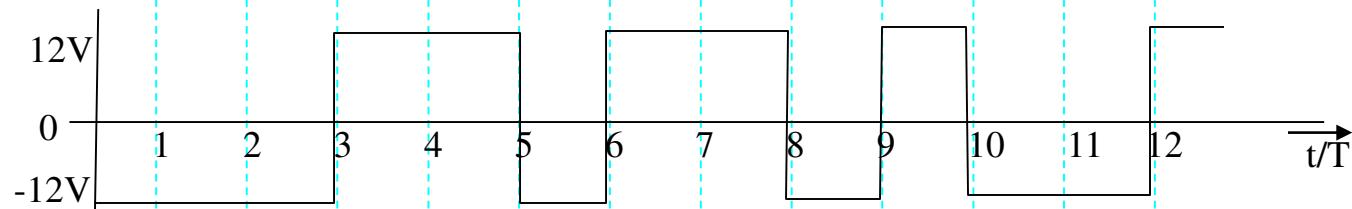
# Beispiele II: Verfeinerung NRZ

Bitmuster 0 0 0 1 0 1 1 0 1 1 1 0 1

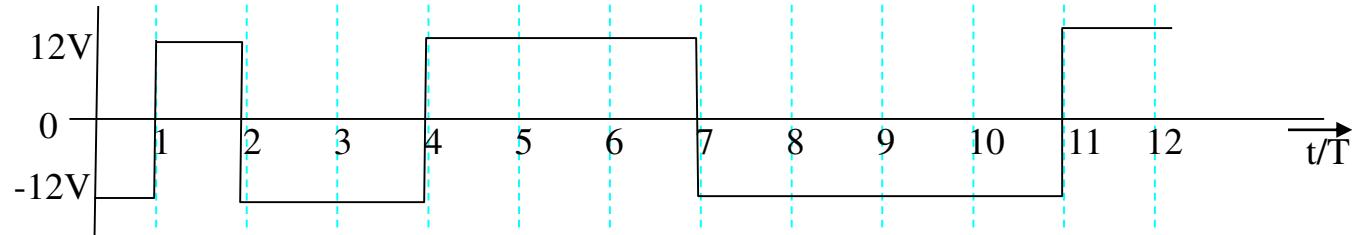
NRZ-L  
(Level)



NRZ-M  
(Mark, 1)  
eine „1“ ändert die Polarität



NRZ-S  
(Space, 0)  
eine „0“ ändert die Polarität

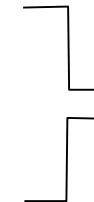
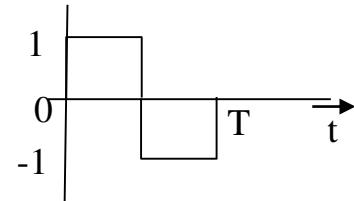


# Beispiele III: Manchester – Code

## Grundverfahren

Verwendeter Impuls

- **Kein Gleichstromanteil**
- Jedoch **doppelte Bandbreite**
- Anwendung: **Ethernet-Systeme**
- erlaubt **Takt-Rückgewinnung**

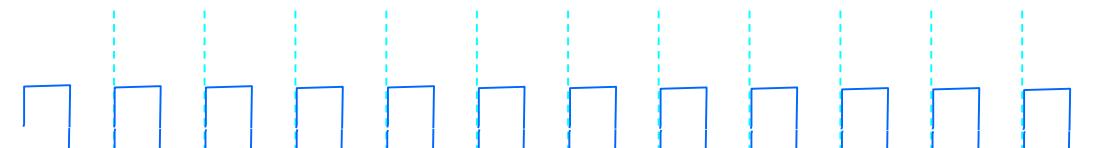


„0“ : fallende Flanke

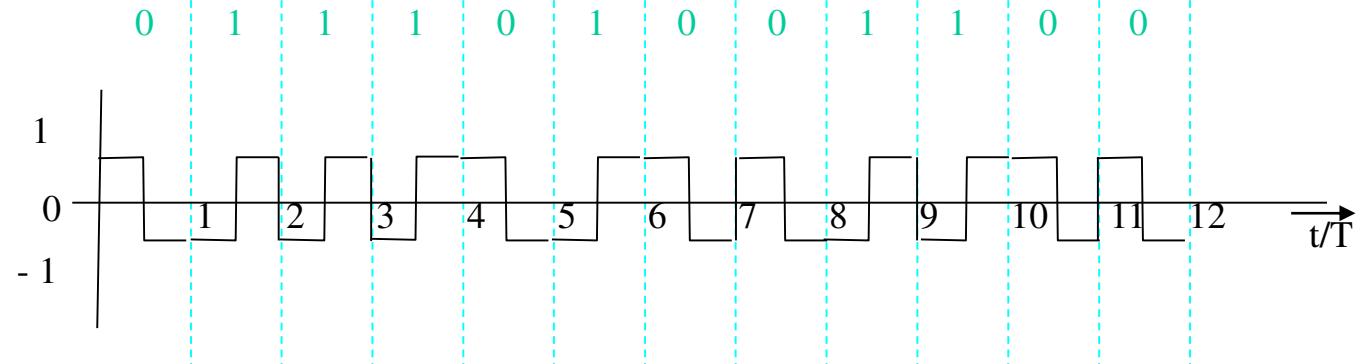


„1“ : steigende Flanke

Takt



Bitmuster



Manchestercodierte  
Pulsfolge

# Beispiele III: AMI – Code

AMI: Alternate Mark Inversion  
Grundverfahren

Merkmale:

- **Kein** Gleichstromanteil
- erlaubt **Takt-Rückgewinnung**  
-> Problem **lange Nullfolgen**
- **modifizierten AMI-Code** Umkehrung der Codierung für „1“ und „0“
- Anwendung:  
modifizierter AMI-Code beim **ISDN-S<sub>0</sub>-Bus**

„0“ : wird durch den 0-Pegel repräsentiert

„1“ : wird abwechselnd durch einen pos. und neg.  
Impuls dargestellt

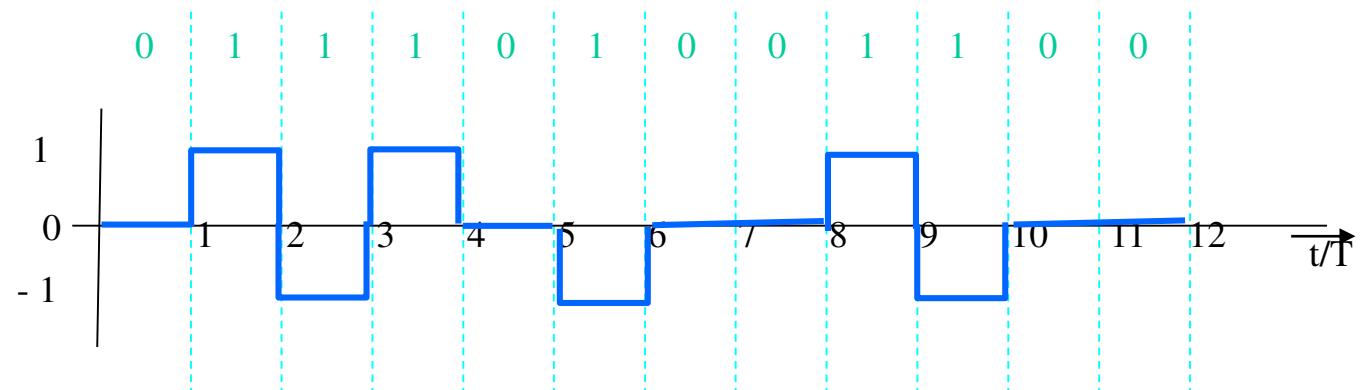
Besonderheit:

Drei physikalische Pegel zur Darstellung von  
zwei logische Werten.

Daher wird der AMI-Code auch als pseudoternärer Code  
bezeichnet.

Bitmuster

AMI-codierte  
Pulsfolge



# Ein elektrisches Signal

Wie kann ein el. Signal beschrieben werden?

Was sind die Parameter eines Signals?

Bestehen alle Signale aus dem gleichen "Stoff"?

Was kann einem Signal auf dem Weg zum Ziel passieren?

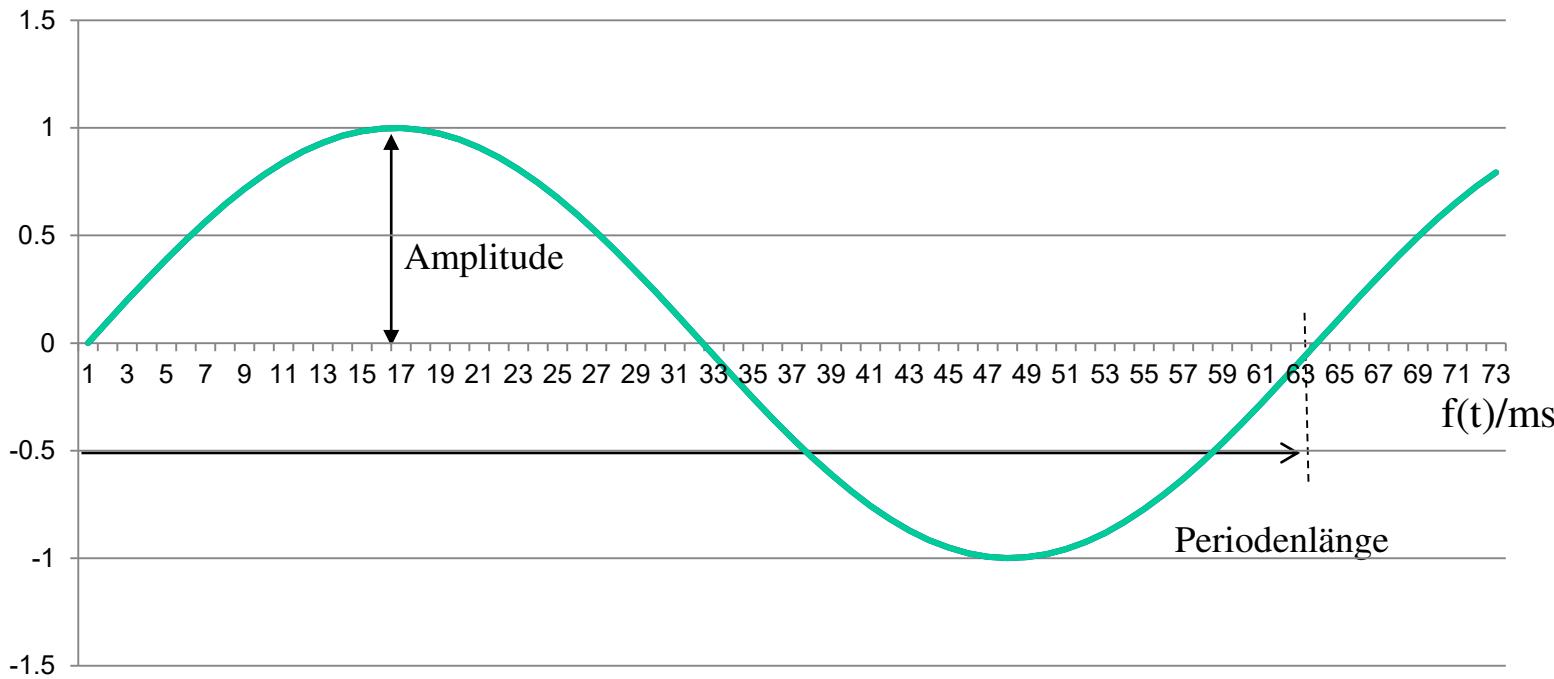
Es kann

gedämpft werden

überlagert werden

gefiltert werden

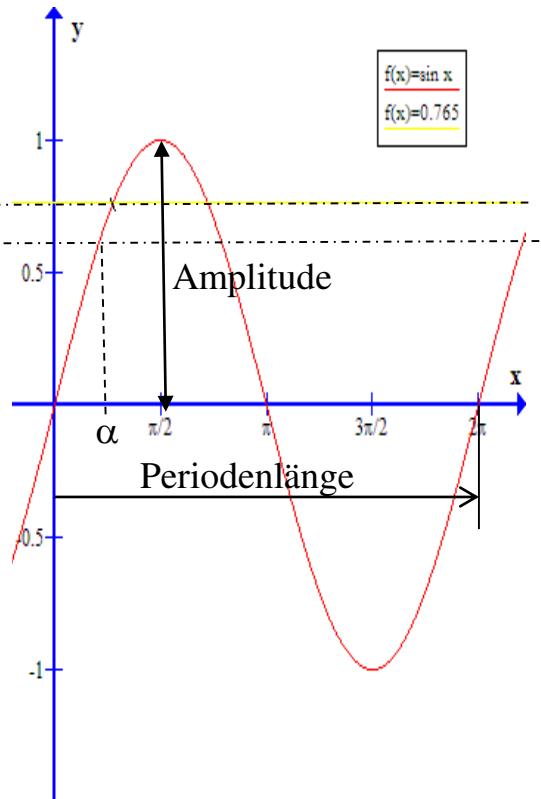
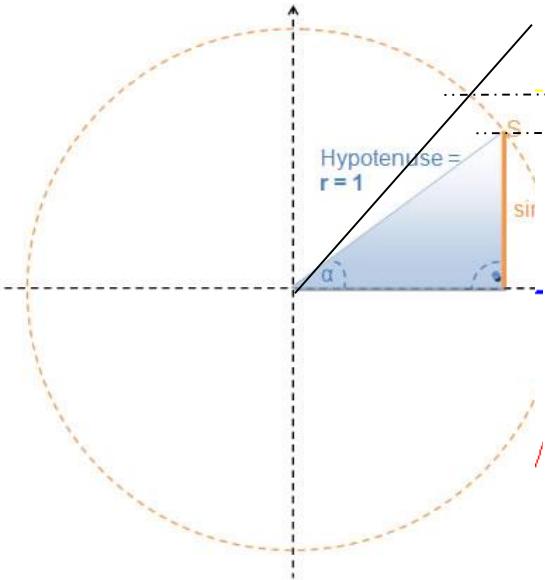
# Kenngrößen eines Signals



Frequenz = ?

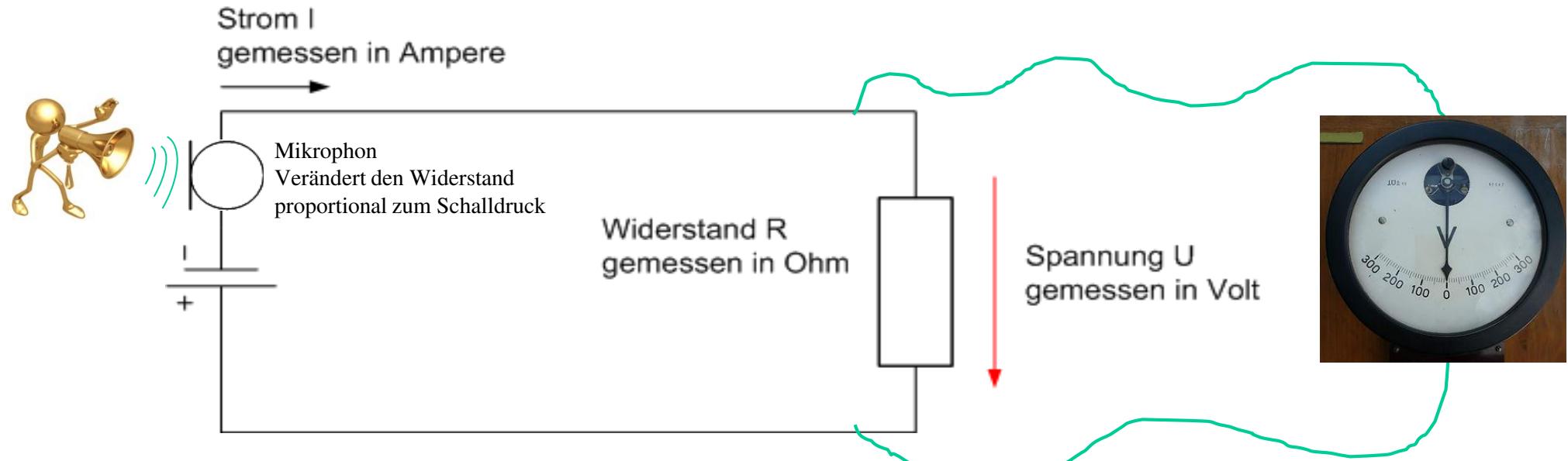
# Kenngrößen eines Signals

## Basiskenntnisse: Sinus und Cosinus



Ein 440 Hz Schallsignal sieht zwar genauso aus, ist aber eine Funktion der Zeit  
und kann beliebige Amplitudenwerte haben. Wie kann man das überführen?  
 $f(t) = ?$

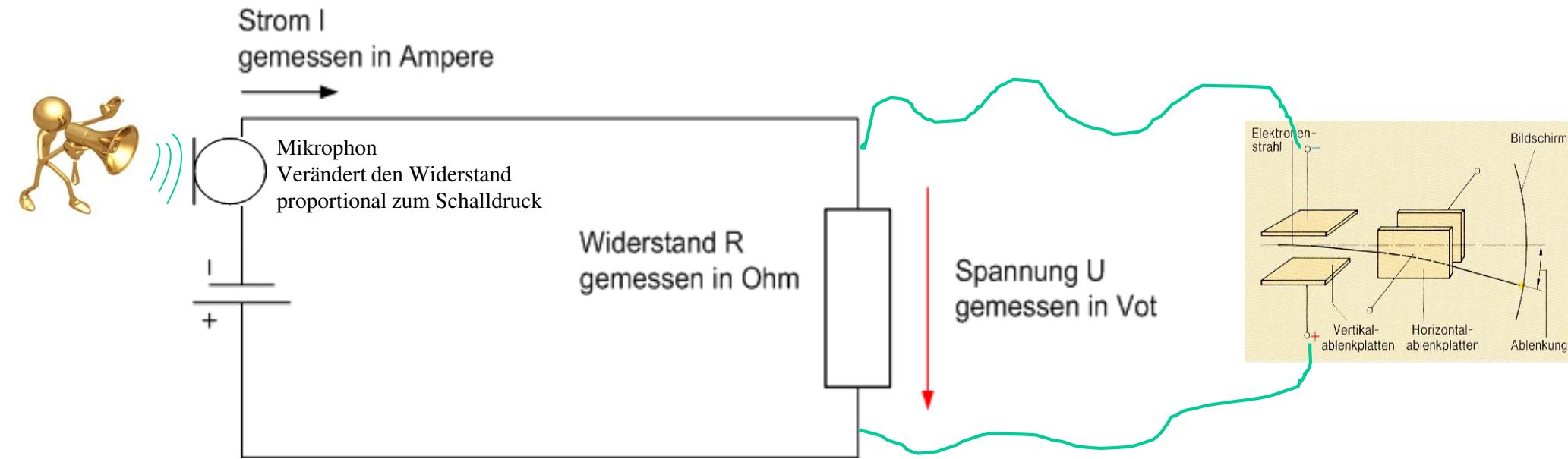
# Sichtbarmachen und Messen elektrischer Spannungen



Problem: Ist die Widerstandsänderung des Mikrofons zu schnell, ist das Zeigerinstrument zu träge.

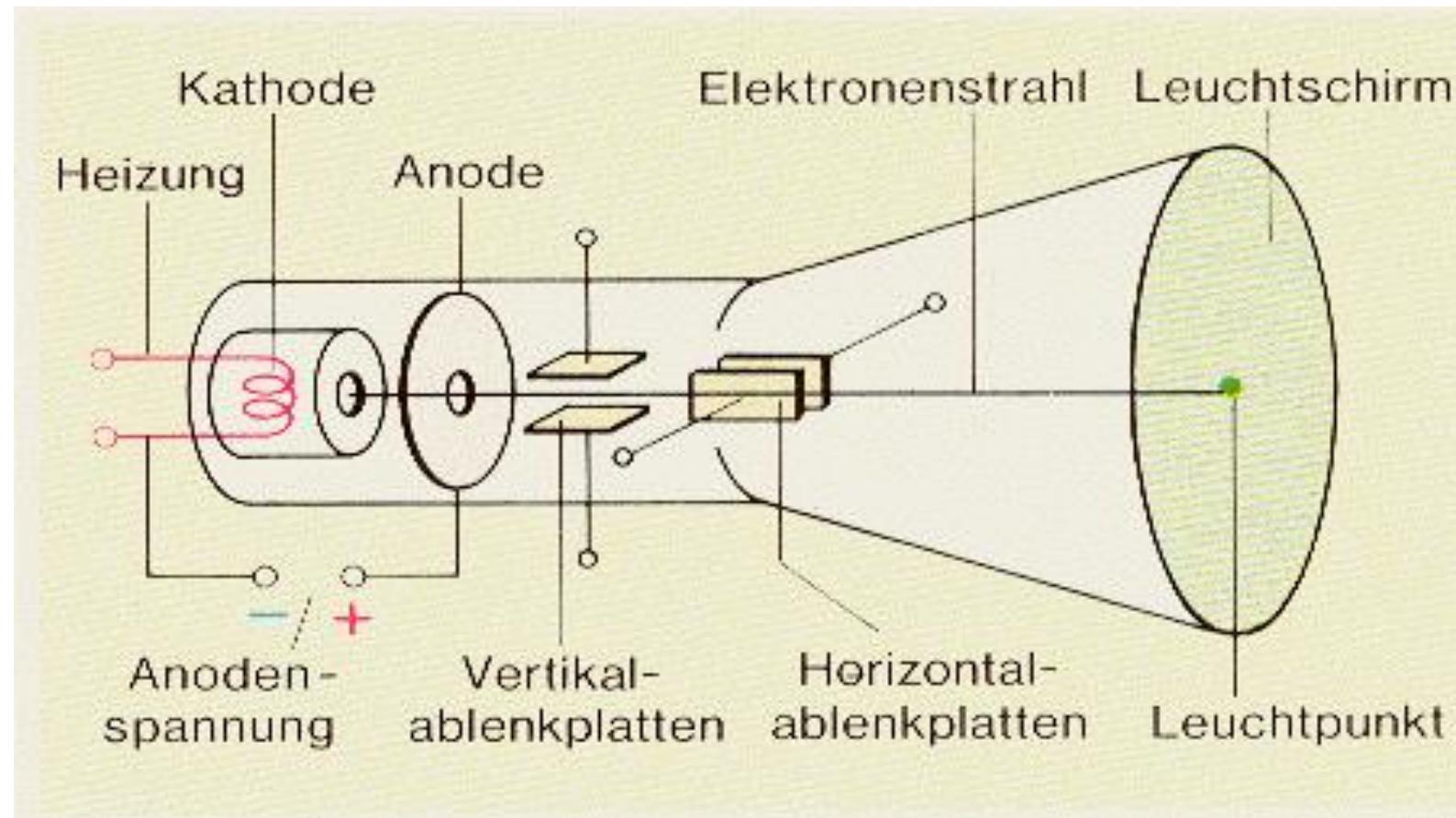
Wunsch: Spannungsverlauf bzw. Schalldruck über der Zeit anschauen!

# Sichtbarmachen und Messen elektrischer Spannungen

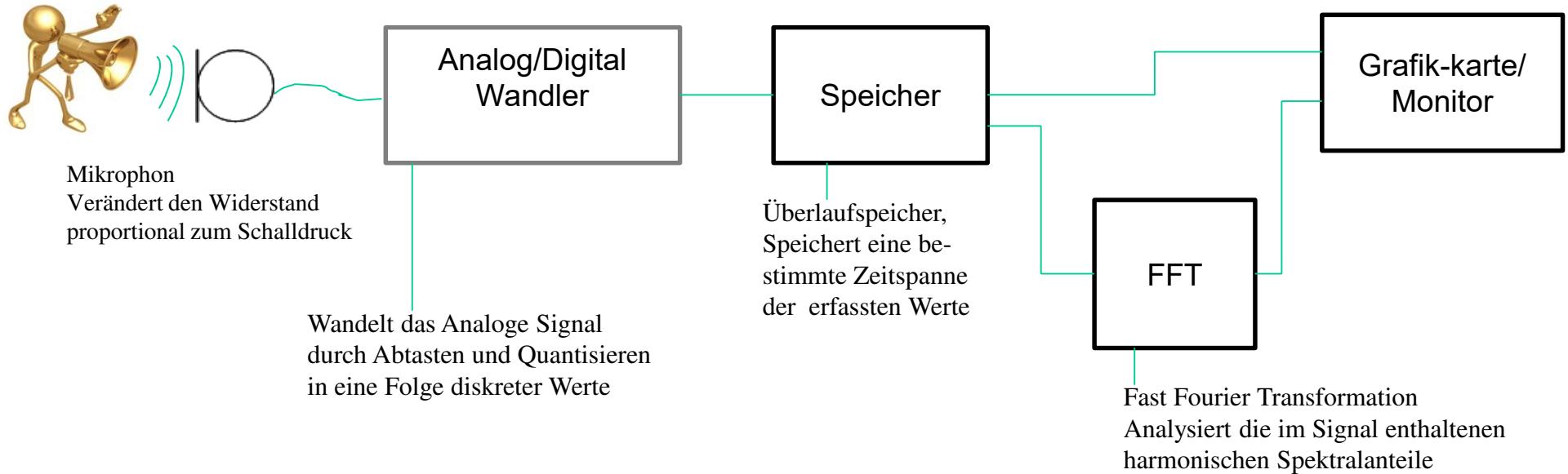


- Zeiger gegen fast masselose Elektronen ersetzen
- schnelle Änderungen Darstellbar, vertikale Ablenkung
- Ablauf über der Zeit durch horizontale Ablenkung
- Bildschirm kann durch Leuchtschicht auftreffende Elektronen darstellen

# Prinzip des Oszilloskop

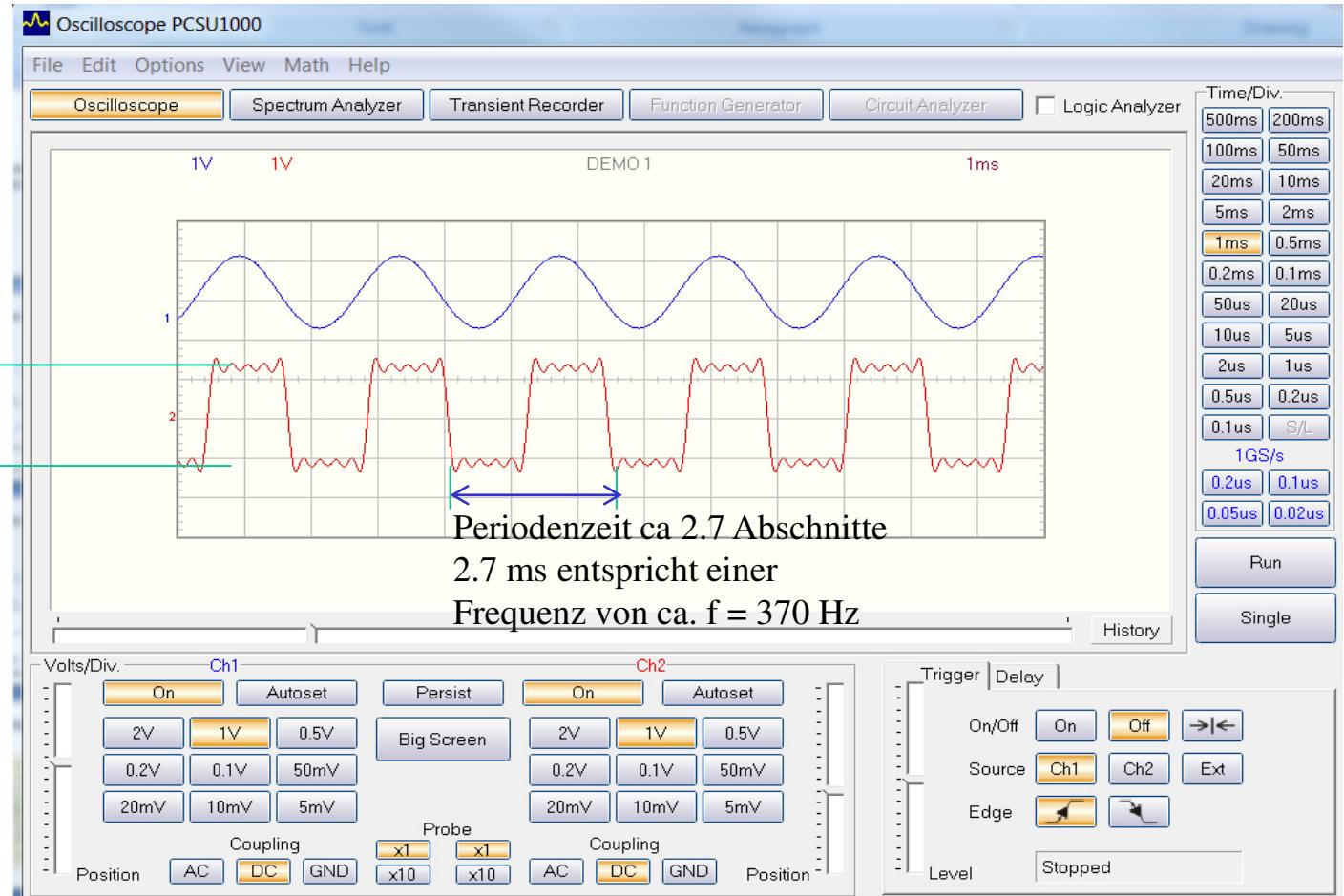


# Prinzip des digitalen Speicher-Oszilloskop

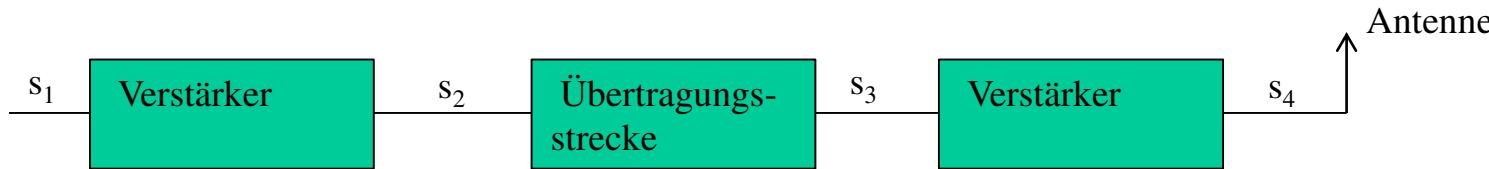


# Übersicht PCSU1000

Amplitude Spitz – Spitz  
 ca 2.3. Stufen das ent-spricht  
 bei gegebener Ein- stellung  
 einem Spannungs-wert von 2.3  
 Volt



# Was ist ein dB?



Sind die einzelnen Komponenten linear so gilt:

$$\frac{s_4}{s_1} = \frac{s_2}{s_1} * \frac{s_3}{s_2} * \frac{s_4}{s_3}$$

Bell wird definiert durch den Logarithmus zur Basis 10 von Ausgangssignal zu Eingangssignal. Ein Bell sind 10 dB

$$s_{Bel} = \log_{10} \frac{s_{out}}{s_{in}}$$

$$1 Bel = 10 dB$$

Beispiel  $s_2 = 20, s_1 = 10$

$$\log_{10} \frac{20}{10} = 0.301 Bell \approx 3 dB$$

Einheit (ITU)	Bedeutung
dBu	Spannungspegel mit der Bezugsgröße $\approx 0.775$ Volt
dBV	Spannungspegel mit der Bezugsgröße 1 V
dBA	A-bewerteter Schalldruckpegel 20 $\mu$ Pa
dBm	A-bewerteter Schallleistungspegel 1 pW
dBm	Leistungspegel mit der Bezugsgröße 1 mW
dBW	Leistungspegel mit der Bezugsgröße 1 W
dB $\mu$	Pegel der elektrischen Feldstärke mit der Bezugsgröße 1 $\mu$ V/m

# Experimente: DasyLab

## Ergebnisse

- Signale gleicher Form aber unterschiedlicher Phase hören sich gleich an sinus, cosinus
- Signale unterschiedlicher Form aber gleicher Frequenz hören sich anders an z.B. hört sich das Rechtecksignal „höher“ an als das Sinussignal
- Welchen Einfluss hat ein Filter?
- Durch einen Filter kann aus einem Rechtecksignal ein Sinussignal gemacht werden
- Durch eine Filterbank kann ein beliebiges Signal in seine Spektralanteile zerlegt werden
- Annahme: in der Realität existieren nur harmonische Schwingungen (sinus bzw. cosinusförmige Signale) alle anderen wahrgenommenen Signale setzen sich letztlich aus diesen Signalen zusammen!
- Jedes beliebige Signal kann durch die Addition von harmonischen Signalen erzeugt werden.

**Schluss:** Wir können jedes beliebige Signal auf zwei gleichwertige Arten beschreiben:

1. **Durch die Periodendauer und Amplitude**  
→ **Zeitbereich**
2. **Durch die Frequenzanteile, die Amplitude sowie deren Phasengang**  
→ **Frequenzbereich**

# Experimente: DasyLab

## Schluss:

- Alle Signale scheinen nur aus sinus- und cosinusförmigen Teilsignalen zu bestehen.
- Wir können jedes beliebige Signal auf zwei gleichwertige Arten beschreiben:
  1. **Durch die Periodendauer und Amplitude**  
→ Zeitbereich
  2. **Durch die Frequenzanteile, die Amplitude sowie deren Phasengang**  
→ Frequenzbereich

Wie kann aus der zeitlichen Beschreibung die Frequenzdarstellung ermittelt werden?



## **Joseph Fourier**

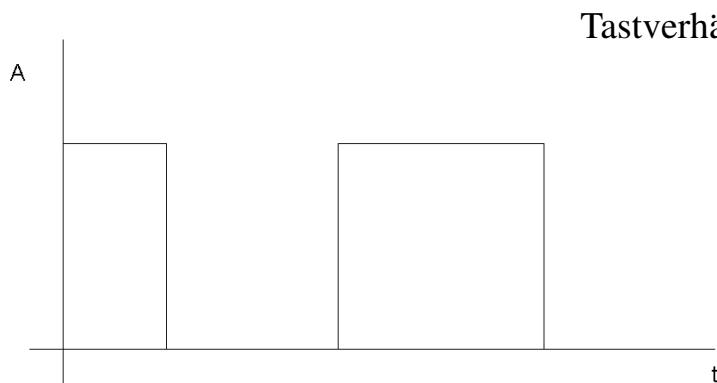
französischer Mathematiker und Physiker  
1768 – 1830

Begründet unter anderem die

- Fourieranalyse und
- Fouriertransformation

# Darstellung von Signalen II

Zeitbereich



Tastverhältnis 0.5

$$f(x) = \sum_{n=0}^{\infty} a_n \cos(nx) + b_n \sin(nx)$$

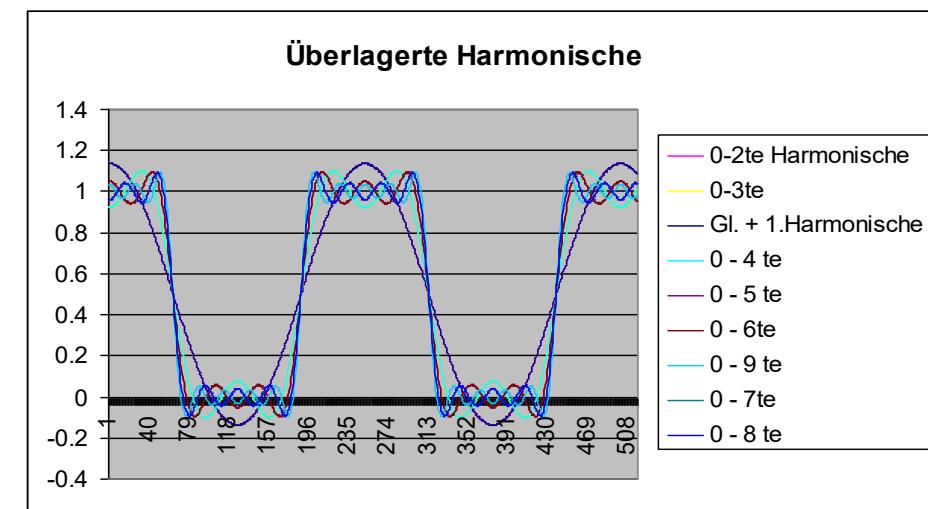
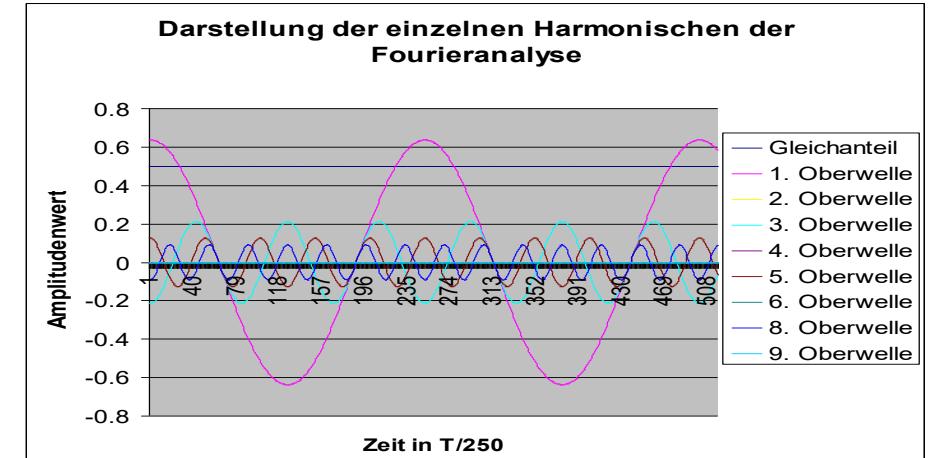
mit

$$a_0 = \frac{1}{2\pi} \int_a^{a+2\pi} f(x) dx$$

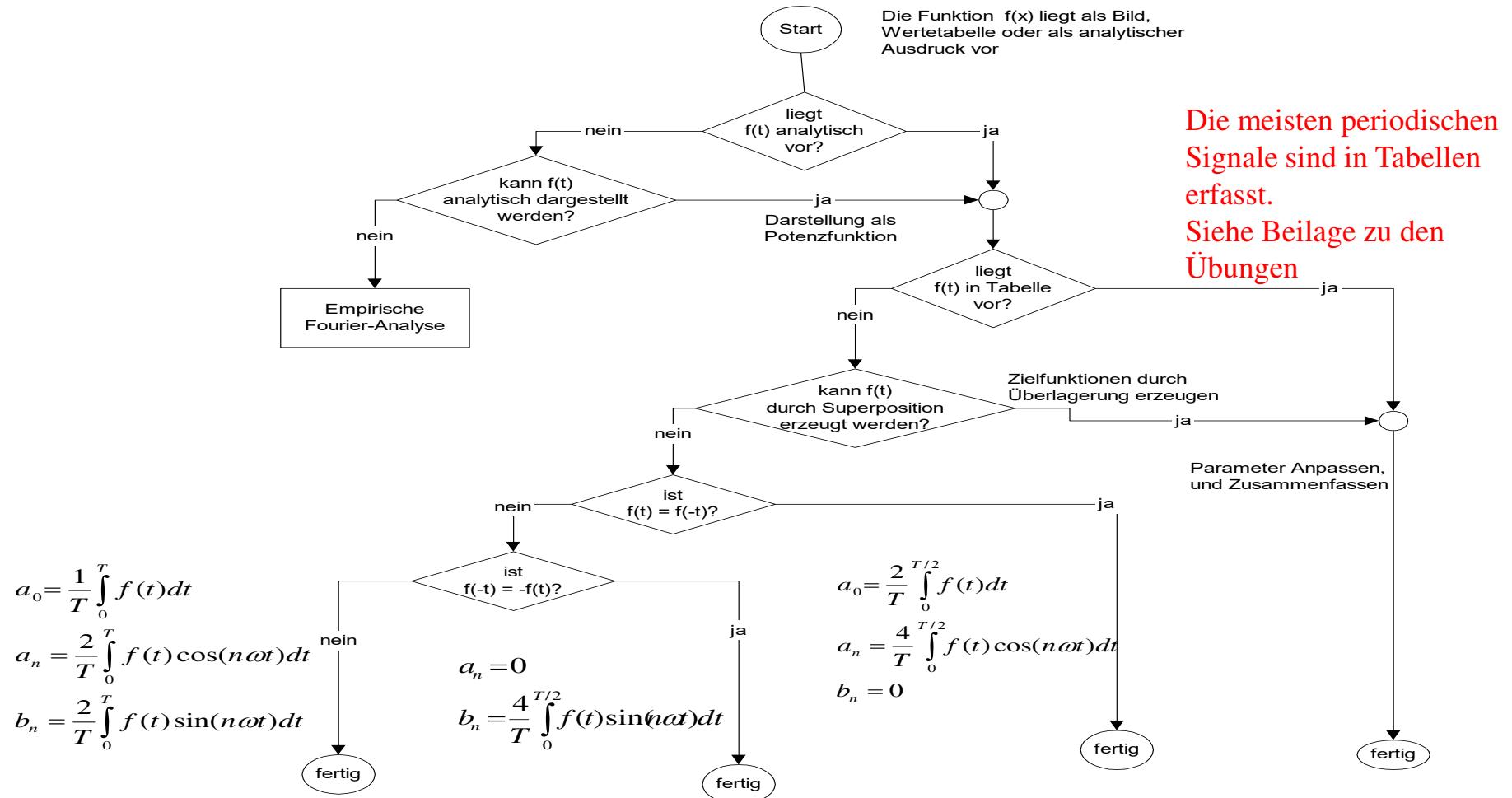
und

$$a_n = \frac{1}{\pi} \int_a^{a+2\pi} f(x) \cos(nx) dx, \quad n \in N$$

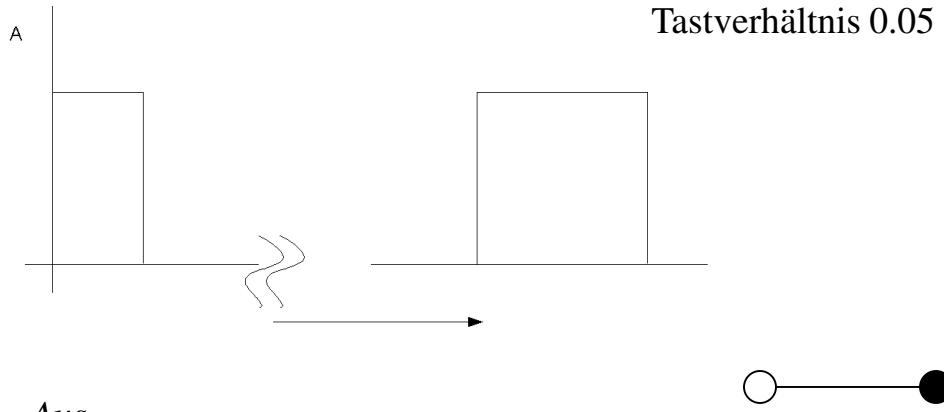
$$b_n = \frac{1}{\pi} \int_a^{a+2\pi} f(x) \sin(nx) dx, \quad n \in N$$



# Vorgehensweise in der Fourier-Analyse



# Darstellung von Signalen III



Aus

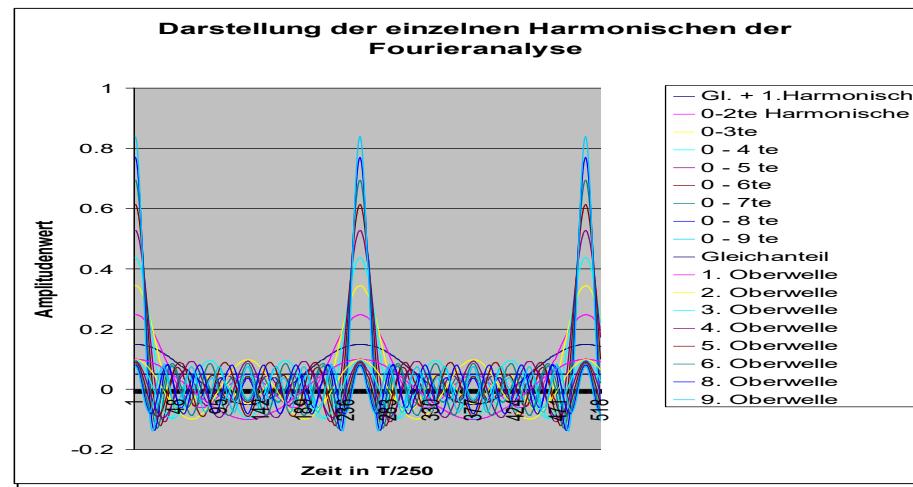
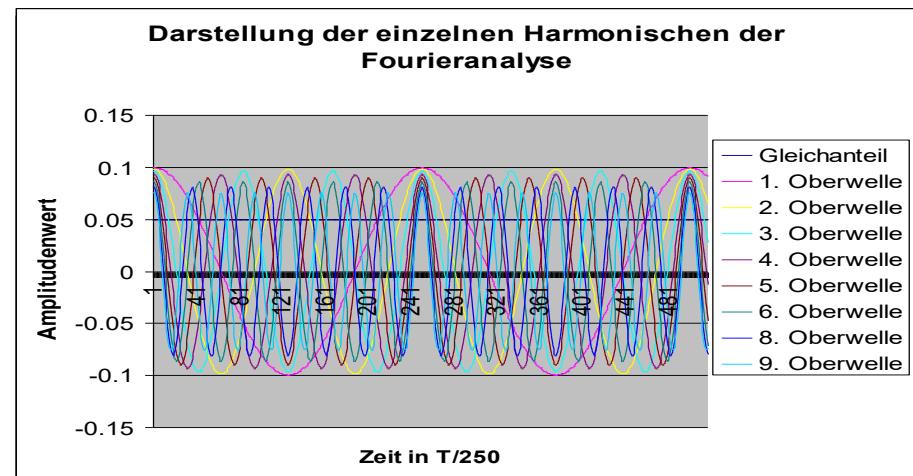
$$f(t) = \sum_{n=-\infty}^{\infty} \left( \frac{1}{T} \int_0^T f(\omega t) e^{-jn\omega t} dt \right) e^{-jn\omega t},$$

folgt

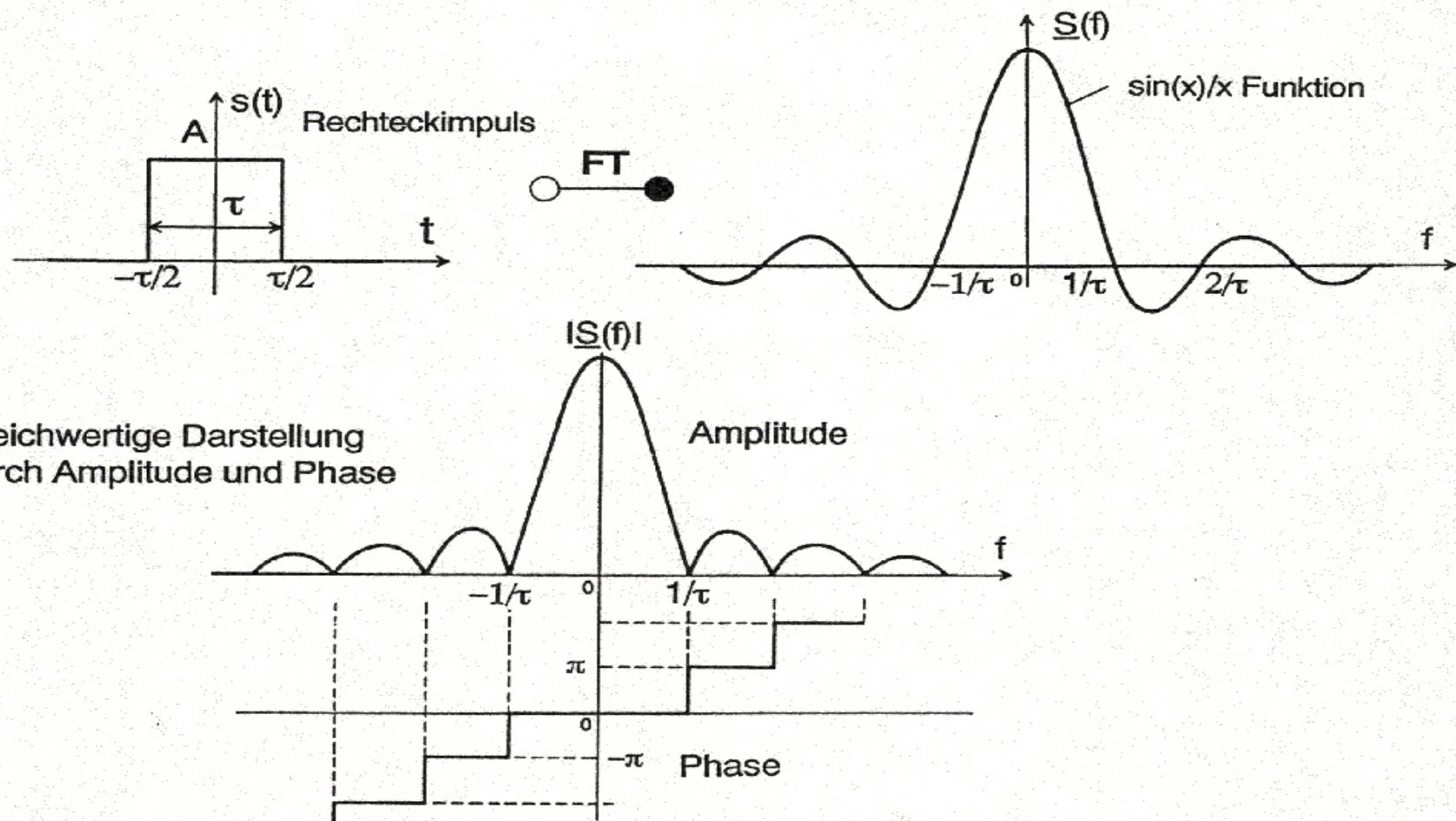
mit  $c_n T_0 \rightarrow S(f)$  und  $f_0 = \frac{1}{T_0} \rightarrow df$  und  $n f_0 \rightarrow f$

$$S(f) = \int_{-\infty}^{+\infty} s(t) e^{-j\omega t} dt \quad \text{Fouriertransformation}$$

$$s(t) = \int_{-\infty}^{+\infty} S(f) e^{j\omega t} df \quad \text{Inverse Fouriertransformation}$$



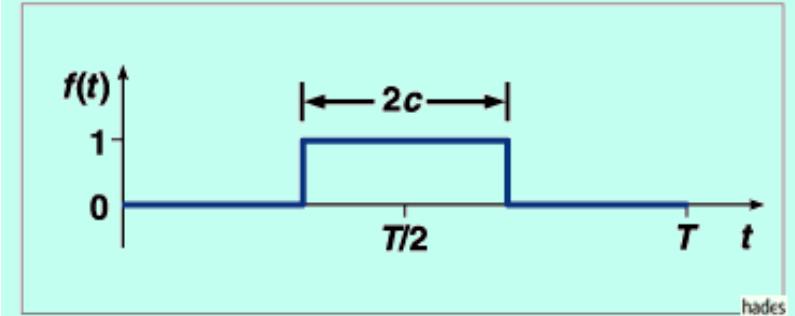
# Darstellung von Signalen IV



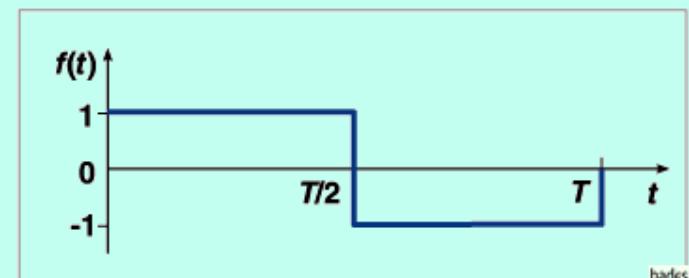
# Tabelle: Beispiel

aus Stöcker, Taschenbuch mathematischer Formeln und moderner Verfahren, Verlag Harri Deutsch, 2007

$$f(t) = \frac{2c}{T} + \frac{2}{\pi} \sum_{n=1}^{\infty} \frac{(-1)^n}{n} \sin \frac{2n\pi c}{T} \cos \frac{2n\pi t}{T}$$



$$f(t) = \frac{4}{\pi} \sum_{n=1,3,5,\dots}^{\infty} \frac{1}{n} \sin \frac{2n\pi t}{T}$$



# Informations- und Codierungstheorie

## *9. Signale und Sprache: Transponieren, Modulieren, Multiplexen*

Computer Networks and



aF&E

Mobile Communication

Prof. Dr.-Ing. Andreas Rinkel  
[andreas.rinkel@hsr.ch](mailto:andreas.rinkel@hsr.ch)

Sprechstunde: Jeden Montag 16:00 bis 17:00, Raum: 6.110

Tel.: +41 (0) 55 2224928

Mobil: +41 (0) 79 3320562

<http://rinkel.ita.hsr.ch>

# Experimentergebnisse: DasyLab

## Antworten:

- Was passiert, wenn zwei harmonische Signale mit einander multipliziert werden?
  - Es entstehen zwei neue Signale mit den Frequenzen
    - $f_1 = f_t - f_n$  und
    - $f_2 = f_t + f_n$
  - $f_1$  wird auch als Kehrlage bezeichnet. Warum?
  - $f_2$  wird auch als Regellage bezeichnet. Warum?
- Das kann genutzt werden, um z.B.
  - ein Sprachsignalsignal zu transponieren, Heliumeffekt,
  - Fledermausschreie hörbar zu machen,
  - die Ressource Frequenz, mehrfach zu nutzen, Frequenzmultiplex. Wie?

# Experimentergebnisse: DasyLab

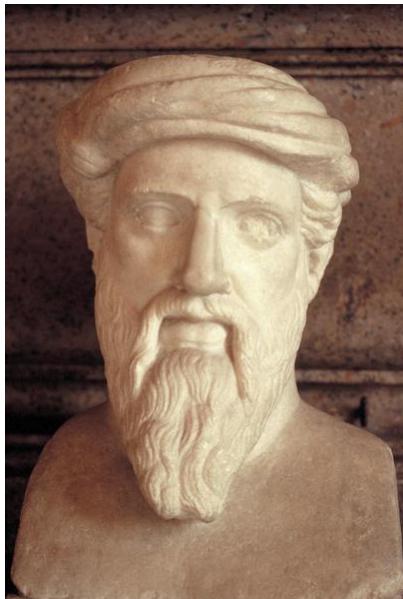
## Antworten:

- Was passiert, wenn das Produkt zweier Signale wieder multipliziert wird und welche Rolle spielt die Phase des aufmodulierten Signals?
  - Ein Teil wird zurück ins Basisband transponiert, ein anderer Teil noch weiter nach oben.
  - Durch einen Tiefpass kann anschliessend das ursprüngliche Signal zurück gewonnen werden.
  - Eine Phasenverschiebung des auf modulierten Signales kann zur Auslöschung führen.
  - Um eine Phasenverschiebung zu vermeiden, wird das Trägersignal mit übertragen.
- Welchen Einfluss haben die Frequenz, die Amplitude der Signale?
  - sie bestimmen die Weite der Verschiebung und den Grad der Verstärkung bzw. Dämpfung
- Kann das auch berechnet werden?

*aber klar !*

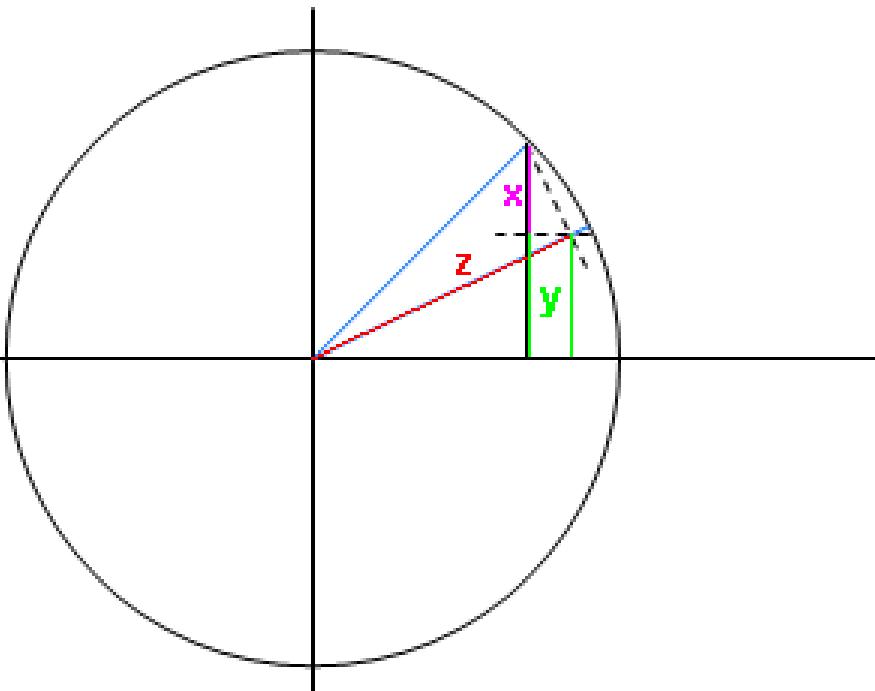
# Mathematisches

Wie kann man die Multiplikation zweier harmonischer Schwingungen berechnen?



**Pythagoras von Samos**  
**Philosoph c.570-c.495:**

Begründet unter anderem die Grundlagen der Trigonometrie



Leiten sie daraus die folgenden Beziehungen ab:

$$\sin(\alpha + \beta) = \sin \alpha \cos \beta + \cos \alpha \sin \beta$$

und bestimmen sie daraus das Ergebnis von

$$\sin \alpha \cos \beta$$

# Informations- und Codierungstheorie

## ***10. Sprache digitalisieren und übertragen***

Computer Networks and



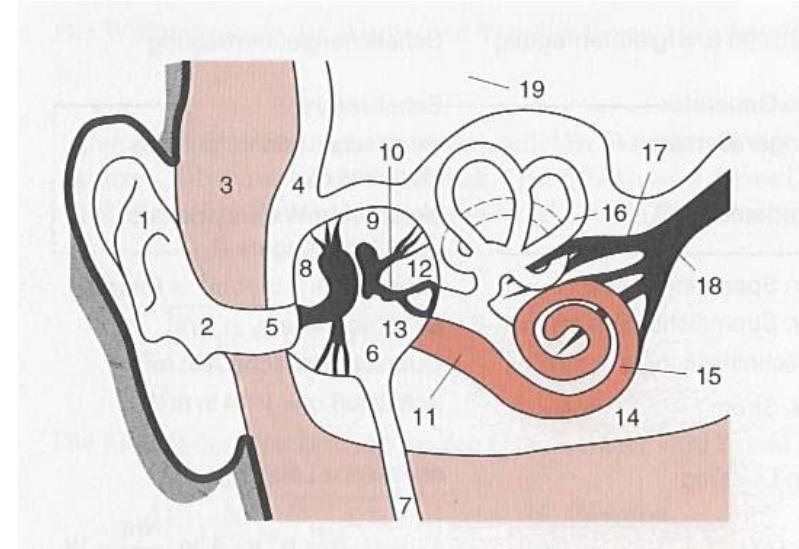
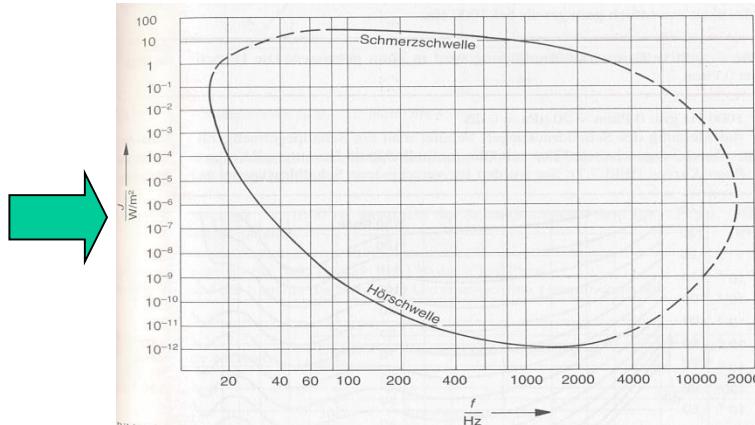
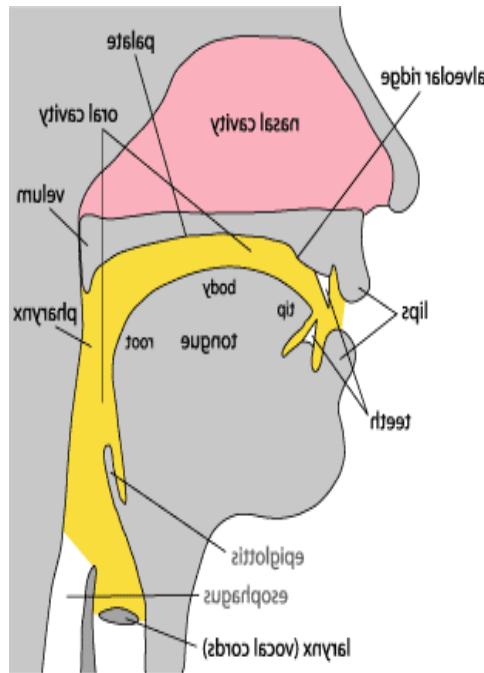
aF&E

Mobile Communication

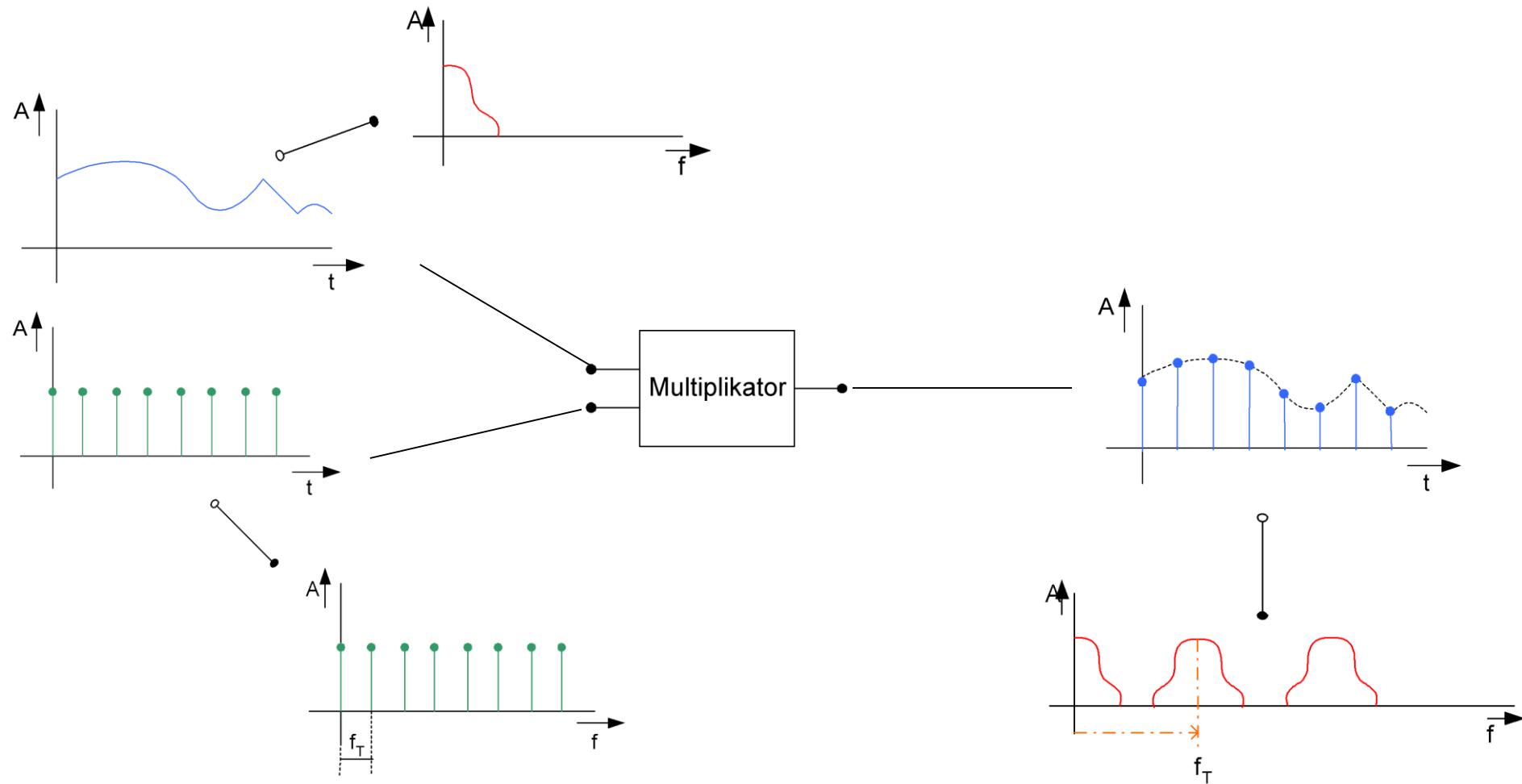
Prof. Dr.-Ing. Andreas Rinkel  
[andreas.rinkel@ost.ch](mailto:andreas.rinkel@ost.ch)

Sprechstunde: Mobil: +41 (0) 79 3320562

# **Eigenschaften des Sprechens und Hörens**



# Analoges Signal: Zeitdiskret



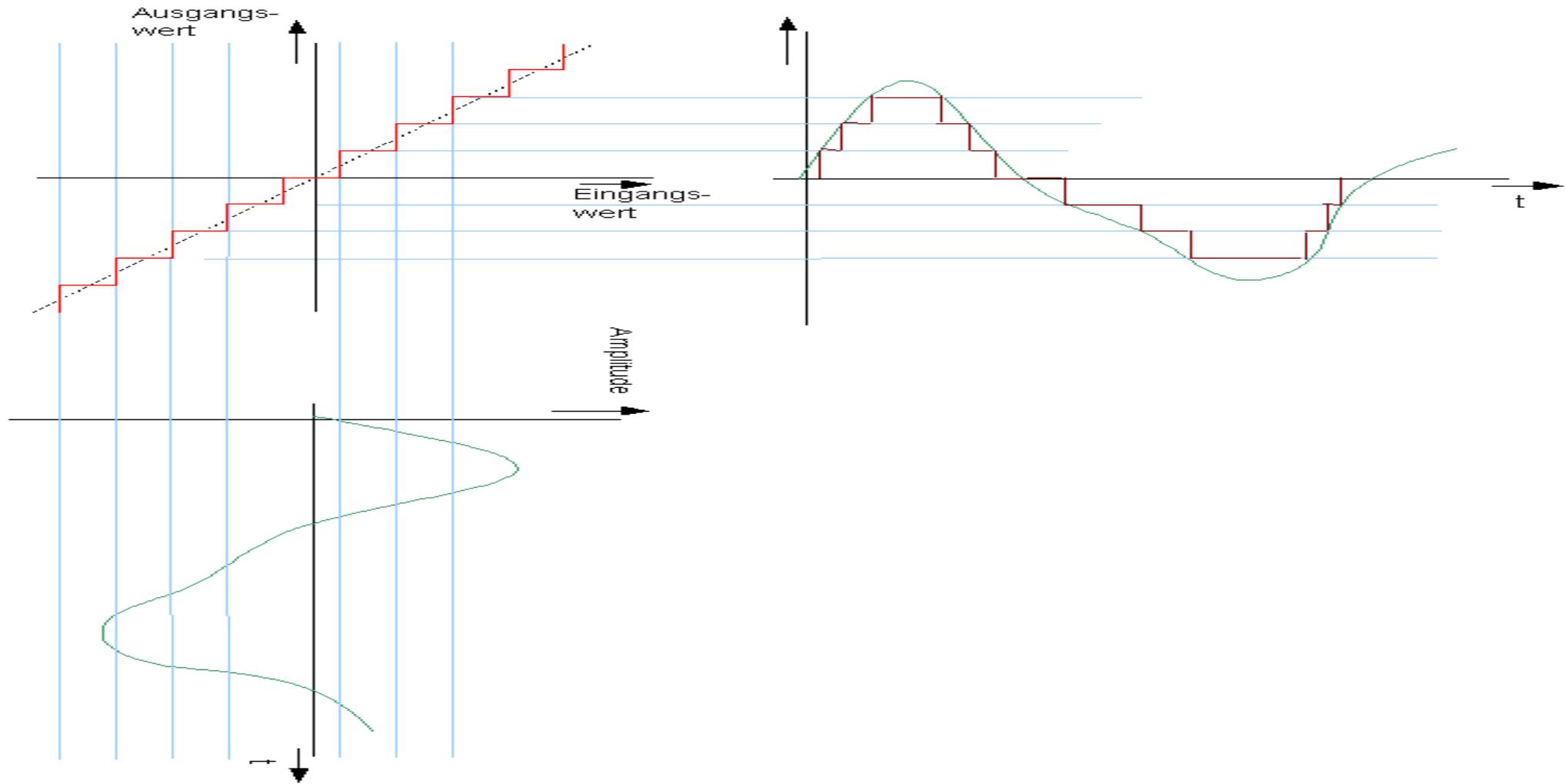
# Vom analogen zum digitalen Signal:

## 1. Schritt abtasten

Antworten:

- Wie kann abgetastet werden?
  - Durch Multiplikation mit einer Impulsfolge doppelter Frequenz.
- Was bedeutet das für das resultierende Signal, wo ist die Sprachinformation?
  - Die Multiplikation entspricht einer Faltung im Frequenzbereich, d.h. das Spektrum wird beliebig oft wiederholt, in Regel- und Kehrlage
- Wie wird aus dem abgetasteten Signal wieder Sprache?
  - Durch einen Tiefpass!
- Wie gross ist der Fehler, der durch die Zeitabtastung entsteht?
  - Kein systematischer Fehler, wenn mit doppelter Frequenz abgetastet wird!

# Analoges Signal: Zeit- und Wertdiskret



# Vom analogen zum digitalen Signal:

## 1. Schritt abtasten

Antworten:

- Abtasten durch Sample-Halte-Glied
- Quantisierungsfehler bei linearer Quantisierung:  
lässt sich nicht verhindern, nur minimieren!

Je höher der Signal-Geräuschabstand ist, desto geringer wird die Störung wahrgenommen.

$$\left| \frac{S}{N} \right|_{db} = 10 \log_{10} 2^{2w} dB \cong 6wdB$$

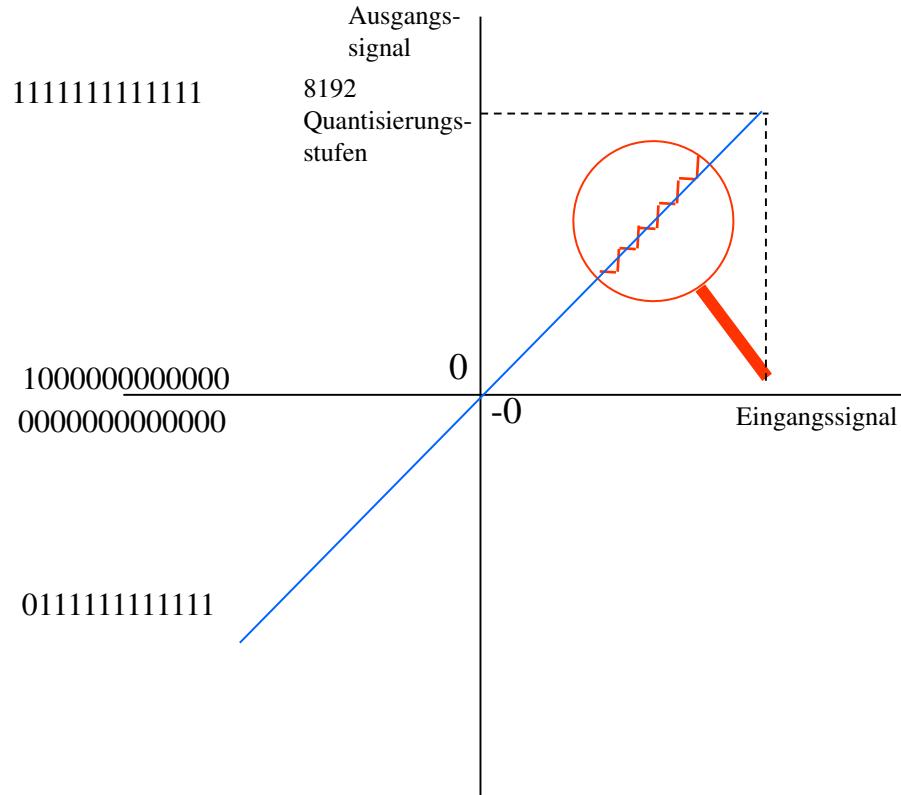
S: Signalstärke  
N: Rauschen,  
W: Anzahl Quantisierungsbits

Können besondere Eigenschaften des menschlichen Hörvermögens bei der Quantisierung ausgenutzt werden ??

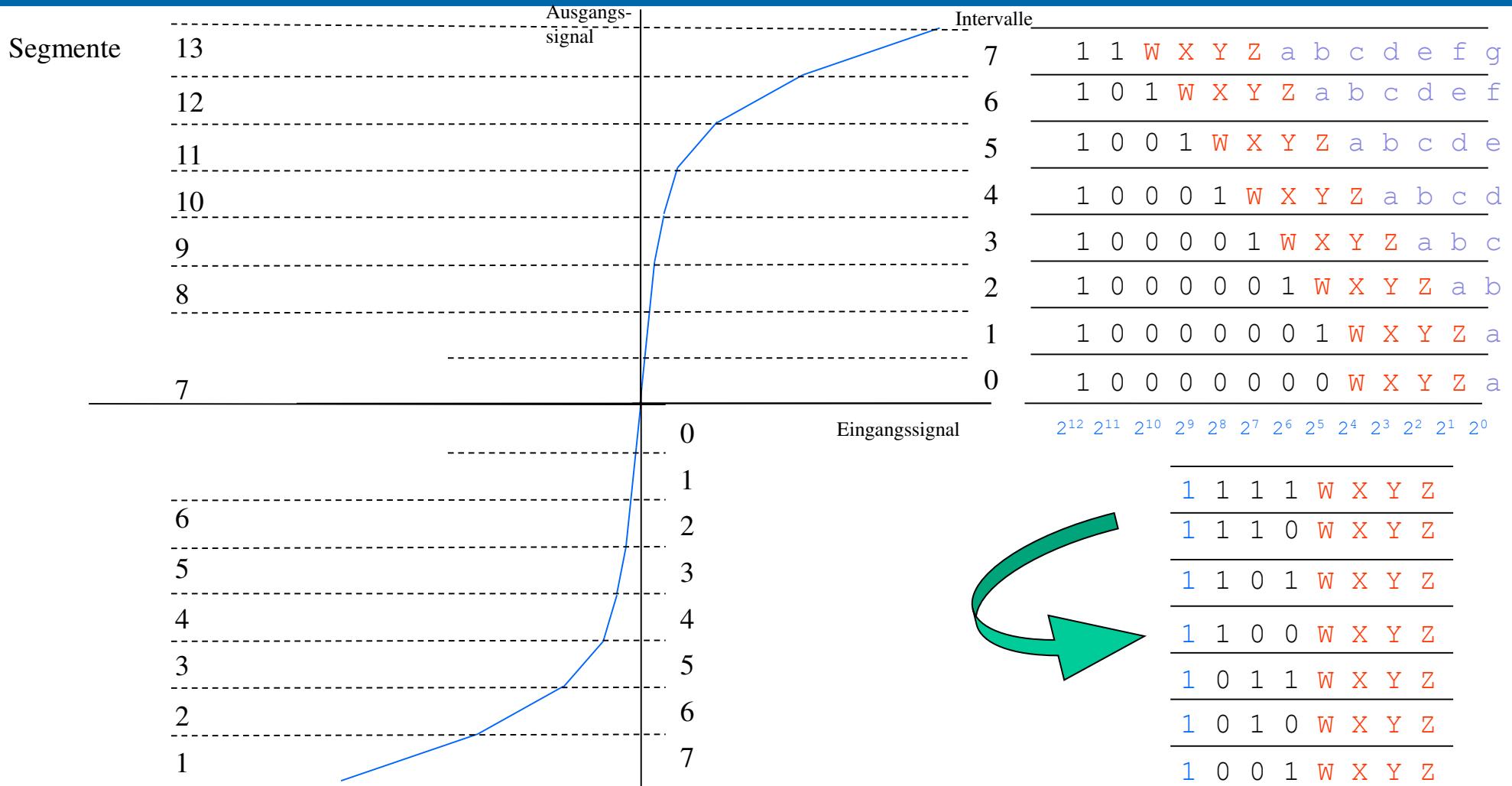
→ *Kompaundierung*

# Kompression – Expandierung oder Kompondierung I

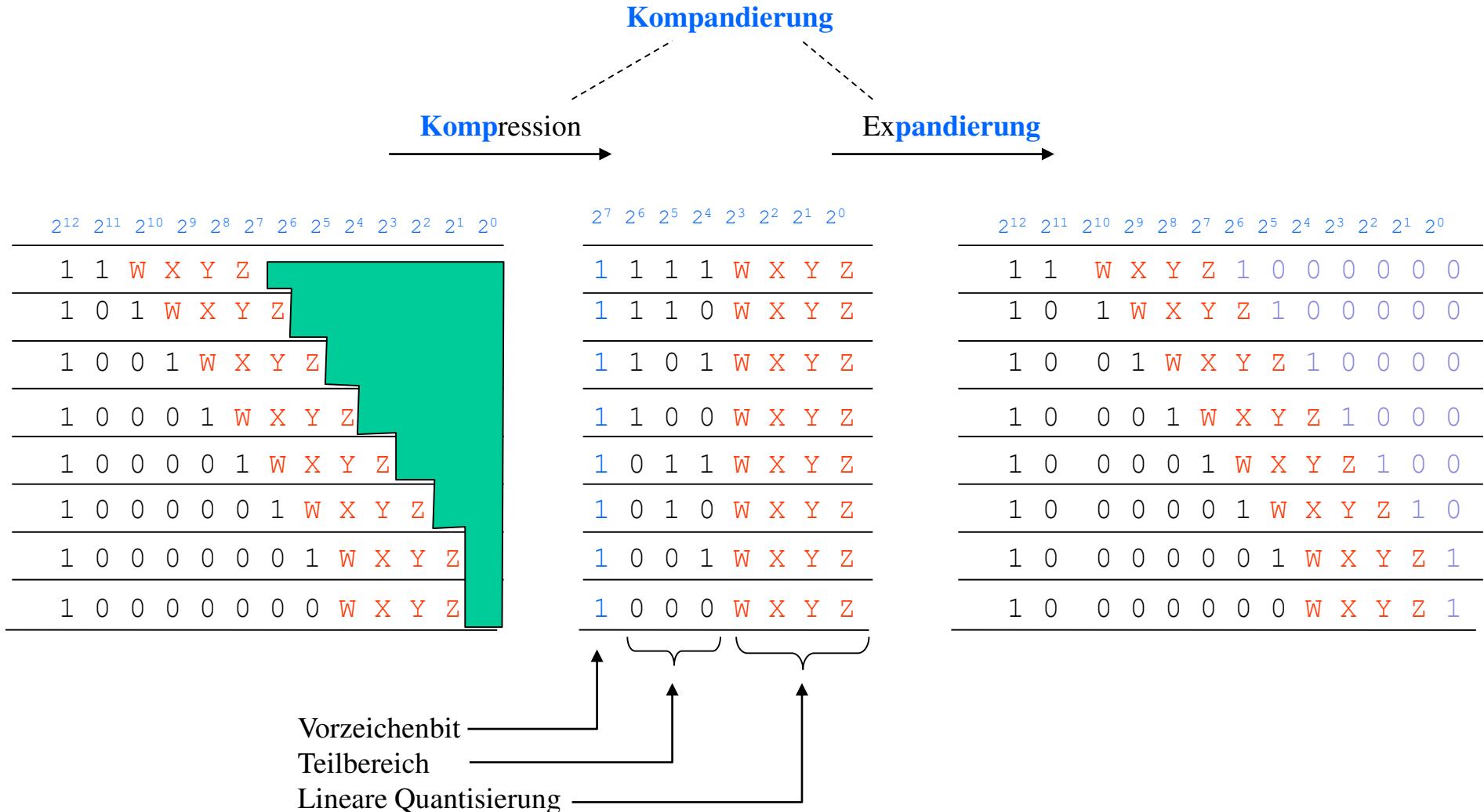
Codierung der linearen  
Quantisierungsstufen  
in 13 bit-Digitalisierung Vorzeichen



# Kompression – Expandierung oder Kompondierung (a-law codec) II



# Kompression – Expandierung oder Kompondierung II



# Enkodierung MPEG-1 Audio

