Kapitel 2: Informationstheorie und Quellcodierung

Wichtigste Quellen:

- Dirk Hoffmann: Einführung in die Informations- und Codierungstheorie bzw. der zugehörigen Vorlesung http://www.dirkwhoffmann.de/ICT/slides/MM_Codes.pdf

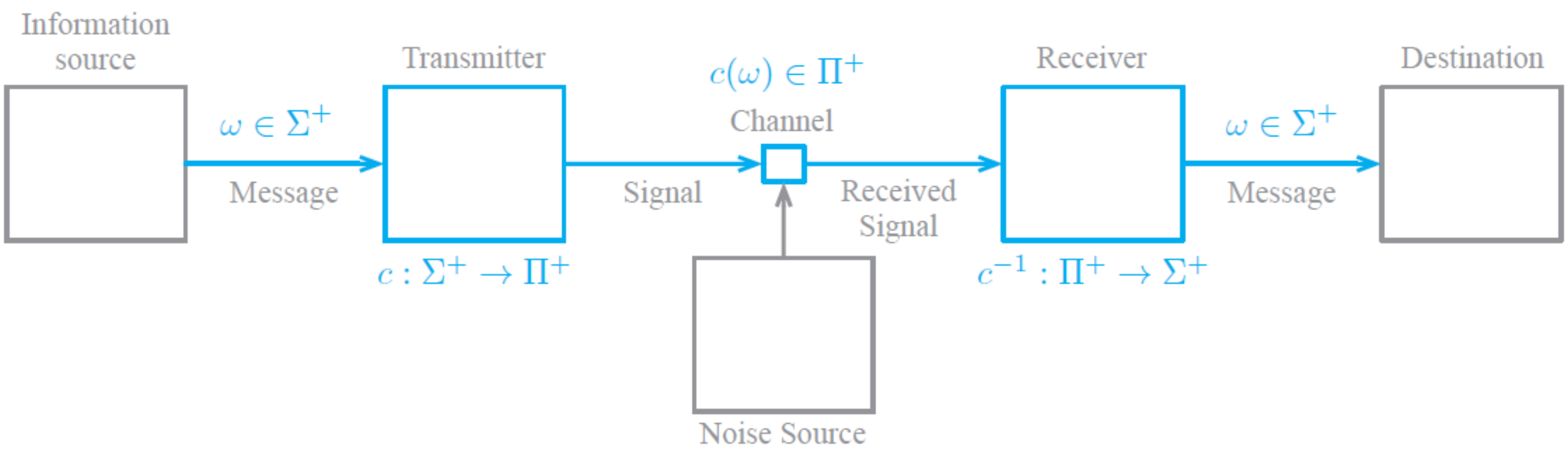
Kapitelstruktur

Was ist Information?

„Eine Aussage oder Beobachtung ist dann informativ, wenn sie uns etwas mitteilt, das uns nicht vorher schon bekannt war. Nur über solche Dinge können wir jedenfalls Information gewinnen, über die bei uns ein gewisses Maß an Nichtwissen oder Ungewissheit besteht. In der Tat **lässt sich Information als dasjenige erklären, das Ungewissheit beseitigt oder reduziert.**"

(Fred Attneave)

Das Shannon'sche Kommunikationsmodell



- Überlegungen zur Kanalkapazität
  - Allgemein: Übertragung von codierten Symbolen von einer Informationsquelle über einen gestörten Übertragungskanal zu einer Informationssenke
  - Speziell: Übertragung von Bitsequenzen
- Sender:
  - in einer Zeitspanne T werden n Bits in den Kanal eingespeist
  - damit können 2^n Bitsequenzen konstruiert werden
- Kanal:
  - wenn n Bits an Information übertragen werden, muss der Empfänger nach T Zeitschritten 2^n verschiedene Signale unterscheiden können
  - eine Halbierung der Signalzahl bedeutet die Übertragung von einem Bit

Das Shannon'sche Kommunikationsmodell

- Etwas mathematischer: wir definieren

$$N(T) = \text{Anzahl der Signale der Länge T}$$

- Ist $N(T) = 2^n$, dann können n Bits übertragen werden und damit überträgt der Kanal $\frac{\log_2 N(T)}{T}$ Informations-Bits pro Zeiteinheit

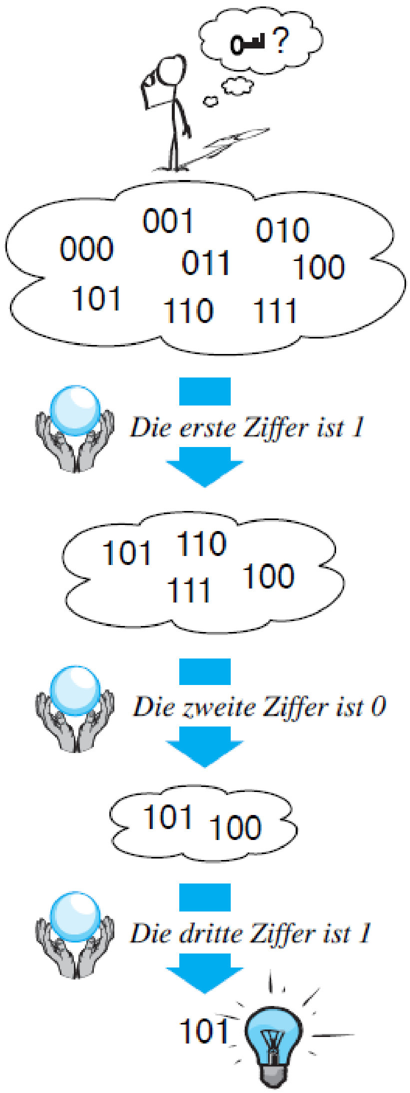
- Definition der Kapazität C eines diskreten Kanals von Claude Shannon (1948):

$$C = \lim_{T \to \infty} \frac{\log_2 N(T)}{T}$$

Claude Shannon
(1916-2001)

Was ist Information

- Aufgabe:
  - Einbruch in ein Computersystem, das mit einem binären Code aus 3 Bits verschlüsselt ist.
  - Alle Codes sind gleichwahrscheinlich.
  - Ein Orakel verrät ein Bit nach dem anderen.
- Frage:
  - Wieviel Information liefert das Orakel?
- Beobachtung:
  - jede Antwort des Orakel halbiert die Anzahl der möglichen Codes
  - die Information einer Antwort ist 1 Bit



Was ist Information?

- Fazit:
  - 8 Möglichkeiten auf 4 Möglichkeiten entspricht einer Information von 1 Bit
  - 8 Möglichkeiten auf 2 Möglichkeiten entspricht einer Information von 2 Bit
  - 8 Möglichkeiten auf 1 Möglichkeiten entspricht einer Information von 3 Bit
- Allgemein:
  - 2^n Möglichkeiten auf 2^m Möglichkeiten einzuschränken entspricht einer Information von $n - m$ Bit
  - n Möglichkeiten auf m Möglichkeiten einzuschränken entspricht einer Information von $ld\ n - ld\ m$ Bits

- Einschränkung von n Möglichkeiten auf m Möglichkeiten entspricht einer Information von

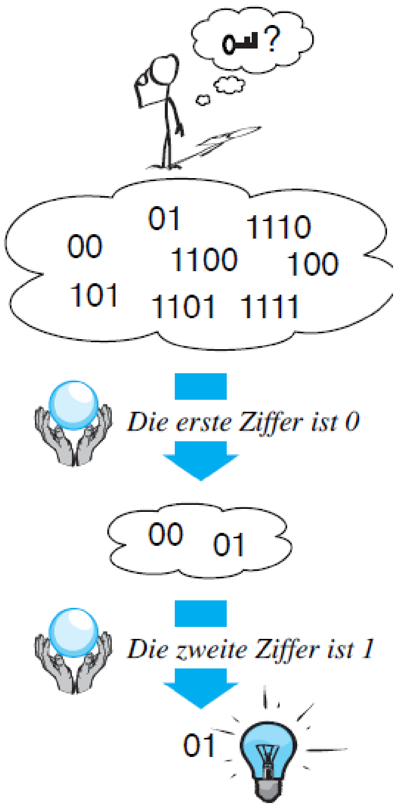
$$ld \frac{n}{m} \text{ Bits}$$

Hinweis: Synonyme Verwendung von $log_2\ x$ und $ld\ x$

Was ist Information?

- Modifizierte Aufgabe
  - Einbruch in ein Computersystem, das mit einem Schlüssel aus der Menge

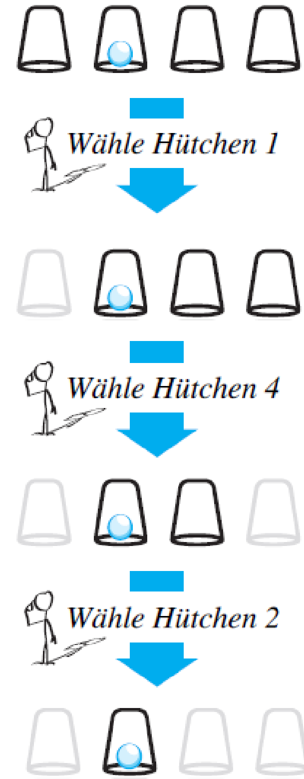
    { 00, 01, 100, 101, 1100, 1101, 1110, 1111 }
    geschützt ist
  - Wiederum können wir ein Orakel fragen, das uns nacheinander die Bits verrät



- Frage:
  - Wie viel Information gewährt uns das Orakel mit jedem Bit?

- Hütchenspiel

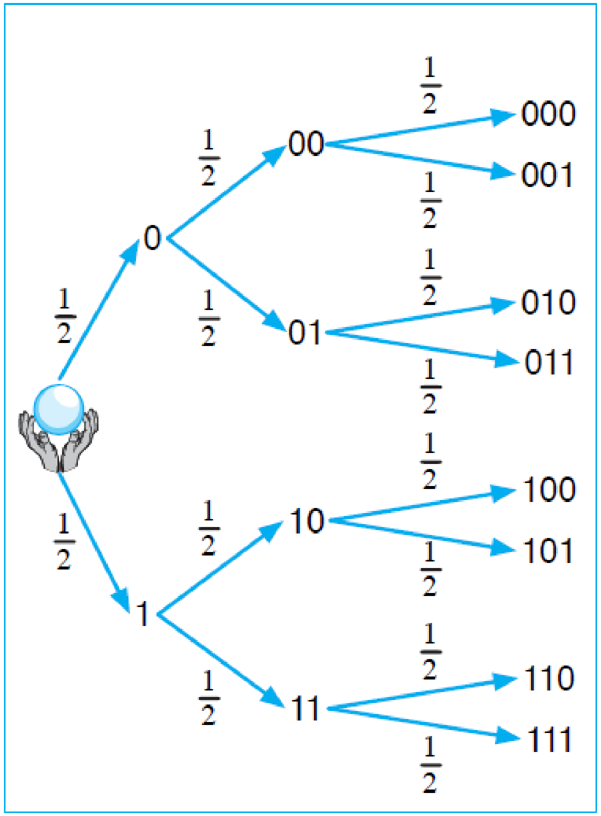
  - Spieler wählt ein Hütchen
    - wird die Perle gefunden ➔ Sieg
    - sonst wird das Spiel fortgesetzt
  - maximal 4 Runden

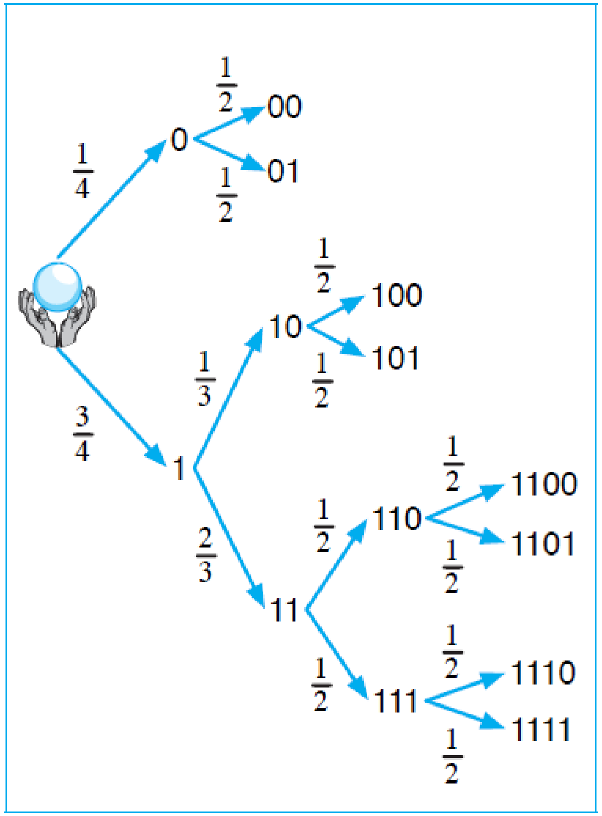


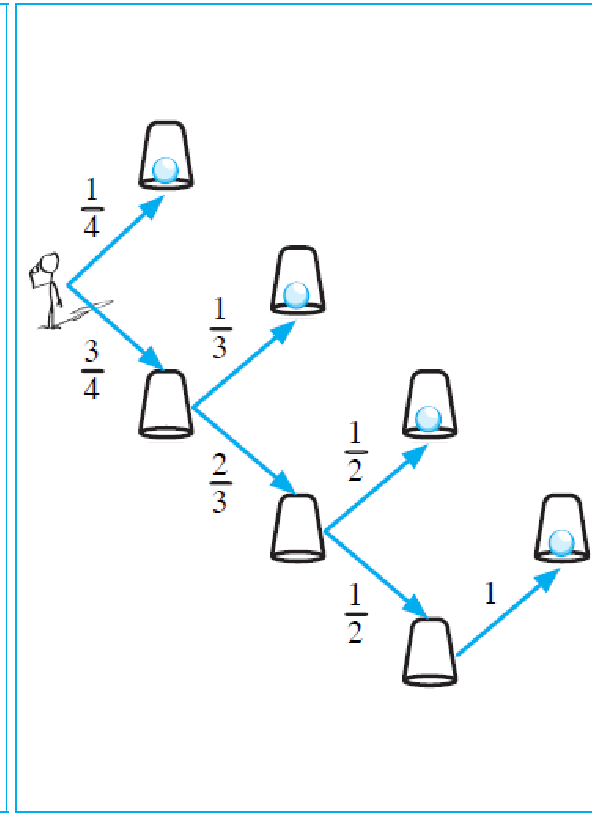
- Frage:
  - Wie viel Information gewinnen wir insgesamt?
  - Wie viel Information gewinnen wir pro Runde?

Auftrittswahrscheinlichkeiten der Ereignisse

Orakelspiel (Variante 1)	Orakelspiel (Variante 2)	Hütchenspiel







Informationsgehalt

- Maß für Information
  - Nachrichtenquelle:
    - Alphabet X mit N Zeichen $x_1, x_2, \dots, x_N$
    - $p(x_i)$: Auftrittswahrscheinlichkeit für Zeichen x_i
  - gesucht: Maßzahl $I_i = f\big(p(x_i)\big)$: Informationsgehalt des Zeichens x_i
- Definition für I_i (Informationsgehalt eines Zeichens)

$$\boxed{I_i = -ld\big(p(x_i)\big) = \mathrm{ld}\,\frac{1}{p(x_i)}}$$

Orakelspiel

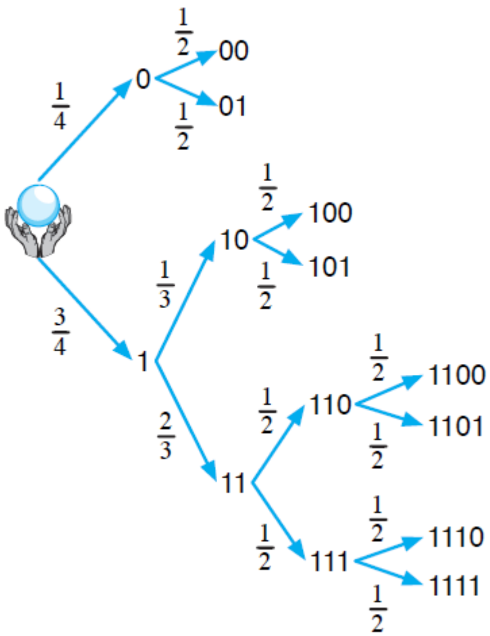
- Gleiche Auftrittswahrscheinlichkeiten:

x_i	$p(x_i)$	$I(x_i)$	Möglichkeiten vorher	Möglichkeiten nachher
0	0,5	1 Bit	8	4
1	0,5	1 Bit	8	4

Orakelspiel

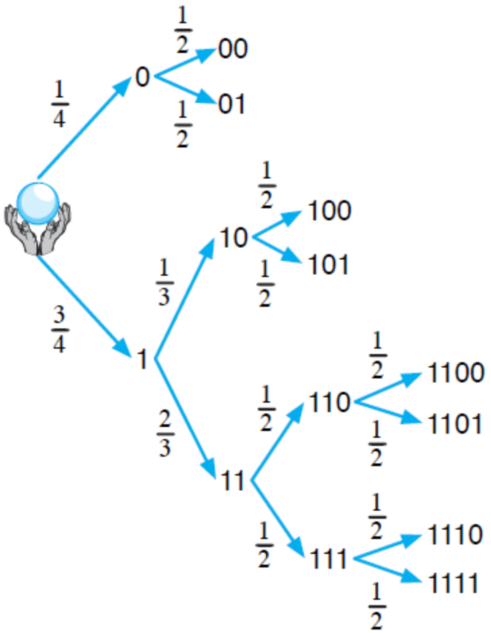
- Unterschiedliche Auftrittswahrscheinlichkeiten

x_i	$p(x_i)$	$I(x_i)$	Möglichkeiten	
			vorher	nachher
0	$1/4$	$ld\left(\frac{1}{1/4}\right) = ld(4) = 2\ Bit$	8	2
1	$3/4$	$ld\left(\frac{1}{3/4}\right) = ld\left(\frac{4}{3}\right) = 0{,}415\ Bit$	8	6
0	$1/3$	$ld\left(\frac{1}{1/3}\right) = ld(3) \approx 1{,}585\ Bit$	6	2
1	$2/3$	$ld\left(\frac{1}{2/3}\right) = ld(1{,}5) \approx 0{,}585\ Bit$	6	4



- Unterschiedliche Auftrittswahrscheinlichkeiten

x_i	$p(x_i)$	$I(x_i)$	Möglichkeiten vorher	Möglichkeiten nachher
0	0,25	$ld(1/0{,}25) = 2\ Bit$	8	2
1	0,75	$ld(1/0{,}75) \approx 0{,}415\ Bit$	8	6
0	0,33	$ld(1/0{,}33) \approx 1{,}585\ Bit$	6	2
1	0,67	$ld(1/0{,}67) \approx 0{,}585\ Bit$	6	4



- Entscheidungspfade
  – 00/01: 2+1=3 Bit
  – 100/101: 0,415+1,585+1=3 Bit
  – 110x/111x: 0,415+0,585+1+1=3 Bit

Additivität des Informationsgehalts

$$I(x_1, x_2, x_3) = I(x_1) + I(x_2) + I(x_3)$$

bei **unabhängigen** Ereignissen

$$p(x_1, x_2, x_3) = p(x_1) \cdot p(x_2) \cdot p(x_3)$$

Entropie

- Definition für I_i (Informationsgehalt eines Zeichens)

$$\boxed{I_i = -ld\big(p(x_i)\big) = \mathrm{ld}\frac{1}{p(x_i)}}$$

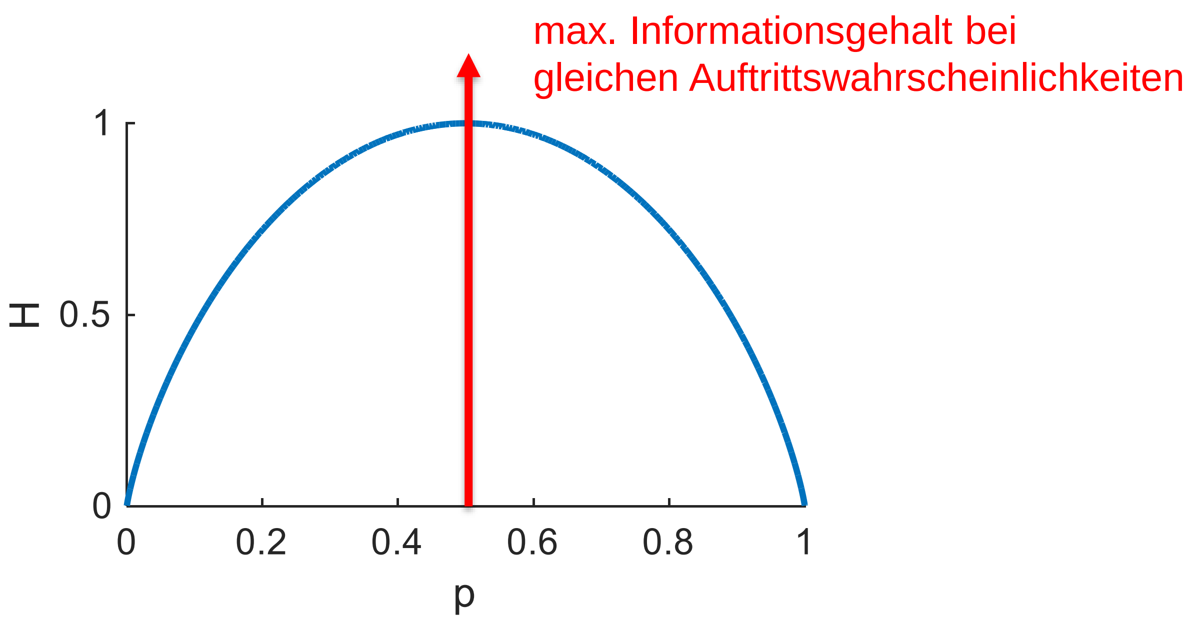
- Definition des mittleren Informationsgehalts einer Nachrichtenquelle

$$\boxed{H(X) = \mathrm{E}[I(X)] = \sum_{i=1}^{N} p(x_i) \cdot I(x_i) = \sum_{i=1}^{N} p(x_i) \cdot \mathrm{ld}\frac{1}{p(x_i)}}$$

  - $H(X)$ bezeichnet die **Entropie** einer Nachrichtenquelle
    - eines Zeichenvorrats X mit Auftrittswahrscheinlichkeiten $p(x_i)$
  - $\mathrm{E}[X]$ bezeichnet den Erwartungswert einer Zufallsvariablen X
    - die Entropie ist der mittlere Informationsgehalt eines zufälligen Zeichens, das die Nachrichtenquelle sendet

Maximaler Informationsgehalt

- Informationsgehalt ist **maximal bei gleicher Auftrittswahrscheinlichkeit** für alle Zeichen
- Beispiel: binärer Kanal mit „0“ und „1“
  - Auftrittswahrscheinlichkeit für „0“: p
  - Auftrittswahrscheinlichkeit für „1“: 1-p
  - Informationsgehalt: $H(X) = p \cdot ld\left(\frac{1}{p}\right) + (1-p) \cdot ld\left(\frac{1}{1-p}\right)$



Beispiel: Informationsgehalt eines Wortes

- Wie groß ist der Informationsgehalts des Worts „HOCHSCHULE“?

Zeichen	H	C	O	S	U	L	E
$p(x_i)$	3/10	2/10	1/10	1/10	1/10	1/10	1/10
$I(x_i)$	1,737	2,322	3,322				

- Informationsgehalt:

$$I_{ges} = 3 \cdot 1{,}737 + 2 \cdot 2{,}322 + 5 \cdot 3{,}322 = 26{,}464$$

- Codierung:
  - Das Wort „HOCHSCHULE“ kann theoretisch mit 27 Bit kodiert werden

Kapitelstruktur

Codierung

- Binäre Codierung
  - jedes Zeichen wird durch eine Bitsequenz repräsentiert
- Binäre Codierung der Nachrichtenquelle „HOCHSCHULE"?
  - 7 Zeichen ➔ 3 Bits

Zeichen	H	C	O	S	U	L	E
Code	000	001	010	011	100	101	110

  - Codierung von „HOCHSCHULE": 000 010 001 000 011 001 000 100 101 110
    - 30 Bit

⇨ Codierung ist nicht optimal

Quellcodierung

- Ziel:
  - Repräsentation von Zeichen mit binären Codes, so dass die benötigte Anzahl Bits minimiert wird
- Grundidee:
  - Verwendung unterschiedlich langer Codes
    - häufiges Zeichen: kurzer Code
    - seltenes Zeichen: langer Code
- Beispiel: Hochschule
  - H:
  - O:
  - C:
  - S:
  - U:
  - L:
  - E:

Quellcodierung

- Ziel:
  - Repräsentation von Zeichen mit binären Codes, so dass die benötigte Anzahl Bits minimiert wird
- Grundidee:
  - Verwendung unterschiedlich langer Codes
    - häufiges Zeichen: kurzer Code
    - seltenes Zeichen: langer Code
- Beispiel: Hochschule
  - `H: 00`
  - `C: 01`
  - `O: 100`
  - `S: 101`
  - `U: 1100`
  - `L: 1101`
  - `E: 111`

→

```
HOCHSCHULE:
00 100 01 00 101 01 00 1100 1101 1110
```

5*2+3*3+2*4=27 Bit

→ Codierung ist optimal

Beispiel: Entropie

- Nachrichtenquelle sendet die Zeichen des Worts „HOCHSCHULE“ mit entsprechenden Auftrittswahrscheinlichkeiten

Zeichen	H	C	O	S	U	L	E
Auftrittswahr-scheinlichkeit	3/10	2/10	1/10	1/10	1/10	1/10	1/10

- Wie groß ist die Entropie der Nachrichtenquelle?

$$H(X) = 0{,}3 \cdot ld\left(\frac{10}{3}\right) + 0{,}2 \cdot ld\left(\frac{10}{2}\right) + 5 \cdot 0{,}1 \cdot ld\left(\frac{10}{1}\right) \approx 2{,}646$$

Zeichen	H	C	O	S	U	L	E
$p(x_i)$	3/10	2/10	1/10	1/10	1/10	1/10	1/10
$I(x_i)$	1,737	2,322	3,322				

Quellcodierung: Mittlere Codewortlänge und Redundanz

- Mittlere Codewortlänge einer Nachricht des Zeichenvorrats X

$$L_C(X) = \sum_{i=1}^{N} p(x_i) \cdot l_C(x_i)$$

$l_C(x_i)$: Länge des binären Codeworts von x_i bei Codierung mit Code C
$\mathrm{p}(x_i)$: Auftrittswahrscheinlichkeit von x_i

  – ein Codewort ist hier die Repräsentation für ein Zeichen des Zeichenvorrats

- Informationsgehalt entspricht Codewortlänge bei optimaler Codierung

- Redundanz einer Codierung
  – Differenz zwischen mittlerer Codewortlänge und mittlerem Informationsgehalt

$$\boxed{R_C(X) = L_C(X) - H(X)}$$

Beispiel: Mittlere Codewortlänge und Redundanz

Zeichen	H	C	O	S	U	L	E
Auftrittswahr-scheinlichkeit	3/10	2/10	1/10	1/10	1/10	1/10	1/10
1. Code	000	001	010	011	100	101	110
2. Code	00	01	110	111	1000	1001	101

- Welcher Code ist besser?
  - Mittlere Codewortlänge
    - Code 1: $L_{C1} = 3,$
      $R_C = 3 - 2{,}646 = 0{,}354$
    - Code 2: $L_{C2} = (0{,}3 + 0{,}2) \cdot 2 + 3 \cdot 0{,}1 \cdot 3 + 2 \cdot 0{,}1 \cdot 4 = 2{,}7$
      $R_C = 2{,}7 - 2{,}646 = 0{,}054$
  - durch geeignete Codierung kann die mittlere Codewortlänge reduziert werden
- Prinzip: häufige Zeichen mit weniger Bits codieren, seltene Zeichen mit mehr Bits codieren
- Algorithmen: Shannon-Fano, Huffman, arithmetische Codierung

Definitionen

- Informationsgehalt eines Zeichens
  - hängt von Auftrittswahrscheinlichkeit $p(x_i)$ ab

$$I(x_i) = \mathrm{ld}\,\frac{1}{p(x_i)}$$

- Informationsgehalt einer Nachricht $\bar{x}$
  - Summe der Informationsgehalte der Zeichen

$$I(\bar{x}) = \sum_{i=1}^{N} I(x_i)$$

- Entropie einer Nachrichtenquelle mit Zeichenvorrat X
  - mittlere Informationsgehalt eines Zeichens

$$H(X) = \sum_{i=1}^{N} p(x_i) \cdot \mathrm{ld}\,\frac{1}{p(x_i)}$$

- Mittlere Codewortlänge eines Zeichen-vorrats X mit Codierung C

$$L_C(X) = \sum_{i=1}^{N} p(x_i) \cdot l_C(x_i)$$

- Redundanz einer Codierung
  - Differenz zwischen Entropie und mittlerer Codewortlänge

$$R_C(X) = L_C(X) - H(X)$$

Kapitelstruktur

2.1 Informationsgehalt

2.2 Quellcodierung

- 2.2.1 Redundanz und Quellcodierung
- **2.2.2 Entropiecodierung**
  - 2.2.2.1 Verfahren nach Shannon und Fano
  - 2.2.2.2 Huffman-Code
  - 2.2.2.3 Arithmetische Codierung
- 2.2.3 Wörterbuch-basierte Verfahren
- 2.2.4 HTTP/2.0 Header-Compression

2.3 Informationsgehalt eines Übertragungskanals

2.4 Zusammenfassung

Entropiecodierung

- Die Entropiecodierung codiert einzelne Zeichen in einem Alphabet entsprechend einer bekannten oder geschätzten Auftrittswahrscheinlichkeit.
- Die verschiedenen Codierungsverfahren dienen dazu, eine Codierung der Zeichen zu ermitteln.
- Eine Codierung muss präfixfrei sein, d.h. kein Codewort darf Präfix eines anderen Codeworts sein, da sonst keine eindeutige Dekodierung möglich ist.

Codierung nach Shannon-Fano

- Algorithmus
  1. Ordnen der Zeichen nach absteigenden Auftrittswahrscheinlichkeiten
  2. Aufteilen der Zeichen in zwei Hälften mit möglichst gleich großen Auftrittswahrscheinlichkeiten
     - bei zwei gleich guten Möglichkeiten soll die Auftrittswahrscheinlichkeit der oberen Hälfte größer als die der unteren Hälfte sein
  3. An das Ende des Codeworts hinzufügen
     a. einer „0“ zu den Codewörtern der Zeichen der oberen Hälfte
     b. einer „1“ zu den Codewörtern der Zeichen der unteren Hälfte
  4. Wiederholen von 2. und 3. für die obere und untere Hälfte
- Beispiel: HOCHSCHULE

Codierung nach Shannon-Fano

- Beispiel (HOCHSCHULE)

Zeichen	H	C	O	S	U	L	E
Auftrittswahrs‘	3/10	2/10	1/10	1/10	1/10	1/10	1/10
1.Schritt	0,5 / 0		0,5 / 1				
2. Schritt	0,3 / 00	0,2 / 01	0,3 / 10			0,2 / 11	
3. Schritt	00	01	0,2 / 100		0,1 /101	0,1 /110	0,1/111
4. Schritt	00	01	0,1 / 1000	0,1 / 1001	101	110	111
Codierung	00	01	1000	1001	101	110	111

- Eigenschaft: Präfixfreiheit
  - kein „kürzeres“ Codewort ist das Präfix eines längeren Codeworts

Codierung nach Huffman (allgemein)

- Algorithmus
  1. Ordne alle Zeichen nach aufsteigenden Auftrittswahrscheinlichkeiten in eine Liste
  2. Wiederhole, bis nur noch ein Zeichen in der Liste steht
     - füge dem Codewort des ersten Zeichens vorne eine „1“ hinzu
     - füge dem Codewort des zweiten Zeichens vorne eine „0“ hinzu
     - kombiniere die ersten beiden Zeichen zu einem Zeichen
       – bestimme die Summe der Auftrittswahrscheinlichkeiten
       – füge das kombinierte Zeichen entsprechend der berechneten Auftrittswahrscheinlichkeit in die Liste ein
         » falls in der Liste bereits Zeichen mit dieser Auftrittswahrscheinlichkeit vorkommen, wird das neue Zeichen nach diesen Zeichen eingefügt
     - lösche die beiden ersten Zeichen aus der Liste

Codierung nach Huffman (auf Papier)

- Liste 1: enthält alle Zeichen und deren Codierung
- Liste 2: enthält alle einzelnen und kombinierten Zeichen mit deren Auftrittswahrscheinlichkeit
- Algorithmus
  1. Schreibe in Liste 1 alle Zeichen nach absteigenden Auftrittswahrscheinlichkeiten untereinander mit Platz links der Zeichen; hier wird später die Codierung der Zeichen notiert
  2. Schreibe in Liste 2 alle Zeichen mit niedrigster Auftrittswahrscheinlichkeit untereinander
  3. Kombiniere die beiden obersten noch nicht betrachteten Zeichen aus Liste 2
     - füge dem Codewort des obersten Zeichens vorne eine „1“ hinzu (in Liste 1)
     - füge dem Codewort des unteren Zeichens vorne eine „0“ hinzu (in Liste 1)
     - addiere die Auftrittswahrscheinlichkeiten der beiden Zeichen
  4. Füge unten in Liste 2 alle einzelnen Zeichen aus Liste 1 an, deren Austrittswahrscheinlichkeit kleiner oder gleich der des kombinierten Zeichens ist
  5. Füge unten in Liste 2 das kombinierte Zeichen an
  6. Wiederhole 3. bis nur noch ein nicht betrachtetes Zeichen vorkommt

Beispiel: HOCHSCHULE

Liste 1

Codierung pro Schritt						Zeichen	Auftrittsw‘	Codewort	Länge	
0	1					H	0,3	01	2	0,6
0	0		0			C	0,2	000	3	0,6
0	0		1			O	0,1	001	3	0,3
1		0		0		S	0,1	100	3	0,3
1		0		1		U	0,1	101	3	0,3
1		1			0	L	0,1	110	3	0,3
1		1			1	E	0,1	111	3	0,3
						Summe	1,0			2,7

Liste 2

Schritt	Zeichen	Auftrittsw‘
	E	0,1
	L	0,1
	U	0,1
	S	0,1
	O	0,1
	C	0,2
1	E/L	0,2
2	U/S	0,2
	H	0,3
3	O/C	0,3
4	EL/US	0,4
5	H/OC	0,6
6	ELUS/HOC	1,0

Beispiel: HOCHSCHULE (kompakter)

Liste 1

Codierung			Zeichen	Auftrittsw‘	Länge	
	0	1	H	0,3	2	0,6
0	0	0	C	0,2	3	0,6
0	0	1	O	0,1	3	0,3
1	0	0	S	0,1	3	0,3
1	0	1	U	0,1	3	0,3
1	1	0	L	0,1	3	0,3
1	1	1	E	0,1	3	0,3
			Summe	1,0		2,7

Liste 2

Zeichen	Auftritts‘
E	0,1
L	0,1
U	0,1
S	0,1
O	0,1
C	0,2
E/L	0,2
U/S	0,2
H	0,3
O/C	0,3
EL/US	0,4
H/OC	0,6
ELUS/HOC	1,0

Darstellung als Baum

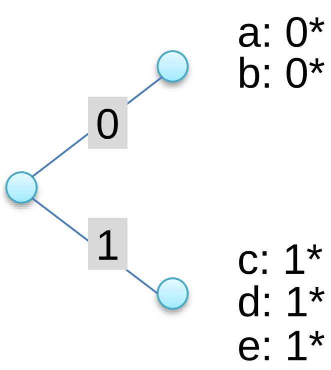
- Nachrichtenquelle:

$\boldsymbol{x_i}$	a	b	c	d	e
$p(x_i)$	12/31	6/31	5/31	4/31	4/31

  - Entropie: $H(X) \approx 2{,}176$

Shannon-Fano

Huffman



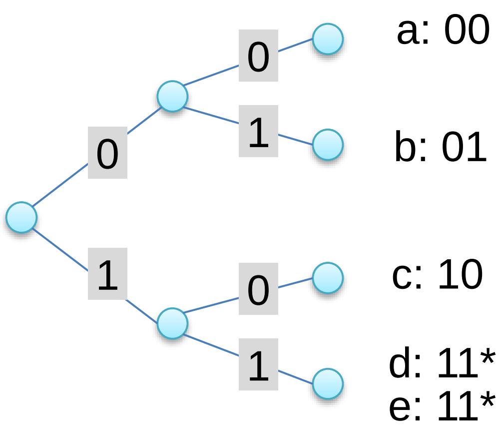
Darstellung als Baum

- Nachrichtenquelle:

x_i	a	b	c	d	e
$p(x_i)$	12/31	6/31	5/31	4/31	4/31

Shannon-Fano

Huffman



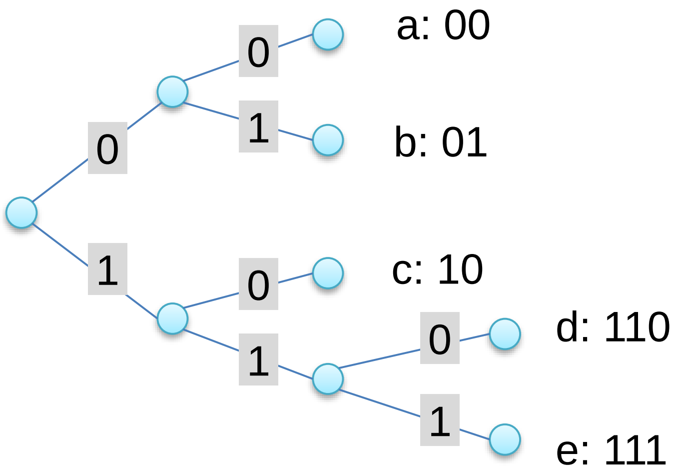
Darstellung als Baum

- Nachrichtenquelle:

x_i	a	b	c	d	e
$p(x_i)$	12/31	6/31	5/31	4/31	4/31

Shannon-Fano

Huffman

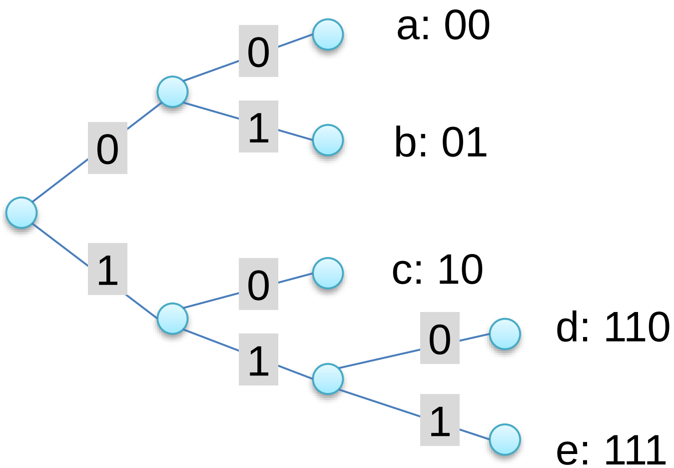


Darstellung als Baum

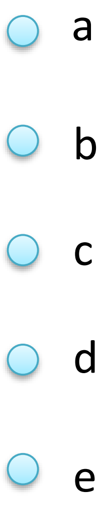
- Nachrichtenquelle:

$\boldsymbol{x_i}$	a	b	c	d	e
$p(x_i)$	12/31	6/31	5/31	4/31	4/31

Shannon-Fano



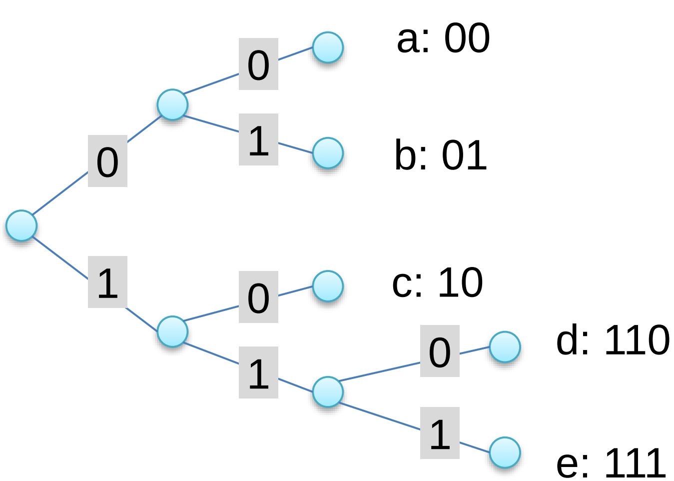
Huffman



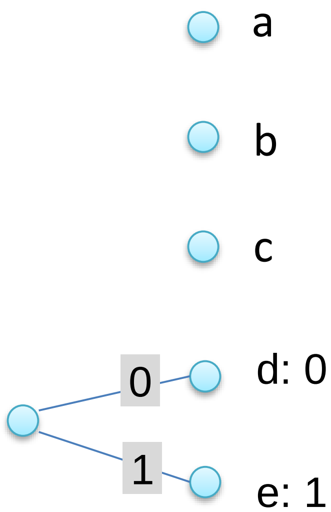
- Nachrichtenquelle:

$\boldsymbol{x_i}$	a	b	c	d	e
$p(x_i)$	12/31	6/31	5/31	4/31	4/31

Shannon-Fano



Huffman

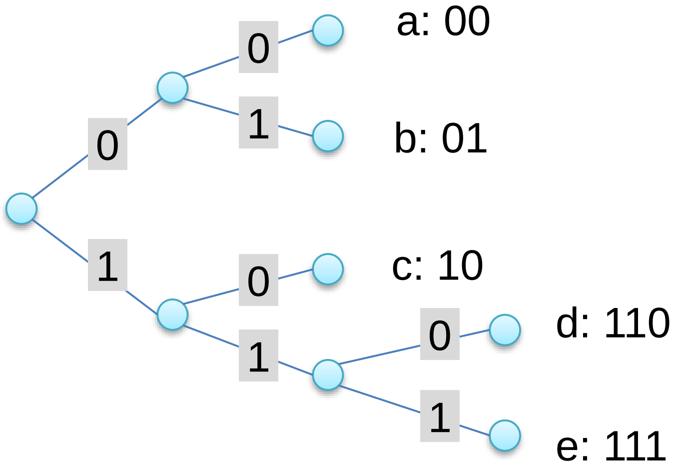


Darstellung als Baum

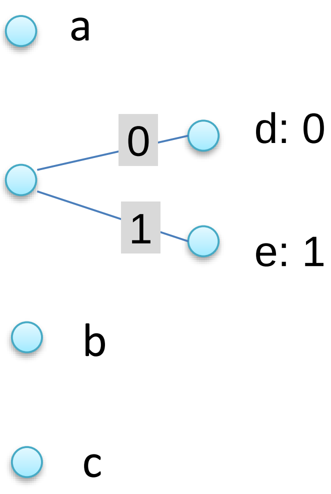
- Nachrichtenquelle:

x_i	a	b	c	d	e
$p(x_i)$	12/31	6/31	5/31	4/31	4/31

Shannon-Fano



Huffman

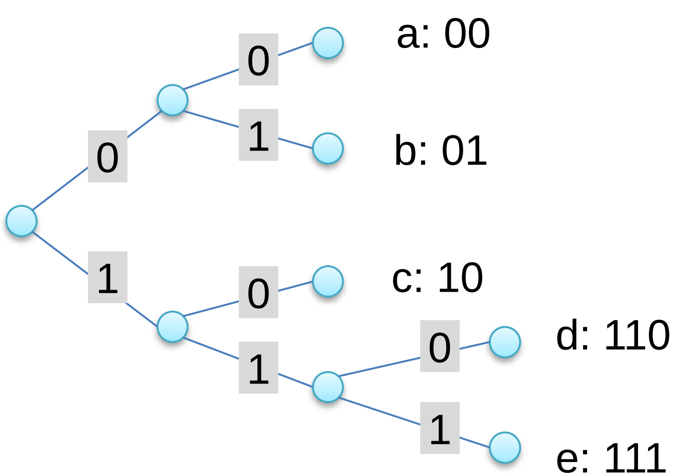


Darstellung als Baum

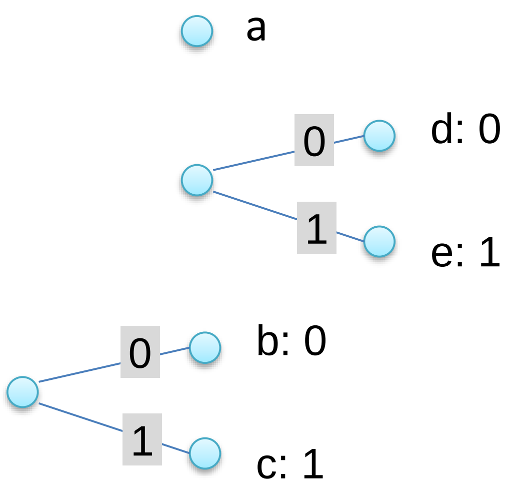
- Nachrichtenquelle:

$\boldsymbol{x_i}$	a	b	c	d	e
$p(x_i)$	12/31	6/31	5/31	4/31	4/31

Shannon-Fano



Huffman

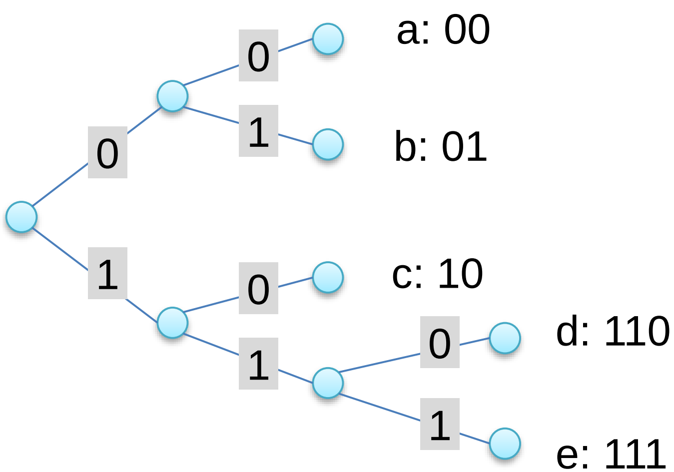


Darstellung als Baum

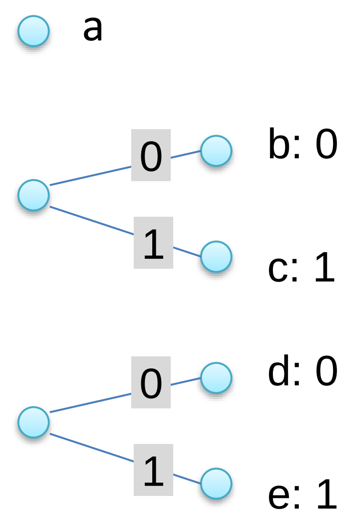
- Nachrichtenquelle:

x_i	a	b	c	d	e
$p(x_i)$	12/31	6/31	5/31	4/31	4/31

Shannon-Fano



Huffman

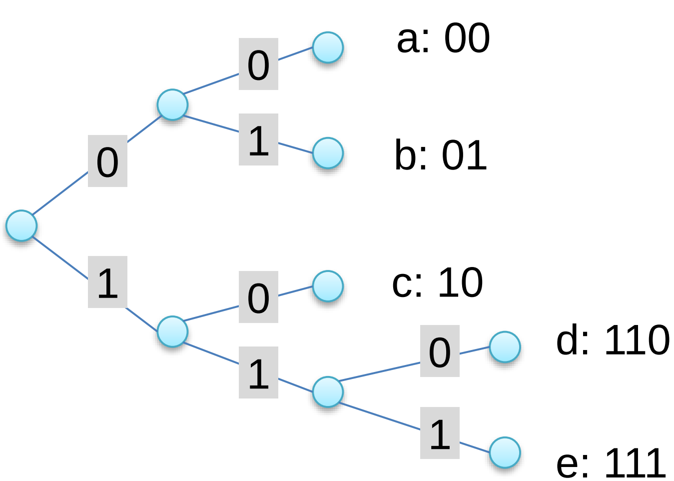


Darstellung als Baum

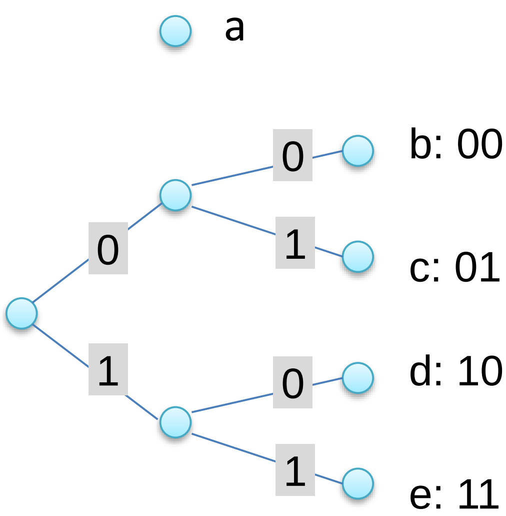
- Nachrichtenquelle:

$\boldsymbol{x_i}$	a	b	c	d	e
$p(x_i)$	12/31	6/31	5/31	4/31	4/31

Shannon-Fano



Huffman

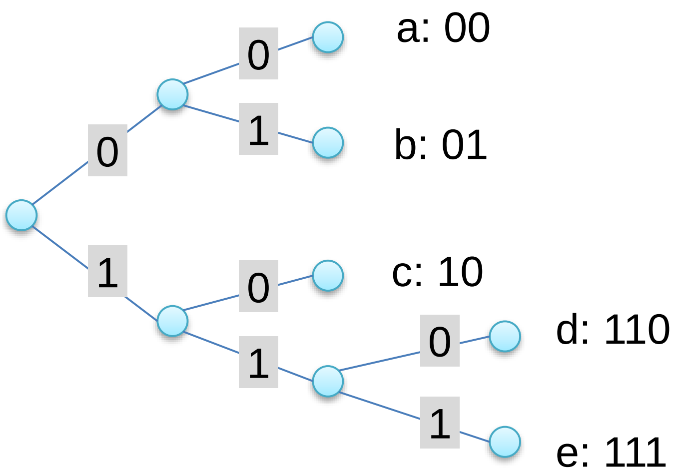


Darstellung als Baum

- Nachrichtenquelle:

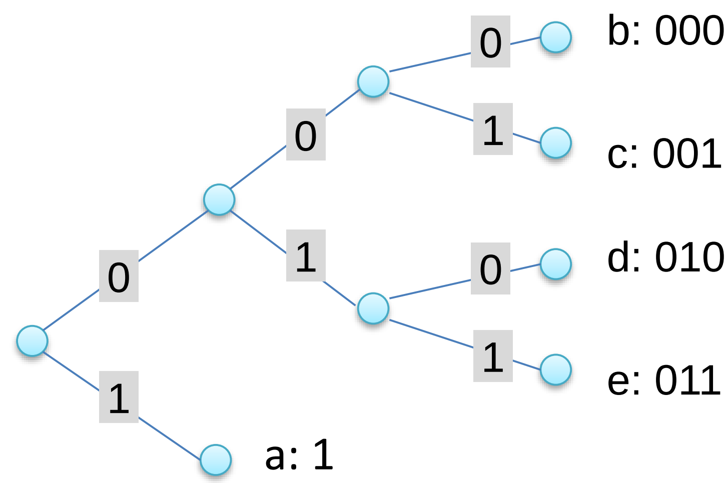
$\boldsymbol{x_i}$	a	b	c	d	e
$p(x_i)$	12/31	6/31	5/31	4/31	4/31

Shannon-Fano



$$L = \frac{23}{31} \cdot 2 + \frac{8}{31} \cdot 3 = \frac{70}{31} = 2{,}258$$

Huffman



$$L = \frac{12}{31} \cdot 1 + \frac{19}{31} \cdot 3 = \frac{69}{31} = 2{,}226$$

Weiteres Beispiel (Tafel)

- Datenquelle

A	B	C	D	E	F	G	H
0,22	0,20	0,16	0,16	0,13	0,06	0,05	0,02

- Mittlere Codewortlänge:
  - Huffman: 2,78
  - Shannon-Fano: 2,82

- Dekodieren Sie die folgenden Nachrichten
  - Huffman: 01011000001001
  - Shannon-Fano: 11100001111111

Arithmetische Codierung

- Huffman-Codierung ist nicht optimal, da jedes Zeichen mit einer ganzzahligen Anzahl Bits dargestellt wird
- Theoretisches Optimum wird mit der arithmetischen Codierung erreicht
- Prinzip der arithmetischen Codierung:
  - Darstellung der gesamten Nachricht als Zahl aus $[0,1)$
  - wiederholte Aufteilung des Intervals $[0,1)$ entsprechend der Auftrittswahrscheinlichkeiten der Zeichen
  - Beispiel: AFFE
- Arithmetische Codierung ist patentrechtlich geschützt (IBM)
  - JPEG unterstützt Huffman-Coding und arithmetische Codierung, de facto sind aber alle JPEG-Bilder mit der Huffman-Codierung komprimiert
    - Hinweis: Huffman-Codierung ist nur ein Teil der JPEG-Codierung

Arithmetische Codierung: Beispiel

- AFFE: p(A)=1/4, p(F)=1/2, p(E)=1/4

0-0,25	0,25-0,75	0,75-1,0
A	F	E

A

alle Wörter mit „A“ liegen im Interval [0;1/4]

F

alle Wörter mit „AF“ liegen im Interval [1/16;3/16]

F

alle Wörter mit „AFF“ liegen im Interval [6/64;10/64]

E

„AFFE“ entspricht einer Zahl aus dem Interval [9/64;10/64]
[0,140625 – 0,15625]

- Codierung:
  - Anzahl Buchstaben und sehr genaue Gleitkommazahl
  - Auftrittswahrscheinlichkeiten an Codierer und Dekodierer bekannt oder in Codierung enthalten

Weiteres Beispiel

- Alphabet:

x_i	A	F	E
p_i	0,25	0,5	0,25
o_i	0	0,25	0,75

- Decodierung: Nachricht $0{,}7$ mit 10 Zeichen

y	x_i
$0{,}7$	F
$(0{,}7 - 0{,}25)/0{,}5 = 0{,}9$	E
$(0{,}9 - 0{,}75)/0{,}25 = 0{,}6$	F
$(0{,}6 - 0{,}25)/0{,}5 = 0{,}7$	F
	E
	F

Arithmetische Codierung: Vorschrift

- Ordnen der Zeichen in beliebiger aber fester Reihenfolge
- Definieren für jedes Zeichen X_i:
  - p_i: Auftrittswahrscheinlichkeit
  - $o_i = \sum_{j=1}^{i-1} p_j$: Summe der Auftrittswahrscheinlichkeiten aller Zeichen mit kleinerem Index
- Codierung einer Nachricht:
  - Variablen: s für Start des Intervalls und d für Breite des Intervalls der Nachricht
  - Initialisierung: $\mathrm{s} = 0, d = 1$
  - Codierung eines Zeichens X_i:
    - $s = s + o_i \cdot d$
    - $d = d \cdot p_i$
- Dekodierung einer Nachricht y mit k Zeichen:
  - für jedes der k Zeichen:
    - bestimme das Zeichen x_i, für das gilt $y \in [o_i; o_{i+1})$
    - $y = (y - o_i)/p_i$

Weiteres Beispiel

- Alphabet:

x_i	A	E	F	M	T	U
p_i	0,1	0,1	0,2	0,5	0,05	0,05
o_i	0	0,1	0,2	0,4	0,9	0,95

- Codierung: AFFE

	s	d
	0	1
A	$0 + 0 \cdot 1 = 0$	$1 \cdot 0{,}1 = 0{,}1$
F	$0 + 0{,}2 \cdot 0{,}1 = 0{,}02$	$0{,}1 \cdot 0{,}2 = 0{,}02$
F	$0{,}02 + 0{,}2 \cdot 0{,}02 = 0{,}024$	$0{,}02 \cdot 0{,}2 = 0{,}004$
E	$0{,}024 + 0{,}1 \cdot 0{,}004 = 0{,}0244$	$0{,}004 \cdot 0{,}1 = 0{,}0004$

Weiteres Beispiel

- Alphabet:

x_i	A	E	F	M	T	U
p_i	0,1	0,1	0,2	0,5	0,05	0,05
o_i	0	0,1	0,2	0,4	0,9	0,95

- Codierung: MAMMUT

	s	d
	0	1
M	$0+0{,}4\cdot 1=0{,}4$	$1\cdot 0{,}5=0{,}5$
A	$0{,}4+0\cdot 0{,}5=0{,}4$	$0{,}5\cdot 0{,}1=0{,}05$
M	$0{,}4+0{,}4\cdot 0{,}05=0{,}42$	$0{,}05\cdot 0{,}5=0{,}025$
M	$0{,}42+0{,}4\cdot 0{,}025=0{,}423$	$0{,}0025\cdot 0{,}5=0{,}0125$
U	$0{,}43+0{,}95\cdot 0{,}0125=0{,}441875$	$0{,}00125\cdot 0{,}05=6{,}25e-4$
T	$0{,}441875+0{,}9\cdot 6{,}25\mathrm{e}-4=0{,}4424375$	$6{,}25e-5\cdot 0{,}05=3{,}125e-5$

Weiteres Beispiel

- Alphabet:

x_i	A	E	F	M	T	U
p_i	0,1	0,1	0,2	0,5	0,05	0,05
o_i	0	0,1	0,2	0,4	0,9	0,95

- Decodierung: $y = 0{,}44245, k = 6$

y	x_i
$0{,}422245$	M
$(0{,}422245 - 0{,}4\,)/0{,}5 = 0{,}0849$	A
$(0{,}0849 - 0\,)/0{,}1 = 0{,}849$	M
$(0{,}849 - 0{,}4\,)/0{,}5 = 0{,}898$	M
$(0{,}898 - 0{,}4\,)/0{,}5 = 0{,}996$	U
$(0{,}996 - 0{,}95\,)/0{,}05 = 0{,}92$	T
$(0{,}92 - 0{,}9\,)/0{,}05 = 0{,}4$	M
$(0{,}4 - 0{,}4\,)/0{,}5 = 0$	A

Kapitelstruktur

Wörterbuch-basierte Verfahren

- Lempel-Ziv:
  - Verwendung der vorherigen Daten als Wörterbuch
  - Einsatzgebiete:
    - PNG
    - kombiniert mit Huffman-Codierung als DEFLATE in ZIP, TIFF, PDF
- Lempel-Ziv-Welch:
  - dynamischer Aufbau eines Wörterbuchs beim Komprimieren
  - Einsatzgebiete:
    - GIF
    - Variante: HTTP\2.0 Header Compression

Lempel Ziv

- Zeichenkette wird über Tripel aus Position, Länge und nächstem Zeichen codiert
  - codiert wird längste Zeichenkette, die im bisherigen Text vorkam
  - Position wird ab der Stelle des codierten Zeichens rückwärts gezählt
  - Maximalwert für die Position beschränkt die Größe des Wörterbuchs
- Beispiel:

```
01234567890123456789012345678901
FISCHERSFRITZFISCHTFRISCHEFISCHE

 (13,5,T) (Position, Anzahl, nächstes Zeichen)
```

Lempel Ziv LZ77 - Beispiel

- Beispiel (ohne Einschränkung für maximale Position und Länge):

```
01234567890123456789012345678901
FISCHERSFRITZFISCHTFRISCHEFISCHE
0,0,F
 0,0,I
       5,1,F
             13,5,T
                   11,3,S
                       20,3,F
                           6,5,''
```

Beispiel Decodierung

- Codierte Nachricht:

  (0,0,B),(0,0,A),(0,0,N),(2,2,E),(,4,3,B),(3,1,U)

- Dekodierung:

(0,0,B)		B
(0,0,A)		BA
(0,0,N)		BAN
(2,2,E)	BAN_	BANANE
(4,3,B)	BANANE_	BANANENANB
(3,1,U)	BANANENANB_	BANANENANBAU

Lempel Ziv LZ77

- Codierung von Strings in Tripel aus
  - Position: n Bit
    - 6 Bit bedeuten eine Referenzwörterbuchlänge von 2^6=64 Zeichen
  - Anzahl: m Bit
    - 5 Bit bedeuten eine maximale Wortlänge von 2^5=32 Bit
  - nächstem Zeichen (wie in der Originalnachricht)
    - benötigte Bits hängen von Anzahl Zeichen im Alphabet ab
- Beispiel:

```
01234567890123456789012345678901
FISCHERSFRITZFISCHTFRISCHEFISCHE
```

  - insgesamt 32 Zeichen codiert in 15 Tripeln aus Position (5 Bit), Anzahl (3 Bit), nächstem Zeichen (Annahme: 8 Bit bei ASCII-Code)
    - Codiert: 15*(5 Bit+3 Bit+8 Bit)=15*16 Bit=240 Bit
    - Original: 32*8 Bit=248 Bit
- Effizient bei längeren Dateien mit Wörterbuchgröße von 2 bis 32kBytes

Lempel-Ziv-Welch

- Lempel-Ziv-Welch: dynamischer Aufbau eines Wörterbuchs beim Komprimieren
- Beispiel: FISCHERSFRITZFISCHTFRISCHEFISCHE

Zeichen	Codierung	Wörterbuch
F	F	0: FI
I	I	1: IS
S	S	2: SC
C	C	3: CH
H	H	4: HE
E	E	5: ER
R	R	6: RS
S	S	7: SF
F	F	8: FR
R	R	9: RI
I	I	10: IT

Zeichen	Codierung	Wörterbuch
T	T	11: TZ
Z	Z	12: ZF
FI	<0>	13: FIS
SC	<2>	14: SCH
H	H	15: HT
T	T	16: TF
FR	<8>	17: FRI
IS	<1>	18: ISC
CH	<3>	19: CHE
E	E	20: EF
FIS	<13>	21: FISC

Lempel-Ziv-Welch

- Wörterbucheinträge bestehen aus Zeichenketten, denen Nummern zugewiesen werden
  - Initial besteht das Wörterbuch aus einem Grundalphabet, das hier nicht weiter betrachtet wird.
  - Im Beispiel von einem leeren Wörterbuch ausgegangen.
- Codierung:
  - suche ab dem zu codierenden Zeichen die längste im Wörterbuch vorkommende Zeichenkette
  - hänge Nummer dieser Zeichenkette an das Codewort an
  - füge <Nummer>+nächstes Zeichen dem Wörterbuch hinzu
  - fahre mit der Codierung ab dem nächsten Zeichens fort
- Dekodierung (ab dem 2. Zeichen):
  - dekodierte Nachricht besteht aus Symbolen y_i, die einer Zeichenkette $x_{i,1}, \ldots, x_{i,K}$ entsprechen
  - bei Dekodierung von Symbol y_{i+1}
    - hänge Zeichenkette $x_{i,1}, \ldots, x_{i,K}$ an die dekodierte Nachricht an
    - füge Zeichenkette $x_{i,1}, \ldots, x_{i,K} y_{i+1,1}$ dem Wörterbuch hinzu

Beispiel Decodierung

- Codierte Nachricht:

  BAN<1>E<2>N<0>U

- Dekodierung:

Nachricht	Ausgabe	Wörterbuch
B	B	
A	A	0: BA
N	N	1: AN
<1>	AN	2: NA
E	E	3: ANE
<2>	NA	4: EN
N	N	5: NAN
<0>	BA	6: NB
U	U	7: BAU

Kapitelstruktur

Anwendungsbeispiel Huffman-Code: HTTP/2.0

- HTTP/2.0 verwendet Kompressionsverfahren zur Übertragung von Header-Feldern

http://http2.github.io/http2-spec/compression.html#static.table.definition

Tabelle für häufige Header-Felder

Index	Header Name	Header Value
1	:authority	
2	:method	GET
3	:method	POST
4	:path	/
5	:path	/index.html
6	:scheme	http
7	:scheme	https
8	:status	200
9	:status	204
10	:status	206
11	:status	304
12	:status	400
13	:status	404
14	:status	500
15	accept-charset	

Huffman-Code für Zeichencodierung

```
'%' ( 37)  |010101          15  [ 6]
'&' ( 38)  |11111000        f8  [ 8]
''' ( 39)  |11111111|010   7fa  [11]
'(' ( 40)  |11111110|10    3fa  [10]
')' ( 41)  |11111110|11    3fb  [10]
'*' ( 42)  |11111001        f9  [ 8]
'+' ( 43)  |11111111|011   7fb  [11]
',' ( 44)  |11111010        fa  [ 8]
'-' ( 45)  |010110          16  [ 6]
'.' ( 46)  |010111          17  [ 6]
'/' ( 47)  |011000          18  [ 6]
'0' ( 48)  |00000            0  [ 5]
'1' ( 49)  |00001            1  [ 5]
'2' ( 50)  |00010            2  [ 5]
'3' ( 51)  |011001          19  [ 6]
'4' ( 52)  |011010          1a  [ 6]
'5' ( 53)  |011011          1b  [ 6]
'6' ( 54)  |011100          1c  [ 6]
```

Beispiel: HTTP Header-Encoding

Header list to encode:

```
:method: GET
:scheme: http
:path: /
:authority: www.example.com
```

Hex dump of encoded data:

```
8286 8441 8cf1 e3c2 e5f2 3a6b a0ab 90f4 | ...A......:k....
ff                                      | .
```

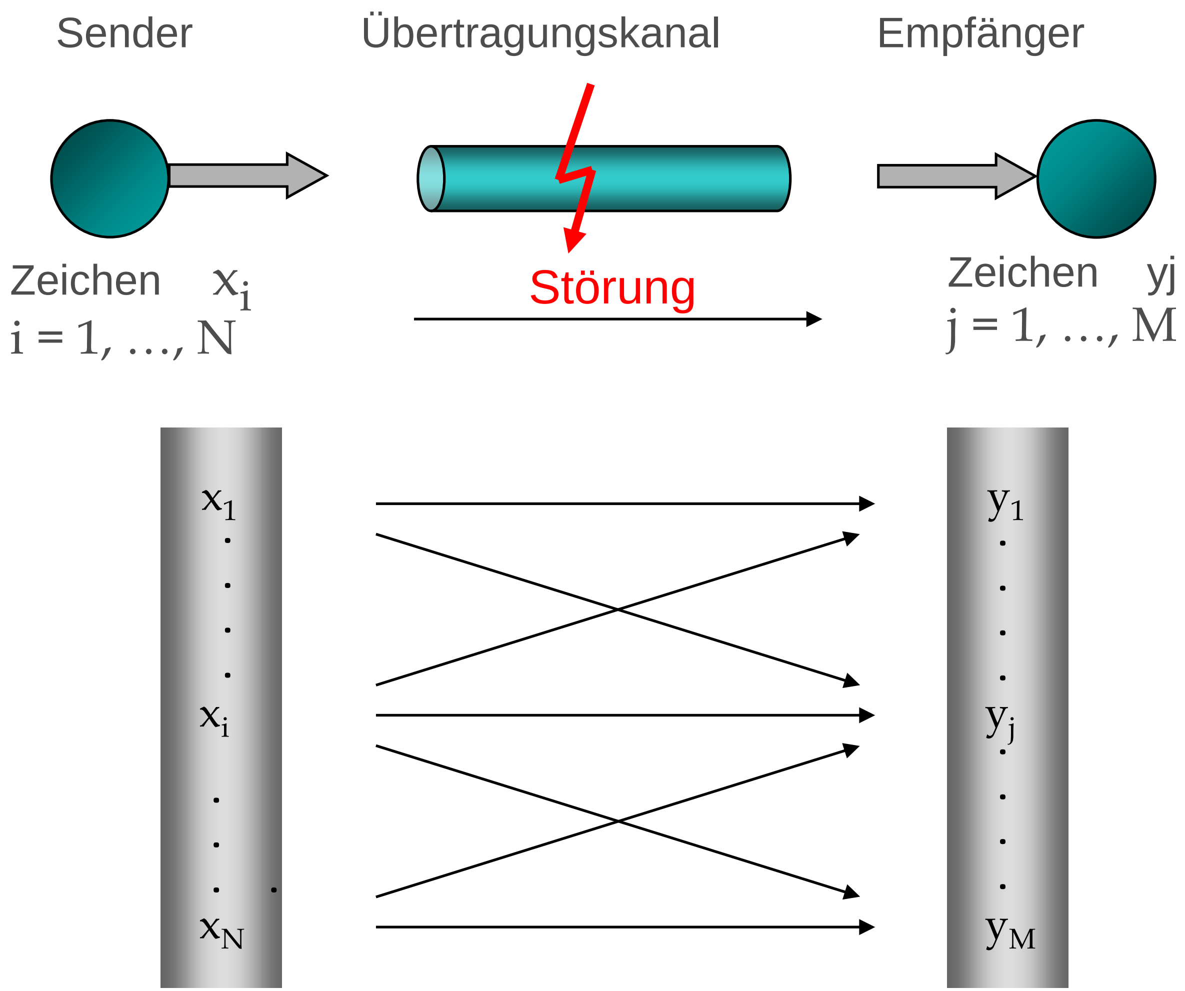
Decoding process:

Codierung über Tabelle

```
82                                      | == Indexed - Add ==
                                        |   idx = 2
                                        | -> :method: GET
86                                      | == Indexed - Add ==
                                        |   idx = 6
                                        | -> :scheme: http
84                                      | == Indexed - Add ==
                                        |   idx = 4
                                        | -> :path: /
41                                      | == Literal indexed ==
                                        |   Indexed name (idx = 1)
                                        |     :authority
8c                                      |   Literal value (len = 12)
                                        |     Huffman encoded:
f1e3 c2e5 f23a 6ba0 ab90 f4ff           | .....:k.....
                                        |     Decoded:
                                        | www.example.com
                                        | -> :authority: www.example\
                                        |   .com
```

Huffman-Codierung

Informationsübertragung



Wie viel Information wird übertragen?

Beispiel

- Quelle:
  - Zeichen „0“ und „1“ mit gleichen Auftrittswahrscheinlichkeiten
  - Entropie der Quelle: 1 Bit
- Übertragungskanal:
  - Verfälschung mit einer Wahrscheinlichkeit von 10%
    - Wahrscheinlichkeit für korrekte Übertragung: 90%
    - Wahrscheinlichkeit für Übertragungsfehler: 10%
- Senke:
  - Zeichen „0“ und „1“ treten auch hier mit Wahrscheinlichkeiten von 50% auf
  - Entropie der Senke: 1 Bit
- Wieviel Information wird übertragen?

Transinformationsgehalt

- Betrachten ein Verbundereignis bestehend aus gesendetem und empfangenem Zeichen
  - Übertragungsfehler: „0“ wird gesendet und „1“ wird empfangen
    - Wahrscheinlichkeit für Verbundereignis: $p(x0, y1) = 0{,}5 \cdot 0{,}1 = 0{,}05$
  - Korrekte Übertragung: „0“ wird gesendet und „0“ wird empfangen
    - Wahrscheinlichkeit für Verbundereignis: $p(x0, y0) = 0{,}5 \cdot 0{,}9 = 0{,}45$
- Transinformationsgehalt einer Übertragung:
  - Gehalt der Information, die aus den empfangenen Zeichen an der Senke über die gesendeten Zeichen an der Quelle geschlossen werden kann

$$I_T(x_i, y_j) = ld \frac{p(x_i, y_j)}{p(x_i) \cdot p(y_j)}$$

  - Transinformationsgehalt für Übertragungsfehler

$$I_T("0","1") = ld \frac{0{,}05}{0{,}5 \cdot 0{,}5} \approx -2{,}322$$

  - Transinformationsgehalt für korrekte Übertragung

$$I_T("0","0") = ld \frac{0{,}45}{0{,}5 \cdot 0{,}5} \approx 0{,}848$$

Transinformationsgehalt

- Transinformationsgehalt eines Übertragungskanals
  - Erwartungswert des Transinformationsgehalts einer zufälligen Übertragung

$$T(X,Y) = \sum_{i=1}^{N} \sum_{j=1}^{N} p(x_i, y_j) \cdot ld \frac{p(x_i, y_j)}{p(x_i) \cdot p(y_j)}$$

  - Beispiel:

$$T(X,Y) = 0{,}05 \cdot I_T(\text{"0"},\text{"1"}) + 0{,}45 \cdot I_T(\text{"0"},\text{"0"})$$
$$+0{,}05 \cdot I_T(\text{"1"},\text{"0"}) + 0{,}45 \cdot I_T(\text{"1"},\text{"1"})$$
$$= 0{,}1 \cdot -2{,}322 + 0{,}9 \cdot 0{,}848 = 0{,}531$$

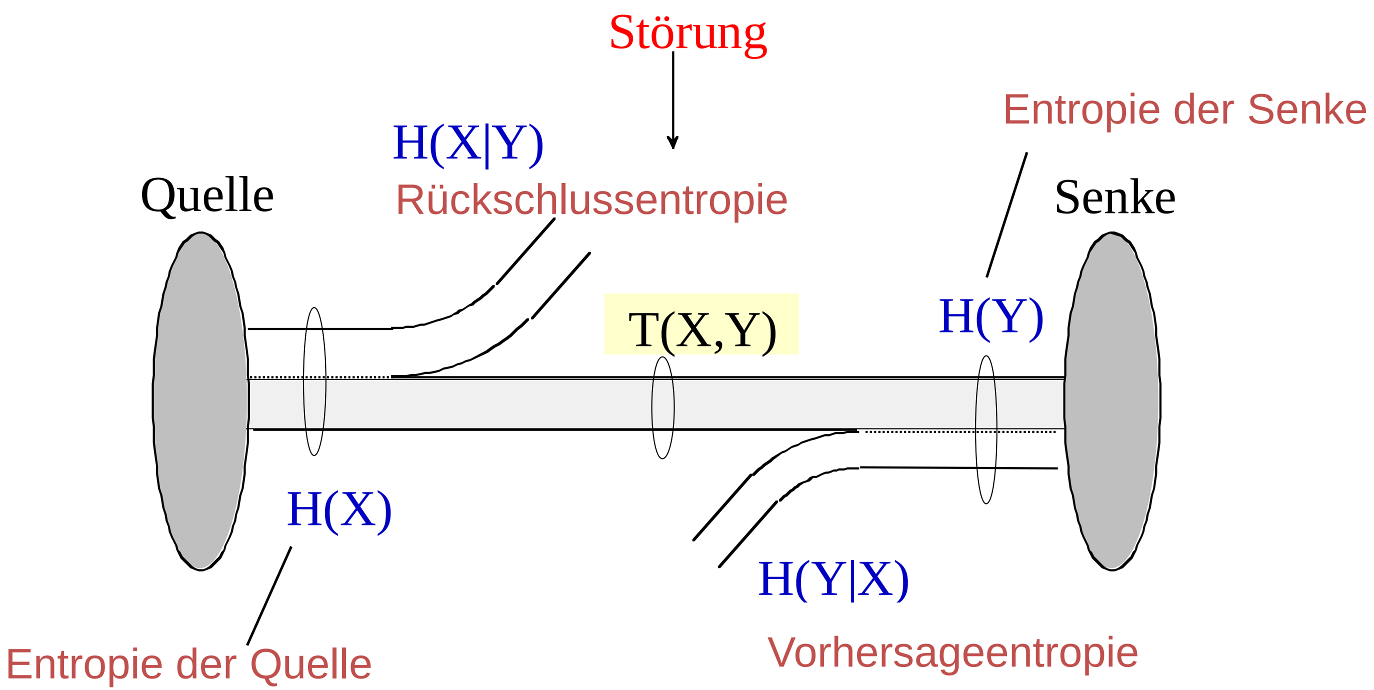
  - Ungestörter Kanal: $p(x_i, y_j) = \begin{cases} p(x_i) & , für\ i = j \\ 0 & i \neq j \end{cases}$

$$I_T(\text{"1"},\text{"1"}) = ld \frac{p(x_i)}{p(x_i) \cdot p(y_j)} = ld \frac{1}{p(y_j)} = I(y_j)$$
$$T(X,Y) = H(X) = H(Y)$$

  - Maximal gestörter Kanal: $p(x_i, y_j) = p(x_i) \cdot p(y_j)$

$$I_T(x_i, y_j) = ld\ 1 = 0$$
$$T(X,Y) = 0$$

Entropieflüsse



- Definitionen:
  - Transinformationsgehalt:
    $$T(X,Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) = H(X) + H(Y) - H(X,Y)$$
  - Rückschlussentropie:
    $$H(X|Y) = \sum_{j=1}^{N} p(y_j) \sum_{I=1}^{N} p(x_i|y_j) \cdot ld \frac{1}{p(x_i|y_j)}$$

Kapitelstruktur

Binärer Symmetrischer Übertragungskanal

- BSC: Binary Symmetric Channel
- Binär:
  - Übertragene Zeichen sind Bits
- Symmetrisch:
  - Bitfehler mit Wahrscheinlichkeit q
    - Verfälschung von „0“ nach „1“ und „1“ nach „0“ sind gleichwahrscheinlich
  - Maximale Entropie der Quelle
    - Zeichen „1“ und „0“ sind gleichwahrscheinlich
- Kanalkapazität: C [bit/s]
  - Dauer eines Bits: 1/C
- Frage: Was ist die Kapazität des Kanals? Wie viel Information kann übertragen werden?

Kapazität des binären Übertragungskanal

Übertragung	(0,0)	(1,1)	(0,1)	(1,0)
Verbundwahrs‘ $p(x_i, y_j)$	$0{,}5 \cdot (1-q)$	$0{,}5 \cdot (1-q)$	$0{,}5 \cdot q$	$0{,}5 \cdot q$
Transinformationsgehalt $I_T(x_i, y_j)$	$ld\left(\frac{0{,}5 \cdot (1-q)}{0{,}5 \cdot 0{,}5}\right)$		$ld\left(\frac{0{,}5 \cdot q}{0{,}5 \cdot 0{,}5}\right)$	
Transinformationsgehalt T(X,Y)	$(1-q) \cdot ld\big(2 \cdot (1-q)\big) + q \cdot ld(2 \cdot q)$			
$ld(2x) = 1 + ld(x)$ $ld(x) = -ld(1/x)$	$1 - (1-q) \cdot ld\frac{1}{1-q} - q \cdot ld\frac{1}{q}$			

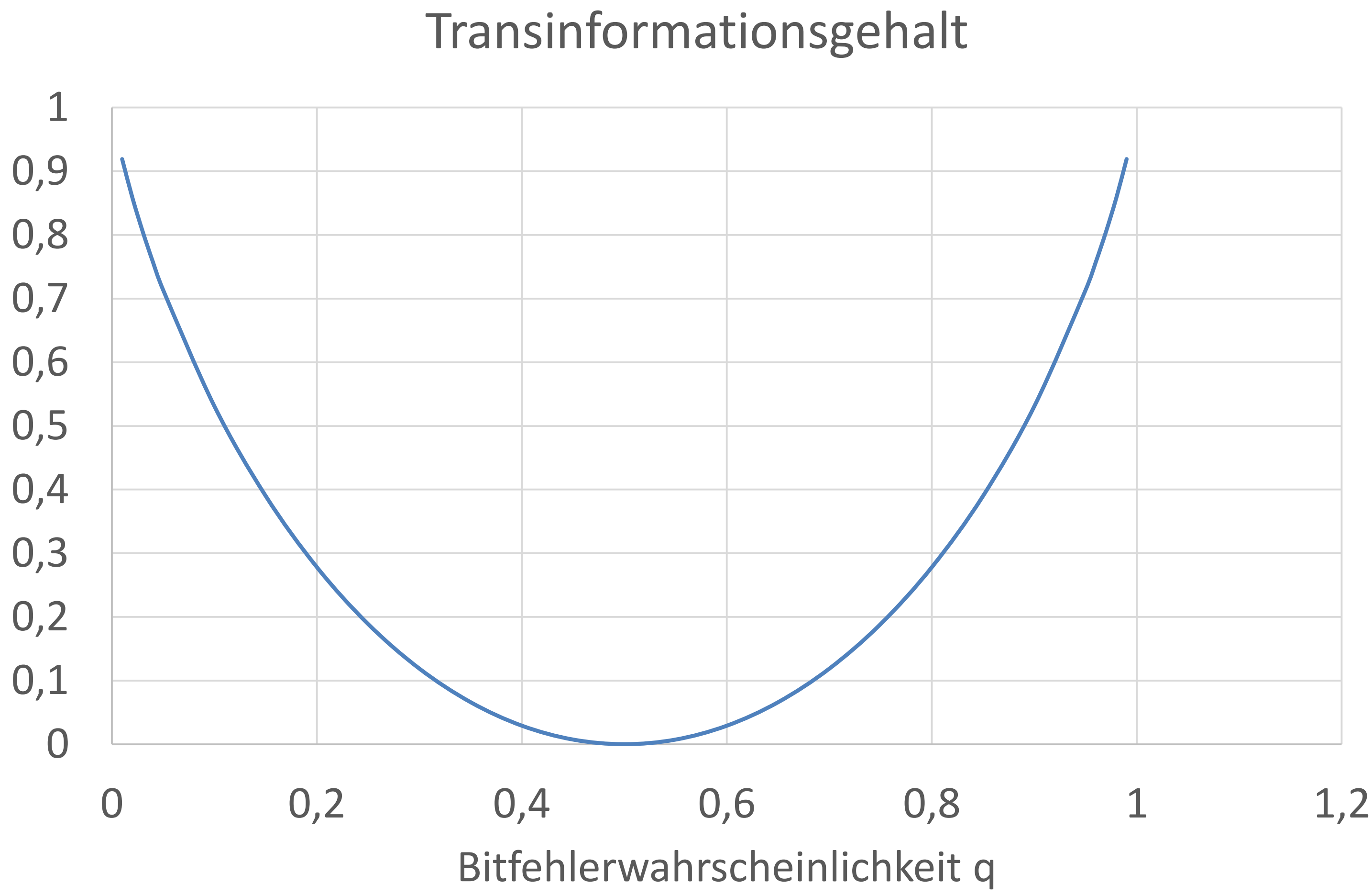
- Transinformationsgehalt des binären Übertragungskanals

$$\boxed{\text{T(X,Y)}=1 - (1-q) \cdot ld\frac{1}{1-q} - q \cdot ld\frac{1}{q}}$$

- Kapazität des binären Übertragungskanals

$$\boxed{C^* = \text{C} \cdot \left\{1 - (1-q) \cdot ld\frac{1}{1-q} - q \cdot ld\frac{1}{q}\right\}}$$

Transinformationsgehalt des binären Übertragungskanal



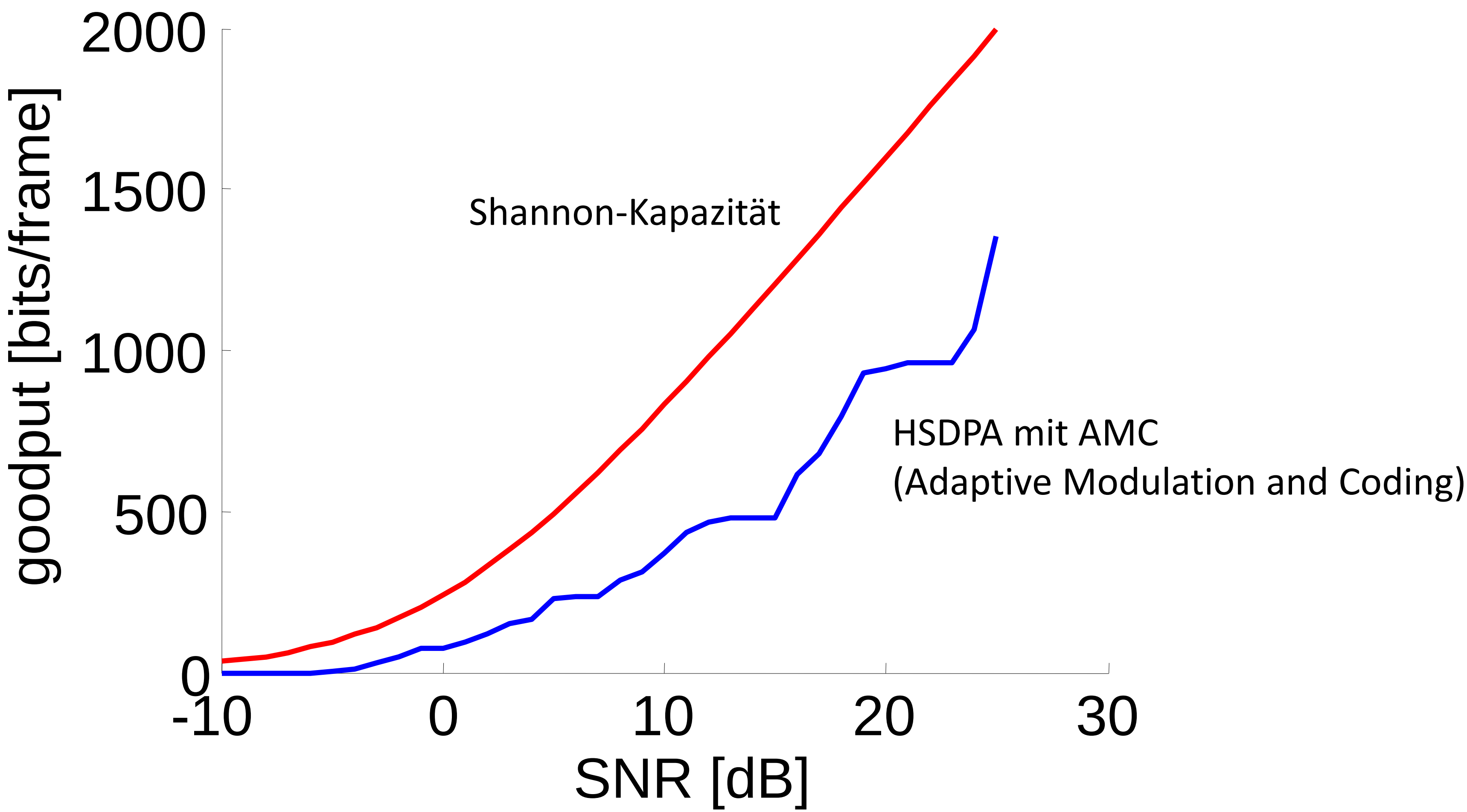
Shannon-Kapazität

- Binärer Übertragungskanal stellt ein Modell dar, in dem die tatsächliche Störung von Signalen auf Bitfehlerwahrscheinlichkeiten abgebildet wird
- In der Realität haben wir einen kontinuierlichen analogen Übertragungskanal, auf dem analoge Signale übertragen und gestört werden
  - Bandbreite des Kanals: B [MHz]
    - z.B. WLAN Kanal mit 20 MHz
  - Signal-Rausch-Verhältnis (SNR, Signal-to-Noise-Ratio): Verhältnis von Leistung des Sendesignals P_S zu Rauschleistung P_N am Empfänger
  - Shannon-Kapazität des Kanals

$$\boxed{C^* = B \cdot \log_2(1 + SNR)}$$

Beispiel: HSDPA

Moderne Übertragungsverfahren erreichen fast die theoretische Shannon-Kapazität – mit MIMO wurde das Shannon-Limit durchbrochen



Kapitelstruktur

2.1 Informationsgehalt

2.2 Quellcodierung

2.3 Informationsgehalt eines Übertragungskanals

2.4 Zusammenfassung

Zusammenfassung

- Informationsgehalt
  - definiert wieviel Information eine Quelle sendet
  - bildet eine obere Grenze für die Performance von Kompressionsalgorithmen
- Kompressionsalgorithmen
  - Entropie-Codierung: Shannon-Fano, Huffman
  - Wörterbuchcodierung: Lempel-Ziv
- Obergrenze für Übertragungskanal
  - Transinformationsgehalt
  - Shannon-Kapazität

$$\boxed{C^* = B \cdot \log_2(1 + SNR)}$$

  - theoretische Kanalkapazität wird mit Kanal- und Leitungscodierung (fast) erreicht