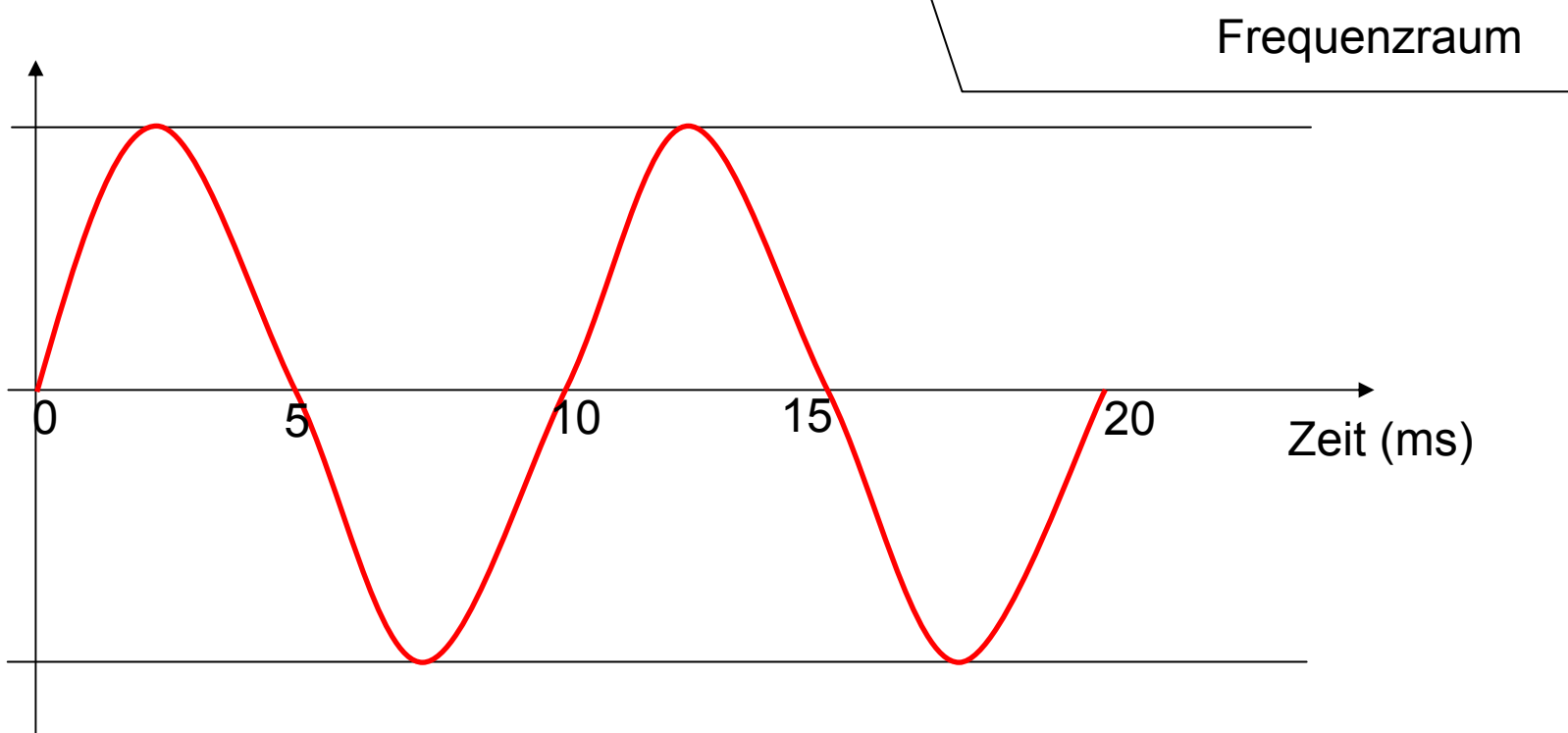


Digitale Medien

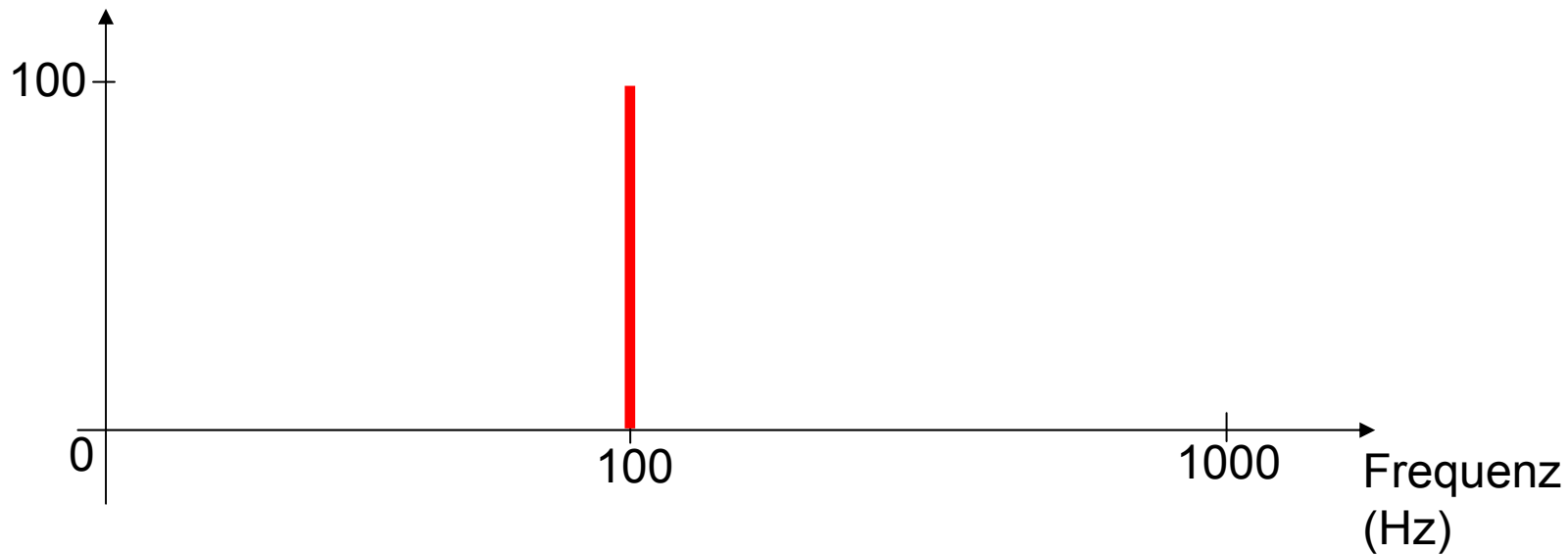
Übung

Heute

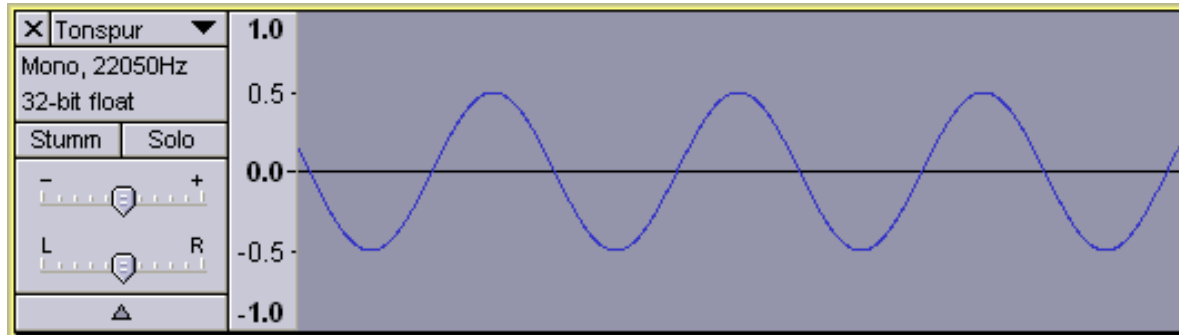
- Arbeiten mit Audio Files
- Wiederholung Codierung



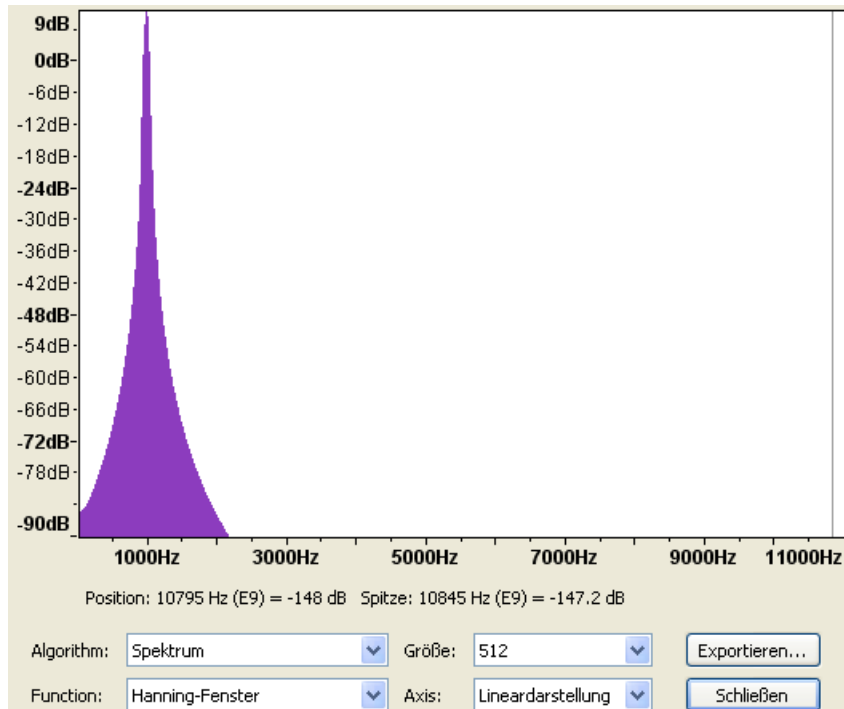
Anteil in %



Wellendarstellung



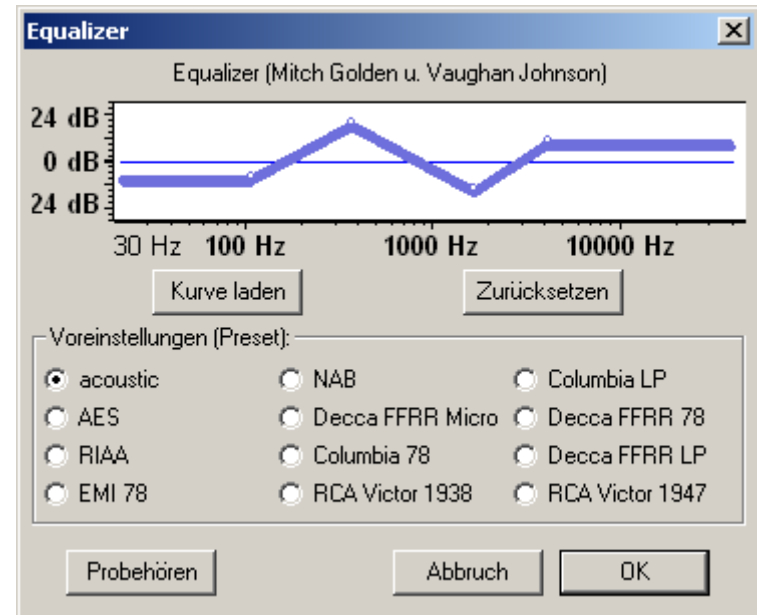
Darstellung im Frequenzraum



Vielzahl von Filtern/Effekten vorhanden

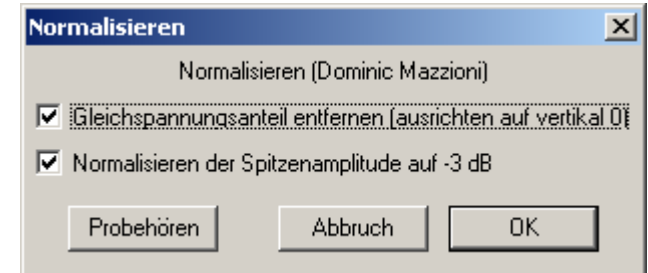
Equalizer

Erlaubt einzelne Bereiche des Frequenzspektrums gezielt lauter oder leiser zu machen.

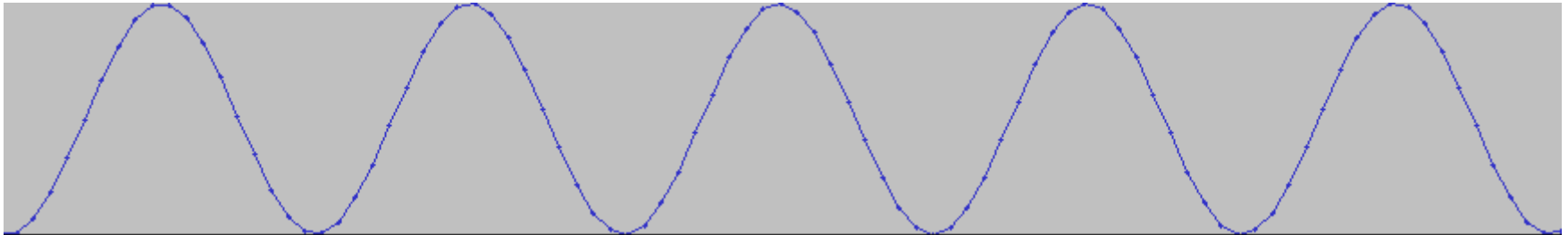
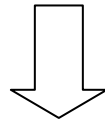
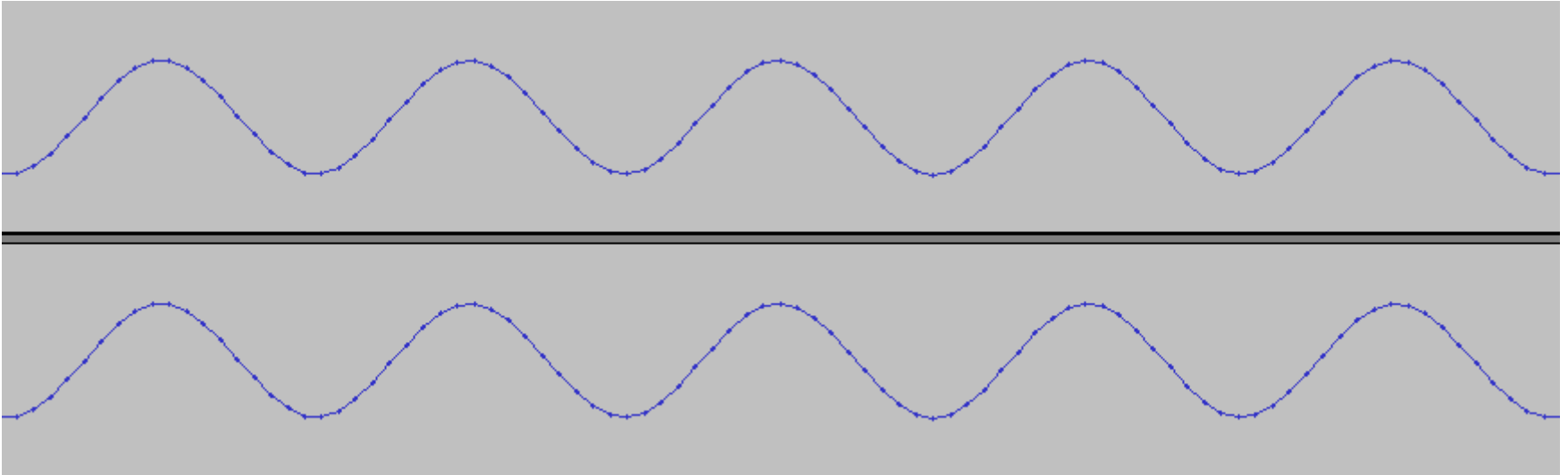


Normalisieren

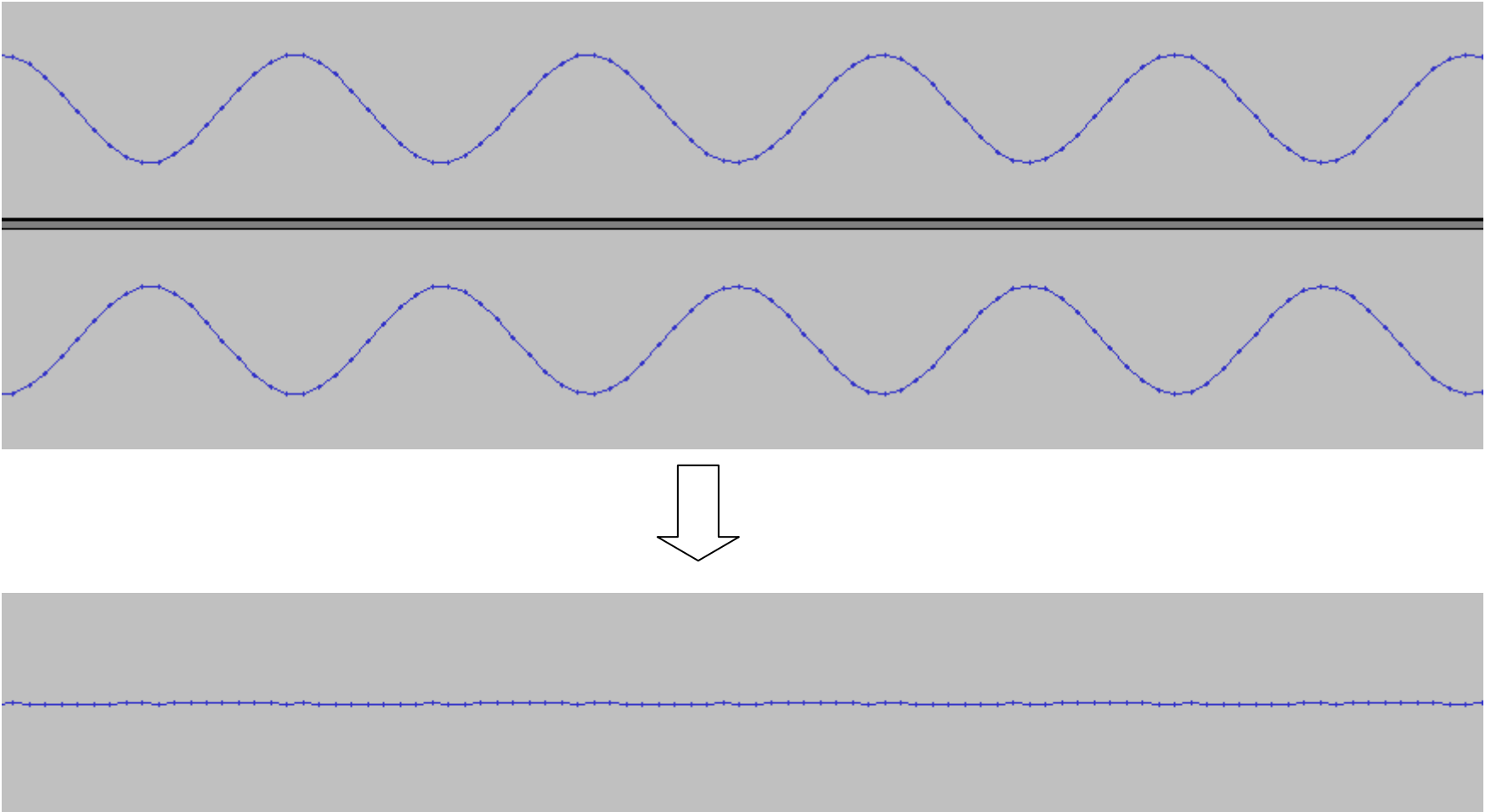
Bringt ein Tonsignal auf eine einheitliche Lautstärke.

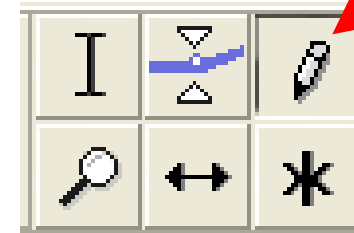


Addition von ähnlichen (kohärenten) Signalen führt zu einer Verstärkung der Amplitude.

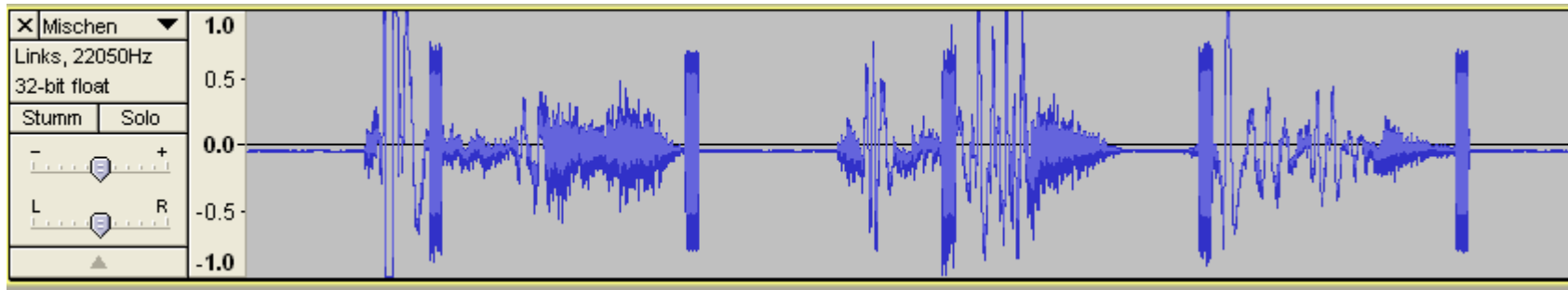


Addition von zeitlich verschobenen Tonsignalen führt zu einer Reduzierung der Amplitude.

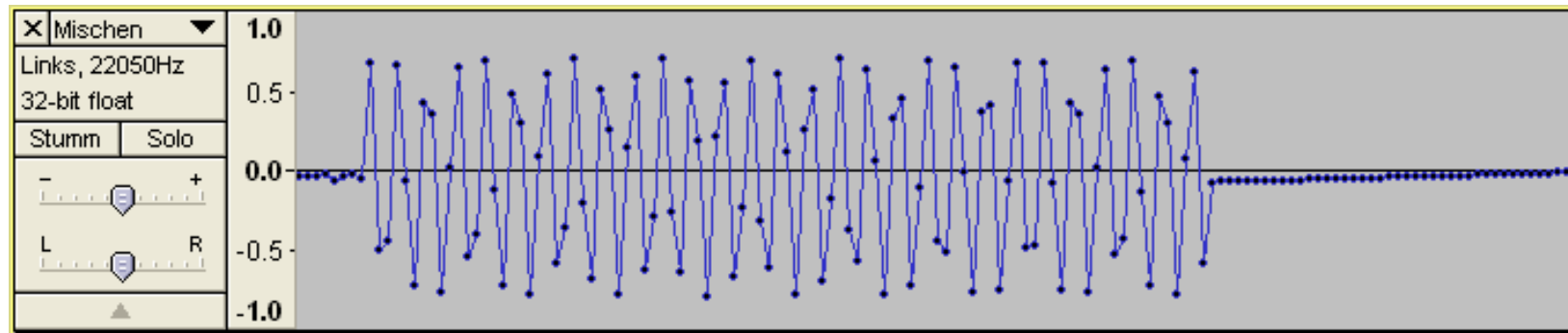




Tonspur mit Störung:



Nahansicht der Störung:



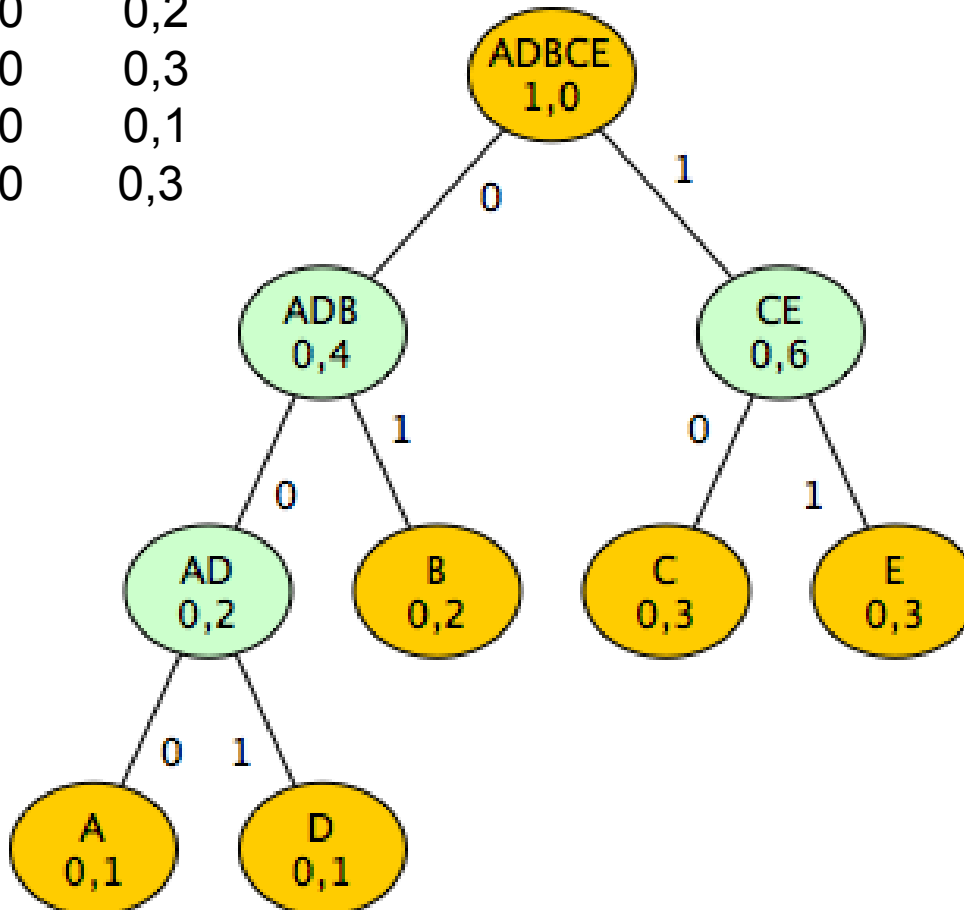
Wiederholung Codierung

- LZW
- Huffman

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:

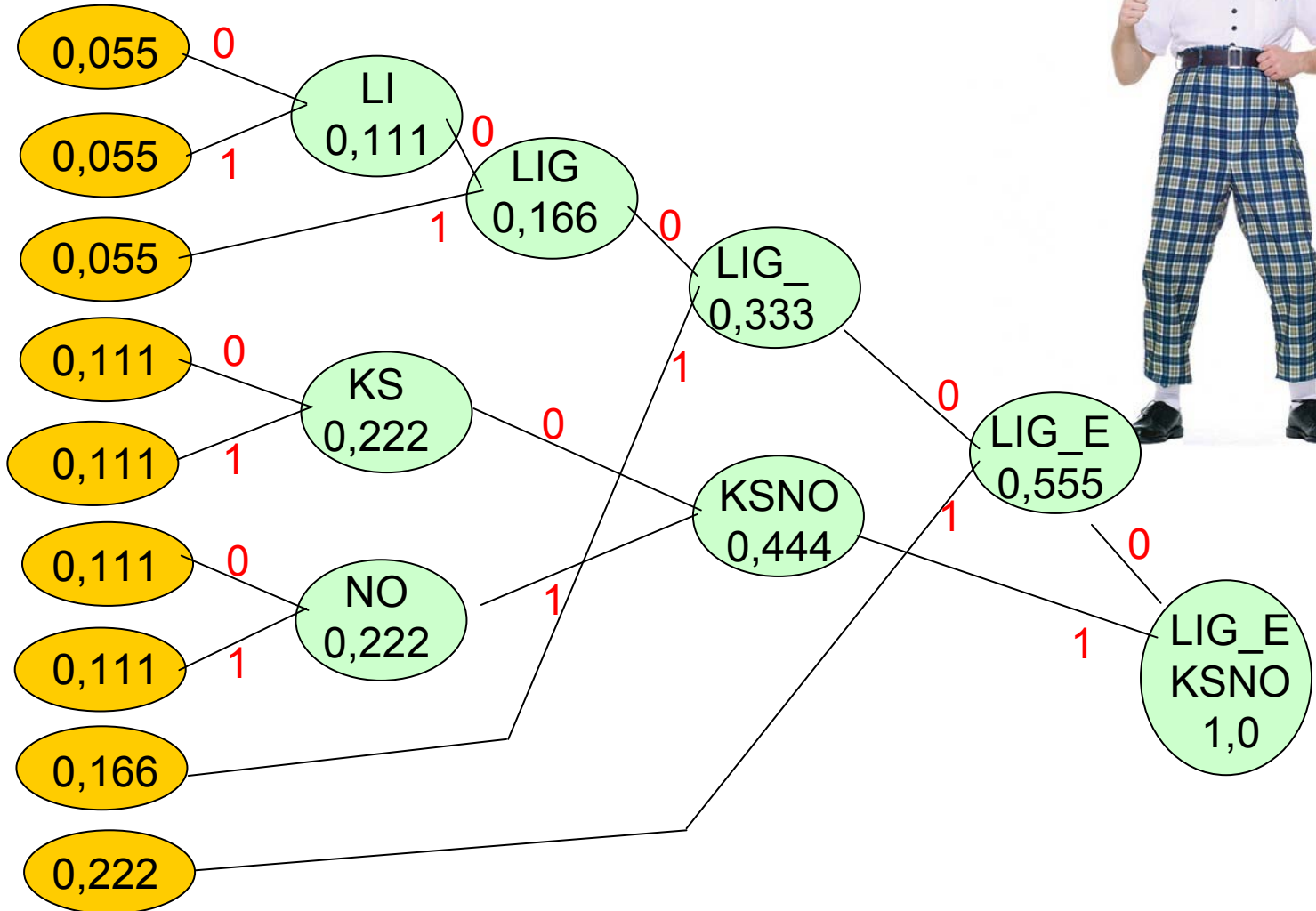
A	000
B	01
C	10
D	001
E	11

NO_ONE_LIKES_GEEKS

Huffman-Codierung



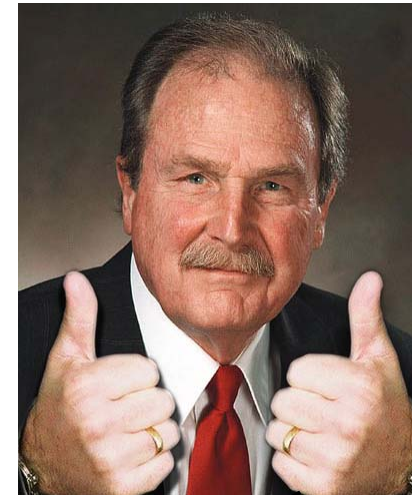
00000	L
00001	I
0001	G
100	K
101	S
110	N
111	O
001	-
01	E



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

p	x = $\text{ld}(1/p)$	$p \cdot x$	$p \cdot c $
0,055	4,170	0,229	0,275
0,055	4,170	0,229	0,275
0,055	4,170	0,229	0,220
0,111	3,170	0,352	0,333
0,111	3,170	0,352	0,333
0,111	3,170	0,352	0,333
0,111	3,170	0,352	0,333
0,166	2,585	0,429	0,498
0,222	2,170	0,481	0,444

$x(a) = \text{ld}(1/p(a))$
 $L = \sum p(a) |c(a)|$
 $H = \sum p(a) x(a)$



Redundanz =
L - H

$H = 3,004$ $L = 3,044$ **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		

Nachricht: kokokoko

LZW-Komprimierung

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)