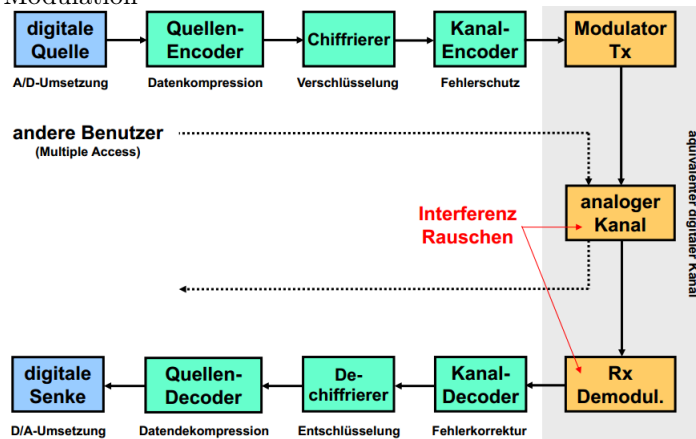


# INFORMATIONSTHEORIE

## Part 1. Kompression

### 1. ELEMENTE IM ÜBERTRAGUNGSSYSTEM

- Quelle/Senke
- Quellencodierung
- Chiffrierung
- Kanalcodierung
- Modulation



## Part 2. Entropie

### 2. DISKRETE INFORMATIONSQUELLEN

Symboldauer	$T$
Symbolrate	$R = 1/T$
Quellensymbol (Zufallsvariable)	$X[n]$
Alphabet	$A = \{x_1, x_2, \dots, x_M\}$
Wahrscheinlichkeit	$P(X = x_m) = P_X(x_m), m = 1, \dots, M$
Wahrscheinlichkeitsverteilung von $X$	$\sum_{m=1}^M P_X(x_m) = 1$

#### 2.0.1. gedächtnislose Quellen.

- DMS (Discrete Memoryless Source), Die Symbole  $X[n]$  sind unabhängig und haben identische Wahrscheinlichkeitsverteilung.
- BMS (Binary Memoryless Source), Die unabhängigen Symbole  $X[n]$  sind 2-wertig, d.h.  $P_X(x_1) = p$  und  $P_X(x_2) = 1 - p$ .
- BSS (Binary Symmetric Source), Die unabhängigen Symbole  $X[n]$  sind 2-wertig und es gilt:  $P_X(x_1) = 0.5$  und  $P_X(x_2) = 0.5$ .

## 3. INFORMATIONSGEHALT

Der Informationsgehalt eines Ereignisses  $X = x_m$  ist wie folgt definiert:

$$I_x(x_m) = \log_2 \left( \frac{1}{P_X(x_m)} \right) [\text{bit}]$$

Für Ereignisse von 2 (oder mehreren) Zufallsvariablen  $X$  und  $Y$  gilt sinngemäss:

$$I_x(x_m) = \log_2 \left( \frac{1}{P_{XY}(x_i, y_k)} \right) [\text{bit}]$$

Für 2 unabhängige Symbole  $X$  und  $Y$  gilt:

$$I_{XY}(x_i, y_k) = I_X(x_i) + I_Y(y_k)$$

## 4. REDUNDANZ

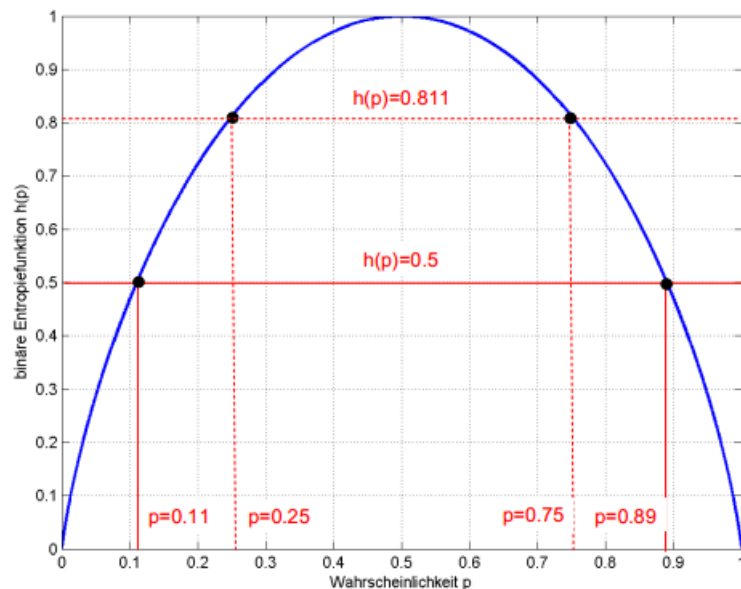
Differenz zwischen maximaler und mittlerer Entropie. Redundanz ( $M$  ist die Anzahl Symbole des Alphabets) Entropie ist maximal  $\log_2(M)$ , wenn  $X$ -Werte gleichverteilt. Möglichst wenig Redundanz am Ausgang des Quellencoders.

$$R = \log_2(M) - H(x)$$

## 5. ENTROPIE

Datenübertragung: die maximale (verlustlose) Kompression = Entropie

$$H_x(x_m) = \sum \log_2 P_x(x_i) \cdot \left( \frac{1}{P_X(x_i)} \right) [\text{bit}]$$



## 5.1. Binäre Entropiekurve.

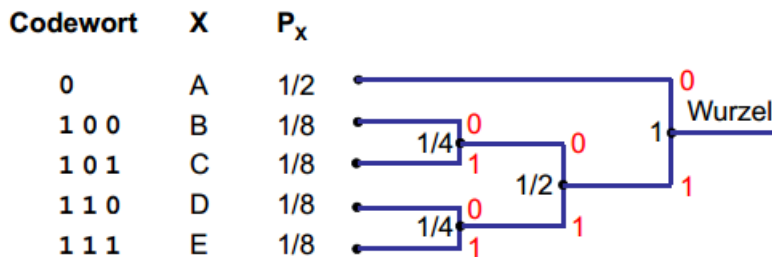
### Part 3. Kompression

#### HUFFMAN CODE

Abhängig von der Quellenstatistik

##### Algorithmus.

- (1) Symbole nach Wahrscheinlichkeiten ordnen und Knoten eines Baums zuweisen
- (2) Zwei Symbole mit kleinster Wahrscheinlichkeit in neuem Symbol zusammenfassen, neuer Knoten hat Summe der Wahrscheinlichkeiten
- (3) Erneutes Reduzieren des Wahrscheinlichkeitsfeldes gem. Schritt 1
- (4) Schritte 2 und 3 wiederholen bis 2 Symbole bzw. Knoten übrig
- (5) Von der Wurzel aus bei jeder Verzweigung nach oben eine „0“ und nach unten eine „1“ eintragen (auch umgekehrt möglich) //Konstruktion Codebuch



$R = \text{Wahrscheinlichkeit} * \text{Codelänge}$  (bsp:  $1 * 1/8 + (3 * 1/8) * 4 = 2$ )

#### LEMPER-ZIV-CODIERUNG

Unabhängig von der Quellenstatistik

##### Algorithmus.

- (1) Eindeutige Unterteilung der Symbolfolge Strings variabler Länge, Unterscheidung nur in 1 Bit
- (2) Encoding eines Strings: [Position des Präfix, neues Bit]

##### LZ77

- (1) Erstes Symbol des Vorschau-Buffers im Such-Buffer suchen
  - (a) rückwärts von rechts nach links
- (2) Token der längsten (letzten) Übereinstimmung ausgeben
  - (a) Token = ( Offset, Länge, nächstes Symbol)
  - (b) Token-Länge:  $\log_2(S+1) + \log_2(L+1) + 8$  typisch :  $11 + 5 + 8 = 24$  Bit
  - (c) wenn keine Übereinstimmung: (0,0, nächstes Symbol)
- (3) 3. Schiebefenster um Länge +1 nach rechts verschieben

##### LZ78

##### LZW

##### • Initialisierung I=[]

- (1) neues Symbol x zu String I hinzufügen =>  $I = Ix$  setzen
  - (a)  $Ix$  im Wörterbuch verzeichnet? Wenn ja, dann zu step 1. sonst zu step 3.
- (2) -
  - (a) Output = Wörterbuch-Pointer von I
  - (b) Neuer Wörterbucheintrag mit Phrase  $Ix$

(c) I = "x" setzen

**Beispiel Encoding.** Text: ABBABABAC

Anfangswörterbuch: 1 : A, 2 : B, 3 : C

Momentane Buchstaben	String I	verzeichnet	WB-Eintrag	Output
A	A	✓		
A	AB	×	4 : AB	1
B	B	✓		
B	BB	×	5 : BB	2
B	B	✓		
B	BA	×	6 : BA	2
A	A	✓		
AB	AB	✓		
AB	ABA	×	7 : ABA	4
A	A	✓		
AB	AB	✓		
ABA	ABA	✓		
ABA	ABAC	×	8 : ABAC	7
C	C	✓		
C	C,eof			3

**Bsp Decoding (Lösung ist in String J).**

68	68	82	99	77	65	82	256	82
D	E	R		M	A	R		R

Input	String I	String J	WB
68	D		
69	D	E	256: DE
82	E	R	257: ER
95	R	_	258: R_
77	_	M	259: _M
65	M	A	260: MA
82	A	R	261: AR
256	R	DE	262: RD
82	DE	R	263: DER

JPG

statt Redundanzreduktion vorallem Irrelevanzreduktion (Qualitätsverlust, häufig jedoch nicht bemerkbar)

**RLE.**

**PN-Sequenzen.** Pseude Noise Sequenzen

*LSFR.* Für Randomgenerator

$$a_0 = (a_{18} + a_5 + a_2 + a_1) \text{ modulo } 2$$