

Übung zur Vorlesung  
**Digitale Medien**

Doris Hausen  
Ludwig-Maximilians-Universität München  
Wintersemester 2011/2012

# Codierung

ASCII:  
aaaaabbbcde 96 Bit

Morse-Code (2 Bit / Symbol):  
. - . - . - . - . - -... -... -... -.-.-... 86 Bit

Laufängerkodierung:  
#a6#b3cde 72 Bit

Huffmankodierung:  
11111101010100100010000 23 Bit

Arithmetische Kodierung:  
000000101010110100111 21 Bit

aaaaabbbcde

# Lauf­längencodierung



AAAAAAAAAAAAAAAAAAAA

20 Zeichen

#A20

4 Zeichen

Dies ist ein Beispieltext.

26 Zeichen

Dies ist ein Beispieltext.

26 Zeichen

Idee: Ersetzen einer Folge gleicher Zeichen durch 1 Zeichen + Zähler

Funktioniert besser mit Bildern als mit Text

# Entropie (1)

AAAAAAAAAAAAAAAA  
AAAAAAAAAAAAAAAA  
AAAAAAAAAAAAAAAA  
AAAAAAAAAAAAAAAA  
AAAAAAAAAAAAAAAA  
AAAAAAAAAAAAAAAA  
AAAAAAAAAAAAAAAA

Jede Nachricht hat einen Informationsgehalt, die *Entropie*.

Generell: Die Entropie gibt an, wie „überraschend“ es ist, in der Nachricht ein bestimmtes Zeichen anzutreffen.

$$p(A) = 1$$

$$\text{Entropie } H = 0$$

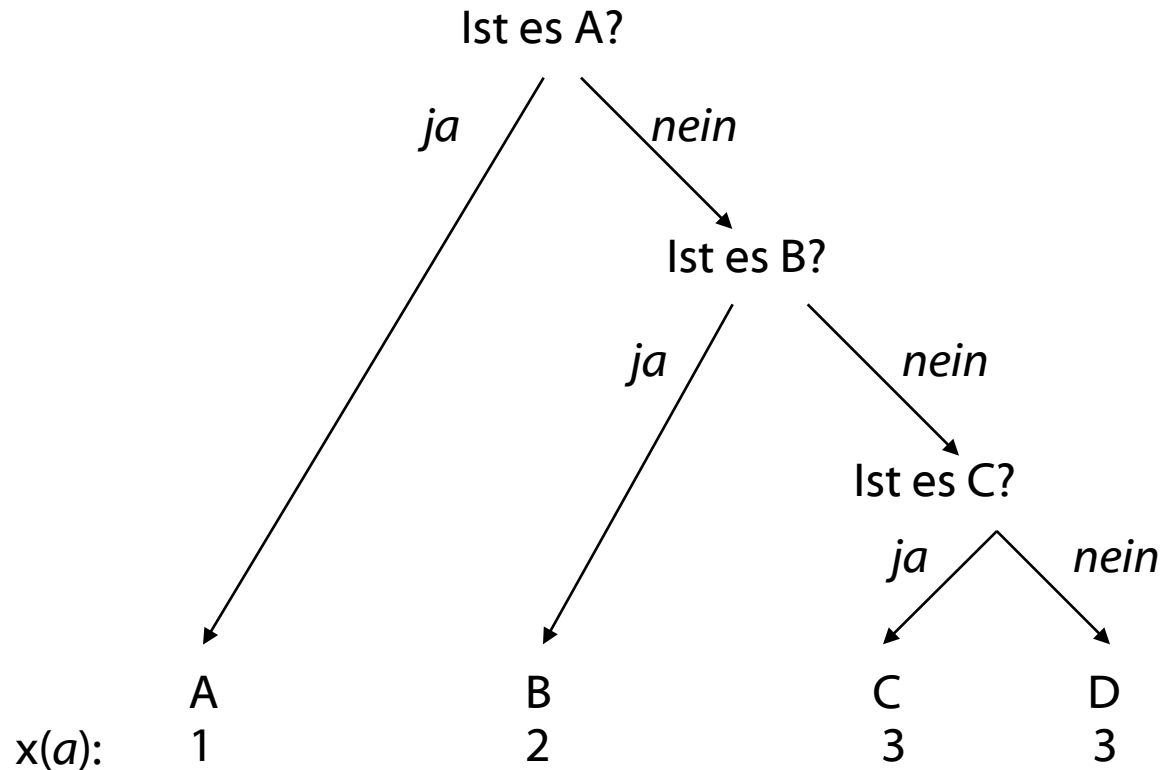
# Entropie (2)

$p(a)$ : Wahrscheinlichkeit, dass  $a$  auftritt  
 $x(a)$ : Anzahl der Entscheidungen für  $a$

```
AAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAA
BBBBBBBBBBBBBBBBBB
BBBBBBBBBBBBBBBBBB
CCCCCCCCCCCCCCCCC
DDDDDDDDDDDDDDDD
```

$p(A) = 0,5$   
 $p(B) = 0,25$   
 $p(C) = 0,125$   
 $p(D) = 0,125$

$$x(a) = \text{ld} ( 1 / p(a) )$$



# Einschub: Zweierlogarithmus

$$a = b^x$$

$$x = \log_b a$$

Beispiele:

$$256 = 2^x$$

$$x = \log_2 256$$

$$x = 8$$

$$1.000.000 = 10^x$$

$$x = \log_{10} 1.000.000$$

$$x = 6$$

$\log_e$  heißt *ln* (natürlich Log.)  
 $\log_2$  heißt *ld* (oder *lg* im Englischen)  
 $\log_{10}$  heißt *lg* (*log* auf Taschenrechnern)

$$\log_b x = \log_y x / \log_y b$$

$$\log_2 x = \ln x / \ln 2$$

$$\log_2 x = \lg x / \lg 2$$

...

$\log_2 256$  im Taschenrechner:

- „256“ eintippen
- „log“ drücken
- „/“ drücken
- „2“ eintippen
- „log“ drücken
- „=“ drücken

Online „Scientific Calculator“:

<http://www.calculator.com>

<http://www.google.com>

# Entropie (3)

	p	x
A	0,5	1
B	0,25	2
C	0,125	3
D	0,125	3

$$H = 1,75$$

Beispielcode:

	c	c
A	00	2
B	01	2
C	10	2
D	11	2

$$L = 2$$

$$R = 0,25$$

$$\text{Entropie } H = \sum p(a) * x(a)$$

Durchschnittlicher Entscheidungsgehalt  
eines Zeichen

(p(a): Wahrscheinlichkeit, dass a auftritt  
x(a): Anzahl der Entscheidungen für a)

$$L = \sum p(a) |c(a)|$$

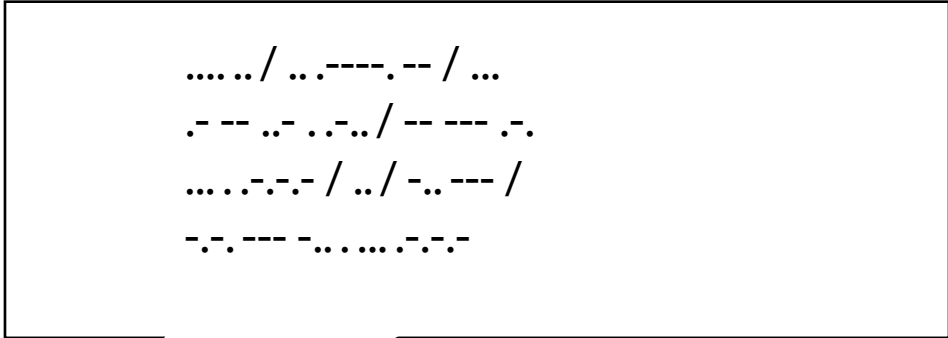
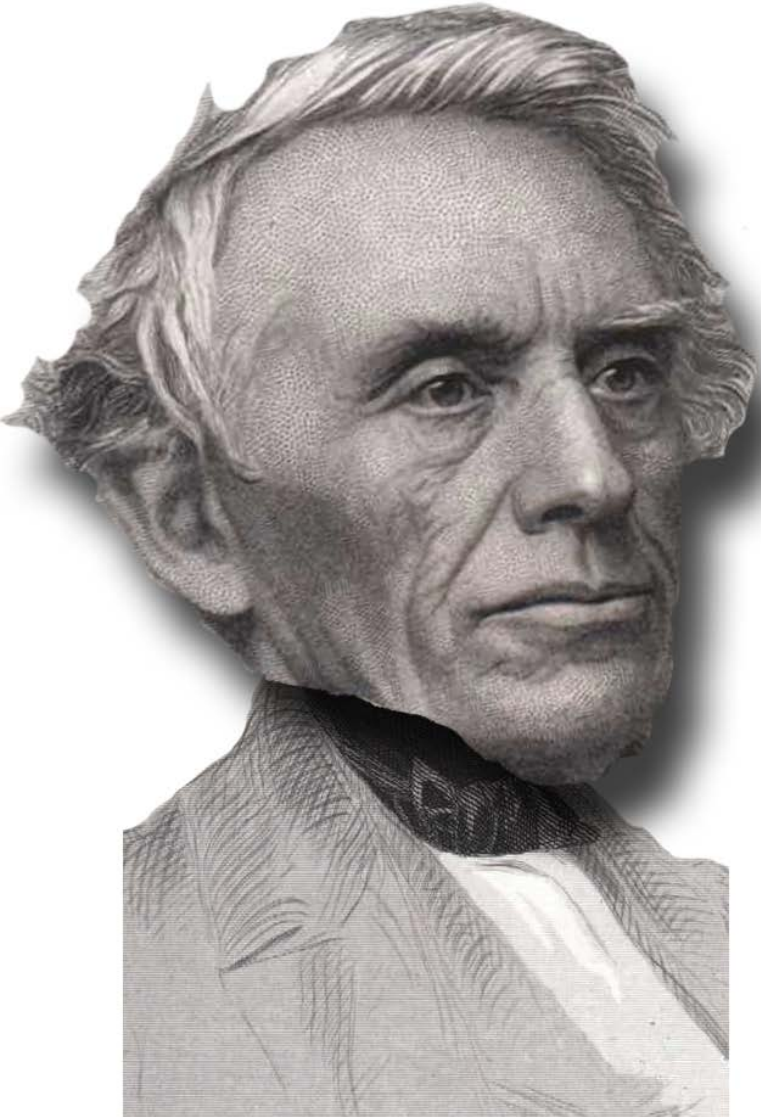
Durchschnittlicher Wortlänge  
pro Zeichen

$$R = L - H$$

Redundanz eines Codes

Je kleiner, desto besser!

# Samuel Morse



## Idee:

Je häufiger ein Zeichen vorkommt,  
desto kürzer das kodierte Symbol  
→ kürzere Nachrichten mit gleichem Inhalt!

Allerdings: Kein binärer Code (kurz ., lang -  
und Pause /)!  
Häufigkeiten falsch eingeschätzt!



# David Huffman



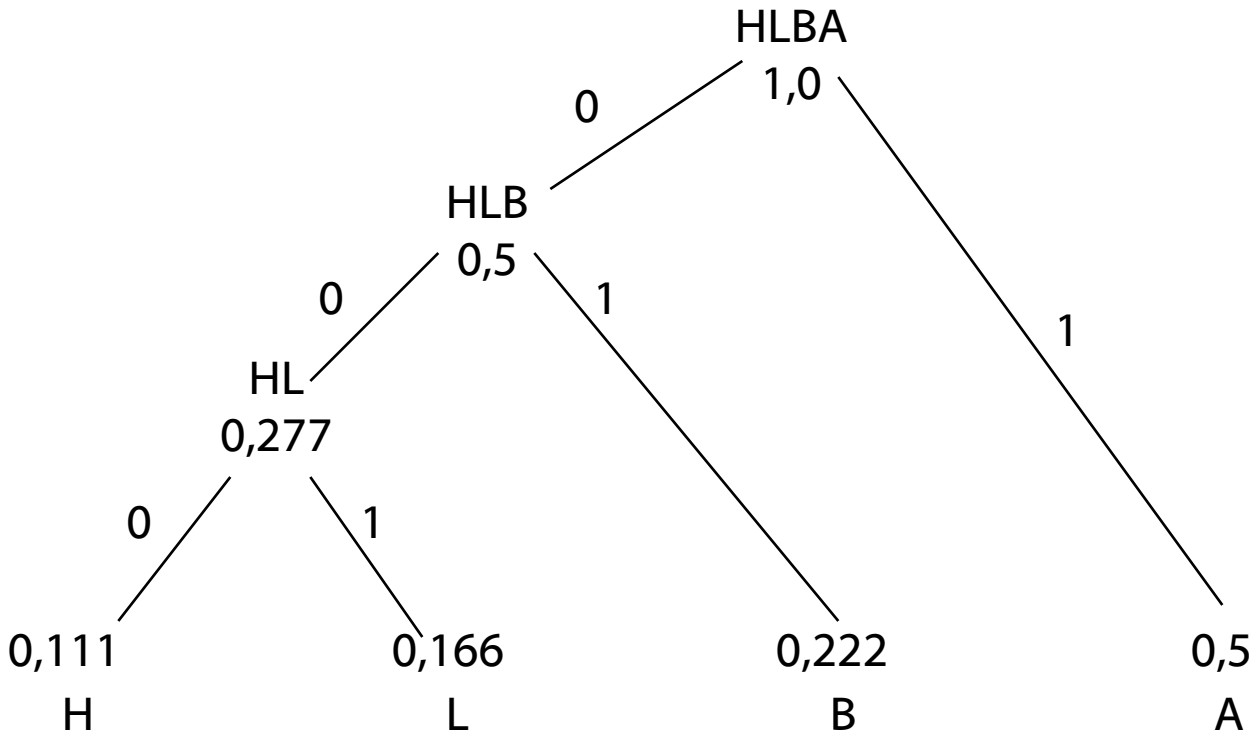
In optimalem Code müssen die beiden Symbole der niedrigsten Häufigkeit mit gleicher Länge codiert sein.

# Huffman Codierung

1. Ermittlung der Häufigkeiten
2. Aufbau des Codebaums (von unten)
3. Code

AAAAHHABBLLABBLAAA

A	9/18	0,5
B	4/18	0,222
L	3/18	0,166
H	2/18	0,111



A	1
B	01
L	001
H	000

# Huffman Codierung

aaaaaabbbcde

Ergebnis:

11111101010100100010000

a	1
b	01
c	001
d	0001
e	0000

Warum nicht dieser kürzere Code?

a	1
b	01
c	10
d	11
e	100

Nicht dekodierbar!  
Fano-Bedingung verletzt!

# Ist der Code optimal?

	p	x
a	0,5	1
b	0,25	2
c	0,083	3,58
d	0,083	3,58
e	0,083	3,58

$$H = 1,896$$

$$x(a) = \text{ld} ( 1 / p(a) )$$

$$H = \sum p(a) x(a)$$

	c	c
a	1 1	2
b	01	2
c	001	3
d	0001	4
e	0000	4

$$L = 1,917$$

$$L = \sum p(a) |c(a)|$$

$$R = L - H$$

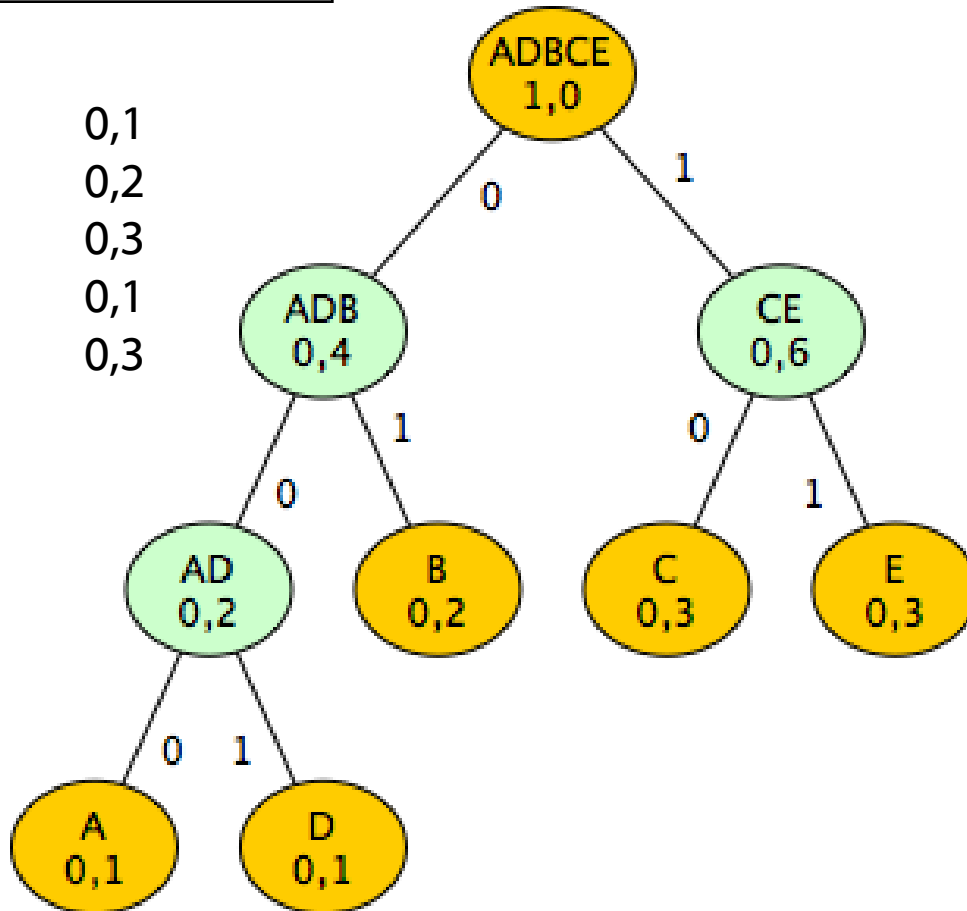
$$R = 0,021$$

Generell: Huffman-Code optimal,  
falls Häufigkeiten negative/Kehrwerte von  
Zweierpotenzen sind, also 0,5 , 0,25 , 0,125

# Anderes Beispiel

CECEDBCABE

A	1/10	0,1
B	2/10	0,2
C	3/10	0,3
D	1/10	0,1
E	3/10	0,3



1. Ermittlung der Häufigkeiten
2. Aufbau des Codebaums
3. Code

Code:

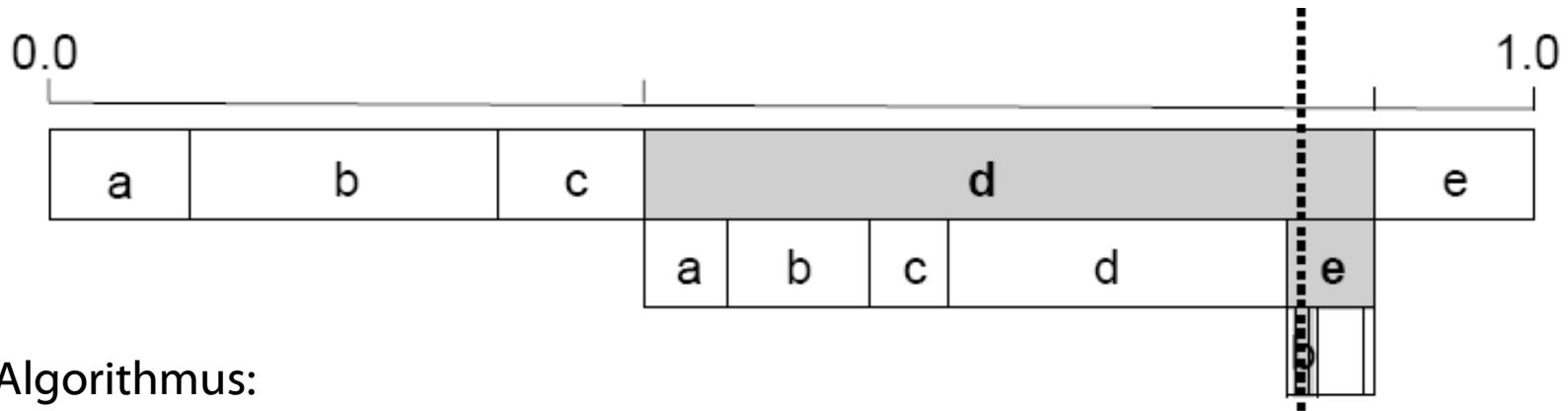
A	000
B	01
C	10
D	001
E	11

# Arithmetische Codierung (1)

D E B

Idee:

Codieren nicht zeichenweise sondern der kompletten Nachricht in einem Zahlenintervall von 0 bis 1. Jedes Zeichen erhält ein Teilintervall je nach Häufigkeit.



Algorithmus:

**real** L = 0.0; **real** R = 1.0;

**Solange** Zeichen vorhanden **wiederhole**

Lies Zeichen und bestimme Zeichenindex i;

**real** B = (R-L); (Intervallbreite)

R = L + B\*R<sub>i</sub>; (Obergrenze)

L = L + B\*L<sub>i</sub>; (Untergrenze)

**Ende Wiederholung;**

L<sub>i</sub> und R<sub>i</sub> sind Ränder eines Zeichens, definiert durch seine Auftrittswahrscheinlichkeit

*Code des Textes ist Zahl im Intervall [L, R]*

# Arithmetische Codierung (2)

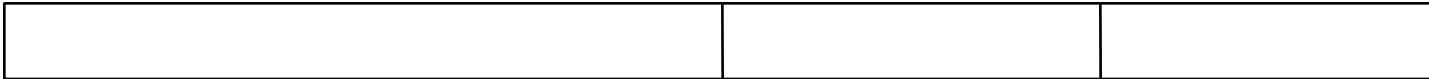
<p>ABCA</p>
-------------

A	0,5	$L_0 = 0$	$R_0 = 0,5$
B	0,25	$L_1 = 0,5$	$R_1 = 0,75$
C	0,25	$L_2 = 0,75$	$R_2 = 1,0$

**A [0;0,5]**

**B [0,5;0,75]**

**C [0,75;1]**



**A [0;0,25]**

**B [0,25;0,375]**

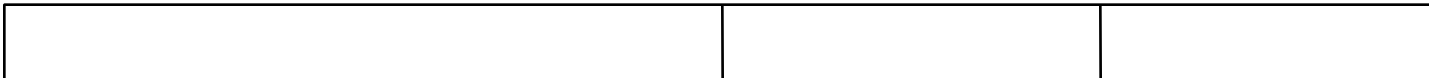
**C [0,375;0,5]**



**A [0,25;0,3125]**

**B [0,3125;0,34375]**

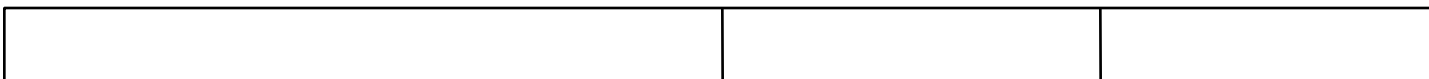
**C [0,34375;0,375]**



**A [0,34375;0,359375]**

**B [0,359375;0,3671875]**

**C [0,3671875;0,375]**



# Arithmetische Codierung (3)

Ergebnisintervall: [0,34375;0,359375]

Untere Grenze in binär:  
0,34375

Obere Grenze in binär:  
0,359375

$$0,34375 \times 2 = \mathbf{0,6875}$$

$$0,6875 \times 2 = \mathbf{1,375}$$

$$0,375 \times 2 = \mathbf{0,75}$$

$$0,75 \times 2 = \mathbf{1,5}$$

$$0,5 \times 2 = \mathbf{1}$$

$$0,359375 \times 2 = \mathbf{0,71875}$$

$$0,71875 \times 2 = \mathbf{1,4375}$$

$$0,4375 \times 2 = \mathbf{0,875}$$

$$0,875 \times 2 = \mathbf{1,75}$$

$$0,75 \times 2 = \mathbf{1,5}$$

$$0,5 \times 2 = \mathbf{1}$$

**0,01011**

**0,010111**

Code endet mit der ersten Ziffer, in der sich  
Ober- und Untergrenze in binär unterscheiden.  
Das führende „0,“ wird weggelassen.

**Code: 010111**