Informations- und Codierungstheorie

Erblasser 1000001 => Was steht hier?
Signal: Änderung physikalischer Größen
Nachricht: Daten (maschinell verarbeitbare Zeichen)
Information: eine mit Bedeutung belegte Nachricht.

Definition 1: Kodierung

Seien Σ und Π zwei endliche Mengen von Symbolen. Eine Codierung c ist eine injektive Abbildung

$$c : \Sigma^+ \Rightarrow \Pi^+.$$

Σ nennt man das Quellenalphabet und Π das Codealphabet.
$\Sigma^+ = \{\sigma_1, \ldots, \sigma_n | n \geq 1, \sigma_i \in \Sigma\}$
$\Pi^+ = \{\pi_1, \ldots, \pi_n | n \geq 1, \pi_i \in \Pi\}$

Beispiel: Ist $\Pi = \{0, 1\}$, so ist c eine binäre Codierung. So wird z. B. bei der 8-Bit ASCII-Codierung einem Zeichen aus Σ in ein Byte $\Pi^8 = \{0, \ldots, 255\}$ zugeordnet.

c muss injektiv sein: $c(u_1) = c(u_2) \rightarrow u_1 = u_2$

Da c injektiv ist, nennt man c auch verlustfrei. Durch die Injektivität lässt sich die Originalnachricht aus der codierten Sequenz eindeutig rekonstruieren. Das ist nicht immer so (verlustbehaftete Codierung, Beispiel?)

Ziel von Codierungen: Übertragung von Nachrichten in Form von Daten, Codierungen können auch zum Zwecke der Komprimierung und Verschlüsselung dienen, meist werden beide Ziele verknüpft.

Beispiel: Lauflängencodierung (RLE)
Sei $\Sigma = \{A, \ldots, Z\}$ und $\Pi = \{0, 1\}$ und $w = KAFFEE$ $\quad c : KAFFEE \rightarrow 1K1A2F2E$

Viele Codierungen sind so beschaffen, dass jedem Element aus Σ eine genau definierte Sequenz von Zeichen aus Π zugeordnet wird. Die Codierungen erfolgen dann zeichenweise. Daher definieren wir:

Definition 2: Code

Sei c eine Codierung. Der von c erzeugte *Code* C ist die Menge

$$C : c(\Sigma) := \{c(\sigma) | \sigma \in \Sigma\}$$

Die Zeichenkette $c(\sigma)$ ist das *Codewort* des Zeichens σ.

D. h. Sei $u = u_0 u_1 \dots u_{n-1}$ ein Wort mit $u \in \Sigma^+$. Dann ist $c(u) = c(u_0 u_1 \dots u_{n-1}) = c(u_0)c(u_1) \dots c(u_{n-1})$ eine zeichenweise Codierung.

Codierung

längenvariabel
(z. B. Huffmann-Codierung)

Blockcodes
(z. B. ASCII-Codierung)

Fehlererkennende und -korrigierende Codes

Paritätscode: Gegeben ist ein Codewort w, z. B.

1 0 0 0 1 0 1 0 | 1 $\Rightarrow$ Erkennung von Einzelfehlern

besser: 11 01 00 00 11 00 11 00

noch besser: 111 000 000 000 111 000 111 000.

$\Downarrow$

010 $\Rightarrow$ Korrektur von Einzelfehlern

Problem: Ziffernvertauschungen werden nicht erkannt.

Der Hamming-Code

Idee: Prüfbits an bestimmter Stelle einfügen und zwar so, dass eine Veränderung eines Bits kein neues Codewort ergibt. $\Rightarrow$ fehlerkorrigierender Blockcode

Hamming-Gewicht: Anzahl der von 0 verschiedenen Stellen ($\hat{=}$ Anzahl 1en)
Hamming-Abstand (Hamming-Distanz) ist ein Maß für die Unterschiedlichkeit von Codewörtern $\Rightarrow$ ist die Anzahl der sich unterscheidenden Stellen zweier Codewörter.

Definition 3: Hamming-Abstand

Sei Σ ein endliches Alphabet sowie $x = (x_1, \dots, x_n)$ und $y = (y_1, \dots, y_n)$ zwei n-lange Wörter aus Σ^n Dann gilt:

$$\Delta(x, y) := |\{j \in \{1, \dots, n\} | x_i \neq x_j\}|$$

Beispiel: 00110 und 10110 $\rightarrow$ 1 oder Hand und Hall $\rightarrow$ 2

Der *Hamming-Abstand eines Codes* ist das Minimum aller Abstände zwischen verschiedenen Wörtern innerhalb eines Codes.

Beispiel: $x = 00110$
>2
$y = 00101$
>3
$x = 01110$

1 } Hamming-Abstand des Codes $= 1$

Aufbau des Hamming-Codes

Idee:	001	
	010	Hamming-Abstand $= 2$
	100	nicht korrigierbar, z.B. 011
	111	aber erkennbar

besser:	01011	
	01100	Hamming-Abstand $= 3$
	10010	korrierbar, z.B. 01111
	10101	

Ratespiel: $\Rightarrow$ Tafel

[7,4]-Hamming-Code
7 Bits $\Rightarrow$ 4 Informationsbits

Der [7,4]-Hamming-Code

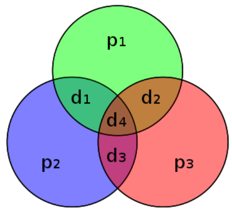
$\vec{x}$	$c(\vec{x})$
0000	0000000
0001	1101001
0010	0101010
0011	1000011
0100	1001100
0101	0100101
0110	1100110
0111	0001111
1000	1110000
1001	0011001
1010	1011010
1011	0110011
1100	0111100
1101	1010101
1110	0010110
1111	1111111

Aufbau: Prüfbits an allen Stellen $2^i, i \in \mathbb{N}^*$

c_1	c_2	c_3	c_4	c_5	c_6	c_7
p_1	p_2	d_1	p_3	d_2	d_3	d_4
2^0	2^1	2^2	2^3	2^4	2^5	2^6

Für die Paritätsbits p_1, p_2, p_3 gilt:

$$p_1 = c_3 \oplus c_5 \oplus c_7$$
$$p_2 = c_3 \oplus c_6 \oplus c_7$$
$$p_3 = c_5 \oplus c_6 \oplus c_7$$



Darstellung der Prüfbits und der zugehörigen Datenbits

Schrittfolge zum Korrigieren von 1-Bit Fehler für den (7,4)-Hamming-Code:

1. Berechnung der Paritätsbits p'_1, p'_2, p'_3 des übertragenen Codewortes w'

2. Verknüpfung der Werte $p'_3 \oplus p_3, p'_2 \oplus p_2, p'_1 \oplus p_1$ und Notierung der Bits (niederwertigstes Bit rechts)

3. Der erhaltenen Binärcode ergibt die Stelle des falschen Bits im Codewort w'

Beispiel: Das Wort $w = 1011010$ wird als $w' = 1011110$ übermittelt.
$p'_1 = c_3 \oplus c_5 \oplus c_7 = 1 \oplus 1 \oplus 0 = 0$
$p'_2 = c_3 \oplus c_6 \oplus c_7 = 1 \oplus 1 \oplus 0 = 0$
$p'_3 = c_5 \oplus c_6 \oplus c_7 = 1 \oplus 1 \oplus 0 = 0$
Berechnung der Fehlerstelle: $p'_3 \oplus p_3, p'_2 \oplus p_2, p'_1 \oplus p_1 = 1 \oplus 0, 0 \oplus 0, 1 \oplus 0 = 101 \Rightarrow$ Fehler an 5. Stelle

Schrittfolge zur Konstruktion von Hamming-Codes mit beliebiger Anzahl von Datenbits?
1. Notiere die Zahlen von $1 \ldots n$ in Binärschreibweise so in eine Tabelle, dass der $i-$te Binärcode in der $i-$ten-Spalte steht.
2. Jeder Spalte mit genau einer 1 ist ein Paritätsbit p_i. Die p_i entsprechen also genau den Zweierpotenz 2
3. Die Berechnung des Wertes von p_i ist die XOR-Verknüpfung der 1 der $i-$ten Zeile eines p_i. Das Paritätsbit p_i wird also über alle Stellen c_j des Codeworts berechnet, in denen an der $i-$ten Stelle der Binärkodierung des Index j eine logische Eins steht.

Aufgabe: Konstruiere den Hammingcode für 11 Datenbits.