Übung zur Vorlesung

Digitale Medien

Doris Hausen

Ludwig-Maximilians-Universität München

Wintersemester 2010/2011

Übungsbetrieb

- Informationen zu den Übungen:
  http://www.medien.ifi.lmu.de/dm
  http://www.die-informatiker.net

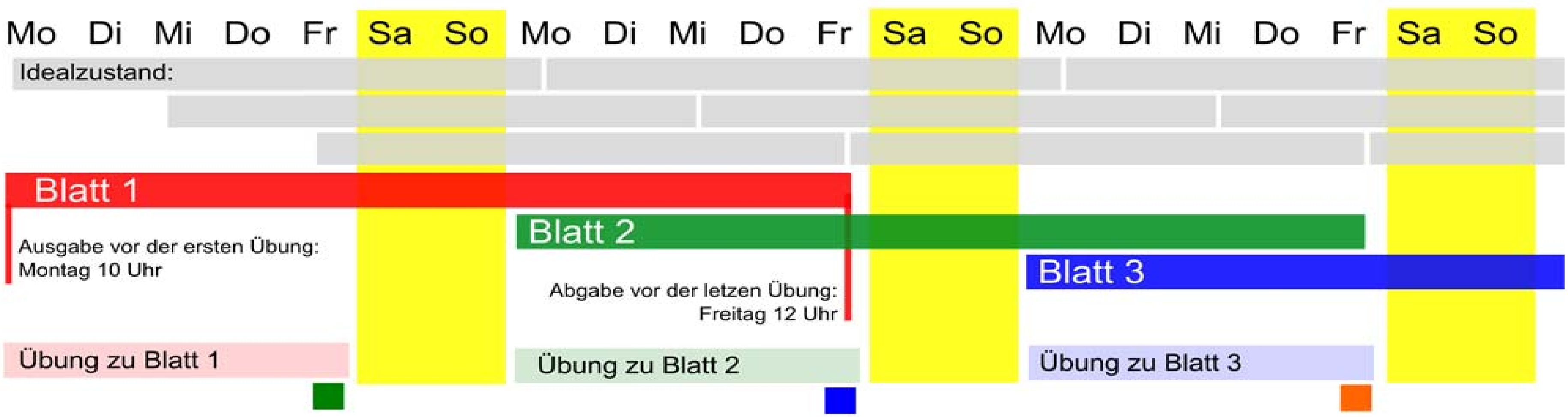
- Praktische Anwendungen des theoretischen Vorlesungsstoffs
- Wichtige Voraussetzung für die Klausur
- Bonuspunkte für die Klausur
  (maximal 15% der Klausurpunkte)

Anmeldung für den Übungsbetrieb

- Anmeldung ist Voraussetzung für Übungsteilnahme
- UniWorx: https://www.pst.ifi.lmu.de/uniworx
- Benötigt Rechnerkennung:
  http://www.rz.ifi.lmu.de/Merkblaetter/RechnerAnmeldung_WS.html
- Anmeldung für die Übungsgruppen:
  ab Freitag, 29.10. 16 Uhr
- Hinweis: Mails von UniWorx werden an
  kennung@cip.ifi.lmu.de versendet
  (regelmäßig Adresse checken oder weiterleiten!)

Übungsblätter

- Ausgabe:
  - jeden Montag ab 10 Uhr
  - auf http://www.medien.ifi.lmu.de/dm/
- Abgabe:
  - jeweils spätestens übernächster Freitag 12 Uhr
  - über https://www.pst.informatik.uni-muenchen.de/uniworx/
  - Achtung: erneutes Hochladen überschreibt die alte Lösung



Benotung

- Klausur am Ende des Semesters
  - Nachholklausur am Ende der Semesterferien (auch zur Notenverbesserung)

- Abgabe der Übungsblätter:
  - freiwillig
  - ABER: dringend empfohlen

- Anreiz:
  Bonuspunkte aus dem Übungsbetrieb (max. 15%)

Plagiate

- Abschreiben
  - von Kommilitonen
  - von anderen Quellen (z.B. Wikipedia)

  Ist in **keiner Weise** erlaubt!
- Wenn Sie erwischt werden, wird das ganze Übungsblatt oder auch alle Übungsblätter mit 0 Punkten bewertet
- Im Zweifelsfall immer vor der Abgabe nachfragen.
- Wir prüfen nach – eventuell auch erst nach Ende der Übungen.

- Erlaubt ist:
  - Zusammen lernen
  - Links zu guten Quellen austauschen
  - Gemeinsames Code Review
  - Tricks verraten

Ansprechpartner

- Gegenseitiger Austausch – auch über das Forum – ist sehr gewünscht.

- Individuelle Probleme (z.B. mit der Korrektur) lassen sich aber meist deutlich besser persönlich klären. Daher bitte direkt den betroffenen Tutor oder die Übungsleitung ansprechen.

1. Übungsblatt

Zahlensysteme
Entropie
Huffman-Codierung

Konvertierung zwischen ganzen Binär- und Dezimalzahlen

*Ganze **Binär**zahlen nach **Dezimal***	*Ganze **Dezimal**zahlen nach **Binär***
Arbeite die Ziffern der Zahl von rechts nach links durch. Falls eine Ziffer an der Position *z* gleich 1 ist (Achtung: Die rechteste Position ist 0 und nicht 1!), rechne 2 hoch *z* und addiere die Lösung zum Gesamtergebnis.	Teile die Zahl durch 2. Der verbleibende Rest ist die nächste Ziffer (fülle von rechts nach links auf!). Höre auf, sobald das Ergebnis 0 wird.

Beispiel:
1001011 nach Dezimal

$$1 \times 2^0 + 1 \times 2^1 + 1 \times 2^3 + 1 \times 2^6$$
$$= 1 + 2 + 8 + 64$$
$$= \mathbf{75}$$

Beispiel:
75 nach Binär

75 / 2 = 37 Rest: **1**
37 / 2 = 18 Rest: **1**
18 / 2 = 9 Rest: **0**
9 / 2 = 4 Rest: **1**
4 / 2 = 2 Rest: **0**
2 / 2 = 1 Rest: **0**
1 / 2 = 0 Rest: **1**

Mehr Infos: http://www.arndt-bruenner.de/mathe/scripts/Zahlensysteme.htm

Konvertierung zwischen Binär- und Dezimalzahlen zwischen 0 und 1

***Binär** nach **Dezimal** (Komma)*

Arbeite die Ziffern der Zahl hinter dem Komma von **links nach rechts** durch. Falls eine Ziffer an der Position *z* gleich 1 ist (Achtung: Die linkeste Position ist diesmal 1!), rechne 2 hoch -*z* und addiere die Lösung zum Gesamtergebnis.

***Dezimal** nach **Binär** (Komma)*

Multipliziere die Zahl mit 2.
Die Zahl vor dem Komma ist die nächste Zahl des Ergebnisses.
Entferne die Zahl vor dem Komma.
Wiederhole das Verfahren, bis nichts mehr rechts vom Komma steht oder sich die Ergebnisse wiederholen.

Beispiel:
0,0101 nach Dezimal

$1 \times 2^{-2} + 1 \times 2^{-4}$
$= 0{,}25 + 0{,}0625$
$= \mathbf{0{,}3125}$

Hinweis:
$2^{-x} = 1 / 2^{x}$

Beispiel:
0,3125 nach Binär

$0{,}3125 \times 2 = \mathbf{0}{,}625$
$0{,}625 \times 2 = \mathbf{1}{,}25$
$0{,}25 \times 2 = \mathbf{0}{,}5$
$0{,}5 \times 2 = \mathbf{1}$

=> **0,0101**

Mehr Infos: http://www.arndt-bruenner.de/mathe/scripts/Zahlensysteme.htm

ASCII Code 1 Byte / Zeichen

Nonsense

Dezimal: 78 111 110 115 101 110 115 101
N o n s e n s e

Binär:

01001110	N
01101111	o
01101110	n
01110011	s
01100101	e
01101110	n
01110011	s
01100101	e

8 Byte

Lauflängencodierung (RLE)

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 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

Jede Nachricht hat einen Informationsgehalt,
die *Entropie*.

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

AAAAAAAAAAAAAAA
AAAAAAAAAAAAAAA
AAAAAAAAAAAAAAA
AAAAAAAAAAAAAAA
AAAAAAAAAAAAAAA
AAAAAAAAAAAAAAA
AAAAAAAAAAAAAAA

$p(A) = 1$

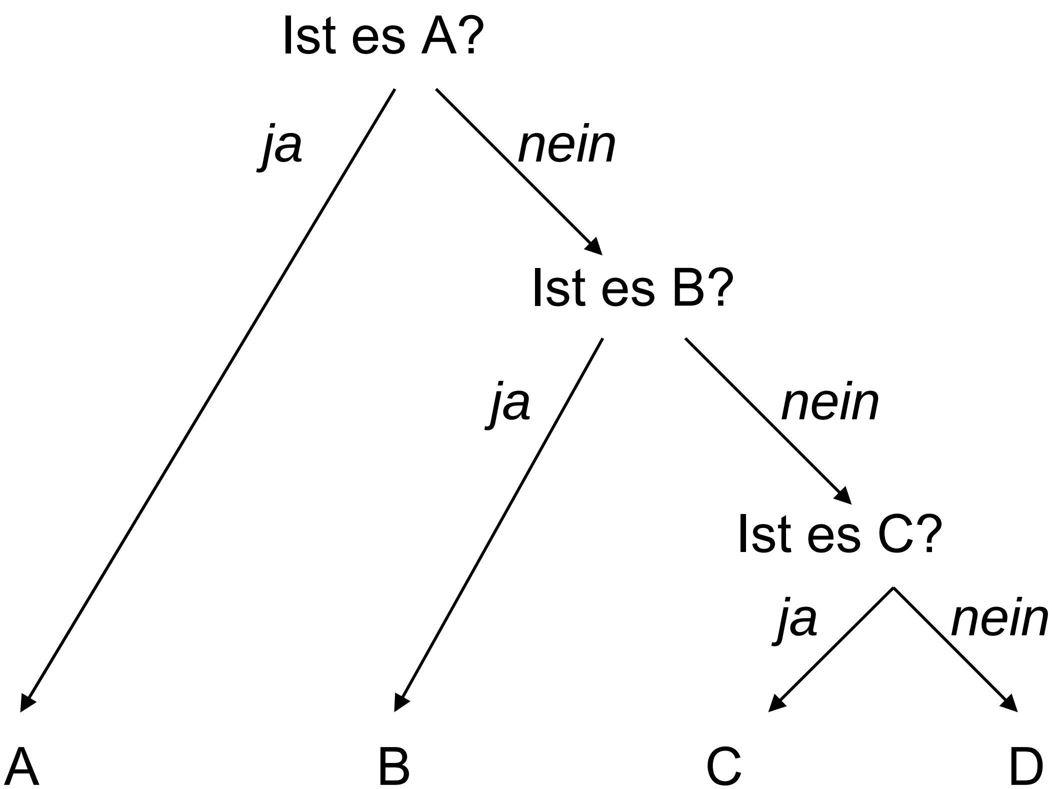
Entropie $H = 0$

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

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

AAAAAAAAAAAAAAA
AAAAAAAAAAAAAAA
AAAAAAAAAAAAAAA
AAAAAAAAAAAAAAA
BBBBBBBBBBBBBBB
BBBBBBBBBBBBBBB
CCCCCCCCCCCCCCC
DDDDDDDDDDDDDDD

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



x(a):	1	2	3	3

Berechnung des Zweier-Logarithmus ld aus dem natürlichen Logarithmus ln

$$a = b^x$$

$$x = \log_b a$$

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

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

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

	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

$$L = 2$$

$$R = 0{,}25$$

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

Durchschnittlicher Entscheidungsgehalt eines Zeichen

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

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

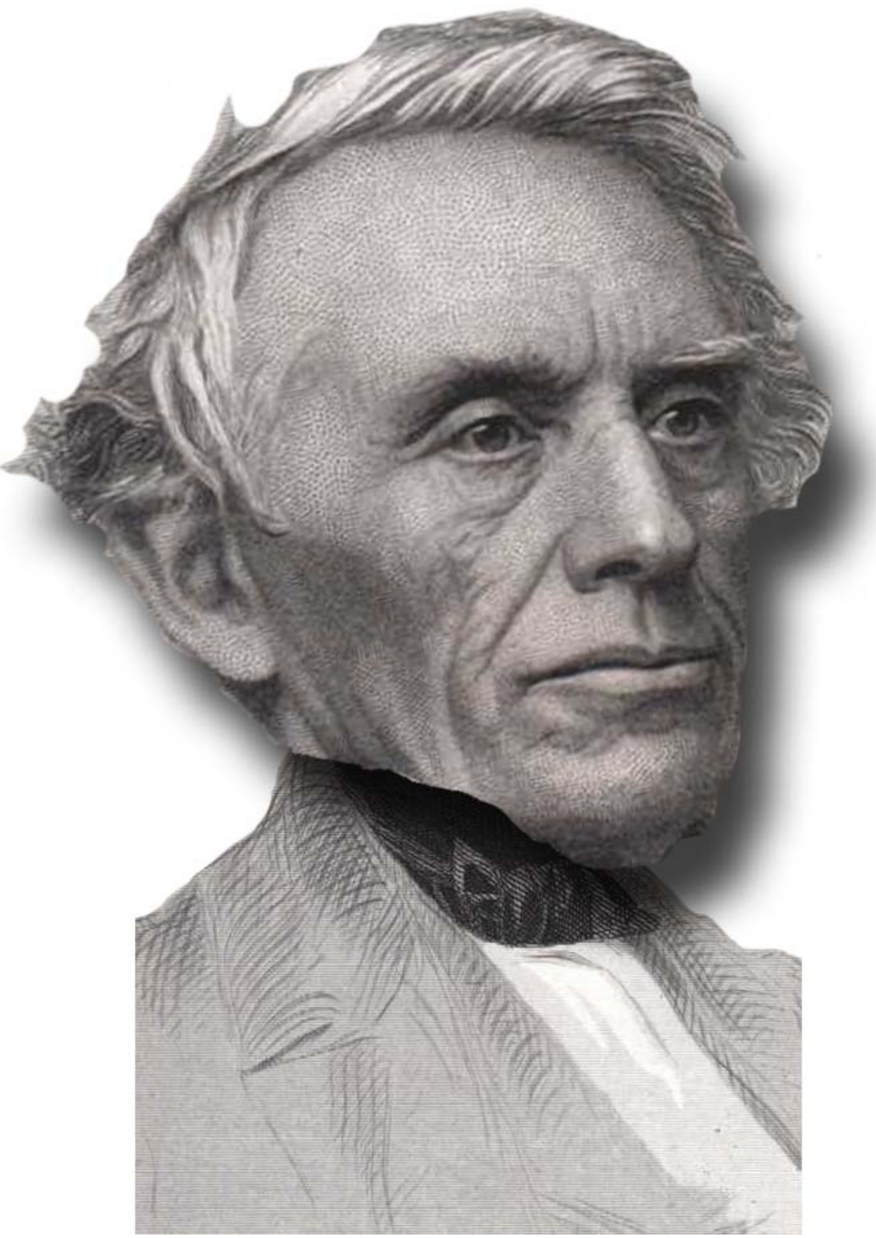
Durchschnittlicher Wortlänge pro Zeichen

$$\boxed{R = L - H}$$

Redundanz eines Codes

Je kleiner, desto besser!

Samuel Morse



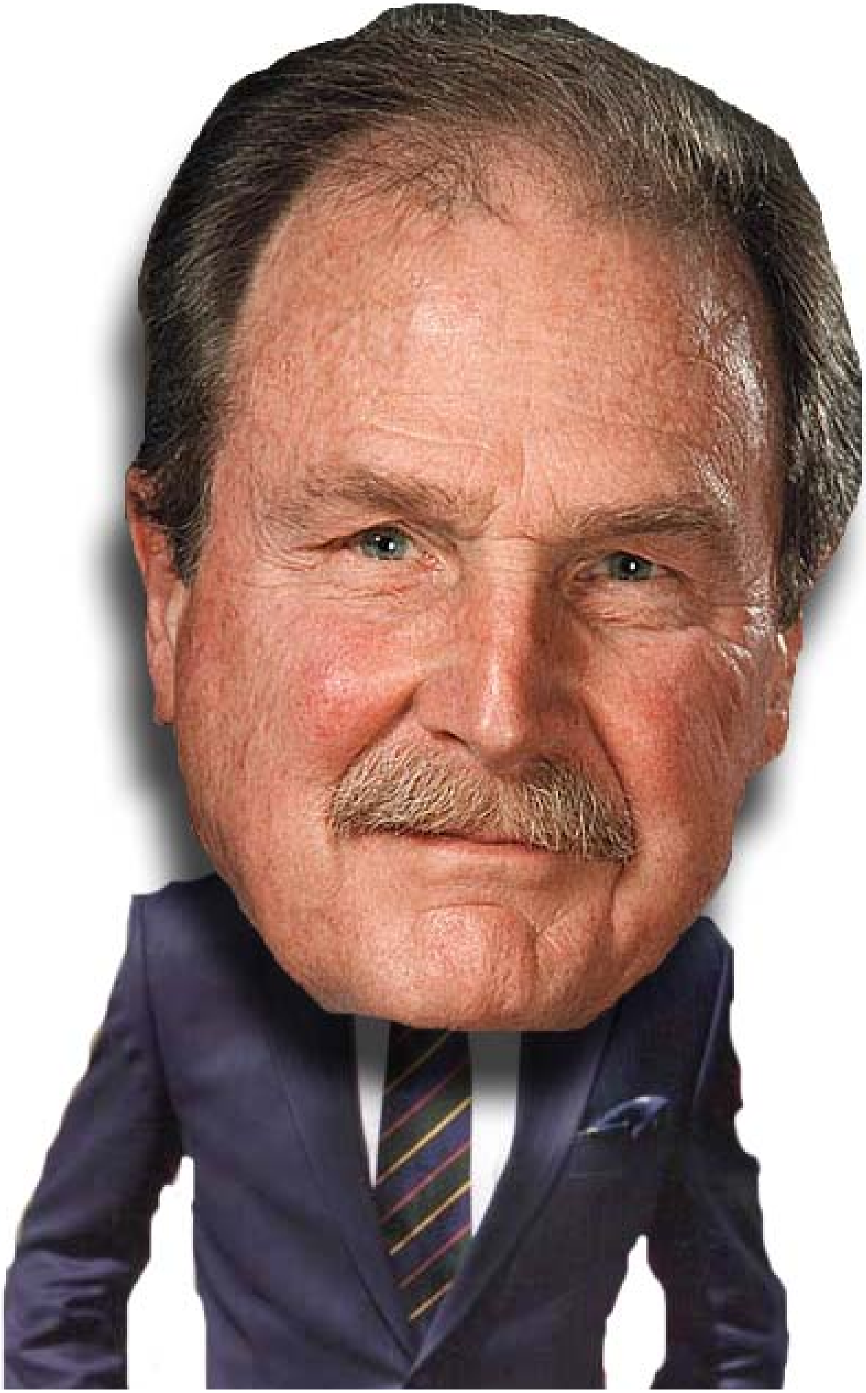
.... .. / .. .----. -- / ...
.- -- ..- . .-.. / -- --- .-.
... . .-.-.- / .. / -.. --- /
-.-. --- -.. . ... .-.-.-

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 A. Huffman:

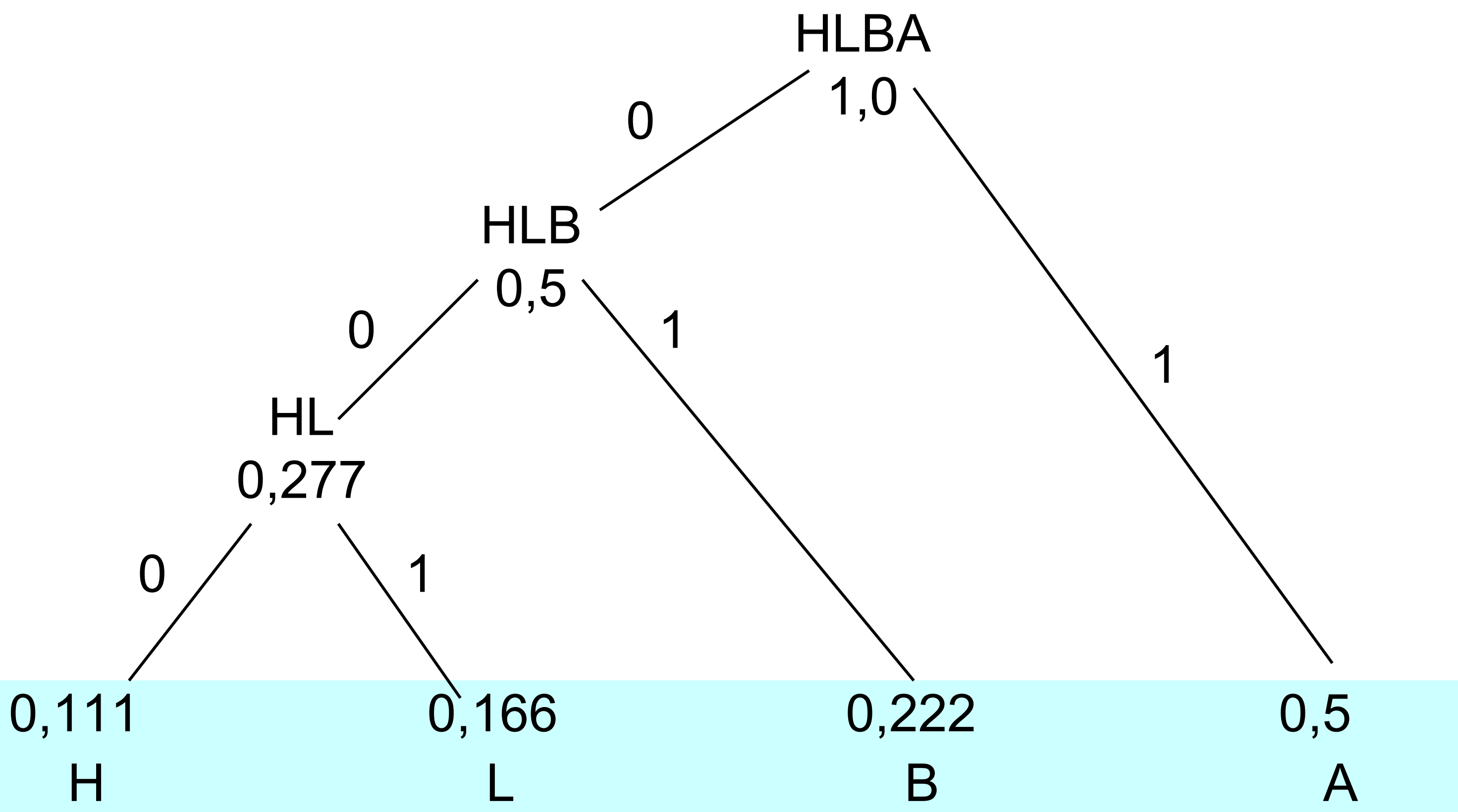


In optimalem Code müssen die beiden Symbole der niedrigsten Häufigkeit mit gleicher Länge codiert sein.

AAAAHHABBLLABBLAAA

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

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



A	1
B	01
L	001
H	000

aaaaaabbbcde

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

a	1
b	01
c	001
d	0001
e	0000

Ergebnis:
11111101010100100010000

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

$$H = 1{,}896$$

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

$$L = 1{,}917$$

$$R = 0{,}021$$

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

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

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

$$R = L - H$$

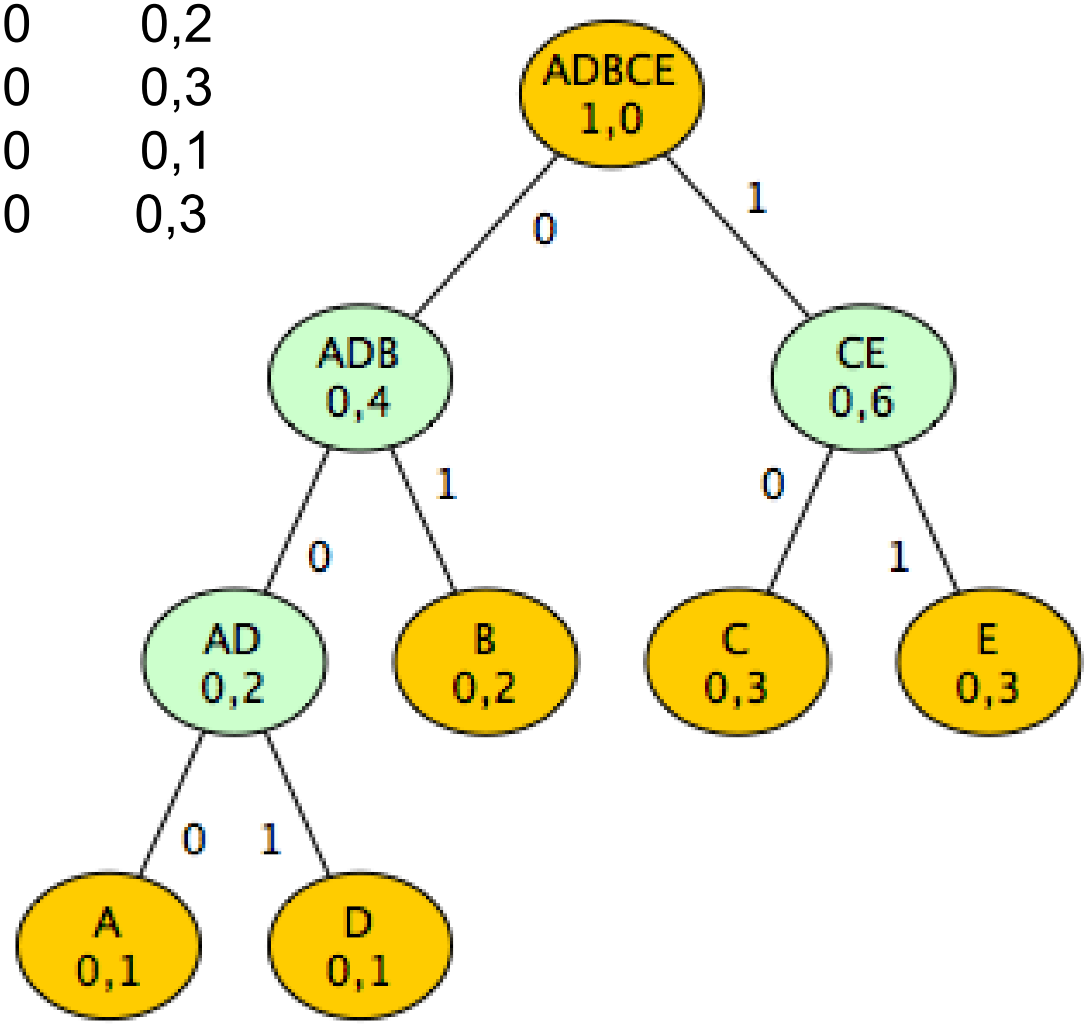
Generell: Huffman-Code optimal,
falls Häufigkeiten negative/Kehrwerte von Zweierpotenzen sind, also 0,5 , 0,25 , 0,125

Anderes Beispiel:

CECEDBCABE

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

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



Code:

A	000
B	01
C	10
D	001
E	11