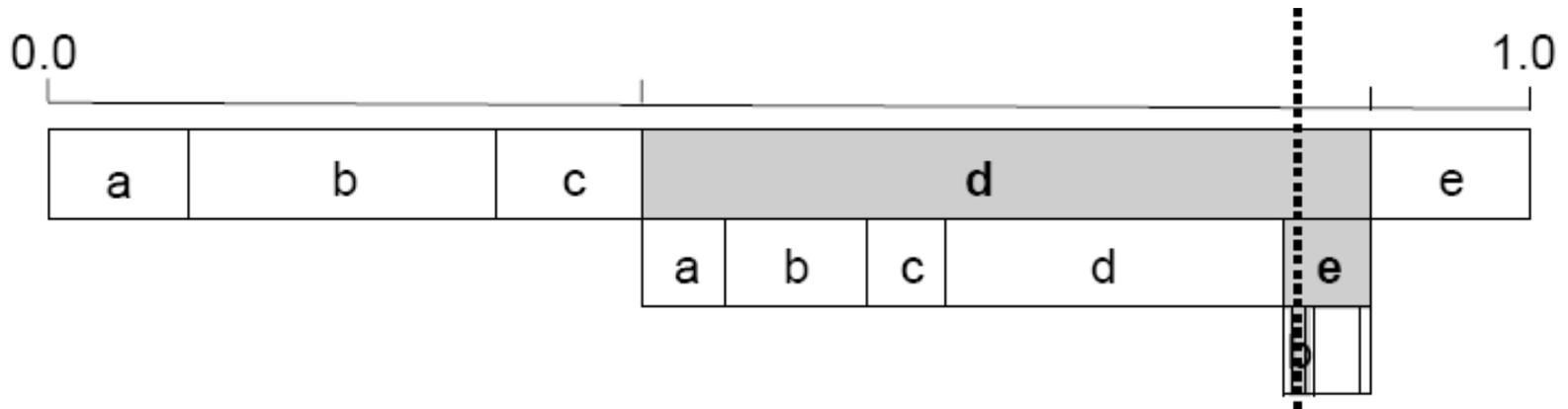


Digitale Medien

Übung

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);
  R = L + B*Ri;
  L = L + B*Li;
Ende Wiederholung;

```

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

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

Ausführliche Einleitung mit Java-Applet:

[Seite von Oliver Schmid](#)


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

CABA

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

Solange Zeichen vorhanden **wiederhole**

Lies Zeichen und bestimme Zeichenindex i;

real B = (R-L);

R = L + B*R_i;

L = L + B*L_i;

Ende Wiederholung;

CABA

A 2/4 = 0,5

B 1/4 = 0,25

C 1/4 = 0,25

A L₀ = 0

B L₁ = 0,5

C L₂ = 0,75

R₀ = 0,5

R₁ = 0,75

R₂ = 1,0

	i	B	L	R
	-	-	0,0	1,0
C	2	1,0	0,75	1,0
A	0	0,25	0,75	0,875
B	1	0,125	0,8125	0,84375
A	0	0,03125	0,8125	0,828125

Wandle untere Grenze nach binär um

Wandle obere Grenze nach binär um

Suche die erste (Binär-)Stelle in der sie sich unterscheiden

Code ist obere Grenze ohne '0,' bis zu einschließlich der Unterscheidungsstelle

Wandle untere Grenze nach binär um

Wandle obere Grenze nach binär um

Suche die erste (Binär-)Stelle in der sie sich unterscheiden

Code ist obere Grenze ohne '0,' bis zu einschließlich der Unterscheidungsstelle

Untere Grenze in binär:

$$0,8125 \times 2 = 1,625$$

$$0,625 \times 2 = 1,25$$

$$0,25 \times 2 = 0,5$$

$$0,5 \times 2 = 1,0$$

Obere Grenze in binär:

$$0,828125 \times 2 = 1,65625$$

$$0,65625 \times 2 = 1,3125$$

$$0,3125 \times 2 = 0,625$$

$$0,625 \times 2 = 1,25$$

$$0,25 \times 2 = 0,5$$

$$0,5 \times 2 = 1,0$$

$$0,8125 = 0,1101$$

$$0,828125 = 0,110101$$



Code: 11010

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

Solange Zeichen vorhanden **wiederhole**

Lies Zeichen und bestimme Zeichenindex i;

real B = (R-L);

R = L + B*Ri;

L = L + B*Li;

Ende Wiederholung;

AABBCCADA

Aufgabe: Codierung bis hier!

A	0,5	$L_0 = 0,0$	$R_0 = 0,5$
B	0,25	$L_1 = 0,5$	$R_1 = 0,75$
C	0,125	$L_2 = 0,75$	$R_2 = 0,875$
D	0,125	$L_3 = 0,875$	$R_3 = 1,0$

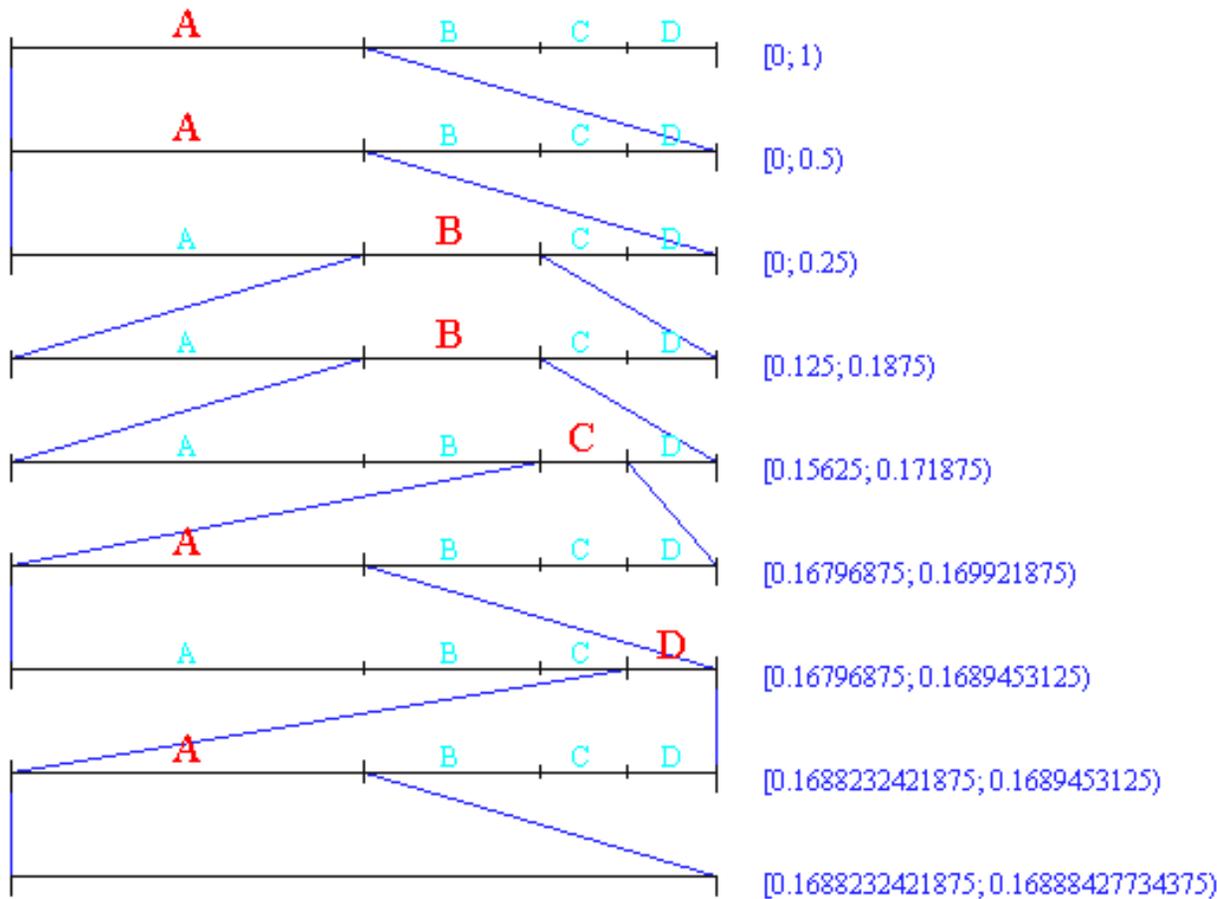
AABBCCADA

	i	B	L	R
	-	-	0,0	1,0
A	0	1,0	0,0	0,5
A	0	0,5	0,0	0,25
B	1	0,25	0,125	0,1875
B	1	0,0625	0,15625	0,171875
C	2	0,015625	0,16796875	0,169921875
...

AABBCADA

AABBCADA

Encoding interval



AABBCCADA

Arithmetische Codierung

untere Grenze:
0,1688232421875

obere Grenze:
0,16888427734375

AABBCCADA

untere Grenze:

0,1688232421875

$$0.1688232421875 \times 2 = 0.337646484375$$

$$0.337646484375 \times 2 = 0.67529296875$$

$$0.67529296875 \times 2 = 1.3505859375$$

$$0.3505859375 \times 2 = 0.701171875$$

$$0.701171875 \times 2 = 1.40234375$$

$$0.40234375 \times 2 = 0.8046875$$

$$0.8046875 \times 2 = 1.609375$$

$$0.609375 \times 2 = 1.21875$$

$$0.21875 \times 2 = 0.4375$$

$$0.4375 \times 2 = 0.875$$

$$0.875 \times 2 = 1.75$$

$$0.75 \times 2 = 1.5$$

$$0.5 \times 2 = 1.0$$

obere Grenze:

0,16888427734375

$$0.16888427734375 \times 2 = 0.3377685546875$$

$$0.3377685546875 \times 2 = 0.675537109375$$

$$0.675537109375 \times 2 = 1.35107421875$$

$$0.35107421875 \times 2 = 0.7021484375$$

$$0.7021484375 \times 2 = 1.404296875$$

$$0.404296875 \times 2 = 0.80859375$$

$$0.80859375 \times 2 = 1.6171875$$

$$0.6171875 \times 2 = 1.234375$$

$$0.234375 \times 2 = 0.46875$$

$$0.46875 \times 2 = 0.9375$$

$$0.9375 \times 2 = 1.875$$

$$0.875 \times 2 = 1.75$$

$$0.75 \times 2 = 1.5$$

$$0.5 \times 2 = 1.0$$

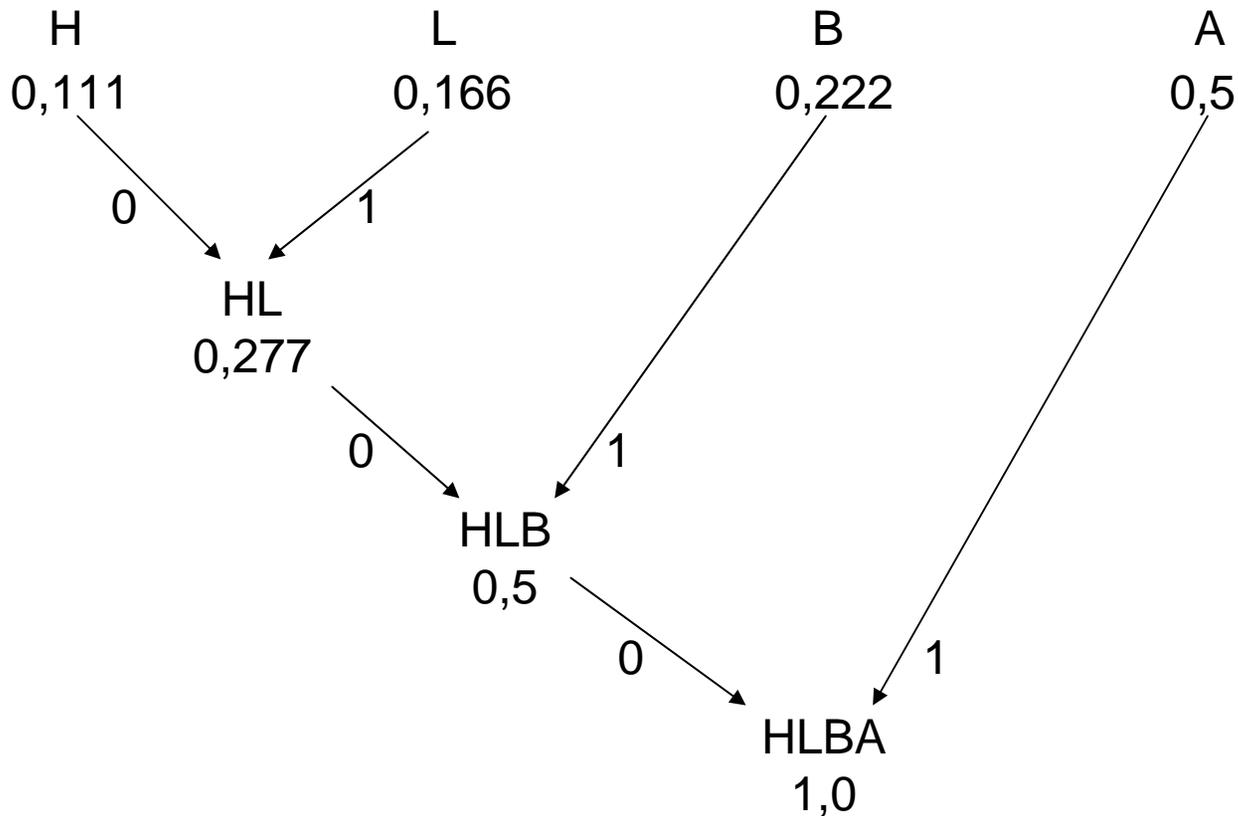
unten: 0,0010101100111

oben: 0,**00101011001111**

AAAAHHABBLLABBLAAA

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

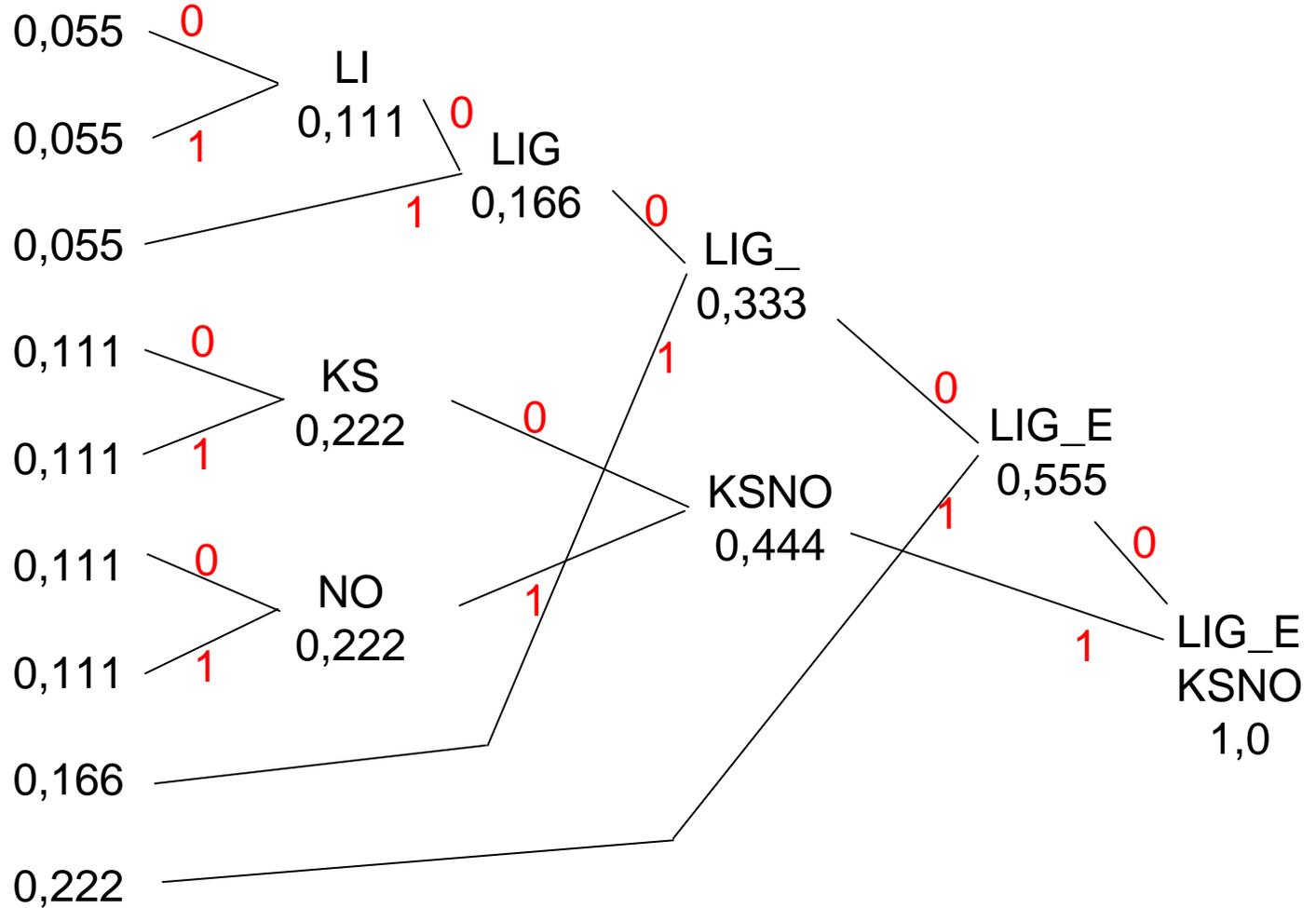
1. Ermittlung der Häufigkeiten
2. Aufbau des Codebaums
3. Code



A	1
B	01
L	001
H	000

NO_ONE_LIKES_GEEKS

00000	L
00001	I
0001	G
100	K
101	S
110	N
111	O
001	-
01	E



		p	x = $\text{ld}(1/p)$	p*x	p* c
00000	L	0,055	4,170	0,229	0,275
00001	I	0,055	4,170	0,229	0,275
0001	G	0,055	4,170	0,229	0,220
100	K	0,111	3,170	0,352	0,333
101	S	0,111	3,170	0,352	0,333
110	N	0,111	3,170	0,352	0,333
111	O	0,111	3,170	0,352	0,333
001	_	0,166	2,585	0,429	0,498
01	E	0,222	2,170	0,481	0,444

$H = 3,004$ $L = 3,044$ $R = 0,04$

Ausgegangen wird von einem Grundwörterbuch (z.B. ASCII-Code)

Algorithmus:

SeqChar p = < Nächstes Eingabezeichen >;

Char k = Nächstes Eingabezeichen;

Wiederhole:

Falls p & < k > in Tabelle enthalten

dann p = p & < k >

sonst trage p & <k> neu in Tabelle ein

 (und erzeuge neuen Index dafür);

 Schreibe Tabellenindex von p auf Ausgabe;

 p = < k >;

Ende Fallunterscheidung;

 k = Nächstes Eingabezeichen;

solange bis Eingabeende

Schreibe Tabellenindex von p auf Ausgabe;

a	97	h	104	o	111	v	118
b	98	i	105	p	112	w	119
c	99	j	106	q	113	x	120
d	100	k	107	r	114	y	121
e	101	l	108	s	115	z	122
f	102	m	109	t	116		
g	103	n	110	u	117		

Lesen (k)	Codetabelle schreiben (p & <k>)	Ausgabe	Puffer füllen (p)

Algorithmus zur Dekomprimierung:

```

SeqChar p := <>;
int k = NächsteEingabezahl;
Schreibe Zeichenreihe mit Tabellenindex k auf Ausgabe;
int old = k;
Wiederhole solange Eingabe nicht leer:
    k = NächsteEingabezahl;
    SeqChar akt = Zeichenreihe mit Tabellenindex k;
    p = Zeichenreihe mit Tabellenindex old (letztes Teilwort);
    Falls Index k in Tabelle enthalten
        dann Char q = erstes Zeichen von akt;
            Schreibe Zeichenreihe akt auf Ausgabe;
        sonst Char q = erstes Zeichen von p;
            Schreibe Zeichenreihe p & <q> auf Ausgabe;
    Ende Fallunterscheidung;
    Trage p & <q> in Tabelle ein
    (und erzeuge neuen Index dafür);
    old = k;
Ende Wiederholung;

```

a	97	h	104	o	111	v	118
b	98	i	105	p	112	w	119
c	99	j	106	q	113	x	120
d	100	k	107	r	114	y	121
e	101	l	108	s	115	z	122
f	102	m	109	t	116		
g	103	n	110	u	117		

Lesen (k)	Ausgabe (akt) (q ist unterstrichen)	Puffer füllen (p)	Codetabelle (p & <q>)	Merken (old)