

Komprimierung

Ingo Blechschmidt,
Michael Hartmann

6. Dezember 2006

Inhalt

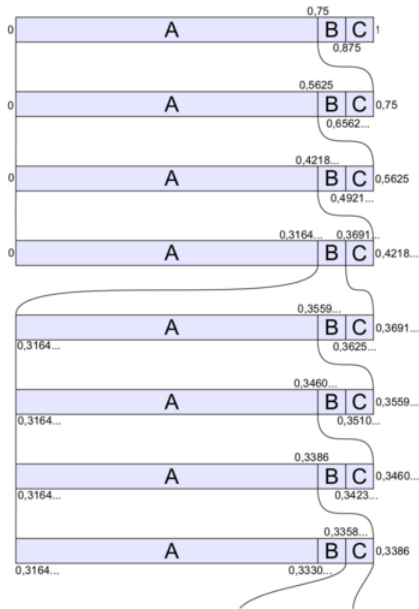
- 1 Lauflängenkodierung
- 2 Arithmetische Kodierung
- 3 Shannon-Fano-Kodierung
 - Grundideen
 - Algorithmus
 - Beispiel
- 4 Huffman-Kodierung
 - Algorithmus
 - Beispiel
 - Verwendung
- 5 Grenzen der Komprimierbarkeit

Laufängenkodierung (RLE)

- Idee: Ersetzung von sich direkt wiederholenden Zeichen durch eine Anweisung
- Beispiel:
aaabbccccccde →
3 a, bb, 6 c, de
- Sehr leicht umsetzbar
- Effizient nur in Spezialfällen, beispielsweise großen einfarbigen Bereichen in Bildern

Arithmetische Kodierung

- 1 Unterteilung eines Einheitsintervalls entsprechend den relativen Häufigkeiten der Zeichen des Texts
- 2 Weitere Unterteilung bis alle Zeichen genutzt
- 3 Kodierung einer beliebigen Zahl des schärfsten Intervalls



Shannon-Fano-Kodierung

Shannon-Fano-Kodierung

Entropiekodierung („Kompressionsverfahren“)



Darstellung *häufiger* Zeichen durch *kurze* Bitfolgen;
Darstellung *seltener* Zeichen durch *lange* Bitfolgen

- Eindeutigkeit der Bitfolgen („Präfixfreiheit“)

Problembeispiel: $A \mapsto 10$ $B \mapsto 01$ $C \mapsto 0$

$ABC \mapsto 10010$ }
 $ACA \mapsto 10010$ } nicht eindeutig

Algorithmus

- 1 Sortierung der Zeichen nach rel. Häufigkeit
- 2 Einteilung der Zeichen in zwei Gruppen, sodass Summen der Häufigkeiten etwa gleich
- 3 So lange fortfahren, bis Entsprechung jedes Zeichens durch einen Pfad im Baum

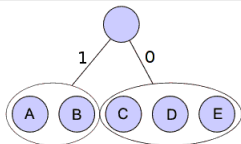
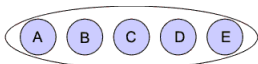
Beispiel

Text (39 Zeichen):

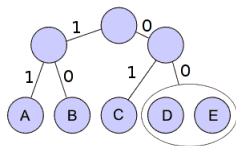
ABADDCCAABABEDAECEBDDDDAAAABAAAABBBCAECECE

Zeichen	A	B	C	D	E
Abs. Häufigkeit	15	7	6	6	5

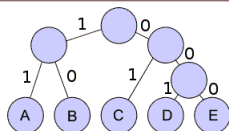
a



c



d



Beispiel

■ Original

(117 bit; Entropie $\approx 0,82$ bit):

```
00000100001101101001000000000
10000011000110001000100010110
11011000000000000001000000000
000001001010000100010100010100
```

Zeichen	A	B	C	D	E
Abs. Häufigkeit	15	7	6	6	5
Benötigte Bits	3	3	3	3	3

Bit	0	1
Abs. Häufigkeit	87	30

A \mapsto 000
 B \mapsto 001
 C \mapsto 010
 D \mapsto 011
 E \mapsto 100

■ Komprimiert

(89 bit; Entropie $\approx 0,99261$ bit (!!)):

```
11101100100101011111110
1110000001110000110001
00100111111111101111111
111010011110000100001000
```

Zeichen	A	B	C	D	E
Abs. Häufigkeit	15	7	6	6	5
Benötigte Bits	2	2	2	3	3

Bit	0	1
Abs. Häufigkeit	40	49

A \mapsto 11
 B \mapsto 10
 C \mapsto 01
 D \mapsto 001
 E \mapsto 000

Huffman-Kodierung: Algorithmus

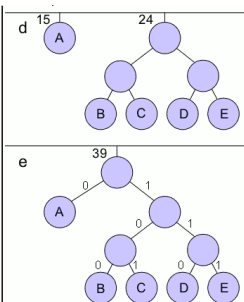
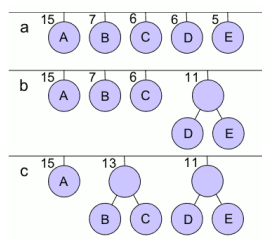
- 1 Wald erstellen mit allen vorkommenden Zeichen
- 2 Neuen Baum erstellen; die beiden Bäume mit geringster Häufigkeit als Blätter nutzen
- 3 So lange fortfahren, bis nur noch ein Baum vorhanden

Beispiel

Text (39 Zeichen):

ABADDCCAABABEDAECEBDDDDAAAABAAAABBBCAECECE

Zeichen	A	B	C	D	E
Abs. Häufigkeit	15	7	6	6	5



Beispiel

■ Original

(117 bit; Entropie $\approx 0,82$ bit):

```
00000100001101101001000000000
10000011000110001000100010110
11011000000000000001000000000
000001001010000100010100010100
```

Zeichen	A	B	C	D	E
Abs. Häufigkeit	15	7	6	6	5
Benötigte Bits	3	3	3	3	3

Bit	0	1
Abs. Häufigkeit	87	30

A \mapsto 000
 B \mapsto 001
 C \mapsto 010
 D \mapsto 011
 E \mapsto 100

■ Komprimiert

(87 bit; Entropie $\approx 0,99762$ bit (!!!)):

```
01000110110101101001000
0100111110011110110011
0110110000010000001001
001010111101111101111
```

Zeichen	A	B	C	D	E
Abs. Häufigkeit	15	7	6	6	5
Benötigte Bits	1	3	3	3	3

Bit	0	1
Abs. Häufigkeit	41	46

A \mapsto 0
 B \mapsto 100
 C \mapsto 101
 D \mapsto 110
 E \mapsto 111

Verwendung

Verwendung von Deflate (LZ77 kombiniert mit Huffman-Kodierung):

- zip
- gzip
- png
- tiff
- pdf
- cab

Grenzen der Komprimierbarkeit

- Entropie als untere Schranke der Komprimierbarkeit
- Beweis der Nichtexistenz eines Perfekten VerfahrensTM:
 - *Annahme*: Existenz eines Verfahrens, dass jeden beliebigen Text um ein Bit verkürzt
 - *Dann*: Rekursive Anwendung denkbar
 - *Schluss*: Komprimierung jedes beliebigen Texts auf ein Bit
 - *Aber*: $256 < \infty!$

Fragen?

Bildquellen

- `http://upload.wikimedia.org/wikipedia/de/d/db/ShannonCodeAlg.png`
- `http://upload.wikimedia.org/wikipedia/de/d/d8/HuffmanCodeAlg.png`
- `http://upload.wikimedia.org/wikipedia/de/5/56/ArithmetischesCodierenBeispiel.png`