I Codierung ohne Rauschen

1. Allgemeines über Codes:

Def:

1. Zeichen sind Elemente einer endlichen Menge von unterschiedlichen Dingen.
   Diese Menge nennt man Zeichenvorrat oder Alphabet
   Diese Dinge können Lichtimpulse, Magnetschw., Spannungen, Töne usw. sein.
2. Ein Wort besteht aus Zeichen eines Alphabets.
   Die Wortlänge gibt an, wieviele Zeichen des Alphabets das Wort hat.
3. M sei ein Alphabet. Dann ist M^* die Menge aller Worte die sich durch Zeichen dieses Alphabets darstellen lassen.
4. Es seien A, B Alphabet. Dann ist die Codierung von Zeichen aus A nach Worten nach B^* definiert durch eine injektive Abb. $\varphi: A \to B^*$
   d.h. jedem Wort aus A wird genau ein Wort aus B^* zugeordnet und jedem Wort aus B^* entspricht genau ein Zeichen aus A.

Beispiel: 7 Bit ASCII-Code:

$R \hat{=} 1011001010$, $B \hat{=} \{1,0\}$, $R \in A$ Codierung

5. Codierung v. Worten aus A
   Es seien A und B Alphabete und es gelte $\varphi(a) \in B^*$ für $a \in A$.
   Es sei $a_1, a_2, \dots, a_k$ ein Wort aus A. Dann ist $\varphi^*: A^* \to B^*$ def. durch:
   $\varphi(a_1, a_2, \dots, a_k) = \varphi(a_1)\varphi(a_2)\varphi(a_3)\cdots\varphi(a_k)$

Beispiel: Ostermann $\in A^*$, $a_i \in A$, $\varphi(\text{Ostermann}) = \varphi(O)\varphi(s)\varphi(t)\cdots\varphi(n)$
$\varphi(a_i) \to B^*$ ist eine injektive Abb.; $\varphi^*(A^*) \to B^*$ muss keine inj. Abb. sein.

6. Ein Alphabet mit nur 2 Zeichen heißt binäres Alphabet
   Ein " " 3 " ternäres Alphabet usw.
   Es sei B ein Alphabet mit q Zeichen, dann schreibt man $q = |B|$

Beispiel: a) Binäre Alphabete: Diese verwendet man in technischen Bereichen um mit Computern zu arbeiten.
b) q-näre Alphabete: Alphabete mit $q > 2$ verwendet man in d. Biologie, Physik etc. Wenn man mit einem zum Problem passenden Alphabet arbeitet (natürlich) (zufällig $q \geq 2$)

2. Decodieren:

Man hat einen Wort von B^* und möchte genau das Urbild des Wortes haben.

a) Das Urbild ist ein Zeichen von A. Da $\varphi: A \to B^*$ eine injekt. Abb. ist das Urbild eindeutig bestimmt

b) Das Urbild sei ein Wort über A z.B. das Wort $a_1a_2a_3\cdots a_k$, $\varphi(a_1\cdots a_k) = \varphi(a_1)\cdots\varphi(a_k) \in B^*$
Man kann nur decodieren, wenn man die Nahtstellen zwischen den Worten $\varphi(a_i)\varphi(a_j)$ kennt
$\varphi(a_i)$, $a_i \in A$ nennt man Codewort.

α) Codes mit fester Wortlänge
⇒ die Nahtstelle zw. den Codeworten kann man durch Abzählen d. Zeichen aus B finden.
⇒ $\varphi^*: A^* \to B^*$ ist eine inj. Abb. ⇒ decodierbar (eindeutig)

β) Codes mit Variabler Wortlänge
Hier lässt sich die Nahtstelle der einzelnen Codeworte nicht immer finden.

Beispiel: -Codes mit fester Wortlänge:
1) 7-Bit-Ascii-Code (128 Zeichen ⇒ 26 groß-, 27 klein Lat., 28 Sonder, 10 Ziffern, 32 Steuer, 6...)
2) Fernschreibcode (5 Bit code, 5 Bit Info, 1 start, 1 stop)
3) Lochstreifencode (7 Bit info, 1 Bit für Lesefehler)
4) 8 Bit EBCDIC-Code (Zeichensatz wie Ascii-Code + 1 Bit)
5) 12-Bit Lochkartencode usw

-Codes mit variabler Wortlänge:
Morsecode: Alphabet hat 3 Zeichen (B) z.B. k. L. pause.

Beispiele:
Es sei $A = \{a_1, a_2, a_3, a_4\}$, $B = \{0,1\}$

a)

a_1	00
a_2	01
a_3	10
a_4	11

b)

a_1	0
a_2	10
a_3	110
a_4	1110

c)

a_1	0
a_2	01
a_3	10
a_4	11

d)

a_1	010
a_2	11000
a_3	01000
a_4	00110

Beispiel a: feste Wortlänge ⇒ decodierbar
Beispiel b bis d: Man betrachte die Folgen v. Codeworten.
β) $\varphi(a_1,a_2,a_3,a_4)$ = 0|10|110|111 bei Code b
γ) $\varphi($ " $)$ = 0|10|0|110 bei Code c
δ) $\varphi($ " $)$ = 0|10|1000|10|1000|10|110 bei Code d

Man betrachte die Wortfolge γ unter dem Code c: $\varphi(a_1a_3a_2a_3) = 0110110 = 0110119 = 0100110$
⇒ nicht decodierbar da Nahtstelle n. eindeutig. $a_1\ a_3\ a_2\ a_3\ \ a_2\ a_1\ a_3\ a_4\ \ a_1\ a_3\ a_1\ a_4$

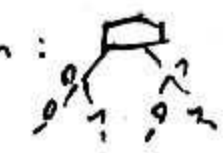
Man betrachte die Wortfolge β unter dem Code b: 0100011000... ankommende Bitfolge
Bei diesem Code ist Decodieren stets möglich, wenn man ein Wortende findet.
Von diesem Wortende lässt sich d. Code v. rechts nach links decodieren.
Man betrachte die Fälle: 1) 0100011000010..., 2) 0100011000110
zu 1): 4 mal 0 ⇒ Wort fängt mit 0 an. 3 Zeichen 0 ⇒ Ende. ⇒ 01000|1000|010...
zu 2): 0|10|0|1|10|0|110... = 0100|10|0|10|110... $a_3\ \ a_2$
$a_1\ \ a_4\ \ a_4\ \ a_3\ \ a_2\ \ a_2$
⇒ links → rechts nicht decodierbar

Def: Suffixcode:
Bei einem Suffixcode ist kein Codewort Endstück eines anderen Codewort.

Satz: Ein Suffixcode ist immer decodierbar. Es lässt sich nur v. rechts nach links decodieren.
Beim Suffixcode hat man eine Verzögerung beim Decodieren (da rechts → links)

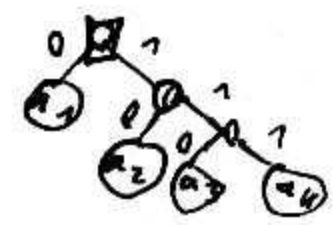
Def (Fano-Bedingung): Ein Code heißt Präfixcode, wenn kein Codewort Anfangsstück eines anderen Codewortes ist.

Satz: Ein Präfixcode ist ohne Verzögerung decodierbar. (Code b ist Präfixcode)
Präfixcodes werden automatisch decodiert mit dem Codebaum:



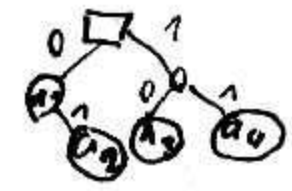
Der Codebaum:
Es sei B ein Alphabet mit q = |B|. Es sei $C = \{a_1, \dots, a_n\}$ ein Code über B^*
Man startet mit der Wurzel. Die Wurzel und jeder Knoten d. Baumes hat maximal q Kinder.
Man trägt die Codeworte in diesen Codebaum ein. Die Urbilder der Codeworte liegen auf Knoten oder Blättern des Baumes. Alle Wege, die zu keinem Urbild eines Codewortes führen, werden aus dem Baum gestrichen.
⇒ Liegen alle Urbilder der Codeworte auf Blättern des Codebaumes, dann hat man einen Präfixcode. Der Code ist eindeutig decodierbar.
Man startet die Decodierung, indem man von der Wurzel den Baum durchläuft bis zu einem Blatt. Damit ist das 1. Codewort decodiert.
Das Urbild steht in dem Blatt. Dann startet man neu. Endet man wieder auf einem Blatt, dann hat man das Urbild des nächsten Codewortes.
Liegt das Urbild eines Codewortes auf einem Knoten und nicht in einem Blatt dann hat man keinen Präfixcode. Der Code ist dann nicht immer eindeutig dec.bar.

Beispiel b: q = |B| = 2, B = {0,1}



Die Urbilder der Codeworte liegen auf Blättern. Man hat einen Präfixcode. Der Code ist eindeutig decodierbar.

Beispiel c: q = |B| = 2, B = {0,1}



Ein Urbild liegt nicht auf einem Blatt, sondern auf einem Knoten.
⇒ Der Code ist kein Präfixcode!
Es kann hier keine Aussage über d. Decodierung gemacht werden

Beispiel d: q = |B| = 2, B = {0,1}

Nicht alle Urbilder liegen auf Blättern
⇒ der Codebaum liefert k. Aussage über d. Decodierung

Ein Suffixcode ist (eindeutig) decodierbar. Ein Suffixcode von hinten gelesen (rechts → links) ist ein Präfixcode ⇒ Man kann einen Suffixcode mit dem Codebaum decodieren
Beispiel d: 
⇒ Alle Urbilder liegen auf Blättern. Der Code ist über den Codebaum (eindeutig) decodierbar.

Andere Codes als Präfix- und Suffixcode decodierbar?
Die Nahtstelle zwischen 2 Codewörtern muss erkennbar sein.

Beispiel: $c = \{1, 101, 1001, 10001\}$
Der Code c ist ein Präfixcode und auch kein Suffixcode. Genau zwischen den Bits 11 liegt die Nahtstelle zwischen 2 Codewörtern ⇒ Der Code ist decodierbar

Satz von Sardina Patterson:
Es sei B ein Alphabet mit $q = |B| \geq 2$. B^* sei die Menge aller Worte über B. B^+ sei identisch mit B^* ohne das leere Wort. Es sei c ein Code über B: $c = \{a_1, \dots, a_n\}$, $a_i \in B^*$.
Es sei $\hat{S}_i = \{w \in B^+ \mid \exists\, w_1 \in c \text{ und } \exists\, w_2 \in S_{i-1} \text{ mit } w_2 = w_1 w\}$
$\tilde{S}_i = \{w \in B^+ \mid \exists\, w_1 \in S_{i-1} \text{ und } \exists\, w_2 \in c \text{ mit } w_2 = w_1 w\}$. Es sei $S = \bigcup S_i$
Der Code ist eindeutig decodierbar ⇔ $c \cap S = \{\}$

Problem: Wie weit muss man die S_i berechnen, um Gewissheit zu haben?
Abbruchbedingungen:
1. Gilt für ein i_0: $S_{i_0} = \{\}$, dann gilt $S_i = \{\}\ \forall i \geq i_0$
2. Gilt für ein i_0: $S_{i_0} = S_{i_0+1}$, dann $S_{i_0+h} = S_{i_0}\ \forall h \in \mathbb{N}$.
3. Gilt für ein i_0: $S_{i_0+j} = S_{i_0}$ $j > 1$, dann $\forall r = 0(1)(j-1)$ ⇒ Man weiß S_{i_0+j} die S_i berechnen
4. Gilt für ein S_{i_0} $c \cap S_{i_0} \neq \{\}$ dann ist der Code nicht decodierbar.

Beispiel: Es sei $c = \{010, 11000, 01000, 00110\}$. Man startet mit $c = S_0$
1. Berechnung v. S_1: $\hat{S}_1 = \{w \in B^+ \mid \exists\, w_1 \in c \text{ und } \exists\, w \in S_0 \text{ mit } w_2 = w_1 w\}$
$w_1 = 010 \Rightarrow w_2 = 01000 \Rightarrow w = 00$
$w_1 = 11000 \Rightarrow \nexists\, w_2$
$w_1 = 01000 \Rightarrow \nexists\, w_2$
$w_1 = 00110 \Rightarrow \nexists\, w_2$
$\Rightarrow \hat{S}_1 = \{00\}$
$\tilde{S}_1 = \hat{S}_1$ weil $S_0 = c$ ⇒ $S_1 = \hat{S}_1 \cup \tilde{S}_1 = \{00\}$
2. Berechnung v. S_2: $\hat{S}_2 = \{w \in B^+ \mid \exists\, w_1 \in c \text{ und } \exists\, w_2 \in S_1 \text{ mit } w_2 = w_1 w\}$
$\tilde{S}_2 = \{w \in B^+ \mid \exists\, w_1 \in S_1 \text{ und } \exists\, w_2 \in c, w_2 = w_1 w\}$
$w_1 = 010 \Rightarrow \nexists\, w_2$
$w_1 = 11000 \Rightarrow \nexists\, w_2$
$w_1 = 01000 \Rightarrow \nexists\, w_2$
$w_1 = 00110 \Rightarrow \nexists\, w_2$
$\Rightarrow \hat{S}_2 = \{\}$
$\tilde{S}_2$: $w_1 = 00 \Rightarrow w_2 = 00110 \Rightarrow w = \{110\} \Rightarrow S_2 = \hat{S}_2 \cup \tilde{S}_2 = \{110\}$
3. Berechnung v. S_3: ... ⇒ $S = \{00, 110\}$ ⇒ $c \cap S = \{\}$

Ein nicht decodierbarer Code lässt sich häufig durch Hinzufügen eines passenden Präfix oder Suffix in einen decodierbaren Code umwandeln. Hierbei wird aber die Wortlänge verändert.
Beispiel: $a_1 = 0$, $a_2 = 010$, $a_3 = 01$, $a_4 = 10$ ⇒ a_4 Suffix v. a_2, a_1 Suffix v. a_3, a_4, a_1 Präfix v. a_2, a_3
Es gilt $a_1 a_4 = a_3$ ⇒ Code nicht decodierbar.
Man hänge an jedes Codewort die Endung 01 an
⇒ $a_1 = 001$, $a_2 = 01001$, $a_3 = 0101$, $a_4 = 1001$ ⇒ Präfixcode.

Satz von Ungleichung von McMillan:
Es sei B ein Alphabet mit $q = |B| \geq 2$. Sei c ein Code über B^* mit Codeworten $a_i \in B^*$
Die Codeworte a_i haben die Wortlängen n_i.
Der Code sei eindeutig decodierbar. Dann gilt $z = \sum_{i=1}^{n} q^{-n_i} \leq 1$ für die Codeworte $(a_1, \dots, a_n)$
Dieses ist eine notwendige Bedingung für die Decodierbarkeit
Beweis:
Es sei nun $n_1 \leq n_2 \leq \dots \leq n_n = L$, $z = \sum_{i=1}^{n} q^{-n_i}$. $z^2 = z \cdot z = \sum_{i_0=1}^{n} q^{-n_{i_0}} \cdot \sum_{i_1=1}^{n} q^{-n_{i_1}} = \sum_{k=i_0+i_1}^{2L} q^{-k} f(2,k)$
$= \sum_{k=1}^{2L} q^{-k} f(2,k)$ mit $f(2,k) = 0$ für $k < i_0 + i_1$ ⇒ $f(2,k) \leq q^k$
Dieses ist eine obere Schranke über die maximale Anzahl von verschiedenen Worten.
$z^r \leq \sum_{k=0}^{rL} q^{-k} q^k$, $(z)^r = \underbrace{z \cdot \dots \cdot z}_{r} = \sum_{k=0}^{rL} q^{-k} f(r,k)$, $k = n_{i_0} + n_{i_1} + \dots + n_{i_r}$.
$f(r,k) \leq q^k$ maximaler Unterschied, d.h. hier Unterschied an 1 jeder Zeichen
$z = \sqrt[r]{z^r} \leq \sqrt[r]{r \cdot L}$, $z \leq \lim_{r \to \infty} \sqrt[r]{rL} = 1 \Rightarrow z \leq 1$

Satz: Ungleichung v. Kraft
Es sei B ein Alphabet mit $q = |B| \geq 2$. Es seien $n_1, \dots, n_m$ m natürliche Zahlen.
und es gelte $\sum_{i=1}^{m} q^{-n_i} \leq 1$.
Dann ex. ein Präfixcode mit Codeworten v. B^* und den Codewortlängen n_i.
Beweis:
$\max_{i=1(1)m} n_i = L$, $\mu(i)$ = Anzahl aller n_i mit $j = n_i$. $z = \sum_{i=1}^{m} q^{-n_i} = \sum_{j=1}^{L} \mu(j) q^{-j} \leq 1 \mid \cdot q^L$
$\sum_{j=1}^{L} \mu(j) q^{L-j} = \mu(L) q^{L-L} + \sum_{j=1}^{L-1} \mu(j) q^{L-j} \leq q^L \Rightarrow 1 \leq \mu(L) \leq q^{L - \sum_{j=1}^{L-1} \mu(j) q^{L-j}}$, $1 < \mu(L) \leq q^L - \sum_{j=1}^{L-1} \mu(j) q^{L-j}$

$0 \le q^l - \sum_{j=1}^{l-1} \mu(j) q^{l-j} \quad :q \Rightarrow 0 \le q^{l-1} - \mu(l-1) q^{l-1-(l-1)} - \sum_{j=1}^{l-2} \mu(j) q^{l-1-j} \quad | + \mu(l-1)$

$0 \le \mu(l-1) < q^{l-1} - \sum_{j=1}^{l-2} \mu(j) q^{l-1-j} \quad : q \Rightarrow \ldots \Rightarrow 0 \le \mu(3) < q^3 - \mu(2) q - \mu(1) q^2 : q$

$0 \le q^2 - \mu(2) - \mu(1) q \quad | + \mu(2) \Rightarrow 0 \le \mu(2) < q^2 - \mu(1) q \quad : q$

$\Rightarrow 0 \le q - \mu(1) \quad | + \mu(1) \Rightarrow \mu(1) \le q$. Nun Codebaum! Wurzel: q Kinder.

$\mu(1)$ dieser Kinder macht man zu Blättern. Hier schreibt man Urbilder der Codewörter Länge 1 rein. Es bleiben $q - \mu(1) > 0$ Kinder, die zu Knoten werden.

Jeder dieser Knoten hat q Kinder. $\Rightarrow$ 2te Stufe: $(q - \mu(1))q$ Kinder

Dann $\mu(2)$ Kinder zu Blättern. Möglich? $\exists \mu(2) \le q^2 - \mu(1) q \Rightarrow \mu(2) \le (q - \mu(1)) q \Rightarrow$ mehr Kinder als Blätter. Man macht $\mu(2)$ Kinder zu Blättern und $(q - \mu(1))q - \mu(2)$ Kinder zu Knoten.

Jeder Knoten hat q Kinder. $\Rightarrow$ 3. Stufe $((q - \mu(1))q - \mu(2)) q$ Kinder $= q^3 - \mu(2) q^2 - \mu(1) q$

Davon $\mu(3)$ Blätter. Möglich. $\mu(3)$ Kinder z. Blättern, Rest Knoten, usw.

Hiermit ist ein Präfixcode konstruiert mit den Wortlängen n_i, $i = 1(1)m$

Anwendung:

Man hat einen eindeutig decodierbaren Code. Dieser Code ist schwer decodierbar. Mit dem Satz von Kraft konstruiert man sich einen Präfixcode mit den gleichen Wortlängen. Dieser Code ist einfach decodierbar

Beispiel:

Die Ungleichung v. McMillan ist nur eine notwendige Bedingung für einen eindeutig decodierbaren Code über B mit $|B| = q = 2$. Es sei $C = \{0\,10, 0\,100, 103\}$ code $\neq$ McMillan genügt: $z = \sum_{i=1}^{3} q^{-n_i} = 2^{-3} + 2^{-4} + 2^{-2} = \frac{7}{16} < 1$. $0\,10, 0\,100 \,|\, 10 = 0\,10\,0\,100\,10$

Informationstheorie von Schlannach

Man betrachte eine Nachrichtenquelle oder auch einfach Quelle.

Die Quelle gibt Signale von sich. Man möchte diese Signale codieren, um sie zu konservieren.

Man unterscheidet zwischen:

a) kontinuierliche Quelle (kont. Signale)

b) diskrete Quelle (diskrete Signale)

Man betrachte endlich viele verschiedene Signale aus diskreten Quellen. $\Rightarrow$ Signalalphabet A.

Um die Auftrittswahrscheinlichkeit der Signale zu erfassen, arbeitet man mit dem Wahrscheinlichkeitsraum $S = (A, p)$.

$p(a)$ für $a \in A$ gibt an (wie häufig (rel. Häufigkeit) Signal a auftritt) mit welcher Wahrschk. ...

Man unterscheidet zwischen:

1) diskrete Quellen mit Gedächtnis. Hier ist p abh. v. vorher & nachher ges. Zeichen.

2) diskrete Quellen ohne Gedächtnis. Hier k. Abhk. v. ges. Zeichen.

Jetzt Quellen ohne Gedächtnis $\Rightarrow$ Wahrsch. auf rel. Häufigkeit

Es gilt nun: $0 \le p(a_i) \le 1 \quad \forall a_i \in A$. $p(a_i) = 1$, falls 100%. $p(a_i) = 0$, falls 0%.

Informationsgehalt eines Zeichens

Der Informationsgehalt eines Zeichens $I(a)$, $a \in A$ ist der Informationszuwachs wenn das Zeichen a aus der Quelle ausgegeben wird. Es gilt:

1) Ist $a \in A$ selten auftretend, dann Inf. gewinn sehr groß, da unerwartet

2) Der Inf. gewinn einer Zeichenkette $I(\tau; a_j; a_k) = \sum_{i=j}^{k} I(a_i)$

3) Der Inf. gewinn eines Zeichens, das jetzt kommen muss, ist 0, da erwartet und eingeplant.

Die einfachste Funktion, die diese Bedingungen an den Inf. gehalt beschr. ist $I(a) = \log\left(\frac{1}{p(a)}\right)$, $a \in A$

$I(a)$ ist bis auf konstanten Faktor eindeutig bestimmt. Es gilt $\frac{1}{(\log_b(x))} = \frac{(\log_c(b))}{(\log_c(x))}$

Beispiel:

In der deutschen Sprache tritt der Buchstabe b mit Wahrscheinlichkeit 0,016 auf.

$\Rightarrow I(b) = \log_2\left(\frac{1}{p(b)}\right) = \log \frac{1}{0{,}016} = 5{,}97$

Def: Mittlere Informationsgehalt

Der mittlere Infgehalt einer Quelle mit $S(A, p)$ ist def. durch:

$$H(S) = \sum_{a \in A} p(a) \log \frac{1}{p(a)} = \sum_{a \in A} p(a) I(a) = -\sum_{a \in A} p(a) \log(p(a))$$

Die Entropie des endlichen Wahrscheinlichkeitsraumes $S = (A, p)$ ist def. durch:

$H(S) = -\sum_{a \in A} p(a) \log(p(a))$ (in Physik Maß für Unordnung eines Systems)

Quellencodierung:

a) Um die Quelle zu erfassen und zu konservieren, später weiter verarbeiten.

b) Quelle zu erfassen und komprimiert abzuspeichern

Jetzt weiter mit b).

Es sei $\varphi : A \to B^*$ ein Code. Die Wortlänge von $\varphi(a)$, $a \in A$ sei $\lambda(\varphi)$
Dann ist die mittlere Wortlänge def. durch $\overline{\lambda}(\varphi, s) = \sum_{a \in A} p(a)\lambda(\varphi)$

Satz: Es gilt: (B Alphabet, $\varphi : A \to B^*$, $|B| = q$)

1) $\overline{\lambda}(\varphi, s) \geq H(s) / \log(q)$
2) $\overline{\lambda}(\varphi, s) = H(s)/\log(q) \Rightarrow p(a) = q^{-\lambda(\varphi(a))} \quad \forall a \in A$

Beweis:
Der Code sei eindeutig decodierbar. $\Rightarrow z = \sum_{a \in A} q^{-\lambda(\varphi(a))} \leq 1$ (McMillan), $M(a) := \frac{q^{-\varphi(a)}}{z}$
Es gilt: $H(s) = \sum_{a \in A} p(a) \log\left(\frac{1}{p(a)}\right) \overset{?}{\geq} \sum_{a \in A} p(a) \log\left(\frac{1}{M(a)}\right) \Leftrightarrow \sum_{a \in A} p(a) \log \frac{1}{p(a)} - \sum p(a) \log\left(\frac{a}{M(a)}\right) \overset{?}{\geq} 0$
$= \sum_{a \in A} p(a) \left(\log\left(\frac{M(a)}{p(a)}\right) - 1\right) \leq 0$