

Inhaltsverzeichnis

1 Einführung in Kodierung	1
1.1 Kodierung nach Shannon	1
1.2 Kodierung nach Fano	1
1.3 Kodierung nach Huffman	2
2 Lineare Blockcodes	3
2.1 Generatormatrix	3
2.2 (Paritäts-) Prüfmatrix	3
2.3 Standard-Array	4
3 Zyklische Codes	5
3.1 Polynom-Kodierung	5
3.2 Separierbare zyklische Codes	5
3.3 Prüfmatrix	6
3.4 Kodierung /Dekodierung in Hardware	6
3.5 Eigenschaften von Polynomen	7
4 Lösungs- und Einfügungsfehler	8
4.1 Solide Codes	8
4.2 Informationsrate	8

1 Einführung in Kodierung

1.1 Kodierung nach Shannon

Vorraussetzung:

- S_1, \dots, S_n Symbole der Quelle
- w_1, \dots, w_n Wahrscheinlichkeiten der Symbole

Algorithmus:

1. Ordne Symbole absteigend nach ihren Wahrscheinlichkeiten $w_1 \geq w_2 \geq \dots \geq w_n$
2. Bestimme die Länge l_i des Codewortes für das Symbol w_i mit $-\log_2 w_i \leq l_i < -\log_2 w_i + 1$
3. Ermittle Wahrscheinlichkeit $W_k = \sum_{i=1}^{k-1} w_i$ für jedes Symbol S_k
4. Stelle W_k als $0.a_1a_2\dots a_{l_k}$ binär dar
5. $a_1a_2\dots a_{l_k}$ is Codewort für S_k

Beispiel (Blocklänge 1):

S_k	A	B	C	D
w_k	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{8}$
l_k	1	2	3	3
W_k	0	$\frac{1}{2}$	$\frac{3}{4}$	$\frac{7}{8}$
W_{k_i}	0.0	0.10	0.110	0.111
Code	0	10	110	111

Blocklänge 2:

S_k	AA	AB	AC	AD	BA	...
w_k	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{8}$...
l_k

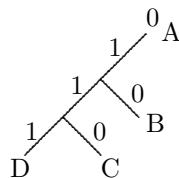
1.2 Kodierung nach Fano

Algorithmus:

1. Ordne Symbole absteigend nach ihren Wahrscheinlichkeiten $w_1 \geq w_2 \geq \dots \geq w_n$
2. Teile Symbole in 2 möglichst gleichwahrscheinliche Gruppen
3. Ordne jeder Gruppe eines der beiden Codesymbole zu
4. Falls die Untergruppen noch nicht einelementig sind: gehe zu 2

Beispiel:

S_k	A	B	C	D
w_k	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{8}$
	0	1		
		0	1	
			0	1
Code	0	10	110	111



1.3 Kodierung nach Huffman

Algorithmus:

1. Ordne Symbole absteigend nach ihren Wahrscheinlichkeiten $w_1 \geq w_2 \geq \dots \geq w_n$
2. Die beiden Symbole mit den kleinsten Wahrscheinlichkeiten werden durch eine binäre Variable unterschieden und gedanklich zu einem neuen Symbol zusammengefasst, dem die Summe der Wahrscheinlichkeiten der beiden zusammengefassten Zeichen zugeordnet wird.
3. Sind mehr als 2 Zeichen übrig: gehe zu 2

Beispiel:

1) 0.22	1) 0.22	1) 0.22	1) 0.22	13) 0.26
2) 0.19	2) 0.19	2) 0.19	2) 0.19	1) 0.22
3) 0.15	3) 0.15	3) 0.15	12) 0.18	2) 0.19
4) 0.12	4) 0.12	11) 0.14	3) 0.15	12) 0.18 14)
5) 0.08	10) 0.10	4) 0.12	11) 0.14 13)	3) 0.15
6) 0.07	5) 0.08	10) 0.10 12)	4) 0.12	
7) 0.07	6) 0.07 11)	5) 0.08		
8) 0.06 10)	7) 0.07			
9) 0.04				
14) 0.33	15) 0.41	16) 0.59	Ende	
13) 0.26	14) 0.33 16)	15) 0.41		
1) 0.22 15)	13) 0.26			
2) 0.19				

2 Lineare Blockcodes

Definition 2.1 Ein Blockcode der Länge n mit k Informationsbits heißt linearer (n, k) Kode, wenn zwei beliebige Kodewörter v, v' summiert ($v \oplus v'$) Kodewort sind.

Definition 2.2 Besteht ein Kodewort aus unverändertem Informationsteil (k -Bit) und $(n-k)$ Bit hinzugefügtem Prüfteil, spricht man von separierbaren (oder systematischen) Blockcodes.

2.1 Generatormatrix

Vorraussetzungen:

- $u = u_1 u_2 \dots u_k$ Informationswort
- $v = v_1 v_2 \dots v_n$ Kodewort
- $C(u) = C(u') \Leftrightarrow u = u'$ C Codierungsfunktion mit $v = C(u)$

Wähle beliebig g_0, \dots, g_{k-1} linear unabhängige Kodewörter aus. Jedes Kodewort lässt sich als Linearkombination $v = u_0 g_0 + u_1 g_1 + \dots + u_{k-1} g_{k-1}$ darstellen oder:

$$G = \begin{pmatrix} g_0 \\ \vdots \\ g_{k-1} \end{pmatrix} = \begin{pmatrix} g_{0,0} & \cdots & g_{0,n-1} \\ \vdots & & \vdots \\ g_{k-1,0} & \cdots & g_{k-1,n-1} \end{pmatrix}$$

Beispiel:

$$G = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \quad \begin{aligned} u &= 1101 \\ v &= u \cdot G \\ &= 1 \cdot g_0 + 1 \cdot g_1 + 0 \cdot g_2 + 1 \cdot g_3 \\ &= 1101000 + 0110100 + 1010001 \\ &= 0001101 \end{aligned}$$

Die Generatormatrix separierbarer Blockcodes enthält die Einheitsmatrix in der Form $G = (M | E)$ oder $G = (E | M)$.

2.2 (Paritäts-) Prüfmatrix

Es gilt $G \cdot H^T = 0$. Ist $G = \begin{pmatrix} p_{0,0} & \cdots & p_{0,n-k-1} & 1 & \cdots & 0 \\ \vdots & & \vdots & \vdots & & \vdots \\ p_{k-1,0} & \cdots & p_{k-1,n-k-1} & 0 & \cdots & 1 \end{pmatrix}$

dann ist $H = \begin{pmatrix} 1 & \cdots & 0 & p_{0,0} & p_{1,0} & \cdots & p_{k-1,0} \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & \cdots & 1 & p_{0,n-k-1} & p_{1,n-k-1} & \cdots & p_{k-1,n-k-1} \end{pmatrix}$

Kurz: $G = (M | E) \Rightarrow H = (E | M^T)$

Beispiel:

$$G = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \Rightarrow H = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix}$$

Datenübertragung:

$$\begin{array}{lcl}
 u \text{ Informationswort} & \xrightarrow{\text{Kodierung}} & v = u \cdot G \quad \text{Kodewort} \\
 & \xrightarrow{\text{Kanal}} & r = v + e \quad \text{Fehler} \\
 & \xrightarrow{\text{Pruefung}} & s = r \cdot H^T \quad \text{Syndrom}
 \end{array}$$

Fehlererkennung/ Fehlerkorrektur

- Wenn r Kodewort ist, so ist $s = \vec{0}$.
- Wenn $s = \vec{0}$, dann ist r Kodewort.
- Wenn der Fehler nicht erkannt wird, muß e Kodewort sein. Somit gibt es $2^k - 1$ Fehler, die nicht erkannt werden können (Prozentsatz $\frac{2^k - 1}{2^n - 1}$).

Definition 2.3 Das Hamminggewicht eines Kodewortes $v = v_0v_1\dots v_{n-1}$ ist die Anzahl seiner Einsen: $w(v) = \sum_{i=0}^{n-1} v_i$. Die Hammingdistanz $d(u, v)$ ist die Anzahl der Komponenten, in denen sich u und v unterscheiden: $d(u, v) = w(u \oplus v)$.

Definition 2.4 Der minimale Hammingabstand eines Codes ist die minimale Distanz zwischen allen Paaren von Kodewörtern: $d_{\min} = \min(d(u, v) \mid u, v \in C, u \neq v)$.

Definition 2.5 Ist die Summe von l Spalten von $H = \vec{0}$, so gibt es einen Kodevektor mit dem Gewicht l . Der minimale Hammingabstand ist gleich dem minimalen Gewicht seiner Kodewörter.

Definition 2.6 Ein Blockcode mit der minimalen Distanz d_{\min} kann alle Fehler e mit $w(e) < d_{\min}$ erkennen und alle Fehler e mit $w(e) = t$ und $2t + 1 \leq d_{\min} \leq 2t + 2$ korrigieren.

2.3 Standard-Array

Algorithmus:

1. Die 2^k Kodevektoren bilden die erste Zeile des Standardarrays.
2. Aus den verbliebenen $2^n - 2^k$ Vektoren wird ein Vektor e_1 mit minimalem Gewicht ausgewählt und die zweite Zeile gewonnen, indem e_1 zu jedem Kodewort (erste Zeile) addiert wird.
3. Aus den verbliebene Vektoren wird ein Vektor e_2 mit minimalem Gewicht gewählt und zu allen Kodewörtern addiert.
4. Continue 2^{n-k} mal, bis alle 2^n Vektoren aufgebraucht sind.

Beispiel:

$$G = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix}$$

Insgesamt ergeben sich $2^5 = 32$ Vektoren = 8 Spalten x 4 Zeilen:

00000	10011	01010	00101	11001	10110	01111	11100
00001	10010	01011	00100	11000	10111	01110	11101
00010	10001	01000	00111	11011	10100	01101	11110
10000	00011	11010	10101	01001	00110	11111	01100

3 Zyklische Codes

Definition 3.1 Ein linearer (n, k) -Blockcode heißt zyklischer Code, wenn jede zyklische Verschiebung eines Kodewortes wieder Kodewort ist, d.h. $c_0c_1\dots c_{n-1} \in C \rightarrow c_{n-1}c_0c_1\dots c_{n-2} \in C$.

3.1 Polynom-Kodierung

Vorraussetzung:

- $x_0x_1\dots x_{k-1}$ Mitteilungswort
- $g_0g_1\dots g_{n-k}$ Koeffizienten
- $v_0v_1\dots v_{n-1}$ Kodewort

$$\text{Generatormatrix: } G = \begin{pmatrix} g_0 & g_1 & \cdots & g_{n-k} & 0 & 0 & \cdots & 0 \\ 0 & g_0 & g_1 & \cdots & g_{n-k} & 0 & \cdots & 0 \\ \vdots & & & & & & & \vdots \\ 0 & 0 & \cdots & 0 & g_0 & g_1 & \cdots & g_{n-k} \end{pmatrix}$$

Kodierung als Polynom:

- $X(Z) = x_0z^{k-1} + x_1z^{k-2} + \dots + x_{k-1}z^0$
- $G(Z) = g_0z^{n-k} + g_1z^{n-k-1} + \dots + g_{n-k}z^0$
- $V(Z) = v_0z^{n-1} + v_1z^{n-2} + \dots + v_{n-1}z^0$
- $V(z) = G(z) \cdot X(Z)$

3.2 Separierbare zyklische Codes

Ziel:

- Kodepolynom ohne Rest durch Generatorpolynom teilbar
- Informationsstellen sind von Kontrollstellen getrennt

Algorithmus:

1. Transformation des Nachrichtenpolynoms $X(z)$ von Grad $k-1$ in das Polynom $X(z) \cdot z^{n-k}$ vom Grad $n-1$.
2. Division von $X(z) \cdot z^{n-k} : G(z)$, Rest $R(z)$ Polynom vom Grad $n-k-1$.
3. Kodierung gemäß $V(z) = X(z) \cdot z^{n-k} + R(z)$.

Beispiel:

- $G(z) = z^3 + z + 1, X(z) = z^3 + 1 \sim 1001$
- $k = 4, n - k = 3, n = 7$
- $X(z) \cdot z^{n-k} = (z^3 + 1) \cdot z^3 = z^6 + z^3$
- $R(z) = (X(z) \cdot z^{n-k}) \bmod G(z) = (z^6 + z^3) \bmod (z^3 + z + 1) = z^2 + z$
- $V(z) = X(z) \cdot z^{n-k} + R(z) = (z^6 + z^3) + (z^2 + z) = z^6 + z^3 + z^2 + z \sim 1001110$

3.3 Prüfmatrix

Algorithmus:

1. $V(z) = v_0 z^{n-1} + v_1 z^{n-2} + \dots + v_{n-2} z + v_{n-1}$
2. $V(z) \bmod G(z) = v_0 z^{n-1} \bmod G(z) + v_1 z^{n-2} \bmod G(z) + \dots + v_{n-2} z \bmod G(z) + v_{n-1} \bmod G(z) = 0$

Beispiel:

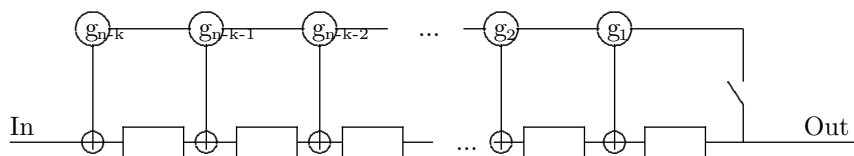
- $G(z) = z^3 + z + 1$
- $V(z) = v_0 z^6 + v_1 z^5 + v_2 z^4 + v_3 z^3 + v_4 z^2 + v_5 z + v_6$
 - $z^6 \bmod (z^3 + z + 1) = z^2 + 1$
 - $z^5 \bmod (z^3 + z + 1) = z^2 + z + 1$
 - $z^4 \bmod (z^3 + z + 1) = z^2 + z$
 - $z^3 \bmod (z^3 + z + 1) = z + 1$
 - $z^2 \bmod (z^3 + z + 1) = z^2$
 - $z \bmod (z^3 + z + 1) = z$
 - $1 \bmod (z^3 + z + 1) = 1$
- $0 = v_0(z^2 + 1) + v_1(z^2 + z + 1) + v_2(z^2 + z) + v_3(z + 1) + v_4 z^2 + v_5 z + v_6$
- $\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} = v_0 \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} + v_1 \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} + v_2 \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} + v_3 \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} + v_4 \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + v_5 \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} + v_6 \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$
- $\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \cdot (v_0 \ v_1 \ v_2 \ v_3 \ v_4 \ v_5 \ v_6)^T$

3.4 Kodierung / Dekodierung in Hardware

Polynom-Multiplikation



Polynom-Division



3.5 Eigenschaften von Polynomen

Definition 3.2 Ein Polynom $P(z)$ heißt reduzibel, wenn es sich als $P(z) = P_1(z) \cdot P_2(z)$, Grad $P_1, P_2 \geq 1$ darstellen läßt. Irreduzibel sonst.

Definition 3.3 Polynome $P_1(z), P_2(z)$ liegen in derselben Restklasse modulo $M(z)$, wenn $P_1(z)$ und $P_2(z)$ den gleichen Rest bei der Division durch $M(z)$ ergeben. Es gibt 2^r verschiedene Restklassen modulo $M(z)$, wenn $M(z)$ den Grad r hat.

Definition 3.4 Die Restklassen modulo $M(z)$, $M(z)$ irreduzibel, bilden einen endlichen Körper bzgl. der komponentenweisen Addition der Polynome und der Multiplikation modulo $M(z)$. Die Anzahl der Elemente ist 2^r . Der Körper heißt $GF(2^r)$.

Definition 3.5 Die Periode eines Modularpolynoms $M(z)$ vom Grad r ist die kleinste Zahl $p > 0$, für die gilt $1 = z^p \pmod{M(z)}$.

Definition 3.6 Ein Polynom $M(z)$ vom Grad r , das die maximale Periode $2^r - 1$ hat, heißt primitiv.

Definition 3.7 Ein Element $a \in GF(2^r)$ ist Wurzel des Polynoms $P(z)$, wenn $P(a) = 0$ gilt. Ist a Wurzel, so ist auch a^2 Wurzel.

Definition 3.8 Das Minimalpolynom $m_a(z)$ des Elements $a \in GF(2^r)$ ist das Polynom kleinsten Grades, das a als Wurzel hat, $m_a(a) = 0$. Ist $a \in GF(2^r)$, so ist der Grad von $m_a(z)$ höchstens r .

Beispiel:

- $M(z) = z^3 + z + 1$, $a = 011 = z + 1$ Modularpolynom, $GF(2^3)$
- $0 = a^0 + b_1 a^1 + b_2 a^2 + b_3 a^3$
- $0 = 001 + b_1(011) + b_2(101) + b_3(100)$
 - $0 = b_2 + b_3 \Rightarrow b_2 = 1$
 - $0 = b_1$
 - $0 = 1 + b_1 + b_2 \Rightarrow b_3 = 1$
- $m_a(z) = 1 + z^2 + z^3$

4 Lösungs- und Einfügungsfehler

4.1 Solide Codes

Vorraussetzungen:

- Alphabet X , endlich, $|X| \geq 2$, $a, b \in X$
- $C \subseteq X^+$ Menge von Codewörtern
- $w \in X^+$ empfangene Nachricht

Definition 4.1 Eine C -Zerlegung von w ist eine Folge $u_0, v_1, u_1, v_2, u_2, \dots, v_n, u_n$ mit

- $v_1, \dots, v_n \in C$, $u_0, \dots, u_n \in X^*$
- $w = u_0, v_1, u_1, v_2, u_2, \dots, v_n, u_n$
- in den u_i kommt kein $x \in C$ als Faktor vor ($u_i \notin X^*CX^*$)

Definition 4.2 C heißt solider Code, wenn jedes $w \in X^+$ genau eine C -Zerlegung hat.

Definition 4.3

- $u, v \in X^+$ überlappen sich, wenn es $x, y, z \in X^+$ gibt, so daß $u = xy$ und $v = yz$ ist.
- C ist überlappungsfrei, wenn keine $u, v \in C$ sich überlappen.
- C ist Infixcode. falls für alle $u, v \in C$ aus $xuy = v$ folgt $x = y = \epsilon$ (und damit $u = v$).

Definition 4.4 $C \subseteq X^+$, $C \neq \emptyset$ ist genau dann ein solider Code, wenn C ein überlappungsfreier Infixcode ist.

4.2 Informationsrate

Definition 4.5 Die Informationsrate r eines systematischen Blockcodes C der Länge n ist $r = \frac{\log_2 |C|}{\log_2 2^n} = \frac{\log_2 |C|}{n}$. Beim linearen (n, k) -Code ist das $\frac{n}{k}$.

Definition 4.6 Bei endlichen Codes $L \subseteq X^+$ ist die Informationsrate $r(L) = \frac{\log_2 |L|}{\frac{s(L)}{|L|}} = \frac{|L| \cdot \log_2 |L|}{s(L)}$ mit $s(L) = \sum_{w \in L} |w|$ (Summe aller Wortlängen).

Definition 4.7 Die minimale Informationsrate $\hat{r}(L)$ ist $\hat{r}(L) = \frac{\log_2 |L|}{\hat{s}(L)}$ mit $\hat{s}(L) = \max(|w| \mid w \in L)$.

Definition 4.8

- $0 \leq \hat{r}(L) \leq r(L)$
- $1 - r(L)$ ist die symbolweise Redundanz von L
- $1 - \hat{r}(L)$ ist die symbolweise maximale Redundanz von L