Übung zur Vorlesung

Digitale Medien

Sarah Tausch
Alice Thudt
Ludwig-Maximilians-Universität München
Wintersemester 2012/2013

Codierung

aaaaaabbbcde

ASCII:
aaaaaabbbcde — 96 Bit

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

Lauflängenkodierung:
#a6#b3cde — 72 Bit

Huffmankodierung:
11111101010100100010000 — 23 Bit

Arithmetische Kodierung:
000000101010110100111 — 21 Bit

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)

AAAAAAAAAAAAAAA
AAAAAAAAAAAAAAA
AAAAAAAAAAAAAAA
AAAAAAAAAAAAAAA
AAAAAAAAAAAAAAA
AAAAAAAAAAAAAAA
AAAAAAAAAAAAAAA

Jede Nachricht hat einen Informationsgehalt,
die *Entropie*.

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

$p(A) = 1$

Entropie $H = 0$

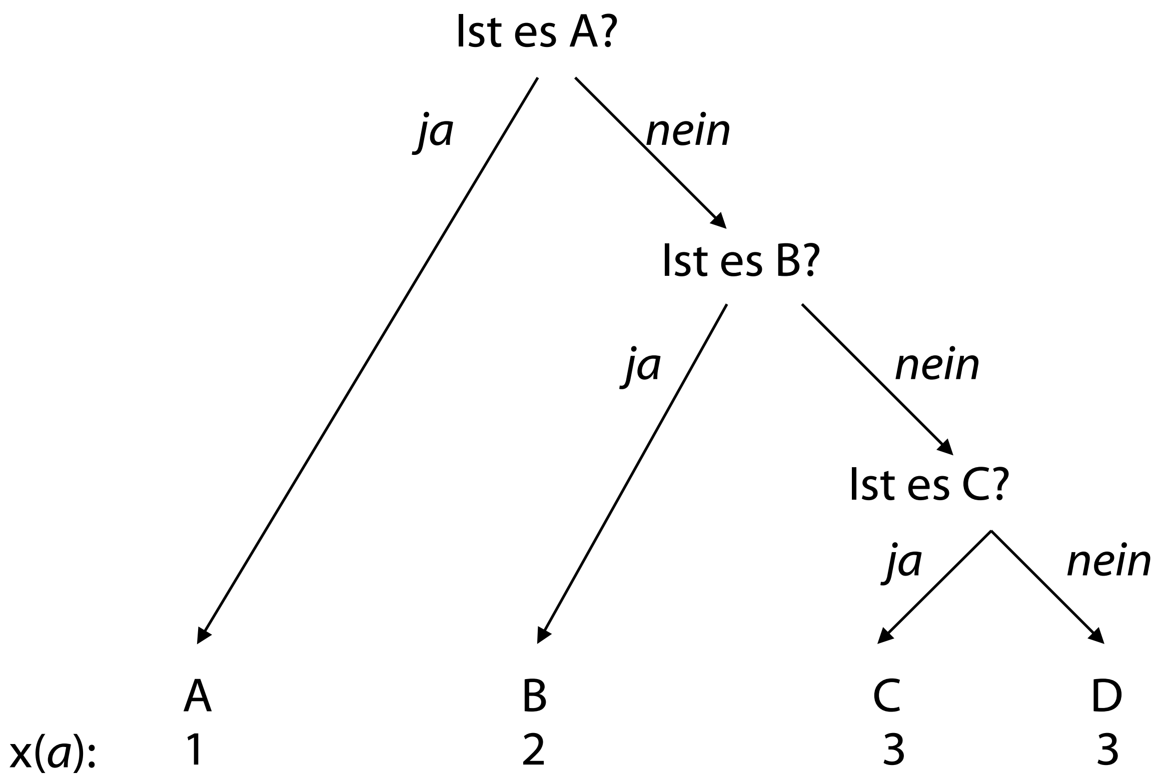
Entropie (2)

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

AAAAAAAAAAAAAA
AAAAAAAAAAAAAA
AAAAAAAAAAAAAA
AAAAAAAAAAAAAA
BBBBBBBBBBBBBB
BBBBBBBBBBBBBB
CCCCCCCCCCCCCC
DDDDDDDDDDDDDD

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

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



Einschub: Zweierlogarithmus

$a = b^x$

$x = \log_b a$

$\boxed{\log_b x = \log_y x / \log_y b}$

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

Beispiele:

$256 = 2^x$
$x = \log_2 256$
$x = 8$

$1.000.000 = 10^x$
$x = \log_{10} 1.000.000$
$x = 6$

$\log_2 256$ im Taschenrechner:

- „ 256 " eintippen
- „ *log* " drücken
- „ / " drücken
- „ 2 " eintippen
- „ *log* " drücken
- „ = " drücken

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

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

$$H = 1{,}75$$

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

Durchschnittlicher Entscheidungsgehalt eines Zeichen

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

Beispielcode:

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

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

Durchschnittlicher Wortlänge pro Zeichen

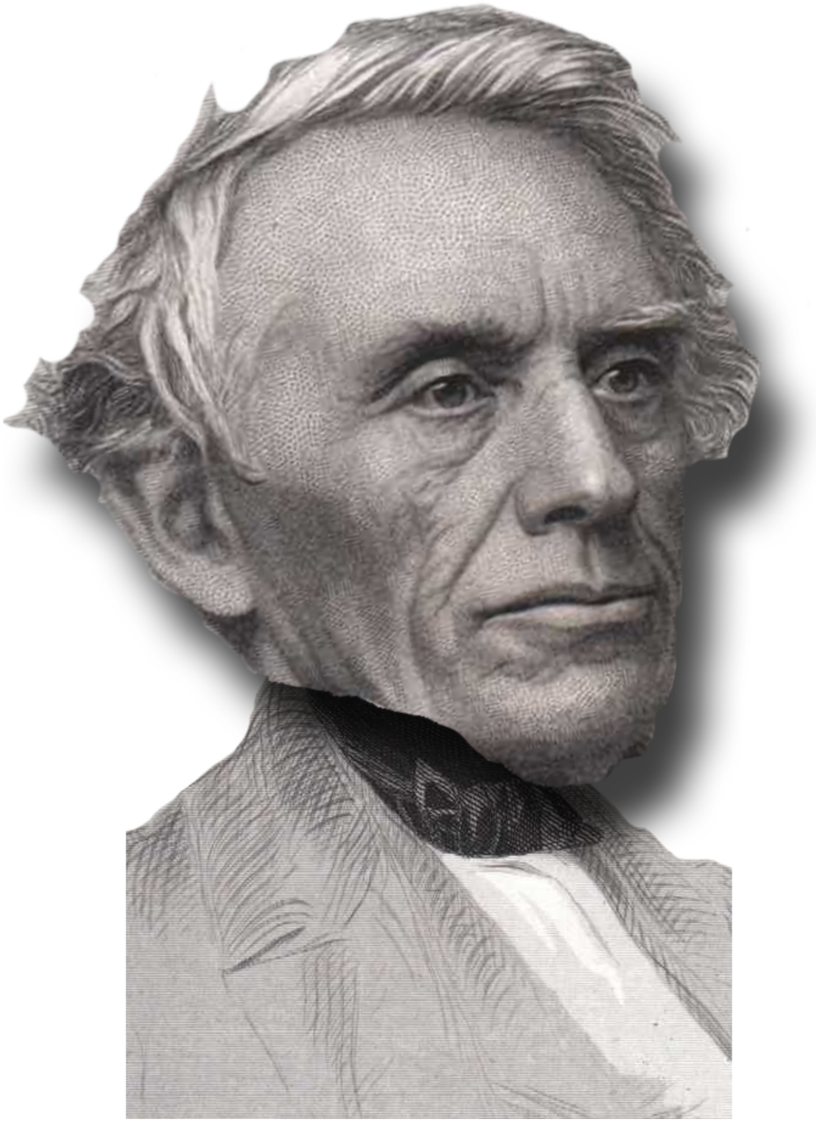
$R = L - H$

Redundanz eines Codes

Je kleiner, desto besser!

$$L = 2 \qquad R = 0{,}25$$

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

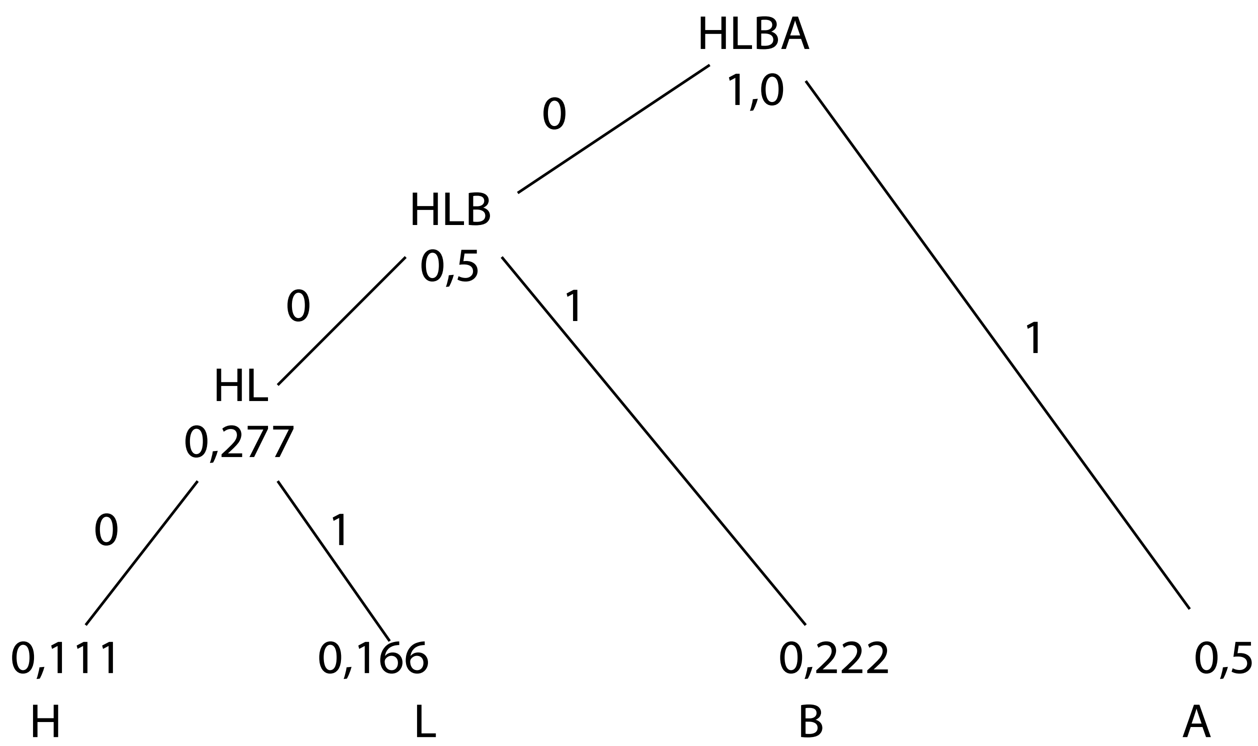
In optimalem Code müssen die beiden Symbole der niedrigsten Häufigkeit mit gleicher Länge codiert sein.

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:
1111110101010010001000 0

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!

1110
=
aac oder dc?

Ist der Code optimal?

	p	x
a	0,5	1
b	0,25	2
c	0,083	3,585
d	0,083	3,585
e	0,083	3,585

$H = 1{,}893$

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

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

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

$L = 1{,}913$

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

$R = L - H$

$R = 0{,}02$

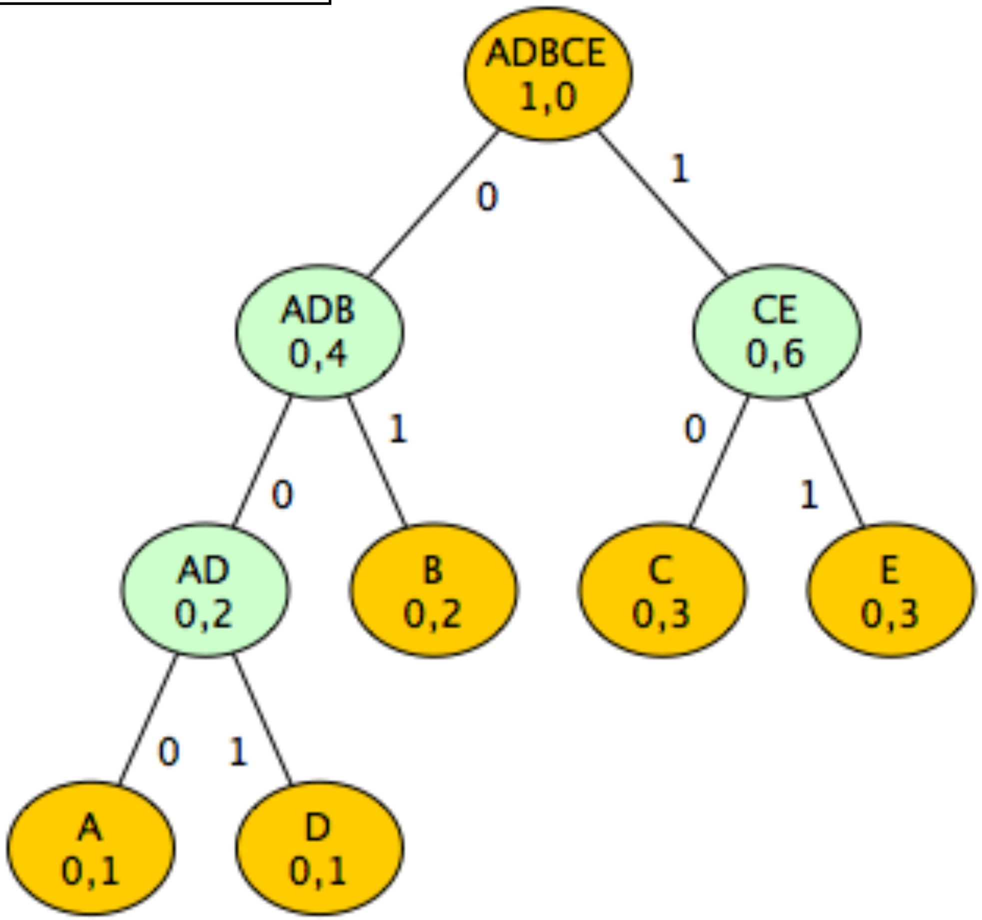
Generell: Huffman-Code optimal,
falls Häufigkeiten negative/Kehrwerte von
Zweierpotenzen sind, also 0,5 , 0,25 , 0,125

Anderes Beispiel

CECEDBCABE

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

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



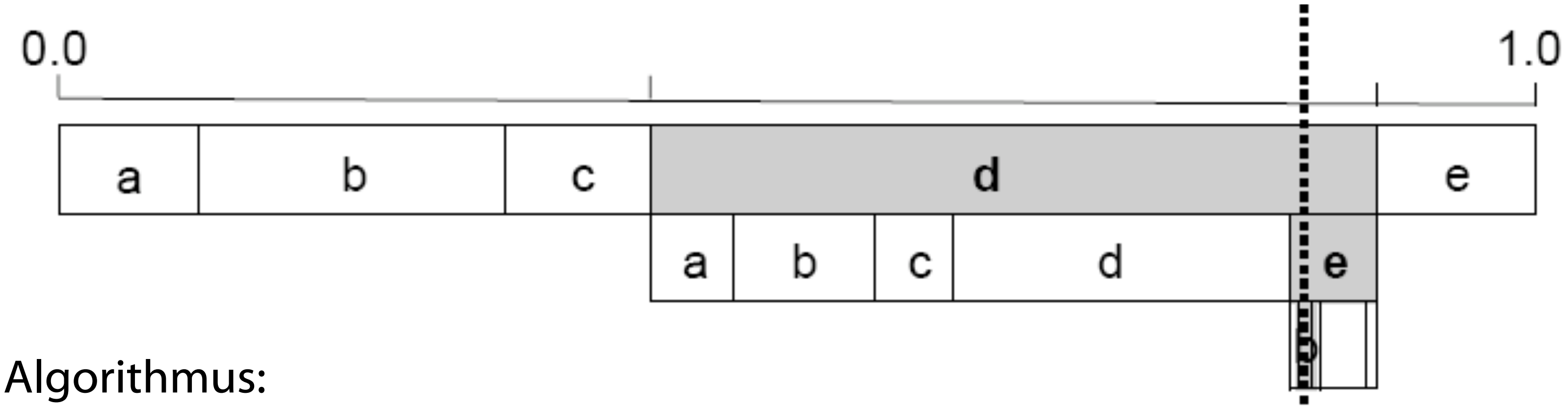
Code:

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*Ri; (Obergrenze)
L = L + B*Li; (Untergrenze)
Ende Wiederholung;

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

L_i und R_i sind Ränder eines Zeichens, definiert durch seine Auftrittswahrscheinlichkeit

Arithmetische Codierung (2)

ABCA

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]

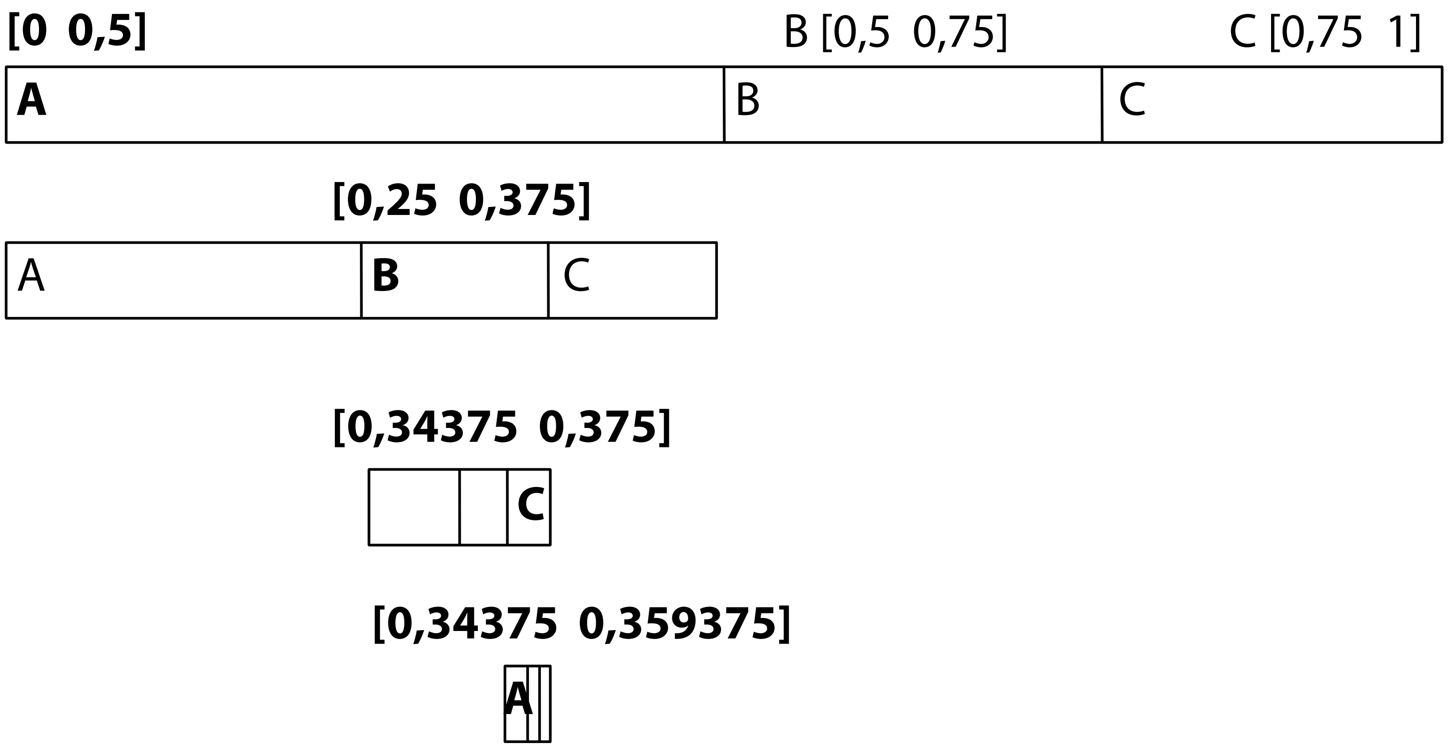
real L = 0.0; **real** R = 1.0;
Solange Zeichen vorhanden **wiederhole**
Lies Zeichen und bestimme Zeichenindex i;
real B = (R–L); (Intervallbreite)
R = L + B*Ri; (Obergrenze)
L = L + B*Li; (Untergrenze)
Ende Wiederholung;

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

Arithmetische Codierung (3)

ABCA

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



Arithmetische Codierung (4)

Ergebnisintervall: [0,34375;0,359375]

Untere Grenze in binär:
0,34375

0,34375 x 2 = **0**,6875
0,6875 x 2 = **1**,375
0,375 x 2 = **0**,75
0,75 x 2 = **1**,5
0,5 x 2 = **1**

0,01011

Obere Grenze in binär:
0,359375

0,359375 x 2 = **0**,71875
0,71875 x 2 = **1**,4375
0,4375 x 2 = **0**,875
0,875 x 2 = **1**,75
0,75 x 2 = **1**,5
0,5 x 2 = **1**

0,010111

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

Code: 010111