4 Codierung diskreter Quellen, Teil II

4.1 Wiederholung: Huffman-Codierung
4.2 Optimalität der Huffman-Codierung
4.3 Nachteile/Probleme der Huffman-Codierung
4.4 Arithmetische Codierung/Range Coding
4.5 Anwendungsspezifische Codierungen
4.6 Zusammenfassung

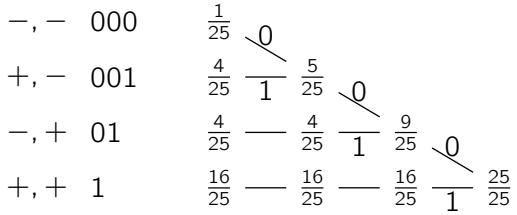
4.1 Wiederholung: Huffman-Codierung

Beispiel 4.1

Es soll eine Quelle $\mathcal{X}$ mit $X = \{+, -\}$ und $p_X(+) = \frac{4}{5}, p_X(-) = \frac{1}{5}$ codiert werden.
Die Entropie $H(\mathcal{X}) = -\frac{4}{5}\operatorname{ld}(\frac{4}{5}) - \frac{1}{5}\operatorname{ld}(\frac{1}{5}) \approx 0{,}7219$ gibt die kleinstmögliche mittlere Codewortlänge an.
Die Huffman-Codierung $\mathcal{C}$ zu einer Quelle mit zwei Symbolen ist die triviale Codierung $+ \mapsto 1, - \mapsto 0$ mit der mittleren Codewortlänge $L(\mathcal{C}) = 1$.

Beispiel 4.2

Nun sei $\mathcal{X}^2$ betrachtet.



$$L(\mathcal{C}) = \left(\frac{1}{25} + \frac{4}{25}\right) \cdot 3 + \frac{4}{25} \cdot 2 + \frac{16}{25} \cdot 1 = \frac{39}{25} = 1{,}56 = 2 \cdot 0{,}78$$

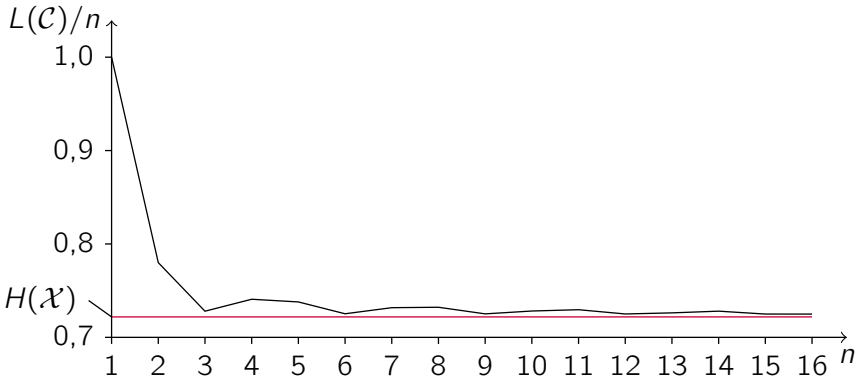
Beispiel 4.3

Nun sei $\mathcal{X}^3$ betrachtet.

$-,-,-$	00000	$\frac{1}{125}$	0													
$+,-,-$	00001	$\frac{4}{125}$	1	$\frac{5}{125}$	—	$\frac{5}{125}$										
$-,+,-$	00010	$\frac{4}{125}$	—	$\frac{4}{125}$	0				0							
$-,-,+$	00011	$\frac{4}{125}$	—	$\frac{4}{125}$	1	$\frac{8}{125}$	1	$\frac{13}{125}$		0						
$-,+,+$	001	$\frac{16}{125}$	—	$\frac{16}{125}$	—	$\frac{16}{125}$	—	$\frac{16}{125}$	1	$\frac{29}{125}$	—	$\frac{29}{125}$				
$+,+,-$	010	$\frac{16}{125}$	—	$\frac{16}{125}$	—	$\frac{16}{125}$	—	$\frac{16}{125}$	—	$\frac{16}{125}$	0			0		
$+,-,+$	011	$\frac{16}{125}$	—	$\frac{16}{125}$	—	$\frac{16}{125}$	—	$\frac{16}{125}$	—	$\frac{16}{125}$	1	$\frac{32}{125}$	1	$\frac{61}{125}$	0	
$+,+,+$	1	$\frac{64}{125}$	—	$\frac{64}{125}$	—	$\frac{64}{125}$	—	$\frac{64}{125}$	—	$\frac{64}{125}$	—	$\frac{64}{125}$	—	$\frac{64}{125}$	1	$\frac{125}{125}$

$$L(\mathcal{C}) = \frac{(1+4+4+4)\cdot 5 + (16+16+16)\cdot 3 + 64\cdot 1}{125} = \frac{273}{125} = 2{,}184 = 3\cdot 0{,}728$$

Für steigende n nähert sich die normierte mittlere Codewortlänge $L(\mathcal{C})/n$ der Entropie $H(\mathcal{X})$.



4.2 Optimalität der Huffman-Codierung

Lemma 4.1

Hat die Codierung $\mathcal{C}'$ zur konstruierten reduzierten Quelle $\mathcal{X}'$ die mittlere Codewortlänge $L(\mathcal{C}')$, so hat die Codierung $\mathcal{C}$ die mittlere Codewortlänge

$$L(\mathcal{C}) = L(\mathcal{C}') + p(x_1) + p(x_2) = L(\mathcal{C}') + p(x'_{1,2}). \tag{4.1}$$

Beweis.

$$L(\mathcal{C}) = \sum_{x\in X} p_X(x) l_x = p_X(x_1) l_{x_1} + p_X(x_2) l_{x_2} + \sum_{\substack{x\in X \\ x\neq x_1, x_2}} p_X(x) l_x \tag{4.2}$$

$$= \big(p_X(x_1) + p_X(x_2)\big) \cdot (1 + l_{x'_{1,2}}) + \sum_{\substack{x'\in X' \\ x'\neq x'_{1,2}}} p_{X'}(x') l'_{x'} \tag{4.3}$$

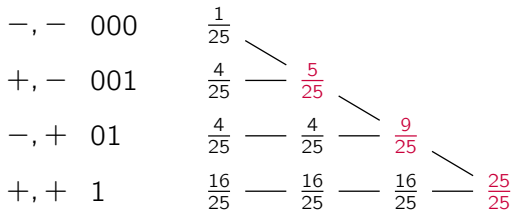
$$= p_X(x_1) + p_X(x_2) + p_{X'}(x'_{1,2}) l'_{x'_{1,2}} + \sum_{\substack{x'\in X' \\ x'\neq x'_{1,2}}} p_{X'}(x') l'_{x'} \tag{4.4}$$

$$= p_X(x_1) + p_X(x_2) + \sum_{x'\in X'} p_{X'}(x') l'_{x'} = p_X(x_1) + p_X(x_2) + L(\mathcal{C}') \tag{4.5}$$

□

Beispiel 4.4

Es sei nochmals die Codierung aus Beispiel 4.2 betrachtet.



Die mittlere Codewortlänge beträgt

$$L(\mathcal{C}) = \left(\frac{1}{25} + \frac{4}{25}\right) \cdot 3 + \frac{4}{25} \cdot 2 + \frac{16}{25} \cdot 1 = \frac{39}{25} = 1{,}56$$

bzw.

$$L(\mathcal{C}) = \frac{5}{25} + \frac{9}{25} + \frac{25}{25} = \frac{39}{25} = 1{,}56.$$

Lemma 4.2

Für eine optimale Codierung $\mathcal{C}$ folgt $l_x \leq l_{x'}$ aus $p_X(x) > p_X(x')$.

Beweis.

Angenommen $l_x > l_{x'}$, und sei $\mathcal{C}'$ aus $\mathcal{C}$ durch vertauschen von $\mathcal{C}(x)$ und $\mathcal{C}(x')$ gebildet. Dann gilt

$$L(\mathcal{C}) - L(\mathcal{C}') = l_x p_X(x) + l_{x'} p_X(x') - l_{x'} p_X(x) - l_x p_X(x') \tag{4.6}$$

$$= (l_x - l_{x'}) \cdot (p_X(x) - p_X(x')) > 0, \tag{4.7}$$

womit $\mathcal{C}$ im Widerspruch zur Voraussetzung nicht optimal wäre. □

Lemma 4.3

Bei einer optimalen Präfix-Codierung $\mathcal{C}$ gibt es zu jedem Codewort $\mathcal{C}(x)$ maximaler Länge ein weiteres Codewort, dass sich nur im letzen Bit von $\mathcal{C}(x)$ unterscheidet (sein Geschwister).

Beweis.

Sei o.B.d.A. $\mathcal{C}(x) = \boldsymbol{c}0$. Ist $\boldsymbol{c}1$ im Code enthalten, trifft die Behauptung offenbar zu. Ist $\boldsymbol{c}1$ nicht im Code enthalten, so lässt sich eine Codierung $\mathcal{C}'$ konstruieren, die $\mathcal{C}$ bis auf $\mathcal{C}'(x) = \boldsymbol{c}$ gleicht und ebenfalls die Präfix-Eigenschaft aufweist. Es gilt $l'_x = l_x - 1$ und damit auch $L(\mathcal{C}') < L(\mathcal{C})$[3], was im Widerspruch zur Voraussetzung steht, dass $\mathcal{C}$ optimal ist. □

Korollar 4.1

Es gibt eine optimale Präfix-Codierung, bei der die Codewörter zu den beiden unwahrscheinlichsten Quellsymbolen Geschwister sind und maximale Länge haben.

[3] Der Einfachheit halber nehmen wir hier und im Folgenden $p_X(x) > 0$ an.

Satz 4.1

Die Huffman-Codierung ist optimal.

Beweis.

Der Beweis erfolgt über Induktion über die Größe des Quellalphabets.
Für nur zwei Quellsymbole ist die Huffman-Codierung offensichtlich optimal.
Es gilt also zu zeigen, dass wenn $\mathcal{C}'$ aus Schritt 3 des Algorithmus optimal ist, auch die gebildete Codierung $\mathcal{C}$ optimal ist.
Strategie: Wir zeigen, dass es eine optimale Codierung $\mathcal{D}$ gibt, die der Huffman-Codierung $\mathcal{C}$ so ähnlich ist, dass auch $\mathcal{C}$ optimal ist.
Nach Korollar 4.1 existiert eine optimale Codierung $\mathcal{D}$, sodass $\mathcal{D}(x_1)$ und $\mathcal{D}(x_2)$ Geschwister sind.
Sei nun $\mathcal{D}'$ die Codierung von $\mathcal{X}'$ mit $\mathcal{D}'(x) = \mathcal{D}(x)$ für $x \neq x_1, x_2$ und $\mathcal{D}'(x'_{1,2})$ dem gemeinsamen Präfix von $\mathcal{D}(x_1)$ und $\mathcal{D}(x_2)$.
Übertragung von Lemma 4.1 liefert $L(\mathcal{D}) = L(\mathcal{D}') + p_{X'}(x'_{1,2})$, und mit $L(\mathcal{C}') \leq L(\mathcal{D}')$ (wegen der Optimalität von $\mathcal{C}'$) folgt

$$L(\mathcal{D}) \geq L(\mathcal{C}') + p_{X'}(x'_{1,2}) = L(\mathcal{C}), \tag{4.8}$$

womit $\mathcal{C}$ optimal ist. □

4.3 Nachteile/Probleme der Huffman-Codierung

- A-Priori-Kenntnis der Verteilungsdichte nötig.
  - In der Praxis kann die Häufigkeitsverteilung der Daten, ggfs. in Blöcken, ermittelt und die Code-Tabelle zum Decoder übertragen werden.
  - Der Overhead bleibt gering, solange die Code-Tabellen nicht zu groß werden $\Rightarrow$ kleine n!
- Setzt Quellen mit unabhängigen Ereignissen voraus bzw. nutzt Abhängigkeiten nicht aus.
  - Kann mit unterschiedlichen Code-Tabellen, je nach vorhergehenden Symbolen gelöst werden.
  - Overhead und Komplexität schnell unpraktikabel.

  $\Rightarrow$ Vorgeschaltete Codierung, die Abhängigkeiten ausnutzt und entfernt.

4.4 Arithmetische Codierung/Range Coding

Die arithmetische Codierung codiert nicht einzelne Symbole, sondern die gesamte Nachricht (bzw. große Blöcke) der Länge n. Das explizite Aufstellen einer Code-Tabelle entfällt.
Einer zu codierenden Sequenz $\boldsymbol{x}$ wird ein Intervall $\big[L(\boldsymbol{x}), U(\boldsymbol{x})\big[$ zugeordnet. Sei $\boldsymbol{x} = \boldsymbol{x'}x$ und auf den Symbolen $x \in X$ eine vollständige Ordnung definiert, so werden die Intervallgrenzen wie folgt bestimmt:

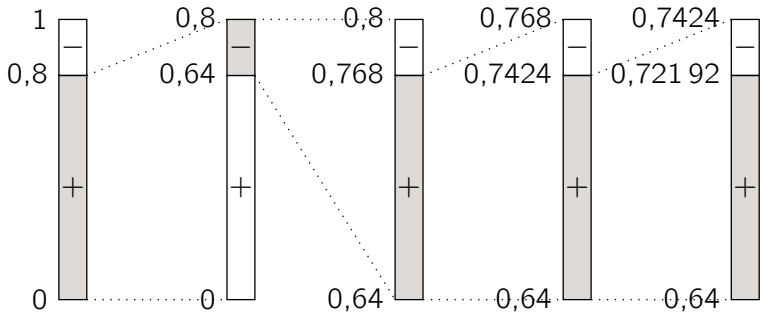
$$L(\boldsymbol{x'}x) = L(\boldsymbol{x'}) + \big(U(\boldsymbol{x'}) - L(\boldsymbol{x'})\big) \cdot \sum_{x'<x} p_X(x') \qquad \text{mit } L(\epsilon) = 0 \tag{4.9}$$

$$U(\boldsymbol{x'}x) = L(\boldsymbol{x'}x) + \big(U(\boldsymbol{x'}) - L(\boldsymbol{x'})\big) \cdot p_X(x) \qquad \text{mit } U(\epsilon) = 1 \tag{4.10}$$

Aus dem Intervall wird die Zahl mit der kürzesten Binärdarstellung ausgewählt, dies ist die zugeordnete Codesequenz $C(\boldsymbol{x})$. Zur Decodierung wird neben den Symbolewahrscheinlichkeiten noch die Nachrichtenlänge n benötigt.

Beispiel 4.5

Es sei wieder die Quelle $\mathcal{X}$ mit $X = \{+, -\}$ und $p_X(+) = \frac{4}{5}, p_X(-) = \frac{1}{5}$ betrachtet.
Es soll die Sequenz $\boldsymbol{x} = + - + + +$ codiert werden.



$L(\boldsymbol{x}) = 0{,}64 = 0{,}101\,000\,11\ldots_2$, $U(\boldsymbol{x}) = 0{,}721\,92 = 0{,}101\,110\,00\ldots_2$
$L(\boldsymbol{x}) \leq 0{,}1011_2 < U(\boldsymbol{x})$, also $C(\boldsymbol{x}) = 1011$

Satz 4.2

Arithmetische Codierung einer Sequenz $\boldsymbol{x} = x_1 x_2 \ldots x_n$ führt auf ein Intervall $[L(\boldsymbol{x}), U(\boldsymbol{x})[$ der Länge

$$U(\boldsymbol{x}) - L(\boldsymbol{x}) = \prod_{i=1}^{n} p_X(x_i). \tag{4.11}$$

Beweis.

Es sei wieder $\boldsymbol{x} = \boldsymbol{x'}x$. Aus

$$L(\boldsymbol{x'}x) = L(\boldsymbol{x'}) + \big(U(\boldsymbol{x'}) - L(\boldsymbol{x'})\big) \cdot \sum_{x'<x} p_X(x') \tag{4.12}$$

$$U(\boldsymbol{x'}x) = L(\boldsymbol{x'}x) + \big(U(\boldsymbol{x'}) - L(\boldsymbol{x'})\big) \cdot p_X(x) \tag{4.13}$$

folgt sofort, dass $U(\boldsymbol{x'}x) - L(\boldsymbol{x'}x) = \big(U(\boldsymbol{x'}) - L(\boldsymbol{x'})\big) \cdot p_X(x)$.
Der Beweis erfolgt nun per Induktion über die Sequenzlänge n. Für $n = 1$, also $\boldsymbol{x'} = \epsilon$, gilt die Aussage offensichtlich.
Gilt die Aussage für $n-1$, also $U(\boldsymbol{x'}) - L(\boldsymbol{x'}) = \prod_{i=1}^{n-1} p_X(x_i)$, so folgt für n unmittelbar

$$U(\boldsymbol{x'}x) - L(\boldsymbol{x'}x_n) = \big(U(\boldsymbol{x'}) - L(\boldsymbol{x'})\big) \cdot p_X(x_n) = \prod_{i=1}^{n} p_X(x_i). \tag{4.14}$$

□

Anstelle der Wahrscheinlichkeiten $p_X(x)$ lässt sich die arithmetische Codierung auch mit den absoluten Symbolhäufigkeiten $f_x \in \{1, \ldots, n\}$ formulieren:

$$L(\boldsymbol{x'}x) = n \cdot L(\boldsymbol{x'}) + \big(U(\boldsymbol{x'}) - L(\boldsymbol{x'})\big) \cdot \sum_{x'<x} f_{x'} \qquad \text{mit } L(\epsilon) = 0 \tag{4.15}$$

$$U(\boldsymbol{x'}x) = L(\boldsymbol{x'}x) + \big(U(\boldsymbol{x'}) - L(\boldsymbol{x'})\big) \cdot f_x \qquad \text{mit } U(\epsilon) = 1. \tag{4.16}$$

Damit gilt $0 \leq L(\boldsymbol{x}) < n^n$, $0 < U(\boldsymbol{x}) \leq n^n$ und $U(\boldsymbol{x}) - L(\boldsymbol{x}) = \prod_{i=1}^{n} f_{x_i} = n^n \cdot \prod_{i=1}^{n} \frac{f_{x_i}}{n}$.
Aufgrund des möglichen Wertebereichs lässt sich jeder Wert im Intervall $[L(\boldsymbol{x}), U(\boldsymbol{x})[$ sicher mit $\lceil \mathrm{ld}(n^n) \rceil$ Bits darstellen. Aus dem Intervall wird derjenige Wert ausgesucht, der auf die längste Folge von Nullen endet; das Codewort ist die Binärdarstellung dieses Wertes ohne die abschließenden Nullen, aber gegebenenfalls inlusive der führenden Nullen der urspünglichen Darstellung mit $\lceil \mathrm{ld}(n^n) \rceil$ Bits.

Beispiel 4.6

Es soll wieder die Sequenz $\boldsymbol{x} = +-+++$ codiert werden. Wir haben $f_+ = 4, f_- = 1$, damit folgt:

$\boldsymbol{x}$	ϵ	$+$	$+-$	$+-+$	$+-++$	$+-+++$
$L(\boldsymbol{x})$	0	0	16	80	400	2000
$U(\boldsymbol{x})$	1	4	20	96	464	2256
$U(\boldsymbol{x}) - L(\boldsymbol{x})$	1	4	4	16	64	256

Zur Darstellung werden (höchstens) $\lceil \mathrm{ld}(5^5) \rceil = 12$ Bits benötigt. Es gilt $L(\boldsymbol{x}) = 2000 = 011111010000_2$ und $U(\boldsymbol{x}) = 2256 = 100011010000_2$. Als Repräsentanten des Intervalls wählen wir $2048 = 100000000000_2$; ohne abschließende Nullen ist daher $C(\boldsymbol{x}) = 1$.
Beachte: $\frac{2000}{5^5} = 0{,}64$ und $\frac{2256}{5^5} = 0{,}721\,92$.

Ist n eine Zweierpotenz, sodass auch n^n einer Zweierpotenz ist, und gilt $f_x = n \cdot p_X(x)$, so sind beide Ansätze äquivalent. Andernfalls weisen sie zwar weitgehend die gleichen Eigenschaften auf, liefern aber im Einzelfall unterschiedliche Ergebnisse.

Satz 4.3

Die (auf ein Quellsymbol bezogene) mittlere Codewortlänge der arithmetischen Codierung konvergiert für große n gegen die Entropie $H(\mathcal{X})$ der Quelle.

Beweisskizze.

Sei f_x die absolute Häufigkeit des Symbols x in einem betrachteten Block der Länge n. Für große n gilt $p_X(x) = \frac{f_x}{n}$ mit verschwindend geringem Fehler.
In der zweiten Variante der arithmetischen Codierung sind $L(\boldsymbol{x})$ und $U(\boldsymbol{x})$ ganze Zahlen und es gilt $0 \leq L(\boldsymbol{x}) < U(\boldsymbol{x}) \leq n^n$. Damit lässt sich jede Zahl im Intervall $\big[L(\boldsymbol{x}), U(\boldsymbol{x})\big[$ mit höchstens $\lceil \mathrm{ld}(n^n) \rceil < \mathrm{ld}(n^n) + 1 = n \,\mathrm{ld}\, n + 1$ Bits darstellen.
Da das Intervall die Länge $U(\boldsymbol{x}) - L(\boldsymbol{x}) = \prod_{i=1}^{n} f_{x_i}$ hat, gibt es in dem Intervall einen Wert, dessen Binärdarstellung auf eine Folge von mindestens $\big\lfloor \mathrm{ld}\left(\prod_{i=1}^{n} f_{x_i}\right) \big\rfloor \geq \mathrm{ld}\left(\prod_{i=1}^{n} f_{x_i}\right) - 1 = \sum_{i=1}^{n} \mathrm{ld}\, f_{x_i} - 1 = \sum_{x \in X} f_x \,\mathrm{ld}\, f_x - 1$ Nullen endet.
Dieser lässt sich also mit höchstens
$n \,\mathrm{ld}\, n + 1 - \left(\sum_{x \in X} f_x \,\mathrm{ld}\, f_x - 1\right) = n \,\mathrm{ld}\, n + 2 - \sum_{x \in X} f_x \,\mathrm{ld}\, f_x$ Bits darstellen.
Bezogen auf ein einzelnes Symbol ergibt sich somit eine mittlere Codewortlänge von höchstens

$$\frac{2}{n} + \mathrm{ld}\, n - \sum_{x \in X} \frac{f_x}{n} \,\mathrm{ld}\, f_x = \frac{2}{n} + \sum_{x \in X} p_X(x) \,\mathrm{ld}\, n - \sum_{x \in X} p_X(x) \,\mathrm{ld}\, f_x$$

$$= \frac{2}{n} - \sum_{x \in X} p_X(x) \,\mathrm{ld}\, p_X(x) = \frac{2}{n} + H(\mathcal{X}). \quad (4.17)$$

4.5 Anwendungsspezifische Codierungen

- Lauflängencodierung, primär für S/W-Bilder
  - Ersetzt folgen gleicher Symbole durch Anzahl und ein Symbol.
  - Beispiel: SSSSSSSSWWWWWSSSSSS $\mapsto$ 8S5W6S
- Prädiktive Ansätze, insbesondere für Audio- und Videodaten
  - Aktueller Abtastwert/aktuelles Bild wird aus Vergangenheitswerten vorhergesagt.
  - Nur die Differenz (Prädiktionsfehler) wird übertragen.
  - Prädiktionsfehler hat in der Regel gehäuft niedrige Amplitudenwerte $\Rightarrow$ geringe Entropie.
- Lempel-Ziv-(Welch-)Codierung, eigentlich für Text, aber recht universell einsetzbar
  - Wird im Folgenden erklärt.

4.5.1 Lempel-Ziv-Welch-Codierung (LZW)

Bei der LZW-Codierung pflegen Codierer und Decodierer ein Wörterbuch, das auf den bereits übertragenen Daten basiert.
Die Codierung läuft nach folgendem Algorithmus ab:

1. Initialisiere Wörterbuch = Symbolvorrat der Quelle (z.B. 7- oder 8-Bit-ASCII).
2. Finde längstes Präfix der noch zu codierenden Symbolsequenz, das im Wörterbuch enthalten ist, und sende zugehörigen Index.
3. Ergänze Wörterbuch um diese Sequenz gefolgt vom nächsten Symbol der Quellsequenz.
4. Wiederhole ab 2.

Beispiel 4.7

Codierung von KATZENTATZE (11 Symbole, $H \approx 2{,}482 \Rightarrow$ mindestens 28 Bit).

KATZENTATZE

Ausgabe der LZW-Codierung	Wörterbuch	
K	0	KA
A	1	AT
T	2	TZ
Z	3	ZE
E	4	EN
N	5	NT
T	6	TA
1	7	ATZ
3		

9 Symbole, $H \approx 2{,}948 \Rightarrow$ 27 Bit

Durch den sukzessiven Aufbau des Wörterbuchs steigt der Gewinn der LZW-Codierung für längere Texte deutlich an.

Beispiel 4.8

Decodierung von KATZENT13:

KATZENT13

Ausgabe der LZW-Decodierung	Wörterbuch	
K		
A	0	KA
T	1	AT
Z	2	TZ
E	3	ZE
N	4	EN
T	5	NT
AT	6	TA
ZE	7	ATZ

Beispiel 4.9

Codierung von `DREI_ZWEI_EINS_MEINS` (20 Symbole, $H \approx 3{,}084 \Rightarrow$ mindestens 62 Bit).

`DREI_ZWEI_EINS_MEINS`

Ausgabe der LZW-Codierung	Wörterbuch		
D	0	DR	
R	1	RE	
E	2	EI	
I	3	I_	
_	4	_Z	
Z	5	ZW	
W	6	WE	
2	7	EI_	16 Symbole, $H \approx 3{,}453 \Rightarrow$ 56 Bit
_	8	_E	
2	9	EIN	
N	10	NS	
S	11	S_	
_	12	_M	
M	13	ME	
9	14	EINS	
S			

4.6 Zusammenfassung

- Die Huffman-Codierung ist optimal, d.h. es gibt keine Codierung, die die gleiche Quelle mit geringerer mittlerer Codewortlänge codieren kann...
- ...solange die Quellsymbole statistisch unabhängig sind. Andernfalls empfiehlt sich eine anwendungsspezifische Codierung.
- Für Texte (und textähnliche Quellen) ist die wörterbuchbasierte LZW-Codierung ein in der Praxis bewährter Algorithmus.