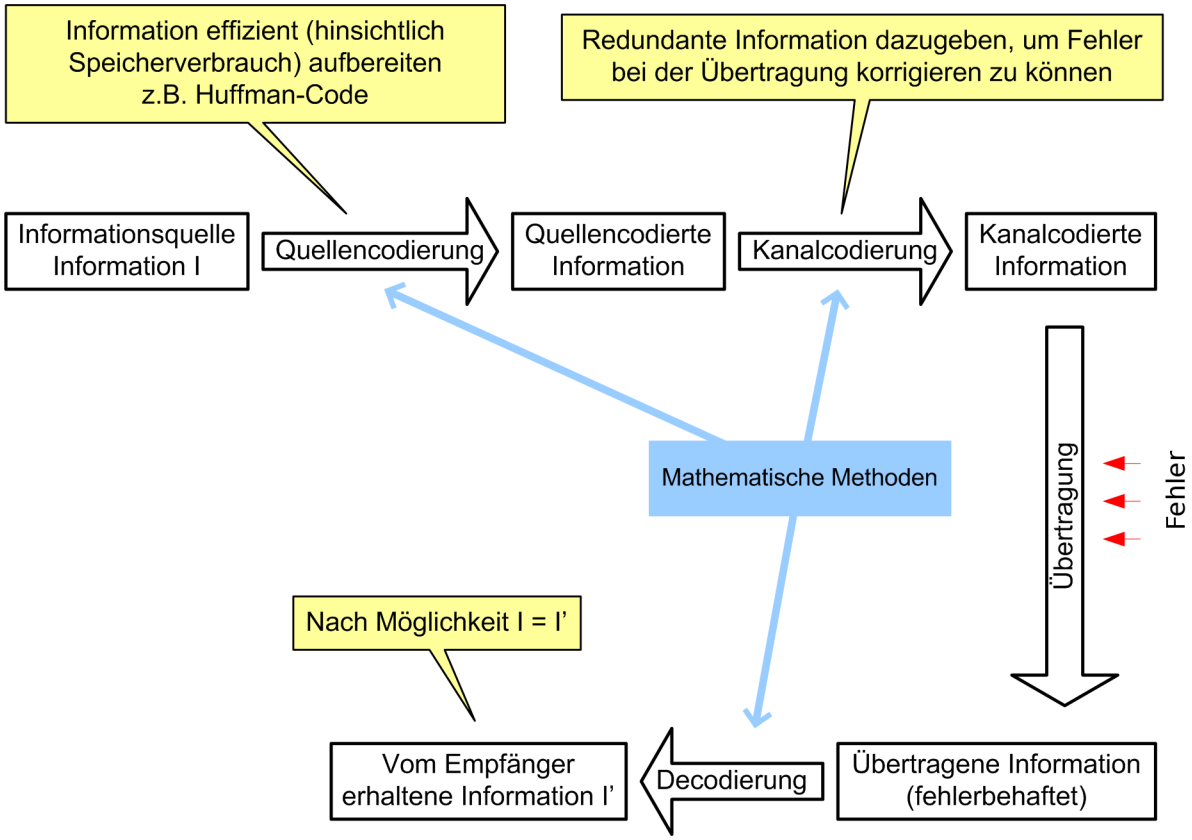
Fehlerkorrigierende Codes

2023S

Gerhard Dorfer


Information effizient (hinsichtlich Speicherverbrauch) aufbereiten z.B. Huffman-Code
Redundante Information dazugeben, um Fehler bei der Übertragung korrigieren zu können
Informationsquelle Information I
Quellencodierung
Quellencodierte Information
Kanalcodierung
Kanalcodierte Information
Mathematische Methoden
Übertragung
Fehler
Nach Möglichkeit I = I'
Vom Empfänger erhaltene Information I'
Decodierung
Übertragene Information (fehlerbehaftet)

Inhaltsverzeichnis

1 Einführende Beispiele

1. Parity check code

Der Code besteht aus (z.B.) 8-stelligen Wörtern über dem Alphabet $\{0,1\}$. An die 7 Nachrichtenbits $a_1a_2a_3\dots a_7$ wird ein Kontrollbit a_8 angehängt, sodass $a_1+a_2+a_3+\dots+a_7+a_8 \equiv 0 \bmod 2$ gilt, d.h. gerade Parität im Wort $a_1\dots a_8$. Wird z.B. beim Abspeichern von Daten verwendet.

Der Code kann einen Bitfehler (oder allgemeiner: eine ungerade Anzahl von Bitfehlern) erkennen. Der Code kann keinen Fehler korrigieren.

2. ISBN-10 Code

Internationale **S**tandard **B**uch**n**ummer; ISBN-10 war die Standardcodierung für Bücher bis zum Jahr 2007, wurde durch ISBN-13 abgelöst, der wie der EAN-Code arbeitet (siehe 3.).

ISBN-10 Codewörter haben 10 Stellen $a_1a_2a_3a_4a_5a_6a_7a_8a_9a_{10}$.

$a_1,\dots,a_9 \in \{0,\dots,9\}$, $a_{10} \in \{0,1,\dots,10\}$, wobei 10 als X codiert wird.

Bedeutung der einzelnen Stellen:
$$\begin{cases} a_1 & \text{Sprache, z.B. deutsch} = 3 \\ a_2\dots a_4 & \text{Verlag} \\ a_5\dots a_9 & \text{Kennnummer für das Buch} \\ a_{10} & \text{Prüfstelle} \end{cases}$$

a_{10} wird so gewählt, dass

$$10\cdot a_1 + 9\cdot a_2 + 8\cdot a_3 + \dots + 1\cdot a_{10} \equiv 0 \bmod 11 \tag{1}$$

Das ist gleichbedeutend mit $1\cdot a_1 + 2\cdot a_2 + \dots + 10\cdot a_{10} \equiv 0 \bmod 11$, denn es gilt $10\cdot a_1 + 9\cdot a_2 + 8\cdot a_3 + \dots + 1\cdot a_{10} = (-1)\cdot a_1 + (-2)\cdot a_2 + (-3)\cdot a_3 + \dots + (-10)\cdot a_{10} \bmod 11$, bzw. wenn man nach a_{10} auflöst: $a_{10} \equiv (1\cdot a_1 + 2\cdot a_2 + \dots + 9\cdot a_9) \bmod 11$.

Die linke Seite von (1) nennt man Prüfsumme, die Gleichung (1) Kontrollgleichung des ISBN-10 Codes.

Der ISBN-10 Code kann Einfachfehler (das ist ein einzelner Fehler an einer nicht bekannten Stelle) erkennen.

Weiters kann er Vertauschungen von 2 Stellen erkennen:
Annahme: $a_1a_2\dots a_{10}$ ist im Code, d.h. $10a_1 + 9a_2 + 8a_3 + \dots + 1a_{10} \equiv 0 \bmod 11$. Die Stelle a_i wird nun mit der Stelle a_j $(i<j)$ vertauscht. Wir nehmen jetzt an, dass der Code diese Vertauschung nicht erkennen kann, d.h. das entstehende Wort $a_1a_2\dots a_{i-1}a_ja_{i+1}\dots a_{j-1}a_ia_{j+1}\dots a_{10}$ ist wieder im Code. Dann muss gelten:
$1a_1+2a_2+\dots+(i-1)a_{i-1}+ia_j+(i+1)a_{i+1}+\dots+(j-1)a_{j-1}+ja_i+(j+1)a_{j+1}+\dots+10a_{10} \equiv 0 \bmod 11$
Subtrahieren von
$1a_1+2a_2+\dots+(i-1)a_{i-1}+ia_i+(i+1)a_{i+1}+\dots+(j-1)a_{j-1}+ja_j+(j+1)a_{j+1}+\dots+10a_{10} \equiv 0 \bmod 11$
ergibt $i(a_j-a_i)+j(a_i-a_j) \equiv 0 \bmod 11$ oder $(j-i)(a_i-a_j) \equiv 0 \bmod 11$,
d.h. 11 teilt $j-i$ oder 11 teilt a_i-a_j (wenn eine Primzahl ein Produkt teilt, muss es auch einen der Faktoren teilen). $j-i$ scheidet aus (da $j-i \in \{1,\dots,9\}$), d.h. 11 teilt a_i-a_j. Da $a_i-a_j \in \{0,\dots,10\}$ ist, ist das nur für $a_i-a_j=0 \Leftrightarrow a_i=a_j$ erfüllt, in diesem Fall ändert die Vertauschung die ISBN aber nicht.

Wichtig ist hier, dass der Modul eine Primzahl ist und dass alle Koeffizienten verschieden sind.

Der Code kann Einfachfehler nicht korrigieren. Ist bei einem Einfachfehler die Fehlerstelle bekannt, so spricht man von einer Auslöschung. Der ISBN-10 kann eine Auslöschung korrigieren, da aus der Kontrollgleichung jede Stelle a_i mit Hilfe der übrigen Stellen ausgerechnet werden kann.

3. EAN-Code

Europäische Artikelnummer

13 Stellen $a_1a_2 \dots a_{13}$, $a_i \in \{0, \dots, 9\}$

Bedeutung der einzelnen Stellen: $\begin{cases} a_1a_2a_3 & \text{Länderpräfix, z.B. Österreich} = 900 \text{ und } 919 \\ a_4 \dots a_i & \text{Hersteller}, i \in \{7, 8, 9\} \\ a_{i+1} \dots a_{12} & \text{Produkt} \\ a_{13} & \text{Prüfstelle} \end{cases}$

Die Prüfstelle a_{13} wird so gewählt, dass

$$a_1 + 3 \cdot a_2 + a_3 + 3 \cdot a_4 + \dots + 3 \cdot a_{12} + a_{13} \equiv 0 \bmod 10 \tag{2}$$

gilt, woraus folgt $a_{13} \equiv -a_1 - 3a_2 - a_3 - \dots - 3a_{12} \bmod 10$.

Der Code kann Einfachfehler erkennen.

Weiters kann der Code phonetische Fehler erkennen: damit sind z.B. Fehler der Form $40 \to 14$, $30 \to 13$ gemeint, allgemein: $a0 \to 1a$ für alle $a \in \{2, 3, \dots, 9\}$.

Der Code kann keinen Fehler korrigieren.

4. 3-fach Wiederholungscode für binäre Nachrichtenwörter der Länge 2

$00 \to 00\,00\,00$
$10 \to 10\,10\,10$
$01 \to 01\,01\,01$
$11 \to 11\,11\,11$

z.B. Nachricht $N = \underline{01}\,\underline{11}\,00\,10 \dots$. Codiert: $\underline{010101}\,\underline{111111}\,000000\,101010 \dots$

Wenn bei der Übertragung eines codierten Blockes mit 6 Bit genau 1 Fehler gemacht wird, dann kann dieser Fehler richtig korrigiert werden. Der Empfänger sieht dazu nach, welcher Zweierblock doppelt vorkommt. Beispiel: gesendet wird $xy\,xy\,xy$, empfangen wird $xz\,xy, xy$. Der Block xy kommt doppelt vor, daher wird zu xy decodiert.

Der Code kann Zweifachfehler pro 6 Bit Block immer erkennen, jedoch im Allgemeinen nicht mehr richtig korrigieren. Zweifachfehler, bei denen ein Fehler an einer geraden Stelle und der andere an einer ungeraden Stelle auftritt, können richtig korrigiert werden. Dazu werden die Mehrheitsverhältnisse an den geraden und den ungeraden Stellen betrachtet. Beispiel: gesendet wird $xy\,xy\,xy$, empfangen wird $wy\,xz, xy$. Kein Block kommt doppelt vor, aber an den ungeraden Stellen hat x die Mehrheit und an den geraden Stellen y, es wird zu xy decodiert. Wenn jedoch beide Fehler an geraden oder ungeraden Stellen auftreten, wird falsch decodiert. Beispiel: gesendet wird $xy\,xy\,xy$, empfangen wird $zy\,xy, zy$: es wird ein Einfachfehler vermutet und falsch zu zy decodiert.

5. Ein weiterer Code

$00 \to 00\,000$
$10 \to 10\,110$
$01 \to 01\,101$
$11 \to 11\,011$
d.h. es werden 3 Bits angehängt.

Je 2 Codewörter unterscheiden sich an mindestens 3 Stellen. Daraus folgt: wenn bei der Übertragung eines Codewortes nur ein Fehler gemacht wird, dann unterscheiden sich alle anderen

Codeworte vom übertragenen Wort noch an mindestens zwei Stellen. Das gesendete Codewort ist das einzige, das sich nur an einer Stelle vom Empfangswort unterscheidet und kann so vom Empfänger richtig rekonstruiert werden.

Beispiel: 01001 wurde empfangen. Abstand zu den vorhandenen Codewörtern:

00000:	2	
10110:	5	
01101:	1	das ist das einzige Codewort mit Abstand 1, dieses Wort wurde wahrscheinlich gesendet. Siehe Nearest Neighbour Decodierung, Kapitel 3.
11011:	2	

Der Code kann Zweifachfehler nicht mehr richtig korrigieren.

Vergleich zum 3-fach Wiederholungscode: Beide Codes können alle Einfachfehler korrigieren, Zweifachfehler im Allgemeinen nicht mehr.

Informationsrate $R = \frac{\text{Anzahl der Nachrichtenstellen}}{\text{Anzahl der Nachrichtenstellen + Kontrollstellen}}$

Der 3-fach Wiederholungscode ist durch das Anhängen von 4 Kontrollstellen entstanden: $R = \frac{2}{2+4} = 0.333$. Der soeben untersuchte Code ist durch das Anhängen von 3 Stellen entstanden: $R = \frac{2}{2+3} = 0.4$; er ist daher besser/effizienter als der 3-fach Wiederholungscode.

Bemerkung: Durch Anhängen von 2 Kontrollbits an zwei Nachrichtenbits können niemals alle Einfachfehler korrigiert werden.

2 Mathematische Grundlagen

Restklassen

Nach der Quellcodierung liegt die Nachricht als Folge $a_1a_2a_3\dots a_n$ über dem Alphabet A (d.h. $a_i \in A,\ i = 1,\dots,n$) vor.

Wichtigster Fall: $A = \{0,1\}$ - Restklassen modulo 2

$+$	0	1
0	0	1
1	1	0

$\cdot$	0	1
0	0	0
1	0	1

$0 = \overline{0} = \{0, \pm2, \pm4, \pm6, \dots\}$
$1 = \overline{1} = \{1, \pm3, \pm5, \pm7, \dots\}$

Allgemein: Restklassen $\bmod m,\ m \in \mathbb{N},\ m \geq 2$

$\mathbb{Z}_m = \{0, 1, 2, 3, \dots, m-1\}$
$0 = \overline{0} = \{0, \pm m, \pm 2m, \dots\}$
$a = \overline{a} = \{a, a \pm m, a \pm 2m, \dots\} = \{a + k\,m \mid k \in \mathbb{Z}\}$

(die Querstriche der Restklassen werden aus Bequemlichkeit meistens weggelassen)

Rechnen $\bmod m$:

$\overline{a}, \overline{b} \in \mathbb{Z}_m$: $\overline{a} + \overline{b} := \underbrace{(a+b)}_{\in\mathbb{Z}} \bmod m$, $\overline{a} \cdot \overline{b} := \underbrace{(a \cdot b)}_{\in\mathbb{Z}} \bmod m$

Beispiel: $m = 5, \quad \mathbb{Z}_5 = \{0, 1, 2, 3, 4\}$

$+$	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

$\cdot$	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

Beispiel: $m = 6, \quad \mathbb{Z}_6 = \{0, 1, 2, 3, 4, 5\}$

$+$	0	1	2	3	4	5
0	0	1	2	3	4	5
1	1	2	3	4	5	0
2	2	3	4	5	0	1
3	3	4	5	0	1	2
4	4	5	0	1	2	3
5	5	0	1	2	3	4

$\cdot$	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	1	2	3	4	5
2	0	2	4	0	2	4
3	0	3	0	3	0	3
4	0	4	2	0	4	2
5	0	5	4	3	2	1

Wir zeigen, dass die Operationen $+$ und $\cdot$ auf $\mathbb{Z}_m$ wohldefiniert sind:

$\overline{a_1} = \overline{a_2}$, $\overline{b_1} = \overline{b_2}$ z.B. $\bmod 5 : \overline{1} = \overline{6} = \overline{-4}$
$\overline{a_1} + \overline{b_1} = \overline{a_1 + b_1}$, $\overline{a_2} + \overline{b_2} = \overline{a_2 + b_2}$

zu zeigen: $\overline{a_1 + b_1} = \overline{a_2 + b_2}$
$\overline{a_1} = \overline{a_2} \Rightarrow \quad a_1 - a_2 = k\,m, \; k \in \mathbb{Z}$
$b_1 - b_2 = l\,m, \; l \in \mathbb{Z}$
$(a_1 + b_1) - (a_2 + b_2) = (k + l)\,m \quad \Rightarrow \overline{a_1 + b_1} = \overline{a_2 + b_2}$

Beweis für die Multiplikation funktioniert analog.

Beispiel: $\bmod 6$ gilt $\overline{2} = \overline{-4}, \quad \overline{4} = \overline{16}$

Addition: $\overline{2} + \overline{4} = \overline{6} = \overline{0}, \quad \overline{-4} + \overline{16} = \overline{12} = \overline{0}$
Multiplikation: $\overline{2} \cdot \overline{4} = \overline{8} = \overline{2}, \quad \overline{-4} \cdot \overline{16} = \overline{-64} = \overline{-66 + 2} = \overline{2}$

Struktur von $\mathbb{Z}_m$: $(\mathbb{Z}_m, +, \cdot)$ ist ein Ring:

- $(\mathbb{Z}_m, +)$ ist eine kommutative Gruppe (abgeschlossen, assoziativ, kommutativ, neutrales Element 0, inverses Element $-x$)
- $(\mathbb{Z}_m, \cdot)$ ist eine kommutative Halbgruppe mit Einselement (abgeschlossen, assoziativ, kommutativ, neutrales Element 1)
- Es gilt das Distributivgesetz: $x \cdot (y + z) = x \cdot y + x \cdot z$

Zusätzlich gilt, wenn der Modul p prim ist:
$(\mathbb{Z}_p \setminus \{0\}, \cdot)$ ist eine kommutative Gruppe. $(\mathbb{Z}_p, +, \cdot)$ ist dann ein Körper.

Beweis für Abgeschlossenheit ($a, b \neq 0 \Rightarrow a \cdot b \neq 0$):
Angenommen, $a \cdot b = 0 \bmod p$. Das heißt, p teilt $a \cdot b$. Dann muss die Primzahl p aber auch einen der beiden Faktoren a oder b teilen $\Rightarrow a \equiv 0 \bmod p$ oder $b \equiv 0 \bmod p$

Das Berechnen des inversen Elements $a^{-1} \bmod p$ erfolgt mit dem **Euklidischen Algorithmus**.

Beispiel: $m = p = 19$, $a = 12$

$$
\begin{array}{rcll}
19 & = & 1 \cdot 12 & + \; 7 \\
12 & = & 1 \cdot 7 & + \; 5 \\
7 & = & 1 \cdot 5 & + \; 2 \\
5 & = & 2 \cdot 2 & + \; \boxed{1} \\
2 & = & 2 \cdot 1 & + \; 0
\end{array}
$$

$\Rightarrow \text{ggT}(19, 12) = 1$ (nicht überraschend, da jede Zahl zu einer größeren Primzahl teilerfremd ist)

Durch Zurückrechnen von unten nach oben im Euklidischen Algorithmus lässt sich der $\text{ggT}(a, b)$ als Linearkombination von a und b darstellen:

$$1 = 5 - 2 \cdot \underbrace{2}_{7-5} = 5 - 2 \cdot (7-5) = 3 \cdot \underbrace{5}_{12-7} - 2 \cdot 7 = 3 \cdot 12 - 5 \cdot \underbrace{7}_{19-12} = 3 \cdot 12 - 5 \cdot (19-12) = 8 \cdot 12 - 5 \cdot 19$$

$1 = 8 \cdot 12 - 5 \cdot 19 \qquad | \bmod 19$
$1 \equiv (8 \cdot 12 - 5 \cdot 0) \bmod 19$
$1 \equiv 8 \cdot 12 \bmod 19 \Rightarrow 8 = 12^{-1} \bmod 19$

weiteres Beispiel: gesucht ist $7^{-1} \bmod 19$
...
$1 = (3 \cdot 19 - 8 \cdot 7) \bmod 19$
$1 \equiv -8 \cdot 7 \bmod 19$
$7^{-1} \equiv -8 \equiv 11 \bmod 19$

Bemerkung: $(\mathbb{Q}, +, \cdot)$, $(\mathbb{R}, +, \cdot)$ und $(\mathbb{C}, +, \cdot)$ sind Körper
$(\mathbb{Z}, +, \cdot)$ und $(\mathbb{R}[x], +, \cdot)$ sind Ringe ($\mathbb{R}[x]$: Polynome in x mit Koeffizienten aus $\mathbb{R}$)

Wichtige Alphabete im Folgenden: $\mathbb{Z}_p$ (p prim), insbesondere $p = 2$, alle endlichen Körper (siehe Kapitel 8).

Hamming-Distanz

In den Beispielen hatten die Codewörter stets eine feste Länge n über dem Alphabet A. z.B. ISBN-10 Code: $n = 10$ $A = \mathbb{Z}_{11}$; EAN-Code: $n = 13$, $A = \mathbb{Z}_{10}$. Allgemein siehe Blockcodes im nächsten Abschnitt.

$A^n = \{a_1 a_2 \dots a_n \mid a_i \in A\}$ heißt **Sequenzraum**. Der Sequenzraum kann als Menge der möglichen Empfangswörter interpretiert werden, wenn Codewörter über dem Alphabet A der Länge n übertragen werden.

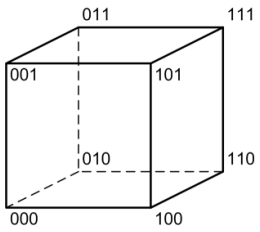
Das Wort $a_1 a_2 \dots a_n$ kann auch als Vektor $(a_1, a_2, \dots, a_n)$ aufgefaßt werden, wenn A ein Körper ist.

Definition: Für $\vec{a} = a_1 a_2 \dots a_n, \vec{b} = b_1 b_2 \dots b_n \in A^n$ ist

$$d(\vec{a}, \vec{b}) = \sharp \{i \in \{1 \dots n\} \mid a_i \neq b_i\}$$

die **Hammingdistanz** von $\vec{a}$ und $\vec{b}$.

Beispiel: $A = \{0, 1\}, n = 3$
$A^3 = \{000, 001, 010, 011, 100, 101, 110, 111\}$



Die Hammingdistanz ist die Anzahl der Kanten zwischen zwei Wörtern.

d ist eine Metrik:

- $d(\vec{a}, \vec{b}) = 0 \Leftrightarrow \vec{a} = \vec{b}$
- $d(\vec{a}, \vec{b}) = d(\vec{b}, \vec{a})$
- $d(\vec{a}, \vec{c}) \leq d(\vec{a}, \vec{b}) + d(\vec{b}, \vec{c})$ Dreiecksungleichung

Blockcodes

Quellencodierte Nachricht $a_1a_2a_3\ldots$ über dem Alphabet A wird in gleich lange Blöcke der Länge k eingeteilt:

$\underbrace{a_1a_2\ldots a_k}_{\text{Länge } k} \mid \underbrace{a_{k+1}a_{k+2}\ldots a_{2k}}_{\text{Länge } k} \mid \underbrace{a_{2k+1}a_{2k+2}\ldots a_{3k}}_{\text{Länge } k} \mid \ldots$

(n,k)-Blockcode: aus einem Nachrichtenblock $a_1a_2\ldots a_k$ der Länge k wird durch eine Codierungsfunktion f_C ein Codewort $c_1c_2\ldots c_n$ der Länge $n, n>k$ erzeugt.

$$f_C: \begin{cases} A^k \to A^n \\ a_1a_2\ldots a_k \to c_1c_2\ldots c_n \end{cases}$$

Damit aus dem Codewort eindeutig das Nachrichtenwort ermittelt werden kann, muss f_C injektiv sein: $\vec{a_1} \neq \vec{a_2} \Rightarrow f_C(\vec{a_1}) \neq f_C(\vec{a_2})$

wichtiger Spezialfall: wenn

$$f_C(a_1a_2\ldots a_k) = \underbrace{a_1a_2\ldots a_k}_{\text{Nachrichtenwort}}\ \underbrace{c_{k+1}\ldots c_n}_{\text{Kontrollstellen}} \quad \text{oder} \quad f_C(a_1a_2\ldots a_k) = \underbrace{c_1\ldots c_{n-k}}_{\text{Kontrollstellen}}\ \underbrace{a_1a_2\ldots a_k}_{\text{Nachrichtenwort}}$$

gilt, dann heißt die Codierung mit f_C **systematisch**. Das Nachrichtenwort kann dann einfach aus dem Codewort durch Streichen der Kontrollstellen abgelesen werden. Ein weiterer Vorteil der systematischen Codierung besteht darin, dass sich Fehler bei der Decodierung eines Empfangswortes beim Übergang vom decodierten Empfangswort zum Nachrichtenwort nicht auch noch verstärken können: Enthält das Codewort, das durch (falsche) Decodierung aus dem Empfangswort entstanden ist, e Fehler, so enthält das zugehörige Nachrichtenwort bei systematischer Codierung $\leq e$ Fehler. Ist die Codierung nicht systematisch, können aus e Fehlern im Codewort beliebig viele Fehler im Nachrichtenwort entstehen.

Die Menge aller möglichen Codeworte $C = \{f_C(a_1\ldots a_k) \mid a_1\ldots a_k \in A^k\}$ nennt man **Code**. $C \subseteq A^n$ (praktisch immer: $C \subset A^n$, da der Code sonst uninteressant wäre), $|C| = |A^k| = |A|^k$.

Beispiel: $(6,4)$-Blockcode über dem Alphabet $A = \{0,1\}$: $|C| = 2^4 = 16$, $|A^n| = 2^6 = 64$

3 Fehlererkennung und Fehlerkorrektur für Blockcodes

Definition. Sei C ein (n,k)-Blockcode. Die **Minimaldistanz** von C ist definiert als

$$d = \min\left\{d(\vec{a},\vec{b}) \mid \vec{a},\vec{b} \in C, \vec{a} \neq \vec{b}\right\}$$

Schreibweise: C ist ein (n,k,d)-Code.

Für $\vec{a} = a_1a_2\ldots a_n \in A^n$ ist $K_t(\vec{a}) := \{\vec{x} \in A^n \mid d(\vec{a},\vec{x}) \leq t\}$ die **Kugelumgebung** von $\vec{a}$ mit Radius t.

Beispiel: $A = \{0,1\},\ n = 3$:

$K_1(000) = \{000, 100, 010, 001\}$
$K_1(111) = \{111, 011, 101, 110\}$

$K_1(000) \cup K_1(111) = A^3$

$K_t(\vec{a})$ enthält alle Wörter, die durch Änderung/Fehler von $\leq t$ Stellen aus $\vec{a}$ entstehen.

Fehlerkorrektur

Ein Code C heißt **t-fehlerkorrigierend**, wenn $K_t(\vec{c_1}) \cap K_t(\vec{c_2}) = \emptyset \quad \forall \vec{c_1}, \vec{c_2} \in C, \vec{c_1} \neq \vec{c_2}$.

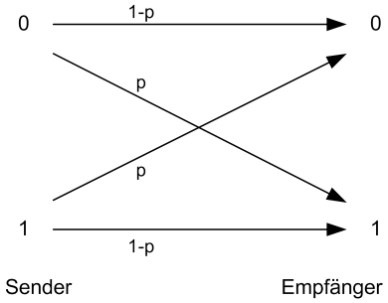
Werden bei $\vec{c} \in C$ bis zu t Stellen verändert, so erhält man $\vec{x} \in K_t(\vec{c}) \Rightarrow \vec{x} \notin K_t(\vec{c_1})$, für alle $\vec{c_1} \in C$, $\vec{c_1} \neq \vec{c}$, also: $d(\vec{x}, \vec{c}) \leq t$ und $d(\vec{x}, \vec{c_1}) > t \; \forall \vec{c_1} \in C$, $\vec{c_1} \neq \vec{c}$.

D.h. $\vec{c}$ ist das eindeutig bestimmte Codewort in C mit kleinstem Hamming-Abstand zu $\vec{x}$.

Wenn der Sender $\vec{c}$ sendet und der Empfänger $\vec{x}$ erhält, so kann der Empfänger eindeutig zu $\vec{c}$ decodieren. Der Empfänger sucht sich dabei immer das Codewort, das zum Empfangswort den kleinsten Hammingabstand hat. Das nennt man **Nearest Neighbour Decodierung**.

Diese Strategie hat folgenden wahrscheinlichkeitstheoretischen Hintergrund:

Wir betrachten der Einfachheit halber einen **Symmetrischen Binärkanal** (q-närer symmetrischer Kanal analog):



Die Werte bei den Pfei len geben die bedingten Übertragungswahrscheinlichkeiten der Buchstaben 0 und 1 an, so ist $p = P(1 \text{ erhalten} \mid 0 \text{ gesendet})$ etc. Eine natürliche Anforderung an die Fehlerwahrscheinlichkeit p ist $p < \frac{1}{2}$ und folglich $p < 1 - p$. p sollte möglichst klein sein. Wäre $p > \frac{1}{2}$, dann kann der Empfänger einfach alle empfangenen Bits umdrehen und erhält wieder Fehlerwahrscheinlichkeit $< \frac{1}{2}$. Ganz schlecht ist es, wenn $p = \frac{1}{2}$. Dann ist keine Information übertragbar.

Die Wahrscheinlichkeit w_t bei der Übertragung eines Wortes der Länge n genau t Fehler an vorgegebenen Stellen zu machen — wobei wir hier voraussetzen, dass die Fehler an den einzelnen Stellen unabhängig voneinander sind (ein Übertragungskanal, der diese Voraussetzung erfüllt, heißt **gedächtnislos**; diese Voraussetzung trifft nicht immer zu, siehe später z.B. bei Compact Disk) — ist gegeben durch

$$w_t = p^t (1-p)^{n-t}.$$

Behauptung: $s > t \Rightarrow w_s < w_t$

$$\begin{array}{rll} p & < 1-p & \mid \quad \ldots^{t-s} \quad (t-s<0 \Rightarrow \text{Ungleichung kehrt sich um}) \\ p^{t-s} & > (1-p)^{t-s} & \mid \quad \cdot p^s (1-p)^{n-t} > 0 \\ p^t (1-p)^{n-t} & > p^s (1-p)^{n-t+t-s} & \\ w_t & > w_s & \end{array}$$

D.h. weniger Fehler sind wahrscheinlicher als mehr Fehler.

Die Nearest Neighbour Strategie entspricht unter den oben angegebenen Voraussetzungen der **Maximum Likelihood** Strategie: darunter versteht man bei gegebenem Empfangswort $\vec{x}$ zu demjenigen Codewort $\vec{c}$ zu decodieren, für das die bedingte Wahrscheinlichkeit $P(\vec{x} \text{ erhalten} \mid \vec{c'} \text{ gesendet})$ maximal unter allen Codewörtern $\vec{c'}$ ist. Es gilt wie zuvor:

$$P(\vec{x} \text{ erhalten} \mid \vec{c'} \text{ gesendet}) = p^{d(\vec{x},\vec{c'})} (1-p)^{n-d(\vec{x},\vec{c'})},$$

und diese Größe ist maximal, wenn $d(\vec{x}, \vec{c'})$ minimal ist.

Die Maximum Likelihood Strategie soll noch mit der Strategie des **Idealen Empfängers** in Beziehung gesetzt werden: Sei $f_C : A^k \to C$ die Codierungsfunktion und $\tilde{f} : A^n \to C$ eine Decodierungsfunktion. Der ideale Empfänger wählt die Decodierungsfunktion $\tilde{f}$ so, dass $P(\tilde{f}(\vec{x})$ gesendet $\mid \vec{x}$ erhalten) für alle $\vec{x} \in A^n$ maximal wird. Dies geschieht im Hinblick darauf, dass die Wahrscheinlichkeit für eine einzelne korrekte Decodierung gegeben ist durch

$$\sum_{\vec{x}\in A^n} P(\tilde{f}(\vec{x}) \text{ gesendet } \wedge\ \vec{x} \text{ erhalten}) = \sum_{\vec{x}\in A^n} P(\tilde{f}(\vec{x}) \text{ gesendet} \mid \vec{x} \text{ erhalten}) \cdot P(\vec{x} \text{ erhalten}).$$

Unter Verwendung der Bayes-Formel erhalten wir

$$P(\vec{c} \text{ gesendet} \mid \vec{x} \text{ erhalten}) = \frac{P(\vec{x} \text{ erhalten} \mid \vec{c} \text{ gesendet}) \cdot P(\vec{c} \text{ gesendet})}{\sum_{\vec{c'}\in C} P(\vec{x} \text{ erhalten} \mid \vec{c'} \text{ gesendet}) \cdot P(\vec{c'} \text{ gesendet})}.$$

Unter der in vielen praktischen Fällen erfüllten Voraussetzung $P(\vec{c'} \text{ gesendet}) = 1/|C|$ vereinfacht sich diese Beziehung zu

$$P(\vec{c} \text{ gesendet} \mid \vec{x} \text{ erhalten}) = \frac{P(\vec{x} \text{ erhalten} \mid \vec{c} \text{ gesendet})}{\sum_{\vec{c'}\in C} P(\vec{x} \text{ erhalten} \mid \vec{c'} \text{ gesendet})}.$$

Da für ein vorgegebenes Empfangswort $\vec{x}$ der Nenner auf der rechten Seite durch die Übertragungswahrscheinlichkeiten des Kanals vorgegeben ist, wird die linke Seite für jene(s) Codewort(e) $\vec{c}$ maximal, für das der Zähler der rechten Seite maximal wird. D.h. der „ideale Empfänger“ verfolgt die Maximum Likelihood Strategie.

Beispiel: $C = \{000, 111\}$ ist 1-fehlerkorrigierend:

$$\left.\begin{array}{l} K_1(000) = \{000, 100, 010, 001\} \\ K_1(111) = \{111, 011, 101, 110\} \end{array}\right\} \text{ disjunkt } \Rightarrow \text{1-fehlerkorrigierend}$$

Alle $\vec{x}$ aus $K_t(\vec{c})$ werden zu $\vec{c}$ decodiert, wenn C t-fehlerkorrigierend ist.

Gegeben: $C \subseteq A^n$. Wie kann man entscheiden, ob C t-fehlerkorrigierend ist? — Die Minimaldistanz $d = \min\{d(\vec{c_1}, \vec{c_2}) \mid \vec{c_1} \neq \vec{c_2}, \vec{c_1} \in C, \vec{c_2} \in C\}$ spielt hierbei die entscheidende Rolle.

Satz: Ein Code C ist genau dann t-fehlerkorrigierend, wenn die Minimaldistanz $d \geq 2t + 1$ ist.
Anders formuliert: C kann bis zu $\lfloor \frac{d-1}{2} \rfloor$ Fehler pro Codewort korrigieren.

Beweis: $\boxed{\Rightarrow}$ C ist t-fehlerkorrigierend. d.h. $K_t(\vec{c_1}) \cap K_t(\vec{c_2}) = \emptyset \quad (\vec{c_1} \neq \vec{c_2}, \vec{c_1} \in C, \vec{c_2} \in C)$.

Angenommen, $d < 2t + 1$ d.h. $d \leq 2t$ bzw. $\lfloor \frac{d}{2} \rfloor \leq \lceil \frac{d}{2} \rceil \leq t$.
Dann gilt: $\exists \vec{c_1}, \vec{c_2} \in C$ $(\vec{c_1} \neq \vec{c_2})$: $d(\vec{c_1}, \vec{c_2}) = d \leq 2t$

$\vec{c_1} = (c_{11}, c_{12}, \ldots, c_{1n})$, $\vec{c_2} = (c_{21}, c_{22}, \ldots, c_{2n})$

Indexmenge $I = \{i_1, \ldots, i_d\} = \{i \mid c_{1i} \neq c_{2i}\}$

sei $\vec{x} = (x_1, x_2, \ldots, x_n)$ definiert durch

$$x_i = \begin{cases} c_{1i} = c_{2i} & i \notin I \\ c_{2i} & i \in \left\{i_1, i_2, \ldots, i_{\lfloor \frac{d}{2} \rfloor}\right\} \\ c_{1i} & i \in \left\{i_{\lfloor \frac{d}{2} \rfloor + 1}, \ldots, i_{d-1}, i_d\right\} \end{cases}$$

$d(\vec{x}, \vec{c_1}) = \lfloor \frac{d}{2} \rfloor \leq t$, $d(\vec{x}, \vec{c_2}) = d - \lfloor \frac{d}{2} \rfloor = \lceil \frac{d}{2} \rceil \leq t$
$\Rightarrow \vec{x} \in K_t(\vec{c_1}) \wedge \vec{x} \in K_t(\vec{c_2})$. Aber $K_t(\vec{c_1}) \cap K_t(\vec{c_2}) = \emptyset$, Widerspruch $\Rightarrow d \geq 2t + 1$.

$\boxed{\Leftarrow}$ Gelte $d \geq 2t + 1$. Angenommen, C ist nicht t-fehlerkorrigierend. Dann gilt: $\exists \vec{c_1} \neq \vec{c_2} \in C$: $K_t(\vec{c_1}) \cap K_t(\vec{c_2}) \neq \emptyset$. Wir wählen ein $\vec{x}$ aus diesem Durchschnitt $\Rightarrow d(\vec{x}, \vec{c_1}) \leq t \wedge d(\vec{x}, \vec{c_2}) \leq t$.

Dreiecksungleichung: $d(\vec{c_1}, \vec{c_2}) \leq \underbrace{d(\vec{c_1}, \vec{x})}_{\leq t} + \underbrace{d(\vec{x}, \vec{c_2})}_{\leq t} \leq 2t$

$\Rightarrow\ 2t+1 \leq d \leq d(\vec{c_1}, \vec{c_2}) \leq 2t$, Widerspruch $\Rightarrow C$ ist t-fehlerkorrigierend.

2. Teil („anders formuliert"): $d \geq 2t+1 \Leftrightarrow t \leq \frac{d-1}{2} \overset{\text{da } t \in N}{\Longleftrightarrow} t \leq \lfloor \frac{d-1}{2} \rfloor$. □

Fehlererkennung

Wir betrachten hier Fehlererkennung in dem Sinne, dass der Empfänger feststellen kann, dass bei der Übertragung ein Fehler passiert ist, d.h. er erhält kein Codewort.

Satz: Ein Code C mit Minimaldistanz d kann alle Fehler an $d-1$ Stellen erkennen, aber nicht alle Fehler an d Stellen.

Beispiel: Code C mit Minimaldistanz $d = 3$:

- kann $\lfloor \frac{d-1}{2} \rfloor = 1$ Fehler korrigieren
- kann $d - 1 = 2$ Fehler erkennen

Code C mit Minimaldistanz $d = 4$:

- kann $\lfloor \frac{d-1}{2} \rfloor = 1$ Fehler korrigieren
- kann $d - 1 = 3$ Fehler erkennen

Unterschied zwischen $d = 3$ und $d = 4$:
Bei $d = 3$ werden manche Zweifachfehler falsch korrigiert: $\vec{c_1} \xrightarrow{\text{2 Fehler}} \vec{x} \to \exists \vec{c_2}$ mit Abstand 1.
Bei $d = 4$ werden Zweifachfehler als solche erkannt: $\vec{c_1} \xrightarrow{\text{2 Fehler}} \vec{x}$ ...hat zu jedem anderen Codewort $\vec{c_2}$ Abstand ≥ 2.

Beispiel: Die Voyager-Raumsonde (1979-1981) verwendete für Fotos von Jupiter und Saturn folgende Codierung: jedem Gitterpunkt eines 600×600 Rasters wurde einer von $2^{12} = 4096$ möglichen Farbwerten zugeordnet. Die binären Nachrichtenwörter wurden in Blöcke der Länge $k = 12$ zerlegt. Die Codierung erfolgte durch einen binären $(24, 12, 8)$-Code (Golay-Code). Dieser Code kann $\lfloor \frac{8-1}{2} \rfloor = 3$ Fehler korrigieren und 4 fach-Fehler als solche erkennen. Informationsrate $\frac{k}{n} = \frac{1}{2}$.

Hamming-Schranke und perfekte Codes

Sei C ein (n, k, d)-Blockcode.

Man möchte, dass die Informationsrate $\frac{k}{n}$ groß ist (Effizienz), und man möchte, dass die Minimaldistanz d groß ist (Korrekturkapazität). Bei festem n wird d (das abhängig von den $n-k$ Kontrollstellen ist) umso kleiner, je größer k ist, und umgekehrt. Folglich gibt es Schranken für die Parameter (n, k, d).

Lemma: $|A| = q,\ \vec{c} \in A^n,\ t \in \mathbb{N}$, dann gilt:
$|K_t(\vec{c})| = \sum_{i=0}^{t} \binom{n}{i}(q-1)^i$ (unabhängig von $\vec{c}$!)

Beweis: $\binom{n}{i}$ gibt die Anzahl der Möglichkeiten an, aus den n Stellen i auszuwählen, wo Unterschiede auftreten.

Das Alphabet hat q Elemente, daher gibt es $q-1$ Möglichkeiten, an einer Stelle einen Fehler zu machen. An i vorgegebenen Stellen kann man folglich $(q-1)^i$ verschiedene Fehler machen.

Daher gilt insgesamt: $|\{\vec{x} \in A^n \mid d(\vec{x}, \vec{c}) = i\}| = \binom{n}{i}(q-1)^i$.

Da $K_t(\vec{c})$ die Vereinigung dieser Mengen für $i = 0, 1, \ldots, t$ ist und diese Mengen disjunkt sind, erhält man die Anzahl der Elemente von $K_t(\vec{c})$ durch einfaches Aufsummieren. □

Satz (Hammig-Schranke, Sphere-Packing Bound): Existiert ein t-fehlerkorrigierender (n,k)-Blockcode C über dem Alphabet A mit $|A| = q$, so gilt: $n - k \geq \log_q \sum_{i=0}^{t} \binom{n}{i}(q-1)^i$.

Beweis: C ist t-fehlerkorrigierend, d.h. $K_t(\vec{c_1}) \cap K_t(\vec{c_2}) = \emptyset \quad \forall \vec{c_1}, \vec{c_2} \in C, \vec{c_1} \neq \vec{c_2}$

$\dot{\bigcup}_{\vec{c} \in C} K_t(\vec{c}) \subseteq A^n$ $\qquad$ $\dot{\bigcup} \ldots$ disjunkte Vereinigung

$$\left|\dot{\bigcup}_{\vec{c} \in C} K_t(\vec{c})\right| = \sum_{\vec{c} \in C} |K_t(\vec{c})| = \underbrace{|C|}_{q^k} \sum_{i=0}^{t} \binom{n}{i}(q-1)^i \leq \underbrace{|A^n|}_{q^n}$$

$$q^k \sum_{i=0}^{t} \binom{n}{i}(q-1)^i \leq q^n \qquad | : q^k \geq 0, \log_q$$

$$\log_q \sum_{i=0}^{t} \binom{n}{i}(q-1)^i \leq \log_q q^{n-k} = n - k$$ □

„Optimalfall“ (im Sinne der Disjunktheit der t-Kugelumgebungen):

$\dot{\bigcup}_{\vec{c} \in C} K_t(\vec{c}) = A^n$, d.h. der Raum wird von den disjunkten Kugelumgebungen vollständig ausgefüllt, die t-Kugelumgebungen um die Codeworte bilden eine Partition von A^n. Der Code heißt dann **t-perfekt**.

Ein t-perfekter Code kann alle t-fach-Fehler (und weniger) korrigieren. Mehr als t Fehler werden aber sicher falsch korrigiert.

Ein Code ist genau dann t-perfekt, wenn er alle t-fach Fehler korrigiert, d.h. Minimaldistanz $d \geq 2t + 1$, und in der Ungleichung der Hamming-Schranke das Gleichheitszeichen gilt.

Beispiel: Binärcode $C = \{000, 111\}, n = 3, k = 1$ (3-fach Wiederholungscode). Der Code ist 1-perfekt. $d = 3 \Rightarrow t = \left\lfloor \frac{d-1}{2} \right\rfloor = 1$ Fehler werden korrigiert.

Allgemein: ein $(2t+1)$-fach Wiederholungscode ($t \in \mathbb{N}$) ist t-perfekt.

Beispiel: $A = \{0, 1\}$

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

Der Binärcode C ergibt sich durch Bildung aller möglichen (bitweisen) Summen der Zeilenvektoren:

$$C = \{ \underbrace{0000000}_{\text{leere Summe}}, \underbrace{1000111, 0100110, 0010101, 0001011}_{\text{1 Summand}}, \underbrace{1100001, 1010010, 1001100}_{\text{2 Summanden}},$$
$$\underbrace{0110011, 0101101, 0011110}_{\text{2 Summanden}}, \underbrace{1110100, 1101010, 1011001, 0111000}_{\text{3 Summanden}}, \underbrace{1111111}_{\text{4 Summanden}} \}$$

An den ersten vier Stellen sieht man, welche Zeilen addiert wurden.

Man kann überprüfen (durch Bestimmung aller Hamming-Distanzen (eine bessere Methode werden wir im Abschnitt über Linearcodes angeben), dass $d = 3 \Rightarrow$ der Code ist 1-fehlerkorrigierend.

C ist sogar 1-perfekt. Wir zeigen, dass in der Formel für die Hamming-Schranke „=“ gilt:
$n = 7$, $q = 2$, $q^k = 16 \Rightarrow k = 4$.
Linke Seite: $n - k = 7 - 4 = 3$. Rechte Seite: $\log_2 \left(\binom{7}{0}1^0 + \binom{7}{1}1^1\right) = \log_2(1 + 7) = 3$.

4 Lineare Codes

Ab jetzt: Alphabet A ist ein Körper, z.B. $A = \mathbb{Z}_p$, p prim ($p = 2$: $\mathbb{Z}_2 = \{0, 1\}$ ... Binärcodes)

Dann haben wir in natürlicher Weise auf $A^n = \{x_1x_2....x_n \mid x_i \in A\}$ eine Vektorraumstruktur:

$\vec{x} = (x_1, x_2, \ldots, x_n)$, $\vec{y} = (y_1, y_2, \ldots, y_n)$, $\lambda \in A$:

$$\vec{x} + \vec{y} = (x_1 + y_1, x_2 + y_2, \ldots, x_n + y_n), \quad \lambda\vec{x} = (\lambda x_1, \lambda x_2, \ldots, \lambda x_n)$$

Beispiel: $A = \mathbb{Z}_3 = \{0, 1, 2\}, n = 2$:

$(1, 2) + (2, 2) = (0, 1)$, $2 \cdot (1, 2) = (2, 1)$.

Für einen Vektorraum muss gelten:

$(A^n, +)$ bildet eine kommutative Gruppe (abgeschlossen, assoziativ, neutrales Element, jedes Element $\vec{x}$ besitzt (additives) Inverses $-\vec{x}$, kommutativ) und für die skalare Multiplikation gilt für alle $\lambda, \mu \in A$:

$$(\lambda + \mu)\vec{x} = \lambda\vec{x} + \mu\vec{x}, \quad (\lambda\mu)\vec{x} = \lambda(\mu\vec{x}), \quad \lambda(\vec{x} + \vec{y}) = \lambda\vec{x} + \lambda\vec{y}, \quad 1\vec{x} = \vec{x}.$$

Sonderfall **Binärcode**: $A = \mathbb{Z}_2 = \{0, 1\}$: skalare Multiplikation ist „trivial“: $0 \cdot \vec{x} = \vec{0}$, $1 \cdot \vec{x} = \vec{x}$.
Weiters: $-1 = 1$ folglich $-\vec{x} = \vec{x}$.

Sei $(A^n, +, (\lambda\cdot)_{\lambda\in A})$ ein Vektorraum. $M \subseteq A^n$ ist ein **Unterraum** (Teilraum), wenn $(M, +, (\lambda\cdot)_{\lambda\in A})$ ein Vektorraum ist, d.h. für alle $\vec{x}, \vec{y} \in M$ und $\lambda \in A$ gilt

$$\vec{x} + \vec{y} \in M \quad \text{und} \quad \lambda\vec{x} \in M$$

(d.h. M ist abgeschlossen bezüglich Addition und skalarer Multiplikation).

Schreibweise für M Unterraum von A^n: $M \leq A^n$.

Ein Code C heißt **Linearcode (linearer Code)**, falls C ein Unterraum von A^n ist, also $C \leq A^n$.

Beispiel: Code 5 aus den einführenden Beispielen $C = \{00000, 10110, 01101, 11011\}$ ist Linearcode über $A = \mathbb{Z}_2$:

Da die skalare Multiplikation bei einem Binärcode trivial ist (man muss nur prüfen, ob $\vec{0} \in C$), brauchen wir nur zu zeigen, dass die Summe zweier Codewörter wieder im Code ist:
$10110 + 01101 = 11011 \in C$, $10110 + 11011 = 01101 \in C$, usw.

Beispiel: $C = \{000, 111\}$ ist ein Linearcode über $A = \mathbb{Z}_2$, aber ist kein Linarcode über $A = \mathbb{Z}_3 = \{0, 1, 2\}$: $2 \cdot (1, 1, 1) = (1, 1, 1) + (1, 1, 1) = (2, 2, 2) \notin C$.
$C' = \{000, 111, 222\}$ ist ein Linearcode über $\mathbb{Z}_3$.

Andere Beispiele für Vektorräume:

- $\left(\mathbb{R}^2, +, (\lambda\cdot)_{\lambda\in\mathbb{R}}\right)$ ... Ebene
  $\left(\mathbb{R}^3, +, (\lambda\cdot)_{\lambda\in\mathbb{R}}\right)$ ... dreidimensionaler Raum
  Unterräume in $\mathbb{R}^2$: Geraden durch den Ursprung (und die trivialen Unterräume $\{0\}, \mathbb{R}^2$)
  Unterräume in $\mathbb{R}^3$: Geraden und Ebenen durch den Ursprung (und die trivialen Unterräume $\{0\}, \mathbb{R}^3$)
- Polynome vom Grad $< n$ mit Koeffizienten aus einem Körper A:

  $$P_n = \{a_0 + a_1x + a_2x^2 + \cdots + a_{n-1}x^{n-1} \mid a_i \in A, i = 0, \ldots, n-1\}$$

  Addition und skalare Multiplikation ist Polynome $a_0 + a_1x + a_2x^2 + \cdots + a_{n-1}x^{n-1}$ genau so definiert wie für das zugehörige n-Tupel $(a_0, a_1, \ldots, a_{n-1})$.

Begriffe in Vektorräumen:

Vektorraum $(V, +, (\lambda\cdot)_{\lambda\in A})$, A ein Körper

$\vec{x} \in V$ heißt **Linearkombination** der Vektoren $\vec{a_1}, \ldots, \vec{a_s} \in V \Leftrightarrow \exists \lambda_1, \ldots, \lambda_s \in A$:

$$\vec{x} = \lambda_1\vec{a_1} + \lambda_2\vec{a_2} + \cdots + \lambda_s\vec{a_s}.$$

Ist jeder Vektor $\vec{x} \in V$ Linearkombination von $\vec{a_1}, \ldots, \vec{a_s}$, dann heißt $\vec{a_1}, \ldots, \vec{a_s}$ **Erzeugendensystem** von V. Man schreibt dafür $V = \langle\vec{a_1}, \vec{a_2}, \ldots, \vec{a_s}\rangle$.
Allgemein: Für $M \subseteq V$ bezeichnet $\langle M\rangle$ das Erzeugnis (die lineare Hülle) von M, ist die Menge aller Linearkombinationen von Elementen aus M. $\langle M\rangle$ ist stets ein Unterraum von V, nämlich der kleinste Unterraum von V, der M enthält.

Beispiel: $A = \mathbb{Z}_3 = \{0, 1, 2\}$, $V = \mathbb{Z}_3^2$, Menge aller Paare aus $\mathbb{Z}_3$; $\vec{a_1} = (1, 1)$, $\vec{a_2} = (1, 2)$.

Behauptung: $\mathbb{Z}_3^2 = \langle\vec{a_1}, \vec{a_2}\rangle$

Bilden aller möglichen Linearkombinationen: $(0, 0), (1, 1), (1, 2), (1, 1) + (1, 2) = (2, 0), 2 \cdot (1, 1) = (2, 2), 2 \cdot (1, 2) = (2, 1), (2, 2) + (2, 1) = (1, 0), (1, 1) + (2, 1) = (0, 2), (2, 2) + (1, 2) = (0, 1)$ ... d.h. es können alle möglichen 9 Vektoren als Linearkombination von $\vec{a_1}$ und $\vec{a_2}$ dargestellt werden.

Beispiel: $A = \mathbb{Z}_2 = \{0, 1\}$, $V = \mathbb{Z}_2^3 = \{(x, y, z) \mid x, y, z \in \mathbb{Z}_2\}$; $\vec{a_1} = (1, 0, 1)$, $\vec{a_2} = (1, 1, 0)$, $\vec{a_3} = (0, 1, 1)$.

Ist $V = \langle\vec{a_1}, \vec{a_2}, \vec{a_3}\rangle$? Nein! Wegen der Beziehung $\vec{a_3} = \vec{a_1} + \vec{a_2}$ lässt sich $\vec{a_3}$ in einer Linearkombination durch $\vec{a_1} + \vec{a_2}$ ersetzen und man erhält nur 4 verschiedene Linearkombinationen von $\vec{a_1}, \vec{a_2}$ und $\vec{a_3}$ und nicht $|V| = 8$.

$\{\vec{a_1}, \vec{a_2}, \ldots, \vec{a_s}\}$ heißt **linear unabhängig** genau dann, wenn $\nexists i \in \{1, \ldots, s\} : \vec{a_i} = \sum_{j=1, j\neq i}^{s} \lambda_j\vec{a_j}$.
Äquivalent dazu sind auch folgende Bedingungen:

- $\lambda_1\vec{a_1} + \lambda_2\vec{a_2} + \cdots + \lambda_s\vec{a_s} = \vec{0} \Leftrightarrow \lambda_1 = \lambda_2 = \cdots = \lambda_s = 0$
- Jeder Vektor im Unterraum $\langle\vec{a_1}, \vec{a_2}, \ldots, \vec{a_s}\rangle$ besitzt eine eindeutige Darstellung als Linearkombination von $\vec{a_1}, \ldots, \vec{a_s}$.

Satz: Sei $(V, +, (\lambda\cdot)_{\lambda\in A})$ ein Vektorraum und $B \subseteq V$, dann sind folgende Aussagen äquivalent:

- B ist ein Erzeugendensystem von V mit möglichst wenigen Vektoren.
- B ist eine linear unabhängige Teilmenge von V mit möglichst vielen Vektoren.
- B ist ein linear unabhängiges Erzeugendensystem.
- Jeder Vektor aus V läßt sich eindeutig als Linearkombination von Vektoren aus B schreiben.

Eine Teilmenge $B \subseteq V$ mit diesen Eigenschaften heißt **Basis** von V. Jeder Vektorraum besitzt eine Basis.

Die **Standardbasis** in $(A^n, +, (\lambda\cdot)_{\lambda\in A})$, $A^n = \{(a_1, a_2, \ldots, a_n \mid a_i \in A\}$ ist gegeben durch $\{\vec{e_1}, \vec{e_2}, \ldots, \vec{e_n}\}$ mit

$\vec{e_1} = (1, 0, 0, 0, \ldots, 0)$, $\vec{e_2} = (0, 1, 0, 0, \ldots, 0)$, $\vec{e_3} = (0, 0, 1, 0, \ldots, 0)$, $\ldots$, $\vec{e_n} = (0, 0, 0, \ldots, 0, 1)$.

Jedes $x \in A^n$ lässt sich eindeutig als Linearkombination aus $\{\vec{e_1}, \vec{e_2}, \ldots, \vec{e_n}\}$ darstellen:

$$x = (x_1, x_2, \ldots, x_n) = x_1 \cdot (1, 0, 0, 0, \ldots, 0) + x_2 \cdot (0, 1, 0, 0, \ldots, 0) + \cdots + x_n \cdot (0, 0, 0, \ldots, 0, 1) = x_1\vec{e_1} + x_2\vec{e_2} + x_3\vec{e_3} + \ldots x_n\vec{e_n}$$

Beispiel:

$\{(1, 1), (1, 2)\}$ ist eine Basis von $V = \mathbb{Z}^3$ (siehe Beispiel zu Erzeugendensystem)

Es reicht zu zeigen, dass sich jeder Vektor einer anderen Basis eindeutig als Linearkombination darstellen läßt.

z.B. Standardbasis $\{(1,0),(0,1)\}$:

$\lambda_1 \cdot (1,1) + \lambda_2 \cdot (1,2) = (1,0)$	$\mu_1 \cdot (1,1) + \mu_2 \cdot (1,2) = (0,1)$
$\lambda_1 + \lambda_2 = 1$	$\mu_1 + \mu_2 = 0$
$\lambda_1 + 2\lambda_2 = 0$	$\mu_1 + 2\mu_2 = 1$
$\Rightarrow \lambda_1 = 2, \lambda_2 = 2$	$\Rightarrow \mu_1 = 1, \mu_2 = 2$

Die Basis eines Vektorraums ist nicht eindeutig bestimmt, aber die Anzahl der Elemente einer Basis ist eindeutig. Diese Anzahl nennt man die **Dimension** von V.

Beispiel: A^n hat Dimension n (die Standardbasis hat n Elemente).

Beispiel: linearer Code $C = \{00000, 10110, 01101, 11011\}$, C ist Vektorraum über $A = \{0,1\}$. Eine Basis ist gegeben durch $\{10110, 01101\} \Rightarrow \dim C = 2$.

Sei C ein linearer Code mit $\dim C = k$. Dann gilt $|C| = |A|^k$:
Sei $B = \left\{\vec{b_1}, \vec{b_2}, \ldots, \vec{b_k}\right\}$ eine Basis von C. Dann gilt:
$\forall \vec{c} \in C \; \exists$ eindeutig $\lambda_1, \lambda_2, \ldots, \lambda_k \in A : \vec{c} = \lambda_1\vec{b_1} + \lambda_2\vec{b_2} + \cdots + \lambda_k\vec{b_k}$.

$|C| = \#$ verschiedener Linearkombinationen von $\left\{\vec{b_1}, \vec{b_2}, \ldots, \vec{b_k}\right\}$.

Für jeden Koeffizienten λ gibt es $|A|$ Möglichkeiten, und daher gilt auf Grund der eindeutigen Darstellung von Elementen in C als Linearkombinationen aus B

$$|C| = \underbrace{|A| \cdot |A| \cdot |A| \ldots |A|}_{k \text{ Faktoren}} = |A|^k.$$

Für einen linearen (n,k)-Blockcode C gilt, dass k — die Anzahl der Nachrichtenstellen von C — mit der Dimension des Codes als Vektorraum übereinstimmt: Die Codierungsfunktion $f_C : A^k \to A^n$ ist injektiv, folglich gilt $|C| = \left|f_C(A^k)\right| \overset{f \text{ injektiv}}{=} \left|A^k\right| = |A|^k$.

Andererseits, habe C als linearer Code die Dimension $\dim C = \overline{k}$ dann folgt (siehe oben) $|C| = |A|^{\overline{k}}$ und daher $k = \overline{k}$.

Also gilt für einen linearen Code immer: Anzahl der Nachrichtenstellen = Dimension des Codes.

Weitere Forderung an die Codierungsfunktion f_C bei Linearcodes: $f_C : A^k \to A^n$ **lineare Funktion**, d.h. für alle Nachrichtenworte $\vec{n_1}, \vec{n_2} \in A^k$ und alle $\lambda \in A$ gilt:

$$\begin{aligned} f_C\left(\vec{n_1} + \vec{n_2}\right) &= f_C(\vec{n_1}) + f_C(\vec{n_2}) \\ f_C(\lambda\vec{n_1}) &= \lambda f_C(\vec{n_1}) \end{aligned}$$

Fordert man allgemein von der Codierungsfunktion $f_C : A^k \to A^n$ neben der Injektivität auch die Linearität, so ist der zugehörige Code $C = f_C\left(A^k\right)$ immer ein Unterraum von A^n, d.h. C ist dann ein linearer Code.

5 Generator- und Kontrollmatrix

Zu jeder linearen Codierungsfunktion $f_C : A^k \to A^n$ gehört eine Matrix $\mathcal{G}$, sodass
$f_C(\vec{n}) = \vec{n} \cdot \mathcal{G} \quad \forall \vec{n} \in A^k$.

$\mathcal{G}$ heißt **Generatormatrix** von $C = f_C(A^k) \subseteq A^n$. $\mathcal{G}$ ist eine $k \times n$ Matrix:

$$\mathcal{G} = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{k1} & c_{k2} & \cdots & c_{kn} \end{pmatrix}$$

Die Zeilenvektoren von $\mathcal{G}$ liegen in C:

$$f_C(1,0,\ldots,0) = (1,0,\ldots,0) \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{k1} & c_{k2} & \cdots & c_{kn} \end{pmatrix} = (c_{11}, c_{12}, \ldots, c_{1n})$$

Allgemein ist die i-te Zeile von $\mathcal{G}$ das Codewort, das zum Nachrichtenwort $(0,\ldots,0,1_{(\text{Stelle } i)},0,\ldots,0)$ gehört.

Die Zeilen von $\mathcal{G}$ bilden eine Basis für den Code C:

$\vec{c} \in C \Rightarrow \exists!\ \vec{x} \in A^k : f_C(\vec{x}) = \vec{c}$

$\vec{x} = (x_1, x_2, \ldots, x_k) \Rightarrow \vec{c} = \underbrace{(x_1, x_2, \ldots, x_k)}_{\substack{x_1(1,0,0,\ldots,0)+ \\ x_2(0,1,0,\ldots,0)+ \\ \cdots + \\ x_k(0,0,\ldots,0,1)}} \mathcal{G} =$

$$x_1 \underbrace{(1,0,0,\ldots,0)\ \mathcal{G}}_{(c_{11},\ldots,c_{1n})} + x_2 \underbrace{(0,1,0,\ldots,0)\ \mathcal{G}}_{(c_{21},\ldots,c_{2n})} + \cdots + x_k \underbrace{(0,0,\ldots,0,1)\ \mathcal{G}}_{(c_{k1},\ldots,c_{kn})} =$$

$x_1(c_{11},\ldots,c_{1n}) + x_2(c_{21},\ldots,c_{2n}) + \cdots + x_k(c_{k1},\ldots,c_{kn})$,
d.h. $\vec{c}$ lässt sich als Linearkombination der Zeilen von $\mathcal{G}$ schreiben, und auf Grund der Injektivität von f_C sind die Zeilen von $\mathcal{G}$ linear unabhängig und diese Darstellung ist eindeutig.

Bei systematischer Codierung bei Linearcodes auf den ersten k Stellen gilt:

$$f_C(a_1,\ldots,a_k) = (a_1,\ldots,a_k,c_1,c_2,\ldots,c_{n-k}) \Leftrightarrow \mathcal{G} = \left(\begin{array}{cccc|c} 1 & 0 & \cdots & 0 & \\ 0 & 1 & \cdots & 0 & \\ \vdots & \vdots & \ddots & \vdots & \mathcal{M} \\ 0 & 0 & \cdots & 1 & \end{array}\right),$$

d.h. die Generatormatrix $\mathcal{G}$ lässt sich dann als Blockmatrix $\mathcal{G} = (\,\mathcal{E}_k \,|\, \mathcal{M}\,)$ schreiben, wobei $\mathcal{E}_k$ die $k \times k$-Einheitsmatrix ist und $\mathcal{M}$ irgendeine $k \times (n-k)$-Matrix. Die Generatormatrix liegt in **kanonischer Form** vor, wenn sie diese Gestalt hat.

Beispiel: (Code 5 aus den einführenden Beispielen)

$$f_C : \begin{cases} 00 & \mapsto 00000 \\ 01 & \mapsto 01110 \\ 10 & \mapsto 10101 \\ 11 & \mapsto 11011 \end{cases}, \ f_C : A^2 \to A^5,\ A = \{0,1\},\ \text{linear und systematisch mit Generatormatrix}$$

$$\mathcal{G} = \left(\begin{array}{cc|ccc} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{array}\right)$$

Beispiel: Binärer (7,4)-Hammingcode (Beispiel eines perfekten Codes)

$$\mathcal{G} = \left(\begin{array}{cccc|ccc} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{array}\right)$$

Unabhängig von der konkreten Codierungsfunktion f_C werden wir jede Matrix, deren Zeilen eine Basis des Codes bilden, als Generatormatrix bezeichnen.

Minimaldistanz von Linearcodes

Sei $d = \min\{d(\vec{c_1}, \vec{c_2}) \mid \vec{c_1}, \vec{c_2} \in C, \vec{c_1} \neq c_2\}$ die Minimaldistanz eines Linearcodes C.

Das **Hamming-Gewicht** eines Wortes $(a_1, \ldots, a_n) \in A^n$ ist definiert durch $w(a_1, \ldots, a_n) := |\{i \mid a_i \neq 0\}|$.

Lemma: Für $\vec{a}, \vec{b}, \vec{c} \in A^n$ gilt:
1. $w(\vec{a}) = d(\vec{a}, \vec{0})$
2. $d(\vec{a}, \vec{b}) = w(\vec{b} - \vec{a})$
3. $d(\vec{a} + \vec{c}, \vec{b} + \vec{c}) = d(\vec{a}, \vec{b})$

Beweis: 1. und 2. ergeben sich unmittelbar aus der Definition, 3. folgt aus 2.

Satz:
Die Minimaldistanz eines Linearcodes ist gleich dem Minimalgewicht $\min\left\{w(\vec{c}) \mid \vec{c} \in C, \vec{c} \neq \vec{0}\right\}$.

Beweis: Wegen 2) im obigen Lemma ist die Hamming-Distanz zwischen zwei verschiedenen Codeworten das Hamming-Gewicht eines Codewortes ungleich $\vec{0}$, woraus Minimalgewicht $\leq$ Minimaldistanz folgt. Umgekehrt ist das Gewicht eines Codewortes ungleich $\vec{0}$ wegen 1) darstellbar als Distanz zwischen zwei verschiedenen Codeworten, also gilt auch Minimaldistanz $\leq$ Minimalgewicht.

Beispiel: (7,4)-Hammingcode

$$\mathcal{G} = \left(\begin{array}{cccc|ccc} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{array}\right)$$

Wir müssen die Gewicht aller Codewörter $\neq \vec{0}$ betrachten, dazu müssen wir alle Linearkombinationen der Zeilen von $\mathcal{G}$ bilden:

Gewicht links von |	Gewicht rechts	Summe
1	≥ 2	≥ 3
2	≥ 1 (da alle Zeilen verschieden)	≥ 3
3	≥ 0	≥ 3
4	3	7

Also ist $d \geq 3$, in der 2. Zeile ist das Gewicht gleich $3 \Rightarrow d = 3$.

Satz (Singelton-Schranke): Für einen (n, k, d)-Blockcode gilt: $k + d \leq n + 1 \Leftrightarrow n - k \geq d - 1$

Beweis: Für einen (n, k, d)-Blockcode C über dem Alphabet A mit $|A| = q$ gilt $|C| = q^k$. Streichen wir bei allen q^k Codeworten aus C die ersten $d-1$ Stellen, so sind wegen der Minimaldistanz d die entstehenden q^k Worte der Länge $n-d+1$ immer noch paarweise verschieden. Da es nur q^{n-d+1} verschiedene Worte der Länge $n-d+1$ gibt, erhalten wir $q^k \leq q^{n-d+1}$ und damit $k \leq n-d+1$.

Ein Code mit $k + d = n + 1$ heißt **MDS-Code** (**M**aximum **D**istance **S**eparable).

Beispiel: Der n-fach Wiederholungscode ist ein MDS-Code:
$a \to a \ldots a$, $G = (1 \ldots 1)$, also ist $k = 1$, $d = n \Rightarrow k + d = n + 1$.

Beispiel: (7,4)-Hammingcode (1-perfekt, optimal im Sinne der Hamming-Schranke),
$d = 3 \Rightarrow k + d = 4 + 3 = 7 < 7 + 1 = 8$, ist also kein MDS-Code.

Beispiel: Reed-Solomon-Codes sind MDS-Codes (später im Detail behandelt).

Inneres Produkt im Sequenzraum A^n

$\vec{a} = (a_1, \ldots, a_n),\ \vec{b} = (b_1, \ldots, b_n) \in A^n$

Inneres Produkt $\vec{a} \cdot \vec{b} = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n \in A$

Eigenschaften:

$$\vec{a} \cdot \vec{0} = 0, \quad \vec{a} \cdot \vec{b} = \vec{b} \cdot \vec{a}, \quad (\lambda \vec{a}) \cdot \vec{b} = \lambda(\vec{a} \cdot \vec{b}), \quad \vec{a} \cdot (\vec{b} + \vec{c}) = \vec{a} \cdot \vec{b} + \vec{a} \cdot \vec{c}$$
$$(\forall \vec{b} \in A^n : \ \vec{a} \cdot \vec{b} = 0) \ \Rightarrow \ \vec{a} = \vec{0} \text{ (Produkt nicht–ausgeartet)}$$

Wir definieren: $\vec{a}$ ist **orthogonal** zu $\vec{b}$ und schreiben dafür $\vec{a} \perp \vec{b} :\Leftrightarrow \vec{a} \cdot \vec{b} = 0$

Bezeichnung inneres Produkt: Es handelt sich beim oben definierten Produkt $\vec{a} \cdot \vec{b}$ i.A. um kein **Skalarprodukt** (auch „inneres Produkt“ genannt) wie es in einem Vektorraum über dem Körper $A = \mathbb{R}$ (oder mit geeigneter Modifikation bei $A = \mathbb{C}$) definiert ist; ein solches muss laut Definition **positiv definit** sein, d. h. es gilt $\vec{a} \cdot \vec{a} > 0$ für alle Vektoren $\vec{a} \neq \vec{0}$. Trotzdem hat sich in der Codierungstheorie die Bezeichnung „inneres Produkt“ für obiges $\vec{a} \cdot \vec{b}$ etabliert.

Im Vektorraum $\mathbb{R}^n$ kann man betrachten:

1. Länge eines Vektors: $\| \vec{a} \| = \sqrt{\vec{a} \cdot \vec{a}} = \sqrt{a_1^2 + a_2^2 + \cdots + a_n^2}$
2. Winkelmessung: $\vec{a}$ und $\vec{b}$ schließen den Winkel ϕ ein $\Leftrightarrow \cos\phi = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \|\vec{b}\|}$

In $\mathbb{R}^n$ gilt: $\| \vec{a} \| = 0 \Leftrightarrow \vec{a} = \vec{0}$.
Das gilt für $A = \mathbb{Z}_p$, oder allgemein für endliche Körper A, i.A. nicht mehr, denn es kann $\vec{a} \cdot \vec{a} = 0$ auch für $\vec{a} \neq \vec{0}$ auftreten. Wenn $\vec{a} \cdot \vec{a} = 0$ dann gilt $\vec{a} \perp \vec{a}$.

Beispiel: $A = \mathbb{Z}_2$, $\vec{a} = (1, 1) \Rightarrow \vec{a} \cdot \vec{a} = 1^2 + 1^2 = 2 = 0$.

Für einen Linearcode C definieren wir den **Dualcode** $C^\perp$ zu C: $C^\perp := \{x \in A^n \mid \vec{x} \cdot \vec{c} = 0 \ \forall \vec{c} \in C\}$

Dualcode $C^\perp$ ist immer ein Linearcode (auch wenn C kein Linearcode ist):

1. $\vec{x}, \vec{y} \in C^\perp \Rightarrow \vec{x} + \vec{y} \in C^\perp : (\vec{x} + \vec{y}) \cdot \vec{c} = \underbrace{\vec{x} \cdot \vec{c}}_{0} + \underbrace{\vec{y} \cdot \vec{c}}_{0} = 0 \ \forall \vec{c} \in C$

2. $\vec{x} \in C^\perp \Rightarrow \lambda \vec{x} \in C^\perp \ \forall \lambda \in A : (\lambda \vec{x}) \cdot \vec{c} = \lambda(\underbrace{\vec{x} \cdot \vec{c}}_{0}) = 0$

Beispiel: $C = \{000, 111\}, A = \{0, 1\}$
$x \in C^\perp$, $x = (x_1, x_2, x_3)$: $(x_1, x_2, x_3) \cdot (1, 1, 1) = 0 \Rightarrow x_1 + x_2 + x_3 = 0$.
Daraus folgt $C^\perp = \{000, 110, 101, 011\}$.

Es reicht zu prüfen: $\vec{x} \in C^\perp \Leftrightarrow \vec{x} \cdot \vec{c_i} = 0 \ \forall \vec{c_i}$ aus Basis $\{\vec{c_1}, \ldots, \vec{c_k}\}$ von C

$C^\perp$ ist auch ein linearer Code. Eine Generatormatrix $\mathcal{H}$ von $C^\perp$ heißt **Kontrollmatrix** für den Code C.

Satz: Wenn C ein (n, k)-Linearcode ist, dann gilt:
1. $\dim C^\perp = n - k$, d.h. $C^\perp$ ist ein $(n, n-k)$-Linearcode
2. $\left(C^\perp\right)^\perp = C$

Beweis:
1. Sei $\{\vec{c_1}, \ldots, \vec{c_k}\}$ eine Basis von C.
$\vec{x} \in C^\perp \Leftrightarrow \forall i \in 1 \ldots k : \vec{x}\vec{c_i} = 0$

In Koordinaten aufgeschrieben bedeutet das Folgendes:

$\vec{x} = (x_1, \ldots, x_n), \qquad \vec{c_i} = (c_{i1}, \ldots, c_{in})$

$$\left.\begin{array}{l} c_{11}x_1 + c_{12}x_2 + \cdots + c_{1n}x_n = 0 \\ c_{21}x_1 + c_{22}x_2 + \cdots + c_{2n}x_n = 0 \\ \ldots \\ c_{k1}x_1 + c_{k2}x_2 + \cdots + c_{kn}x_n = 0 \end{array}\right\} \; k \text{ Gleichungen, } n \text{ Variable } x_1, \ldots, x_n.$$

Diese k Gleichungen sind linear unabhängig, da die Koeffizientenmatrix die Generatormatrix $\mathcal{G}$ von C ist, und der Rang $\operatorname{rg}(\mathcal{G}) = k$ (die Zeilen von $\mathcal{G}$ sind eine Basis von C und daher linear unabhängig).

Die Lösungen $(x_1, \ldots, x_n)$ bilden daher einen Unterraum der Dimension $n - \operatorname{rg}(\mathcal{G}) = n - k$.

2. $C \subseteq \left(C^\perp\right)^\perp$: $\vec{c} \in C \Rightarrow \vec{c} \cdot \vec{c_\perp} = 0 \; \forall \vec{c_\perp} \in C^\perp \Rightarrow \vec{c} \in \left(C^\perp\right)^\perp$.

Weiters gilt nach 1.: $\dim \left(C^\perp\right)^\perp = n - (n-k) = k$. Weil aber $\underbrace{C}_{\text{Dim. } k} \subseteq \underbrace{\left(C^\perp\right)^\perp}_{\text{Dim. } k}$ und beide die Dimension k haben, folgt $\left(C^\perp\right)^\perp = C$ □

Ein Code C heißt **selbstdual**, wenn $C = C^\perp$. Aufgrund des vorigen Satzes gilt für einen selbstdualen Code $k = n/2$, folglich muß die Länge n gerade sein.

Beispiel: $C = \{0000, 1100, 0011, 1111\}$ ist ein selbstdualer Binärcode.

Berechnung der Kontrolmatrix $\mathcal{H}$:

C ist ein (n,k)-Code, $\mathcal{G}$ Generatormatrix von C, ist eine $k \times n$-Matrix,
$\mathcal{H}$ Kontrollmatrix von C, ist eine $(n-k) \times n$-Matrix

1. Liegt $\mathcal{G} = (\mathcal{E}_k | \mathcal{M})$ in kanonischer Gestalt vor

$\Rightarrow \mathcal{H} = \left(-\mathcal{M}^T | \mathcal{E}_{n-k}\right)$ (siehe Übungen). Das nennt man die kanonische Gestalt der Kontrollmatrix.

Beispiel: $A = \mathbb{Z}_3$, $\mathcal{G} = \left(\begin{array}{cc|ccc} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{array}\right) \to \mathcal{H} = \left(\begin{array}{cc|ccc} -1 & 0 & 1 & 0 & 0 \\ -1 & -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 & 1 \end{array}\right) = \left(\begin{array}{cc|ccc} 2 & 0 & 1 & 0 & 0 \\ 2 & 2 & 0 & 1 & 0 \\ 0 & 2 & 0 & 0 & 1 \end{array}\right)$.

2. Alle anderen Fälle führen wir durch Zeilen und Spaltenumformungen auf Fall 1. zurück.

Beispiel: $A = \mathbb{Z}_2$

$$\mathcal{G} = \left(\begin{array}{ccc|ccc} 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \end{array}\right)$$

Zeile 1 von Zeile 2 abziehen: $\mathcal{G} = \left(\begin{array}{ccc|ccc} 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 \end{array}\right)$

Zeile 2 von Zeile 3 abziehen: $\mathcal{G} = \left(\begin{array}{ccc|ccc} 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{array}\right)$

3. und 4. Spalte tauschen: $\mathcal{G}' = \left(\begin{array}{ccc|ccc} 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{array}\right)$

Durch das Vertauschen der Spalten erzeugt $\mathcal{G}'$ einen anderen Code C'. Eine Kontrollmatrix von C' ist nach 1. gegeben durch

$$\mathcal{H}' = \left(\begin{array}{ccc|ccc} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \end{array}\right).$$

Um eine Kontrollmatrix $\mathcal{H}$ von C zu bekommen, müssen die durchgeführten Spaltenvertauschungen rückgängig gemacht werden:

$$\mathcal{H} = \left(\begin{array}{ccc|ccc} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 \end{array}\right)$$

Zusammenhang zwischen $\mathcal{G}$ und $\mathcal{H}$:

$\mathcal{G} \cdot \mathcal{H}^T = \mathcal{O}$ $\qquad$ $\mathcal{O} \ldots k \times n-k$–Nullmatrix

Grund: Das Element in der Zeile i und der Spalte j ergibt sich durch Multiplikation der Zeile i von $\mathcal{G}$ mit der Spalte j von $\mathcal{H}^T$ = Zeile j von $\mathcal{H}$. Die Zeilen von $\mathcal{G}$ und $\mathcal{H}$ bilden aber die Basen von C und $C^\perp$ und sind daher orthogonal zueinander.

Diese Beziehung charakterisiert die Kontrollmatrix: Wenn $\mathcal{H}$ eine Matrix mit $n-k$ Zeilen, n Spalten und $\operatorname{rg}(\mathcal{H}) = n-k$ ist und $\mathcal{G} \cdot \mathcal{H}^T = \mathcal{O}$ gilt $\Rightarrow \mathcal{H}$ Kontrollmatrix.

Charakterisierung der Codeworte mit Hilfe der Kontrollmatrix:

$\vec{c} \in C \Leftrightarrow \vec{c} \cdot \mathcal{H}^T = \vec{0}$:

wenn $\mathcal{H} = \begin{pmatrix} \vec{x}_1 \\ \vdots \\ \vec{x}_{n-k} \end{pmatrix}$, dann ist $\{\vec{x}_1, \ldots, \vec{x}_{n-k}\}$ eine Basis von $C^\perp$

$\vec{c} \cdot \mathcal{H}^T = \vec{0} \Leftrightarrow \vec{c} \cdot \vec{x}_1 = 0, \ldots, \vec{c} \cdot \vec{x}_{n-k} = 0 \Leftrightarrow \vec{c} \cdot \vec{x} = 0 \ \forall \vec{x} \in C^\perp \Leftrightarrow \vec{c} \in (C^\perp)^\perp = C$

Aus der Kontrollmatrix kann auch die Minimaldistanz eines Linearcodes bestimmt werden.

Satz: Wenn $\mathcal{H}$ Kontrollmatrix eines (n,k)-Linearcodes C ist, dann gilt: d ist Minimaldistanz von $C \Leftrightarrow$ je $d-1$ Spalten von H sind linear unabhängig und es gibt d Spalten, die linear abhängig sind.

Beweis: $\mathcal{H} = \left(\vec{h_1}\vec{h_2} \ldots \vec{h_n}\right)$ $\qquad$ $\ldots (n-k)$ Zeilen

$\vec{c} = (c_1, c_2, \ldots, c_n) \in C \Leftrightarrow \vec{c}\,\mathcal{H}^T = \vec{0}$

$\vec{c}\,\mathcal{H}^T = c_1\vec{h_1} + c_2\vec{h_2} + \cdots + c_n\vec{h_n} = \vec{0}$

$d = d(C)$: $\quad \exists\, \vec{c} \in C,\, w(\vec{c}) = d > 0 \Rightarrow \exists\, d$ Spalten von $\mathcal{H}$, die linear abhängig sind.

Angenommen, es gäbe $d-1$ linear abhängige Spalten. Dann kann man die Linearkombination $\cdots = \vec{0}$ mit Hilfe dieser Spalten aufschreiben, die Koeffizienten würden dann ein Codewort $\neq \vec{0}$ ergeben, das ein Gewicht $< d$ hätte, d ist aber die Minimaldistanz.) $\quad \square$

Folgerung: Sind alle Spalten der Kontrollmatrix $\mathcal{H}$ eines Binärcodes verschieden und $\neq \vec{0}$, dann kann C alle 1-fach-Fehler korrigieren. (Es sind dann je zwei Spalten linear unabhängig $\Rightarrow d \geq 3$.) Verallgemeinerung der Bedingung für Nicht-Binärcodes: Die Spalten von $\mathcal{H}$ dürfen keine Vielfachen voneinander sein.

6 Standardkorrekturschema für Linearcodes

$C \leq A^n$ Unterraum, $\vec{a} \in A^n$, dann heißt
$\vec{a} + C = \{\vec{a} + \vec{c} \mid \vec{c} \in C\}$ Nebenraum (Nebenklasse) von $\vec{a}$ bezüglich C

Nebenräume $\{\vec{a} + C \mid \vec{a} \in A^n\}$ bilden Partition von A^n, d.h. $(\vec{a_1} + C) \cap (\vec{a_2} + C) = \emptyset$ oder $\vec{a_1} + C = \vec{a_2} + C$, Letzteres gilt $\Leftrightarrow \vec{a_1} - \vec{a_2} \in C$.

Beispiel:

$A = \mathbb{Z}_2, C = \{000, 101\}$

$000 + C = \{000, 101\} = 101 + C$
$100 + C = \{100, 001\} = 001 + C$
$010 + C = \{010, 111\} = 111 + C$
$110 + C = \{110, 011\} = 011 + C$

$\dim C = k$, $|A| = q \Rightarrow$ es gibt q^{n-k} veschiedene Nebenklassen:
$|C| = q^k, \forall \vec{a} \in A^n$: $|\vec{a} + C| = q^k, \#$ Nebenklassen $\cdot\, q^k = q^n \Rightarrow \#$ Nebenklassen $= q^{n-k}$

Wir untersuchen jetzt die **Decodierung** eines Linearcodes C und werden feststellen, dass der wesentliche Vorteil der Linearität darin besteht, das die Decodierung nicht für jedes Empfangswort einzeln erfolgen muss sondern nur einmal pro Nebenklasse von C:

Sei $\vec{x_0} \in A^n$ Empfangswort und $\vec{c_0} \in C$ mit der Eigenschaft dass $d(\vec{x_0}, \vec{c_0}) = \min\{d(\vec{x_0}, \vec{c}) \mid \vec{c} \in C\}$ ($\vec{c_0}$ muss nicht eindeutig bestimmt sein!) $\Rightarrow$ Empfänger decodiert $\vec{x_0}$ zu $\vec{c_0}$ (Nearest Neighbor Decodierung)

$\vec{e_0} := \vec{x_0} - \vec{c_0}$ heißt dann **Fehlervektor** zu $\vec{x_0}$. Es gilt dann $\vec{x_0} = \vec{c_0} + \vec{e_0}$ und $d(\vec{x_0}, \vec{c_0}) = w(\vec{x_0} - \vec{c_0}) = w(\vec{e_0})$.

Behauptung: Zu allen $\vec{x} \in \vec{e_0} + C$ gehört derselbe Fehlervektor $\vec{e_0}$, d.h. Empfänger decodiert $\vec{x} = \vec{e_0} + \vec{c}$ zu $\vec{c}$:

Es gilt $d(\vec{x}, \vec{c}) = w(\vec{x} - \vec{c}) = w(\vec{e_0})$.

Betrachten $d(\vec{x}, \vec{c'})$ für ein beliebiges $\vec{c'} \in C$.

Es gilt allgemein $\forall\, \vec{x}, \vec{y}, \vec{z} \in A^n : d(\vec{x}, \vec{y}) = d(\vec{x} + \vec{z}, \vec{y} + \vec{z})$.

Aus $\vec{e_0} = \vec{x_0} - \vec{c_0} = \vec{x} - \vec{c}$ folgt $\vec{x_0} - \vec{x} = \vec{c_0} - \vec{c}$.

Daher gilt $d(\vec{x}, \vec{c'}) = d(\vec{x} + (\vec{x_0} - \vec{x}), \vec{c'} + (\vec{c_0} - \vec{c})) = d(\vec{x_0}, \underbrace{\vec{c'} + \vec{c_0} - \vec{c}}_{\in C}) \geq w(\vec{e_0}) = d(\vec{x}, \vec{c})$.

Folgerung 1: Speziell haben wir für $\vec{c} = \vec{0}$:

$$w(\vec{e_0}) = d(\vec{e_0}, \vec{0}) \leq d(\vec{e_0}, -\vec{c}) = w(\vec{e_0} + \vec{c}) \quad \forall \vec{c} \in C,$$

also ist $w(\vec{e_0}) = \min\{w(\vec{e_0} + \vec{c}) | \vec{c} \in C\}$.

Folgerung 2: Wenn $w(\vec{e}) = \min\{w(\vec{e} + \vec{c}) | \vec{c} \in C\}$, dann gilt für alle $\vec{c}, \vec{c}_1 \in C$:

$$d(\vec{e} + \vec{c}, \vec{c}_1) = w(\vec{e} + \underbrace{\vec{c} - \vec{c}_1}_{\in C}) \geq w(\vec{e}) = d(\vec{e} + \vec{c}, \vec{c}),$$

also kann $\vec{e} + \vec{c}$ zu $\vec{c}$ decodiert werden.

Schlussfolgerung: Man wählt aus jeder Nebenklasse N ein Wort $\vec{e}$ mit minimalem Gewicht aus, $N = \vec{e} + C$, und decodiert das Empfangswort $\vec{e} + \vec{c}$ zum Codewort $\vec{c}$.

Das Fehlerwort $\vec{e}$ heißt **Nebenklassenanführer**. Jedes Wort aus N mit minimalem Gewicht ist als Anführer geeignet. Das Korrekturschema, das sich aus dieser Vorgangsweise ergibt, heißt **Standardkorrekturschema**.

Beispiel: $C = \{00000, 10110, 01101, 11011\}$

$\vec{e}\downarrow \vec{c}\rightarrow$	00000	10110	01101	11011
00000	00000	10110	01101	11011
10000	10000	00110	11101	01011
01000	01000	11110	00101	10011
00100	00100	10010	01010	11111
00010	00010	10100	01111	11001
00001	00001	10111	01100	11010
11000	11000	01110	10101	00011
10001	10001	00111	11100	01010

Die Nebenklassenanführer mit Gewicht 2 sind nicht mehr eindeutig bestimmt!

Korrektur: 1) Empfangswort 10100, steht in der Zeile mit Anführer 00010, wird also decodiert zu $10100 - 00010 = 10110$ (= Codewort, das in der gleichen Spalte wie das Empfangswort steht). Am Anführer sieht man die Stelle(n), die korrigiert wird/werden.
2) 01110 wird zu 10110 decodiert, dabei werden die ersten beiden Stellen korrigiert. Wenn man in dieser Nebenklasse 00011 als Anführer gewählt hätte, würde man zu 01101 decodieren.

Die Anführer (= ausgewählte Fehlerwörter) beschreiben genau die Fehler, die richtig korrigiert werden.

Vereinfachung durch Verwendung der Kontrollmatrix:

$C \ldots (n, k, d)$ Code, Generatormatrix $\mathcal{G} \ldots k \times n$ Matrix,
Kontrollmatrix $\mathcal{H} \ldots (n-k) \times n$ Matrix

Für $\vec{x} \in A^n$ definieren wir das **Syndrom** von $\vec{x}$ als $s_H(\vec{x}) := \vec{x} \cdot \mathcal{H}^T \in A^{n-k}$.

Es gilt: $\vec{c} \in C \Leftrightarrow \vec{c} \cdot \mathcal{H}^T = \vec{0}$, also $\vec{c} \in C \Leftrightarrow s_H(\vec{c}) = \vec{0}$.

Allgemein: $s_H(\vec{x}) = s_H(\vec{y}) \Leftrightarrow \vec{x} + C = \vec{y} + C$, d.h. das Syndrom von $\vec{x}$ kennzeichnet die Nebenklasse von $\vec{x}$:

$s_H(\vec{x}) = s_H(\vec{y}) \Leftrightarrow \vec{x} \cdot \mathcal{H}^T = \vec{y} \cdot \mathcal{H}^T \Leftrightarrow (\vec{x} - \vec{y}) \cdot \mathcal{H}^T = \vec{0} \Leftrightarrow \vec{x} - \vec{y} \in C \Leftrightarrow \vec{x} + C = \vec{y} + C$

$s_H : A^n \to A^{n-k}$ ist offensichtlich eine lineare Abbildung und surjektiv weil $\mathrm{rg}\mathcal{H} = \dim C^\perp = n - k$.

Vorgangsweise für Korrekturschema:

Man braucht für jedes der q^{n-k} verschiedenen Syndrome in A^{n-k} einen Anführer für die zugehörige Nebenklasse. Dazu bestimmt man zuerst die Syndrome der Fehlerwörter vom Gewicht 1 und solange man verschiedene Syndrome erhält, kann man diese Worte als Nebenklassenanführer nehmen.

Dann macht man mit Fehlerworten vom Gewicht 2,3,... weiter, bis man alle möglichen Syndrome erhalten hat. Beim Suchen von Anführern für bisher noch nicht erhaltene Syndrome benützt man, dass die Syndromfunktion s_H linear ist. Das heißt, man versucht noch nicht erhaltene Syndrome als Linearkombination von Syndromen von Fehlerwörtern kleinstmöglichen Gewichts zusammenzusetzen und erhält so die fehlenden Anführer. Damit hat man schließlich für jede Nebenklasse einen Anführer mit minimalem Gewicht gefunden.

Decodierung: Empfängt man $\vec{x} \in A^n$, so berechnet man das Syndrom $s_H(\vec{x})$ und sucht sich dasjenige Fehlerwort $\vec{e}$ mit $s_H(\vec{e}) = s_H(\vec{x})$ und decodiert $\vec{x}$ zu $\vec{x} - \vec{e} = \vec{c} \in C$.

Beispiel: $C = \{00000, 10110, 01101, 11011\}$

$$\mathcal{G} = \left(\begin{array}{cc|ccc} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \end{array}\right) \qquad \Rightarrow \qquad \mathcal{H} = \left(\begin{array}{cc|ccc} 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{array}\right)$$

$s_H(00000) = (000)$

$$s_H(10000) = (10000) \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = (110)$$

$s_H(01000) = (101)$

$s_H(00100) = (100)$

$s_H(00010) = (010)$

$s_H(00001) = (001)$

Da die Syndrome aller Worte vom Gewicht 1 verschieden sind, korrigiert der Code Einfachfehler.

Noch fehlende Syndrome: (111) und (011)

$(111) = (101) + (010) \Rightarrow s_H(01010) = (111)$ oder
$(111) = (110) + (001) \Rightarrow s_H(10001) = (111)$

$(011) = (110) + (101) \Rightarrow s_H(11000) = (011)$ oder
$(011) = (010) + (001) \Rightarrow s_H(00011) = (011)$

D.h. für die beiden letzten Syndrome kann ein Fehlerwort mit Gewicht 2 als Anführer gewählt werden, dieses ist aber in beiden Fällen nicht eindeutig bestimmt.

Decodierung: Empfangswort $\vec{x} = (11111)$, das zugehörige Syndrom ist $s_H(\vec{x}) = (100) = s_H(\underbrace{(00100)}_{\vec{e}})$,

also wird $\vec{x}$ decodiert zu $\vec{x} - \vec{e} = (11011)$.

Bemerkung: Für Binärcodes gilt: Die Spalten der Kontrollmatrix sind die Syndrome der Fehlerworte vom Gewicht 1. Sind also die Spalten von $\mathcal{H}$ alle verschieden, so liegen alle Fehlerworte vom Gewicht 1 in verschiedenen Nebenklassen; diese können als Anführer ausgewählt werden $\Rightarrow$ der Code kann alle Einfachfehler korrigieren.

Für beliebige Lineararcodes gilt: Die Syndrome der Fehlerworte vom Gewicht 1 sind genau die Vielfachen (mit Faktor $\neq 0$) der Spalten von H; sind also je zwei verschiedene Spalten von H linear unabhängig, so korrigiert der Code Einfachfehler.

Binäre Hamming Codes

Wir haben als Beispiel schon den $(7, 4)$-Hamming Code mit Generatormatrix

$$G = \left(\begin{array}{cccc|ccc} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{array}\right)$$

kennengelernt. Als kanonische Kontrollmatrix erhält man

$$\mathcal{H} = \left(\begin{array}{cccc|ccc} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{array}\right).$$

Wie wir schon festgestellt haben, ist dieser Code 1-perfekt. Das erkennen wir auch an den Spalten von $\mathcal{H}$, die genau alle von $\vec{0}$ verschiedenen Elemente von $\mathbb{Z}_2^{n-k} = \mathbb{Z}_2^3$ durchlaufen. Da die Spalten von $\mathcal{H}$ genau die Syndrome der Fehlerworte vom Gewicht 1 sind, können wir aus jeder von C verschiedenen Nebenklasse genau ein Wort vom Gewicht 1 als Anführer auswählen, also korrigiert dieser Code genau alle Einfachfehler.

Diese Vorgangsweise funktioniert ganz allgemein: Für beliebiges $r \geq 0$ bestehe die Matrix $\mathcal{H}$ spaltenweise aus allen Vektoren in $\mathbb{Z}_2^r \setminus \vec{0}$ (die Reihenfolge der Vektoren ist vorläufig nicht so wichtig). $\mathcal{H}$ ist damit eine $r \times (2^r - 1)$-Matrix und hat klarerweise den Rang r. Mit den obigen Überlegungen betreffend Syndrome und Fehlerkorrektur sieht man, dass $\mathcal{H}$ die Kontrollmatrix eines 1-perfekten $(2^r - 1, 2^r - 1 - r)$-Binärcodes ist. Diese Klasse von Codes für $r \in \mathbb{N}$ nennt man **binäre Hamming Codes**.

- Binäre Hamming Codes haben Minimaldistanz $d = 3$: Wäre $d \geq 4$, so könnten 2-fach Fehler als solche erkannt werden; andererseits hat jedes Empfangswort auf Grund der 1-Perfektheit Hammingdistanz ≤ 1 zu einem Codewort, Widerspruch.
- Wie man aus der Konstruktion der binären Hamming Codes über die Kontrollmatrix sieht, gibt es bis auf Äquivalenz, d.h. bis auf eine Permutation der Stellen, genau einen 1-perfekten $(2^r - 1, 2^r - 1 - r)$-Binärcode.
- Hamming Codes über beliebigen Körpern werden wir in den Übungen diskutieren.

7 Polynomcodes, zyklische Codes

Bis jetzt hatten wir eine Vektorraum-Struktur auf Codeworten (Summe, skalare Vielfache). Jetzt wollen wir zusätzlich noch eine multiplikative Struktur einführen.

Definition: Ein Linearcode C über dem Körper A heißt **zyklisch**, wenn gilt:

$$(c_0, c_1, \ldots, c_{n-1}) \in C \Rightarrow (c_{n-1}, c_0, c_1, \ldots, c_{n-2}) \in C,$$

d.h. durch zyklisches Vertauschen der Stellen eines Codewortes erhält man wieder ein Codewort.

Beispiel: $C = \{0000, 1010, 0101, 1111\}$ ist ein zyklischer Code.

Warum betrachtet man zyklische Codes? — Noch effizienter als mit Matrizen kann man mit Polynomen rechnen ($\rightarrow$ Schieberegister).

Zyklische Codes lassen sich mit Hilfe von Polynomen gut darstellen:

Idee: Wir identifizieren das Wort / den Vektor $\vec{c} = (c_0, c_1, \ldots, c_{n-1}) \in A^n$ mit dem Polynom $c(x) = c_0 + c_1 x + c_2 x^2 + \cdots + c_{n-1} x^{n-1} \in A[x]$ mit Grad $\leq n - 1$.

Damit haben wir die gewünschte zusätzliche multiplikative Struktur in der Polynommultiplikation gefunden, wir können im Polynomring $(A[x], +, \cdot)$ operieren. Für die Addition und skalare Multiplikation (jeweils komponentenweise) ändert sich nichts im Vergleich zum Vektorraum $(A^n, +, (\lambda\cdot)_{\lambda \in A})$. Allerdings haben wir bei der Polynommultiplikation das Problem, dass sich die Gradbeschränkung „Grad $\leq n - 1$“ nicht von zwei Faktoren $c_1(x)$ und $c_2(x)$ auf das Produkt $c_1(x) \cdot c_2(x)$ vererbt:

$$\operatorname{grad}(c_1(x) \cdot c_2(x)) = \operatorname{grad} c_1(x) + \operatorname{grad} c_2(x)$$

Daher fassen wir diese Polynome als Elemente von $A[x]/(x^n-1)$ auf, d.h. als Elemente des Polynomringes $A[x]$ faktorisiert nach dem von x^n-1 erzeugten Hauptideal:

Addition von Polynomen (wie gewöhnlich in $A[x]$):
$(c_0+c_1x+\cdots+c_{n-1}x^{n-1})+(d_0+d_1x+\cdots+d_{n-1}x^{n-1})=$
$(c_0+d_0)+(c_1+d_1)x+\cdots+(c_{n-1}+d_{n-1})x^{n-1}$

Multiplikation mit Reduktion $\bmod(x^n-1)$: $x^n\equiv 1$, $x^{n+1}\equiv x$, $x^{n+2}\equiv x^2,\ldots$:

Beispiele:

- $x\cdot(c_0+c_1x+c_2x^2+\cdots+c_{n-1}x^{n-1})=c_0x+c_1x^2+c_2x^3+\cdots+c_{n-2}x^{n-1}+c_{n-1}\underbrace{x^n}_{\equiv 1}\equiv$
  $c_{n-1}+c_0x+c_1x^2+\cdots+c_{n-2}x^{n-1}=(c_{n-1},c_0,c_1,\ldots,c_{n-2})$,
- $x^2\cdot(c_0,\ldots,c_{n-1})=(c_{n-2},c_{n-1},c_0,\ldots c_{n-3})$,
- $\left(d_0x+d_1x^2\right)(c_0,\ldots,c_{n-1})=d_0\,(c_{n-1},c_0,\ldots,c_{n-2})+d_1\,(c_{n-2},c_{n-1},c_0,\ldots,c_{n-3})$

Man sieht, dass einer Multiplikation mit x in $A[x]/(x^n-1)$ eine zyklische Vertauschung der Koeffizienten entspricht. Daher sind zyklische Codes in Polynomdarstellung abgeschlossen gegenüber einer Multiplikation mit x, und folglich auch abgeschlossen bzgl. Multiplikation mit x^i, $i\in\mathbb{N}$.

Da zyklische Codes voraussetzungsgemäß auch linear sind, heißt das, dass sie abgeschlossen sind bezüglich der Multiplikation mit beliebigen Polynomen, d. h.

$$\forall c(x)\in C\ \forall p(x)\in A[x]:\ c(x)\cdot p(x)\bmod(x^n-1)\in C.$$

Diese Eigenschaft, abgeschlossen bezüglich Multiplikation mit beliebigen Polynomen zu sein, charakterisiert offensichtlich zyklische Codes.

Andere Interpretation für Multiplikation mit Reduktion $\bmod(x^n-1)$:
$r(x)=(c(x)\cdot p(x))\bmod(x^n-1)$ ist der (eindeutig bestimmte) Rest bei der Division von $c(x)\cdot p(x)$ ($\cdot$ ist hier die Multiplikation in $A[x]$) durch das Polynom x^n-1 (es gibt eine weitreichende Analogie zwischen den ganzen Zahlen $\mathbb{Z}$ und dem Polynomring $A[x]$ über einem Körper A).

Exkurs über allgemeine Division mit Rest für Polynome:
gegeben $a(x),b(x)$ Polynome in $A[x]$, $b(x)\neq 0$
$\Rightarrow\exists|\ q(x),r(x):\ a(x)=q(x)\,b(x)+r(x)$ mit $r(x)=0$ oder $\operatorname{grad} r(x)<\operatorname{grad} b(x)$.

Als Quotient geschrieben: $\frac{a(x)}{b(x)}=q(x)+\frac{r(x)}{b(x)}$.

Beispiel: $A=\mathbb{Z}_2,\ a(x)=x^4+x^3+x,\ b(x)=x^2+1$

$$\begin{array}{rrrrrcl}
(x^4 & +x^3 & +0x^2 & +x & +0) & : & (x^2+1)=x^2+x+1\\
-(x^4 & & +x^2) & & & & \\
\hline
 & x^3 & +x^2 & +x & & & \\
 & -(x^3 & & +x) & & & \\
\hline
 & & x^2 & & & & \\
 & & -(x^2 & & +1) & & \\
\hline
 & & & & 1 & & \text{Rest}
\end{array}$$

Also erhalten wir: $x^4+x^3+x=(x^2+x+1)\cdot(x^2+1)+1$ bzw. $\frac{x^4+x^3+x}{x^2+1}=x^2+x+1+\frac{1}{x^2+1}$.

Satz: Sei C zyklischer Code der Länge n über dem Körper A. Dann gilt:

1. C entspricht in eindeutiger Weise einem Ideal in $A[x]/(x^n-1)$.
2. Es existiert genau ein normiertes Polynom (normiert heißt: Führungskoeffizient ist 1) $g(x) \neq 0$ minimalen Grades in C. Dieses Polynom heißt **Generatorpolynom** von C.
3. Das Generatorpolynom $g(x)$ ist Teiler von x^n-1, d.h. $\exists\, h(x) \in A[x] : g(x) \cdot h(x) = x^n - 1$, $h(x)$ heißt **Kontrollpolynom** von C.
4. Jedes Codewort/-polynom $c(x)$ kann eindeutig in der Form $c(x) = f(x) \cdot g(x)$ geschrieben werden, wobei $\operatorname{grad} f(x) < n - \operatorname{grad} g(x)$ gilt, d.h. $g(x)$ teilt jedes Codepolynom und umgekehrt kann man $g(x)$ mit beliebigem Polynom mit Grad $< (n - \operatorname{grad} g(x))$ multiplizieren und erhält wieder ein Codepolynom.

Beweis: ad 1) Sei C ein zyklischer Code. Wir definieren

$$I := \{c(x) + p(x) \cdot (x^n - 1) \mid c(x) \in C, p(x) \in A[x]\}.$$

Man rechnet direkt nach, dass I ein Ideal in $A[x]$ ist mit $(x^n-1) \subseteq I$, exemplarisch zeigen wir $i(x) = c(x) + p(x) \cdot (x^n-1) \in I$ und $a(x) \in A[x]$ impliziert $i(x) \cdot a(x) \in I$:

$$i(x) \cdot a(x) = \underbrace{c(x)a(x)}_{=q(x)(x^n-1)+r(x)} + p(x)a(x)(x^n-1) = r(x) + (q(x) + p(x)a(x))(x^n-1),$$

wobei das Restpolynom $r(x)$ von $c(x)a(x)$ bei Division durch x^n-1 auf Grund der Zyklizität in C liegt. Damit hat $i(x) \cdot a(x)$ die Gestalt „Codepolynom plus Vielfaches von x^n-1“, liegt also in I.

Nach einem bekannten Isomorphiesatz der Algebra gilt

$$A[x]/I \cong (A[x]/(x^n-1))/(I/(x^n-1)),$$

also kann man dem Ideal I in $A[x]$ das Ideal $I/(x^n-1)$ in $A[x]/(x^n-1)$ zuordnen.

Umgekehrt kann jedem Ideal J in $A[x]/(x^n-1)$ das Ideal

$$I := \{i(x) \in A[x] \mid i(x) + (x^n-1) \in J\}$$

in $A[x]$ zugeordnet werden, und $I \cap \{p(x) \mid \operatorname{grad} p(x) \leq n-1\}$ liefert einen zyklischen Code.

ad 2) $A[x]$ ist als Polynomring über einem Körper ein Euklidischer Ring und damit ein Hauptidealring, damit wird das Ideal I von einem normierten Polynom minimalen Grades $g(x)$ erzeugt. Dieses $g(x)$ ist eindeutig bestimmt: Ang. $\exists\, g_1(x), g_2(x)$ normiert, minimaler Grad

$$\left.\begin{array}{l} g_1(x) = 1\,x^l + c_{l-1}x^{l-1} + \cdots + c_0 \\ g_2(x) = 1\,x^l + d_{l-1}x^{l-1} + \cdots + d_0 \end{array}\right\} \in C$$

C linear $\Rightarrow g_1(x) - g_2(x) \in C$
$g_1(x) - g_2(x) = (c_{l-1} - d_{l-1})\,x^{l-1} + \cdots + (c_0 - d_0)$

$\operatorname{grad}(g_1 - g_2) < \operatorname{grad} g_1$, $g_1 - g_2 \in C \Rightarrow g_1(x) - g_2(x) = 0 \Rightarrow g_1(x) = g_2(x)$

ad 3) Aus $I = (g(x)) \supseteq (x^n-1)$ folgt sofort $g(x) | (x^n-1)$, oder man rechnet nach:

$g(x) \mid x^n - 1$: Division mit Rest für $a(x) = x^n - 1$, $b(x) = g(x)$
$x^n - 1 = g(x)\,q(x) + r(x) \quad$ | Reduktion $(x^n \equiv 1)$
$0 \equiv g(x)\,q(x) + r(x)$

$$\underbrace{\underbrace{g(x)}_{\in C}\,q(x)}_{C \text{ zyklisch} \Rightarrow \in C} \equiv -\underbrace{r(x)}_{\in C} \Rightarrow r(x) \in C$$

Ang. $r(x) \neq 0$: man kann $r(x)$ normieren (mit Faktor multiplizieren sodass Führungskoeffizient 1 wird) $\rightarrow \tilde{r}(x)$

$\tilde{r}(x)$ normiert, kleinerer Grad als $g(x)$, $\tilde{r}(x) \in C \rightarrow$ Widerspruch zur Wahl von $g(x)$!

Also ist $r(x) = 0 \Rightarrow x^n - 1 = g(x)\, q(x) \Rightarrow g(x) \mid x^n - 1$.

ad 4) Für beliebiges $f(x) \in A[x]$ mit $\operatorname{grad} f(x) < n - \operatorname{grad} g(x)$ gilt $g(x) \cdot f(x) \in C$ weil C zyklisch und $\operatorname{grad} g(x) + \underbrace{\operatorname{grad} f(x)}_{< n - \operatorname{grad} g(x)} < n$.

Umgekehrt gilt für jedes $c(x) \in C$, dass $g(x) \mid c(x)$ weil $I = (g(x))$.

□

Beispiel:

Code mit $n = 4$ und Generatormatrix $\mathcal{G} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}$

$C = \Big\{\underbrace{0000}_{0}, \underbrace{1010}_{1+x^2}, \underbrace{0101}_{x+x^3}, \underbrace{1111}_{1+x+x^2+x^3}\Big\}$, also ist das Generatorpolynom $g(x) = x^2 + 1$.

$g(x) \cdot f(x) : \operatorname{grad} f(x) < n - \operatorname{grad} g(x) = 4 - 2 = 2 \Rightarrow f(x) \in \{0, 1, x, x+1\}$:

$$\begin{aligned} g(x) \cdot 0 &= 0 \\ g(x) \cdot 1 &= g(x) \\ g(x) \cdot x &= x^3 + x \\ g(x) \cdot (x+1) &= x^3 + x^2 + x + 1 \end{aligned}$$

$x^n - 1 = x^4 + 1$, $(x^4 - 1) : g(x) = (x^4 + 1) : (x^2 + 1) = \underbrace{x^2 + 1}_{h(x)}$, 0 Rest.

Beispiel:

$C = \{0000, 0011, 0101, 0110, 1001, 1010, 1100, 1111\}$

Behauptung: C zyklisch

1100 beginnt von hinten mit 2 Nullen, kein anderes Codewort hat mehr Nullen am Ende $\Rightarrow$ $g(x) = 1 + x$

$(x^4 + 1) : (x + 1) = x^3 + x^2 + x + 1, \quad 0$ Rest
(wenn ein Rest bleibt: sicher kein zyklischer Code)

$$\begin{aligned} g(x) \cdot 0 &= 0 \\ g(x) \cdot 1 &= x + 1 = g(x) \\ g(x) \cdot x &= x^2 + x \\ g(x) \cdot (x+1) &= x^2 + 1 \\ g(x) \cdot x^2 &= x^3 + x^2 \\ g(x) \cdot (x^2 + 1) &= x^3 + x^2 + x + 1 \\ g(x) \cdot (x^2 + x) &= x^3 + x \\ g(x) \cdot (x^2 + x + 1) &= x^3 + 1 \end{aligned}$$

$\Rightarrow C$ ist zyklisch.

Um alle zyklischen Codes der Länge n zu bestimmen, muss man alle normierten Teiler von $x^n - 1$ als Generatorpolynome bestimmen.

Der Polynomring $A[x]$ über dem Körper A ist (so wie $\mathbb{Z}$) ein Euklidischer Ring und damit ein ZPE-Ring. Folglich lässt sich jedes Polynom eindeutig (bis auf die Reihenfolge der Faktoren) als Produkt von irreduziblen Polynomen (entsprechen den Primzahlen in $\mathbb{Z}$) schreiben.

$p(x) \in A[x]$ heißt **irreduzibel**, wenn gilt: $p(x) = a(x) \cdot b(x)$ mit $a(x), b(x) \in A[x] \Rightarrow \operatorname{grad} a(x) = 0$ oder $\operatorname{grad} b(x) = 0$, d.h. $a(x)$ oder $b(x)$ sind eine Konstante. Ein irreduzibles Polynom lässt sich also nicht als Produkt von Polynomen mit echt kleinerem Grad darstellen.

Lemma: $x - a \mid p(x) \Leftrightarrow p(a) = 0$

Beweis: $\Rightarrow$: $p(x) = (x-a)q(x) \Rightarrow p(a) = \underbrace{(a-a)}_{=0} q(a) = 0$.

$\Leftarrow$: Division mit Rest von $p(x)$ durch $(x-a)$:
$p(x) = (x-a)q(x) + r(x)$ wobei $r(x) = 0$ oder $\operatorname{grad} r(x) < \operatorname{grad}(x-a) = 1 (\Rightarrow r(x) = C)$
Also $p(x) = (x-a)q(x) + C$, und daher
$p(a) = 0 = \underbrace{(a-a)}_{=0} q(a) + C \Rightarrow 0 = C \Rightarrow r(x) = 0 \Rightarrow (x-a) \mid p(x)$. □

Folgerung: Ein Polynom vom Grad 2 oder 3 ist genau dann irreduzibel über A, wenn es keine Nullstellen in A besitzt.

Beispiel: $n = 4$, $A = \mathbb{Z}_2$: $x^n - 1 = x^4 - 1 = (x-1)^4$

Mögliche Generatorpolynome $g(x)$:

1	$\leftrightarrow C$ enthält alle Vielfachen von $1 \Rightarrow C = \mathbb{Z}_2^4$
$(x-1)$	$\leftrightarrow C$ enthält alle Worte mit gerader Parität
$(x-1)^2 = x^2 + 1$	$\leftrightarrow C = \{0000, 1010, 0101, 1111\}$
$(x-1)^3 = x^3 + x^2 + x + 1$	$\leftrightarrow C = \{0000, 1111\}$
$(x-1)^4 = x^4 - 1$	$\leftrightarrow C = \{0000\}$

Anmerkung: Das Polynom $x-1$ ist immer ein Generatorpolynom, der zugehörige zyklische Code enthält alle Worte mit Summe der Komponenten gleich 0.

Dimension eines zyklischen Codes: $k = n - \operatorname{grad} g(x)$, d.h. $\operatorname{grad} g(x) = n - k$:

Sei das Generatorpolynom vom Grad r: $g(x) = 1x^r + g_{r-1}x^{r-1} + g_{r-2}x^{r-2} + \cdots + g_0$.

Da $\underbrace{f(x)}_{\operatorname{grad} < n-r} \mapsto \underbrace{f(x)\,\underbrace{g(x)}_{\operatorname{grad} r}}_{\operatorname{grad} < n}$ eine bijektive Zuordnung zwischen Nachrichtenwörtern in A^k und Codewörtern in A^n, ist folglich $k = n - r = n - \operatorname{grad} g(x) = \operatorname{grad} h(x)$.

Generatormatrix eines zyklischen Codes:

$$\mathcal{G} = \left.\begin{pmatrix} g_0 & g_1 & g_2 & \cdots & g_{n-k-1} & 1 & 0 & \cdots & 0 \\ 0 & g_0 & g_1 & \cdots & \cdots & g_{n-k-1} & 1 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & & & \ddots & \ddots & \\ 0 & \cdots & 0 & g_0 & g_1 & g_2 & \cdots & g_{n-k-1} & 1 \end{pmatrix}\right\} k \text{ Zeilen}$$

Für das **Kontrollpolynom** $h(x)$ gelten folgende Eigenschaften:

1) $c(x) \in C \Leftrightarrow c(x)\,h(x) \equiv 0 \bmod(x^n - 1)$:
$c(x) \in C \Leftrightarrow g(x) \mid c(x) \Leftrightarrow \underbrace{h(x)\,g(x)}_{x^n-1} \mid h(x)\,c(x)$

2) **Syndrom** mit Hilfe des Kontrollpolynoms $h(x)$:

$$s_h(p(x)) = p(x) \cdot h(x) \bmod(x^n - 1)$$

$p(x) + C = q(x) + C \Leftrightarrow p(x) - q(x) \in C \Leftrightarrow g(x) \mid p(x) - q(x) \Leftrightarrow$
$h(x) \cdot g(x) = x^n - 1 \mid (h(x) \cdot p(x) - h(x) \cdot q(x))$
Vorteil dieser Methode: dieses Syndrom kann mit Schieberegistern sehr schnell berechnet werden

3) Alternative Syndromberechnung mit Hilfe des Generatorpolynoms $g(x)$:

$$s_g(p(x)) = p(x) \bmod g(x)$$

$p(x) \in C \Leftrightarrow g(x) \mid p(x) \Leftrightarrow p(x) \equiv 0 \bmod g(x)$ und s_g ist linear;
Vorteil dieser Methode: die möglichen Syndrome sind alle Polynome mit Grad $< n - k$ (bei s_h sind es alle Vielfachen von $h(x)$ mit Grad $< n$). Bei der „händischen" Berechnung ausgehend von den Syndromen der Fehlerworte mit kleinem Gewicht sieht man gleich, welche Syndrome noch fehlen und kann diese linear aus bereits berechneten Syndromen zusammensetzen.

Beispiel: $n = 4$, $g(x) = x^2 + 1 \Rightarrow h(x) = x^2 + 1$

$$C = \Big\{\underbrace{0}_{0000}, \underbrace{x^2+1}_{1010}, \underbrace{x^3+x}_{0101}, \underbrace{x^3+x^2+x+1}_{1111}\Big\}$$

Syndrome nach 2): $s_h(p(x)) = p(x) \cdot h(x) \bmod (x^4 - 1)$

$p(x)$	Syndrom $s_h(p(x))$
0	0
1	$x^2 + 1$
x	$x^3 + x$
x^2	$x^2(x^2+1) = x^4 + x^2 \equiv x^2 + 1$ $(\rightarrow$ dieses Syndrom gibt es schon$)$
x^3	$x^3(x^2+1) = x^5 + x^3 \equiv x^3 + x$ $(\rightarrow$ dieses Syndrom gibt es schon$)$
$x + 1$	$x^3 + x^2 + x + 1$

Syndrome nach 3): $s_g(p(x)) = p(x) \bmod g(x)$

$g(x) = x^2 + 1 \equiv 0 \Rightarrow x^2 \equiv 1$, $x^3 \equiv x$

$p(x)$	Syndrom $s_g(p(x))$
0	0
1	1
x	x
x^2	1 $(\rightarrow$ dieses Syndrom gibt es schon$)$
x^3	x $(\rightarrow$ dieses Syndrom gibt es schon, es fehlt noch das Syndrom $x + 1)$
$x + 1$	$x + 1$ (oder: $p(x) = x^3 + x^2$ liefert auch das Syndrom $x + 1)$

8 Endliche Körper

In diesem Abschnitt sammeln wir ein paar grundlegende Eigenschaften von endlichen Körpern bzw. Körpern im Allgemeinen. Vorausgesetzt werden hier Grundkenntnisse über Körper wie sie etwa in der LVA Algebra (für Techn. Math.) vermittelt werden.

Satz: Zu jeder Primzahlpotenz p^m gibt es (bis auf Isomorphie) genau einen endlichen Körper mit p^m Elementen, und alle endlichen Körper sind von dieser Gestalt.

Bezeichnung für Körper mit p^m Elementen: $\mathbb{F}_{p^m} = \mathrm{GF}(p^m)$ (GF ... Galois-Feld)

$\mathbb{F}_q$ mit $q = p^m$ kann aufgefasst werden als der Zerfällungskörper des Polynoms $T^q - T$, es gilt sogar

$$T^q - T = \prod_{\alpha \in \mathbb{F}_q} (T - \alpha)$$

d. h. $\mathbb{F}_q$ besteht genau aus den Nullstellen des Polynoms $T^q - T$.

Konstruktion von $\mathbb{F}_{p^m}$, p prim, $m \geq 1$:

- $m = 1$: $p^m = p$, $\mathbb{F}_p = \mathbb{Z}_p$ ... Restklassenring mod p
- $m > 1$: man sucht ein irreduzibles Polynom $s(T)$ über $\mathbb{Z}_p$ mit $\operatorname{grad} s(T) = m$ und rechnet mit Polynomen aus $\mathbb{Z}_p[T]$ modulo $s(T)$

Irreduzible Polynome vom Grad $m > 1$ über $\mathbb{Z}_p$ existieren immer:

Satz: Die Anzahl der normierten irreduziblen Polynome über dem Körper $\mathbb{F}_q$ vom Grad $m \in \mathbb{N}$ ist gegeben durch die Formel

$$N_q(m) = \frac{1}{m} \sum_{t|m} \mu(t) \cdot q^{m/t}.$$

Hierbei bezeichnet $\mu(t)$ die Möbius-Funktion definiert durch:

$$\mu(t) = \begin{cases} 1 & , \ t = 1 \\ (-1)^k & , \ t = p_1 \cdot \ldots \cdot p_k \ (p_i \text{ prim und paarweise verschieden}) \\ 0 & , \ \text{sonst} \end{cases}$$

Beweis: Durch die Zerlegung von $T^{q^m} - T$ in irreduzible Faktoren zeigt man die Formel

$$q^m = \sum_{t|m} t \cdot N_q(t),$$

daraus schließt man mit der Möbiusschen Umkehrformel auf die obige Formel für $N_q(m)$ (Übung). □

Durch eine einfache Abschätzung sieht man $N_q(m) \geq 1$ (Übung).

Beispiel: $\mathbb{F}_4$

$4 = 2^2$, gesucht ist daher in $\mathbb{Z}_2$ ein irreduzibles Polynom vom Grad $m = 2$

Alle Polynome vom Grad 2: $\underbrace{T^2}_{T \cdot T}, \underbrace{T^2+1}_{(T+1)^2}, \underbrace{T^2+T}_{T(T+1)}, \underbrace{T^2+T+1}_{\text{irreduzibel}}$

Bemerkung: Ein Polynom $s(T)$ vom Grad 2 oder 3 ist irreduzibel über dem Körper $A \Leftrightarrow s(T)$ hat keine Nullstelle in A.

$s(T) = T^2 + T + 1$

Rechnen modulo $s(T)$: $T^2 + T + 1 \equiv 0 \Rightarrow T^2 = -T - 1 = T + 1$
$T^3 \equiv T^2 + T = T + 1 + T = 1$

$\mathbb{F}_4 = \{0, 1, T, T+1\} = \{0, \underbrace{1}_{T^3}, T, T^2\}$

$+$	0	1	T	$T+1$
0	0	1	T	$T+1$
1	1	0	$T+1$	T
T	T	$T+1$	0	1
$T+1$	$T+1$	T	1	0

$\cdot$	0	1	T	$T+1$
0	0	0	0	0
1	0	1	T	$T+1$
T	0	T	$T+1$	1
$T+1$	0	$T+1$	1	T

Darstellung der Elemente in $\mathbb{F}_4$: $\{0, 1, T, T+1\}$ oder $\{0, 1, T, T^2\}$

Aus der zweiten Darstellung sieht man: T (bzw. genauer müsste man sagen die Restklasse $T + (T^2 + T + 1)$) ist ein **primitives Element** in $\mathbb{F}_4$, das heißt: jedes Element $\neq 0$ läßt sich als Potenz von T darstellen.

Satz: Jeder endliche Körper besitzt ein primitives Element.

Beweis: Wir zeigen zunächst:
BH 1. Existieren in einer kommutativen Gruppe $(G, \cdot, 1, ^{-1})$ Elemente α, β mit $\mathrm{ord}(\alpha) = k$, $\mathrm{ord}(\beta) = l$, so $\exists \gamma \in G$: $\mathrm{ord}(\gamma) = \mathrm{kgV}(k, l)$:

Es existieren teilerfremde Zahlen $\bar{k}, \bar{l}$, sodass $\mathrm{kgV}(k, l) = \bar{k} \cdot \bar{l}$ mit $\bar{k}|k$ und $\bar{l}|l$: seien die Primfaktorzerlegungen von k und l gegeben durch $k = \prod_{i=1}^{n} p_i^{e_i}$, $l = \prod_{i=1}^{n} p_i^{f_i}$, $0 \leq e_i, f_i$, und seien die Primfaktoren $p_1, \ldots, p_n$ so geordnet, dass

$$\mathrm{kgV}(k, l) = \prod_{i=1}^{n} p_i^{\max\{e_i, f_i\}} = \prod_{i=1}^{r} p_i^{e_i} \prod_{i=r+1}^{n} p_i^{f_i}.$$

Dann erfüllen $\bar{k} := \prod_{i=1}^{r} p_i^{e_i}$ und $\bar{l} := \prod_{i=r+1}^{n} p_i^{f_i}$ offensichtlich das Gewünschte.

Wir betrachten jetzt die Elemente $\bar{\alpha} = \alpha^{k/\bar{k}}$ und $\bar{\beta} = \beta^{l/\bar{l}}$. Für diese gilt offensichtlich $\mathrm{ord}(\bar{\alpha}) = \bar{k}$ und $\mathrm{ord}(\bar{\beta}) = \bar{l}$.

Weiters gilt $\mathrm{ord}(\bar{\alpha} \cdot \bar{\beta}) = \bar{k} \cdot \bar{l} = \mathrm{kgV}(k, l)$: $(\bar{\alpha} \cdot \bar{\beta})^{\bar{k} \cdot \bar{l}} = (\underbrace{\bar{\alpha}^{\bar{k}}}_{=1})^{\bar{l}} \cdot (\underbrace{\bar{\beta}^{\bar{l}}}_{=1})^{\bar{k}} = 1$, und aus $(\bar{\alpha} \cdot \bar{\beta})^m = 1$ folgt $\bar{\alpha}^m = \bar{\beta}^{-m}$; durch Bilden der $\bar{k}$-ten Potenz erhält man $\bar{\beta}^{-m\bar{k}} = 1 \Rightarrow \bar{l}|m \cdot \bar{k} \Rightarrow \bar{l}|m$; analog erhält man durch Bilden der $\bar{l}$-ten Potenz, dass $\bar{\alpha}^{-m\bar{l}} = 1 \Rightarrow \bar{k}|m \cdot \bar{l} \Rightarrow \bar{k}|m$, woraus schließlich wegen $\mathrm{ggT}(\bar{k}, \bar{l}) = 1$ folgt, dass $\bar{k} \cdot \bar{l}|m$. Damit ist BH 1 bewiesen.

Sei jetzt k die maximale Ordnung eines Elementes in $\mathbb{F}_q \setminus \{0\}$. Aus BH 1 folgt sofort, dass alle Ordnungen von Elementen $\alpha \in \mathbb{F}_q \setminus \{0\}$ ein Teiler von k sein müssen. Das bedeutet weiters, dass alle Elemente aus $\mathbb{F}_q \setminus \{0\}$ Nullstellen des Polynoms $p(T) = T^k - 1$ sind. Weil das Polynom $p(T)$ als Polynom über einem Körper höchstens $\mathrm{grad}\, p(T) = k$ Nullstellen haben kann, folgt $k \geq q - 1$. Andererseits gilt nach dem Satz von Lagrange in $(\mathbb{F}_q \setminus \{0\}, \cdot)$, dass $k|(q-1)$, woraus wir insgesamt $k = q - 1$ erhalten, d. h. die multiplikative Gruppe des Körpers $\mathbb{F}_q$ ist zyklisch, es gibt also ein primitives Element. □

Beispiel: Gesucht sind primitive Elemente in $\mathbb{Z}_5 = \{0, 1, 2, 3, 4\}$

0 ... nicht primitiv (alle Potenzen von 0 sind 0)
1 ... nicht primitiv (alle Potenzen von 1 sind 1)
2: $2^1 = 2, 2^2 = 4, 2^3 = 3, 2^4 = 1$ ... primitiv
3: $3^1 = 3, 3^2 = 4, 3^3 = 2, 3^4 = 1$ ... primitiv
4: $4^2 = 1$ ... nicht primitiv

Definition: Ein irreduzibles Polynom $s(T)$ über dem endlichen Körper A heißt **primitiv**, wenn das Element $T + (s(T))$ im Körper $A[T]/(s(T))$ primitiv ist.

Beispiel: $\mathbb{F}_9$
$9 = 3^2 \rightarrow$ Primkörper ist $\mathbb{Z}_3$, $m = 2$

1) $s(T) = T^2 + 2T + 2$ ($s(0) = 2$, $s(1) = 2$, $s(2) = 1$ ... keine Nullstelle $\Rightarrow$ irreduzibel)

Rechnen modulo $s(T)$: $T^2 + 2T + 2 \equiv 0 \Rightarrow T^2 \equiv -2T - 2 = T + 1$

Ist $s(T)$ primitiv?

$T^1 = T$
$T^2 = T + 1$
$T^3 = T^2 + T = T + 1 + T = 2T + 1$
$T^4 = 2T^2 + T = 2(T + 1) + T = 2 = -1$
$T^5 = 2T$
$T^6 = 2T^2 = 2(T + 1) = 2T + 2$
$T^7 = 2T^2 + 2T = 2(T + 1) + 2T = T + 2$
$T^8 = (T^4)^2 = 1$
$\Rightarrow s(T)$ ist primitiv ($\mathbb{F}_9 = \{0, 1, 2, T, T + 1, T + 2, 2T, 2T + 1, 2T + 2\}$

So wird effektiv multipliziert: $(T + 2) \cdot (2T + 1) = T^7 \cdot T^3 = T^{10} = T^2 = T + 1$

2) $s_1(T) = T^2 + 1$ ist auch irreduzibel über $\mathbb{Z}_3$ ($s_1(0) = 1$, $s(1) = 2$, $s(2) = 2$), allerdings ist $s_1(T)$ nicht primitiv:

$T^1 = T$, $T^2 = -1 = 2$, $T^3 = 2T$, $T^4 = (T^2)^2 = (-1)^2 = 1$, $T^5 = T$, ...

Lemma: Ein irreduzibles Polynom $s(T) \in \mathbb{F}_q[T]$ vom Grad m ist primitiv genau dann, wenn $s(T)$ teilt nicht das Polynom $T^r - 1$ für alle $r|(q^m - 1)$ und $r < q^m - 1$.

Beweis: Es gilt $T^r \equiv 1$ im Körper $\mathbb{F}_q[T]/(s(T)) \cong \mathbb{F}_{q^m}$ genau dann, wenn $T^r - 1 \equiv 0 \bmod s(T)$, also $s(T)|(T^r - 1)$. Weil die Ordnung von T in der multiplikativen Gruppe von $\mathbb{F}_{q^m}$ ein Teiler der Gruppenordnung sein muss, ist diese Bedingung nur für $r|(q^m - 1)$ zu überprüfen. Damit entspricht die Primitivität von $s(T)$ genau der angegebenen Bedingung. □

Sei $(L, +, \cdot)$ ein Körper und $K \subseteq L$ ein **Unterkörper** (d. h. K bildet mit den von L auf K eingeschränkten Operationen $+$ und $\cdot$ einen Körper). Dafür schreiben wir zur Abkürzung $K \leq L$ und nennen gleichbedeutend L einen **Erweiterungskörper** von K.

Beispiel: $\mathbb{R}$ ist ein Unterkörper von $(\mathbb{C}, +, \cdot)$, d. h. $\mathbb{R} \leq \mathbb{C}$.

Ist $K \leq L$, so können wir den größeren Körper L in natürlicher Weise als Vektorraum über den kleineren Körper K auffassen. Die Dimension dieses Vektorraums wird mit $\dim_K L$ bezeichnet.

Beispiel: $\mathbb{C}$ als Vektorraum über $\mathbb{R}$ hat Dimension 2, d. h. $\dim_{\mathbb{R}} \mathbb{C} = 2$.

Für endliche Körper gilt: $L = \mathbb{F}_{p^m}$ hat als Unterkörper genau die Körper $K = \mathbb{F}_{p^r}$ mit $r|m$: Wegen der Eindeutigkeit der Darstellung von Vektoren aus L als Linearkombination von Elementen einer Basis mit Koeffizienten aus K gilt mit $k = \dim_K L$: $p^m = |L| = |K|^k = (p^r)^k = p^{rk}$, also folgt $r|m$. Andererseits folgt aus $r|m$ sofort $(p^r - 1)|(p^m - 1)$, und daraus erhält man dann $(T^{p^r} - T)|(T^{p^m} - T)$. Da, wie bereits erwähnt, $\mathbb{F}_{p^i}$ genau aus den Nullstellen von $T^{p^i} - T$ besteht, impliziert das $\mathbb{F}_{p^r} \leq \mathbb{F}_{p^m}$.

Beispiel: $\mathbb{F}_{16} = \mathbb{F}_{2^4}$ hat genau die (echten) Unterkörper $\mathbb{F}_2$ und $\mathbb{F}_4$; $\mathbb{F}_8$ ist kein Unterkörper von $\mathbb{F}_{16}$.

Sei $K \leq L$, $\dim_K L =: k < \infty$ und $\alpha \in L$. Dann gibt es ein Polynom $p(T) \in K[T]$ mit $\operatorname{grad} p(T) \leq k$ mit $p(\alpha) = 0$: Die $k + 1$ Elemente $1, \alpha, \alpha^2, \ldots, \alpha^k$ sind linear abhängig über K.

Definition: Das normierte Polynom kleinsten Grades $m_\alpha(T) \in K[T]$ mit $m_\alpha(\alpha) = 0$ heißt **Minimalpolynom** von α über K.

Das Minimalpolynom $m_\alpha(T)$ von α über K ist eindeutig bestimmt und es gilt: ist $p(T) \in K[T]$ mit $p(\alpha) = 0$, so folgt $m_\alpha(T)|p(T)$ in $K[T]$ (Polynomdivision mit Rest).

Das Minimalpolynom $m_\alpha(T)$ ist irreduzibel über K (aus einer Zerlegung von $m_\alpha(T)$ in zwei Polynome aus $K[T]$ mit kleinerem Grad folgt durch Einsetzen von α sofort ein Widerspruch zur Minimalität des Grades von $m_\alpha(T)$).

9 Reed-Solomon Codes

Alphabet $A = \mathbb{F}_q$ endlicher Körper, $q = p^m$, p prim

sei T **primitives Element** in $\mathbb{F}_q$, d.h. $A = \{0, 1, T, T^2, T^3, \ldots, T^{p^m-2}\}$, $T^{p^m-1} = 1$, d. h. wir rechnen multiplikativ in der zyklischen Grupppe $(\mathbb{F}_q \setminus \{0\}, \cdot)$ mit $q - 1 = p^m - 1$ Elementen

wir betrachten in der Folge Codes mit der **festen Länge** $n = q - 1 = p^m - 1$, d. h. die Länge ist durch das Alphabet gegeben (Nachteil!)

in $A = \mathbb{F}_q$ haben wir die Zerlegung

$$x^n - 1 = x^{p^m-1} - 1 = (x-1)(x-T)(x-T^2)\ldots(x-T^{p^m-2})$$

daraus folgt,dass alle Elemente aus $\mathbb{F}_q \setminus \{0\}$ Nullstellen von $x^n - 1$ sind

Definition: Ein Code über $\mathbb{F}_q$ der Länge $n = q - 1$ mit Generatorpolynom
$g(x) = \underbrace{(x - T^i)(x - T^{i+1})\ldots(x - T^{i+n-k-1})}_{\text{Grad } n-k}$ (i beliebig) heißt Reed-Solomon-Code (RS Code).

Wesentlich ist: das Generatorpolynom muss $n - k$ aufeinanderfolgende Potenzen von T als Nullstellen haben.

Satz: Mit obigen Bezeichnungen gilt für den zugehörigen Reed-Solomon-Code C:

1. $p(x) \in C \Leftrightarrow p(T^i) = p(T^{i+1}) = \cdots = p(T^{i+n-k-1}) = 0$
2. $\mathcal{H} = \begin{pmatrix} 1 & T^i & T^{2i} & \ldots & T^{(n-1)i} \\ 1 & T^{i+1} & T^{2(i+1)} & \ldots & T^{(n-1)(i+1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & T^{i+n-k-1} & T^{2(i+n-k-1)} & \ldots & T^{(n-1)(i+n-k-1)} \end{pmatrix}$
   ist eine Kontrollmatrix für den Reed-Solomon-Code.
3. für die Minimaldistanz gilt $d = n - k + 1$ (MDS-Code, optimal im Sinne der Singleton-Schranke)

Beweis:

1. $g(x) = (x - T^i)(x - T^{i+1})\ldots(x - T^{i+n-k-1})$
$p(x) \in C \Leftrightarrow g(x) \mid p(x) \Leftrightarrow T^i, T^{i+1}, \ldots, T^{i+n-k-1}$ müssen Nullstellen von $p(x)$ sein.

2. $p(x) \in C \Leftrightarrow p(T^{i+j}) = 0 \quad \forall j = 0, 1, \ldots, n-k-1$
$\underbrace{p(x)}_{\leftrightarrow \vec{p}} \in C \Leftrightarrow \vec{p}\,\mathcal{H}^T = \vec{0}$
$\vec{p}\,\mathcal{H}^T = \big(p(T^i), p(T^{i+1}), \ldots, p(T^{i+n-k-1})\big) = \vec{0} \Leftrightarrow p(x) \in C$

3. d Minimaldistanz $\Leftrightarrow$ je $d - 1$ Spalten der Kontrollmatrix sind linear unabhängig und es existieren d Spalten die linear abhängig sind. Man zeigt: beliebige $n - k$ Spalten von $\mathcal{H}$ sind linear unabhängig ($\rightarrow$ Vandermondsche Determinante, Übungen) $\square$

Beispiele: 1) $A = \mathbb{Z}_5$, primitives Element 2, Generatorpolynom

$$g(x) = (x-1)(x-2) = x^2 + 2x + 2,$$

$n = q - 1 = 4$, $k = n - \operatorname{grad} g(x) = 4 - 2 = 2$, $d = n - k + 1 = 3$, also $(4, 2, 3)$-Code über $\mathbb{Z}_5$, ist MDS Code (nicht 1-perfekt), Generatormatrix $G = \begin{pmatrix} 2 & 2 & 1 & 0 \\ 0 & 2 & 2 & 1 \end{pmatrix} \sim \begin{pmatrix} 1 & 0 & 2 & 2 \\ 0 & 1 & 1 & 3 \end{pmatrix}$,

Kontrollmatrix $H = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 2^2 & 2^3 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 3 \end{pmatrix}$,

Kontrollpolynom $h(x) = (x - 2^2)(x - 2^3) = x^2 + 3x + 2$

2) $A = \mathbb{F}_8 = \mathbb{F}_{2^3}$, primitives Element T mit $T^3 = T + 1$, $T^4 = T^2 + T$, $T^5 = T^2 + T + 1$, $T^6 = T^2 + 1$, $T^7 = 1$, Generatorpolynom

$$g(x) = (x - 1)(x - T)(x - T^2)(x - T^3) = x^4 + T^2x^3 + T^5x^2 + T^5x + T^6,$$

$n = q - 1 = 7$, $k = n - \operatorname{grad} g(x) = 7 - 4 = 3$, $d = n - k + 1 = 5$, also $(7, 3, 5)$-Code über $\mathbb{F}_8$, ist MDS Code (nicht 2-perfekt):
Vergleich der Werte in der Hamming-Schranke: $n - k = 4$ und $\log_q(\sum_{i=0}^{t} \binom{n}{i}(q-1)^i) = \log_8(\sum_{i=0}^{2} \binom{7}{i}7^i) = \log_8(1079) < 4$.

Generatormatrix $G = \begin{pmatrix} T^6 & T^5 & T^5 & T^2 & 1 & 0 & 0 \\ 0 & T^6 & T^5 & T^5 & T^2 & 1 & 0 \\ 0 & 0 & T^6 & T^5 & T^5 & T^2 & 1 \end{pmatrix}$,
Anzahl der Codeworte $8^3 = 512$, $c(x) \in C \Leftrightarrow c(1) = c(T) = c(T^2) = c(T^3) = 0$,

Kontrollmatrix $H = \begin{pmatrix} 1 & 1 & \dots & 1 \\ 1 & T & \dots & T^6 \\ 1 & T^2 & \dots & T^{2\cdot 6} \\ 1 & T^3 & \dots & T^{3\cdot 6} \end{pmatrix}$,
Kontrollpolynom $h(x) = (x - T^4)(x - T^5)(x - T^6) = x^3 + T^2x^2 + x + T = (x^7 - 1)/g(x)$

erweiterter Reed-Solomon-Code: wird beim Generatorpolynom eines RS-Codes

$$g(x) = (x - T^i)(x - T^{i+1}) \dots (x - T^{i+n-k-1})$$

$i = 1$ und $k \geq 1$ gewählt, dann kann man durch Hinzufügen einer geeigneten Prüfstelle die Länge und die Minimaldistanz um 1 erhöhen:

Satz: Sei C ein (n, k, d) RS-Code mit Generatorpolynom $g(x) = (x - T)(x - T^2) \dots (x - T^{n-k})$, $k \geq 1$. Dann ist der Code

$$\hat{C} := \{c_0c_1 \dots c_n \mid c_0c_1 \dots c_{n-1} \in C \text{ und } \sum_{j=0}^{n} c_j = 0\}$$

ein $(n + 1, k, d + 1)$ Code (erweiterter RS-Code).

Beweis: Sei $c(x) \in C$ mit minimalem Gewicht d. Wir zeigen, dass gilt $c(1) = \sum_{j=0}^{n-1} c_j \neq 0$:

Aus $c(x) \in C$ folgt $c(x) = a(x) \cdot g(x)$, folglich ist $c(1) = a(1) \cdot g(1)$. Auf Grund der Voraussetzungen gilt $g(1) \neq 0$. Wäre $a(1) = 0$, so wäre $c(x)$ ein Vielfaches von $g_1(x) := (x - 1) \cdot g(x)$, welches Generatorpolynom eines RS-Codes mit Parametern $(n, k - 1, d + 1)$ ist, also wäre das Gewicht von $c(x)$ mindestens $d + 1$, Widerspruch. Also gilt $c(1) \neq 0$.

Daraus folgt nun, dass die an $c(x)$ angehängte Prüfstelle $c_n \neq 0$, also das zugehörige Wort $c_0c_1 \dots c_n \in \hat{C}$ Gewicht $d + 1$ hat. □

Folgerung: Erweiterte RS-Codes sind ebenfalls MDS Codes.

Besondere Bedeutung haben Reed-Solomon-Codes mit $q = 2^m$:

jedes Element aus $\mathbb{F}_{2^m} \cong \mathbb{Z}_2[T]/(p(T))$ ($p(T)$ irreduzibles Polynom über $\mathbb{Z}_2$ vom Grad m) kann dargestellt werden als Polynom vom Grad $\leq m - 1$ und entspricht damit einem binären m-Tupel

bei festgehaltenem $p(T)$ wird durch diese Zuordnung aus einem (n, k, d)-Code C über $\mathbb{F}_{2^m}$ ein $(nm, km, \geq d)$ Binärcode C_2; die untere Schranke d für die Minimaldistanz von C_2 kann durch

eine kleine Modifikation verbessert werden zur unteren Schranke $2d$: ist $c_0c_1\dots c_{n-1} \in C$, so ordnen wir jedem c_j das zugehörige m-Tupel in $\mathbb{Z}_2^m$ zu und hängen für jedes j ein Prüfbit an, sodass das entstehende $(m+1)$-Tupel gerade Parität hat. Das entstehende Binärwort hat dann Länge $(m+1)\cdot n$ und Gewicht $\geq 2d$ (weil jedes $c_j \neq 0$ mindestens 2 bits $\neq 0$ induziert). Der so konstruierte Binärcode $\hat{C}_2$ hat also die Parameter $(n(m+1), km, \geq 2d)$.

Geht man von einem erweiterten RS-Code über $\mathbb{F}_{2^m}$ mit Parametern $(n+1, k, d+1)$ aus, so erhält man mit der selben Vorgangsweise einen $((n+1)(m+1), km, \geq 2(d+1))$ Binärcode.

Beispiel: Aus einem $(15, 10, 6)$ RS-Code C über $\mathbb{F}_{16}$ erhält man einen $(75, 40, \geq 12)$ Binärcode $\hat{C}_2$, bzw. aus dem $(16, 10, 7)$ erweiterten RS-Code C über $\mathbb{F}_{16}$ erhält man einen $(80, 40, \geq 14)$ Binärcode $\hat{C}_2$.

Reed-Solomon-Codes über $\mathbb{F}_{2^m}$, die in diesem Sinne als Binärcodes aufgefasst werden, sind besonders gut geeignet, um sogenannte **Bündelfehler** zu korrigieren.

Bündelfehler der Länge i: i aufeinanderfolgende Positionen sind von Fehlern betroffen, wobei die erste und die i-te Stelle jedenfalls fehlerhaft sind, die Stellen dazwischen können/müssen aber nicht falsch sein

			1	2	3	4	5	.	.	.	i							1	2	3	4	5	.	.	.	i			
.	.	.	*	*	*	*	*	.	.	.	*	.	.	.	$\rightarrow$	.	.	.	×	×	*	×	*	.	.	.	×	.	.	.

Codewort in Reed-Solomon-Code über $\mathbb{F}_{2^m}$ gestörtes Codewort

Code über $\mathbb{F}_2$: jedem Element aus $\mathbb{F}_{2^m}$ wird ein binäres m-Tupel zugeordnet

(Buchstabe $\in \mathbb{F}_{2^m}$)	(Buchstabe $\in \mathbb{F}_{2^m}$)	(Buchstabe $\in \mathbb{F}_{2^m}$)	...	Länge n	über $\mathbb{F}_{2^m}$
$\downarrow$	$\downarrow$	$\downarrow$			
(binäres m-Tupel)	(binäres m-Tupel)	(binäres m-Tupel)	...	Länge $n \cdot m$	über $\mathbb{F}_2 = \mathbb{Z}_2$

Beispiel: Wenn ein Reed-Solomon-Code über $\mathbb{F}_{2^m}$ 5 Fehler korrigiert, kann der zugehörige Binärcode Bündelfehler der Länge $4m+1$ korrigieren, unter günstigen Umständen sogar Bündelfehler der Länge $5m$. (Bündelfehler der Länge $4m+1$ umfassen höchstens 5 m-Tupel. Wenn der Bündelfehler günstig liegt, umfaßt er genau 5 m-Tupel und kann auch noch korrigiert werden.)

RS Codes werden in der Praxis sehr häufig verwendet, z.B. bei der Audio/Daten Compact Disk, DVD, ADSL (Modems), DVB (digitales Fernsehen), ...; beispielsweise setzt der Chip AHA 4210 RSVP RS Codes über $\mathbb{F}_{2^8}$ ein und hat eine Codier-/Decodierleistung von 30 Mbyte/s (2001)

wichtige Klassen von Codes, die RS Codes verallgemeinern sind

- BCH Codes (arbeiten mit einem ähnlichen Konstruktionsprinzip, zyklisch, flexibel in ihrer Länge),
- Goppa Codes, welche – im Gegensatz zu RS bzw. BCH Codes – auch asymptotisch gut sind (d. h. es gibt Folgen von Goppa Codes mit Parametern (n, k_n, d_n), $n \in \mathbb{N}$, sodass die relativen Codeparameter k_n/n (Informationsrate) und d_n/n (relative Minimaldistanz) beide positive Grenzwerte haben für $n \to \infty$); Nachteil von Goppa Codes: es existieren noch nicht so effiziente Algorithmen zur Codierung/Decodierung wie bei RS- bzw. BCH-Codes.

10 Die Codierung auf der Compact Disk

Produkt-Code: gegeben seien zwei lineare Codes über demselben Alphabet A:

$$C_1 \dots [n_1, k_1, d_1]\text{-}Code, \quad C_2 \dots [n_2, k_2, d_2]\text{-}Code,$$

oBdA werde systematische Codierung auf den ersten k_i Stellen von C_i, $i = 1, 2$, vorausgesetzt

der Produktcode $C_1 \otimes C_2$ besteht aus allen Matrizen, die wie folgt entstehen:

1. man startet mit einer $k_1 \times k_2$-Matrix M mit beliebigen Einträgen aus A (Nachrichtenmatrix),
2. die k_2 Spalten der Dimension k_1 von M werden jeweils mit $n_1 - k_1$ Kontrollstellen so ergänzt, dass die entstehenden Spalten in C_1 liegen, so erhält man eine $n_1 \times k_2$-Matrix M_1
3. die n_1 Zeilen der Dimension k_2 von M_1 werden jeweils mit $n_2 - k_2$ Kontrollstellen so ergänzt, dass die entstehenden Zeilen in C_2 liegen

es gilt (siehe Übungen):

- der Code $C_1 \otimes C_2$ ist linear,
- $C_1 \otimes C_2$ ist ein $[n_1 n_2, k_1 k_2, d_1 d_2]$-Code,
- die Schritte 2. und 3. können in umgekehrter Reihenfolge angewendet werden (mutatis mutandis), ohne dass sich der Produktcode $C_1 \otimes C_2$ ändert

Beispiel: C_1: $G_1 = \left(\begin{array}{cc|c} 1 & 0 & 1 \\ 0 & 1 & 1 \end{array}\right)$ $[3,2,2]$-Code, $G_2 = \left(\begin{array}{ccc|ccc} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{array}\right)$, $[6,3,3]$-Code

Produktcode $C_1 \otimes C_2$:

z.B. $M = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} \to M_1 = \left(\begin{array}{ccc} 1 & 1 & 0 \\ 0 & 1 & 1 \\ \hline 1 & 0 & 1 \end{array}\right) \to \left(\begin{array}{ccc|ccc} 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ \hline 1 & 0 & 1 & 0 & 1 & 1 \end{array}\right) \in C_1 \otimes C_2$

Verkürzung eines Codes:

C ein $[n,k,d]$-Code über A, unter dem um die ersten m Stellen ($m \leq k$) verkürzten Code von C versteht man den Code

$$C_m := \{x_{m+1} \dots x_n \mid \underbrace{0 \dots 0}_{m \text{ Stellen}} \; x_{m+1} \dots x_n \in C\}$$

offensichtlich ist C_m ein linearer $[n-m, k-m, \geq d]$-Code, wenn die ersten m Stellen von C Nachrichtenstellen sind

Folgerung: Aus einem MDS Code wird durch Verkürzung wieder ein MDS Code

Beispiel: C mit Generatormatrix $G = \left(\begin{array}{ccc|ccc} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{array}\right)$, $[6,3,3]$-Code

1-te Stelle kürzen liefert Code mit Generatormatrix $G_1 = \left(\begin{array}{cc|ccc} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \end{array}\right)$, $[5,2,3]$-Code,

3-te Stelle kürzen liefert Code mit Generatormatrix $G_2 = \left(\begin{array}{cc|ccc} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{array}\right)$, $[5,2,3]$-Code.

Digitalisierung von Musik: das Analogsignal wird 44.100 mal pro Sekunde bemustert; Rate von 44,1 kHz ist notwendig um Frequenzen bis zu ungefähr 20 kHz hören zu können, da allgemein die Abtastfrequenz doppelt so groß sein muß, wie die höchste aufzunehmende Tonfrequenz (Satz von Nyquist);

sollten höhere Frequenzen als 22 kHz vorkommen, müßten diese mit einem Tiefpassfilter vor der Bemusterung herausgefiltert werden; ansonsten kommt es bei Vorliegen höherer Frequenzen zu Fehlern (z.B. zu sogenanntem Aliasing, Bsp.: werden Filmaufnahmen von sich schnell drehenden Speichenrädern gemacht, hat man bei der Wiedergabe den Eindruck, die Räder drehen sich in die andere Richtung als bei der Aufnahme)

jedes der 44.100 Muster (samples) gibt eine von 2^{16} möglichen Amplituden an, d.h. pro Muster 2 bytes Information; da man üblicherweise 2 Kanäle hat (Stereo), erhält man 4 bytes pro Muster;

als Alphabet wird $A = \mathbb{F}_{2^8} = \mathbb{F}_{256}$ verwendet, d.h. jedem byte entspricht ein Buchstabe; die Buchstabenfolge wird in Blöcke der Länge 28 eingeteilt; auf diese Blöcke wird ein Produktcode aus einem $[28, 24, 5]$-Code C_1 und einem $[32, 28, 5]$-Code C_2 über $\mathbb{F}_{2^8}$ angewendet, etwas vereinfacht dargestellt werden jeweils 24 Blöcke der Länge 28 zeilenweise zu einer Matrix M (Nachrichtenmatrix) zusammengefasst und diese Matrix dann dem Produktcode $C_1 \otimes C_2$ unterworfen: aus M entsteht durch die Codierung eine 28×32-Matrix, deren Spalten in C_1 und Zeilen in C_2 liegen

die Codes C_1 und C_2 entstehen durch Verkürzung aus einem **Reed-Solomon Code** über $\mathbb{F}_{2^8}$: man startet mit einem $[255, 251, 5]$-RS-Code,

— durch Verkürzung um 227 Stellen entsteht $C_1 \ldots [28, 24, 5]$-Code, bzw.
— durch Verkürzung um 223 Stellen entsteht $C_2 \ldots [32, 28, 5]$-Code;

der **Produktcode** $C_1 \otimes C_2$ hat die Minimaldistanz $5 \cdot 5 = 25$, er kann als solcher bis zu 12 Fehler pro Codewort ($= 28 \times 32$-Matrix) korrigieren; durch geschicktes **Zusammenspiel aus Spalten- und Zeilencodierung** leistet der Code aber viel mehr.

Beispiel: nehmen wir an, es treten bei einem Codewort ($= 28 \times 32$-Matrix) 21 Fehler auf, die sich folgendermaßen verteilen: 12 Spalten mit einem Fehler, je eine Spalte mit 2, 3 bzw. 4 Fehlern, und 17 korrekte Spalten;

die Spalten wurden mit C_1, einem $[28, 24, 5]$-Code codiert, d.h. C_1 kann 2-fach Fehler korrigieren; wir verwenden C_1 allerdings nur um **1-fach Fehler** zu **korrigieren**; Empfangsworte mit Hammingdistanz ≥ 2 zu allen Codeworten werden nur als **fehlerhaft markiert**; das bewirkt, dass Spalten mit 2 bzw. 3 Fehlern als falsch erkannt werden aber nicht korrigiert werden; bei einer Spalte mit 4 Fehlern gibt es zwei Möglichkeiten:
1) es kann sein, dass der Korrekturalgorithmus glaubt einen 1-fach Fehler korrigiert zu haben, während in diesem Fall tatsächlich eine 5-te falsche Stelle hinzugefügt wurde,
2) es wird ein Fehler an mind. 2 Stellen erkannt

überall dort, wo die Spaltendecodierung einen Fehler erkennt (mind. 2 Fehler), konstatiert der Decodierungsalgorithmus für alle Zeilen eine **Auslöschung**, das ist ein Fehler an der jeweiligen Spaltenposition;

Bilanz nach Korrektur der 32 Spalten: 17+12=29 Spalten sind korrekt (jene die ohnehin korrekt waren bzw. jene mit nur einem Fehler), 2 Spalten enthalten eine Auslöschung (das sind jene mit 2 und 3 Fehlern in der Spalte), und für die eine Spalte mit 4 Fehlern nehmen wir an, dass ein fünfter Fehler dazugekommen ist (der Fall, dass dort auch eine Auslöschung erkannt wurde, ist einfacher zu behandeln, wie man im folgenden sieht);

wir betrachten nun die Zeilen, die mit C_2, einem $[32, 28, 5]$-Code codiert sind: in jeder Zeile gibt es zwei Auslöschungen und in 5 der 28 Zeilen gibt es einen weiteren Fehler an einer unbekannten Stelle, d.h. jede Zeile hat max. zwei Auslöschungen und einen weiteren Fehler, da die Minimaldistanz gleich 5 ist, kann C_2 alle Zeilen richtig decodieren, und damit konnten insgesamt 21 Fehler korrigiert werden (obwohl $C_1 \otimes C_2$ als Linearcode mit $d \geq 25$ a priori nur alle 12-fach Fehler korrigieren kann;

allgemein gilt für eine **Kombination** von **Fehlern** (an nicht bekannten Stellen) und **Auslöschungen**: ein Code mit Minimaldistanz d kann eine Kombination von i Auslöschungen und j Fehlern an unbekannten Stellen korrigieren kann, wenn gilt $i + 2j < d$ (Übung);

diese Produkt-Codierung heißt **CIRC** (**c**ross **i**nterleave **R**S-**c**ode)
interleaving: durch zeilenweises Verarbeiten der Matrixwörter auf der CD jedoch zuerst spaltenweises Decodieren splitten sich Bündelfehler auf mehrere Wörter auf (das tatsächlich verwendete

Verfahren ist etwas komplizierter (interleaving wird mehrfach angewendet) mit noch besserer Verteilung von Bündelfehlern, aber im Prinzip ähnlich)

Leistung von CIRC:

- **Bündelfehler** mit einer Länge von bis zu 4000 bits (entspricht 2,47 mm Spurlänge bzw. 1,9 Millisek.) können vollständig korrigiert werden
- Bündelfehler mit max. Länge von 13.700 bits (entspricht 8,5 mm Spurlänge) können interpoliert werden: wenn Fehler erkannt aber nicht eindeutig korrigiert werden können, werden diese Stellen markiert; sind vorangehende und nachfolgende Stelle(n) bekannt, wird mit Polynomfunktionen interpoliert; bei zu vielen nicht korrigierbaren Stellen wird Signal stumm geschaltet (plus fade out/in)
- falsch korrigierte Fehler treten bei einer **bit error rate** (BER) von 10^{-4} praktisch nicht auf, bei BER von 10^{-3} erhält man weniger als 1 Fehler in 750 Stunden
- ein Kriterium für die Korrekturqualität ist die sog. Interpolationsrate: das ist die Anzahl der interpolierten samples bei gegebener BER in einer gewissen Zeitspanne; unter CIRC erhält man bei BER $= 10^{-4}$: 1 sample alle 10 Std, bei BER $= 10^{-3}$: 1000 samples pro min
- es gibt verschieden leistungsstarke (=rechenaufwändige) Korrekturstrategien bei CIRC, je nachdem, wie man C_1 und C_2 zusammenarbeiten läßt, man kann z.B. C_1 auch 2-fach Fehler korrigieren lassen und sich diese Stellen merken (bringt in manchen Fällen noch mehr)

weitere technische Details:

- 7 Muster entsprechen 28 bytes, werden zeilenweise zu 32 bytes codiert, dazu kommt noch ein Extrabyte, welches Kontroll- und Anzeigeinformation enthält (unter anderem auf welcher Spur sich die CD gerade befindet), insgesamt hat man damit 33 bytes also 264 bits;
- die Speicherung der Daten auf der CD erfolgt als 5 km lange spiralförmige Spur (innen beginnend), die Spur besteht aus einer Abfolge von **Pits** (Vertiefungen von 0,13 μm) und **Lands** (eben);
- die Spur wird optisch von einem Laserstrahl gescannt; Übergang von Pit/Land (P/L) oder umgekehrt wird als 1 interpretiert, dazwischen Nullen; 1 bit entspricht einer Länge von 0,3 μm

  Bsp.: Pit 1,8 μm gefolgt von Land 0,9 μm entspricht der Folge 100000 100
- Messung erfolgt so, dass der Laserstrahl bei Land fast vollständig reflektiert wird, bei Pit tritt Interferenz auf (Tiefe ist 1/4 der Wellenlänge, ergibt bei Überlagerung einen Unterschied von 1/2 Wellenlänge) und es wird wenig reflektiert;

daraus ergeben sich folgende Anforderungen für die Abfolge von Pits und Lands:

1. jeder Wechsel P/L muß mindestens 3 bits d.h. 0,9 μm lang sein, damit der Laser, der auf einen Durchmesser von 1,3 μm fokussiert ist Übergänge richtig registrieren kann, d.h. zwischen zwei Einsen müssen mindestens zwei Nullen vorkommen,
2. Lands dürfen nicht zu lang sein, damit der Laserstrahl auf der Spur gehalten und die Sychronisation gewährleistet werden kann, konkret dürfen zwischen zwei Einsen höchstens 10 Nullen vorkommen;

daher können die codierten Zeilen bestehend aus 33 bytes nicht als solche gespeichert werden, sondern werden einer "eight to fourteen" Modulation (EFM) unterworfen: jedem byte = 8 bits wird eine Bitfolge der Länge 14 zugeordnet

warum Länge 14? — erst ab Länge 14 gibt es genug, nämlich 267 Wörter, die den Anforderungen 1. und 2. von oben genügen (mind. 2, höchstens 10 Nullen zwischen zwei Einsen), von diesen 267 werden nur $256 = 2^8$ benötigt

weiteres Problem: wenn man zwei solche Wörter der Länge 14 zusammenhängt, müssen 1. und 2. immer noch erfüllt sein; dazu reicht es, wenn man dazwischen 3 merging bits einfügt (werden so gewählt, dass ungefähr gleich viele Pits und Lands auftreten), also eigentlich wird eine $8 \to 17$ Modulation vorgenommen; nachdem 33 bytes so moduliert wurden, werden letztlich noch 27 Synchronisationsbits eingefügt (identifizieren den Beginn der nächsten Codesequenz)

eine Zeile: 7 Muster $\to 33 \times 8 = 264$ codierte Datenbits $\to 33 \times 17 + 27 = 588$ Kanalbits

eine Matrix: 24×7 Muster $\to 28 \times 588 = 16.464$ Kanalbits

1 Sekunde (44.100 Hz) $\to \frac{44.100}{168} \times 16.464 = 4.321.800$ bits ≈ 540 kbyte auf der Tonspur

(Literatur: K. Pohlmann, Compact Disk Handbuch, IWT Verlag, München, 1994.)

11 BCH Codes

Im gesamten Abschnitt 11 seien n und q teilerfremde natürliche Zahlen.

Definition: Die i-te **Kreisteilungsklasse** von q modulo n ist definiert durch:

$$C_i = \{iq^j \in \mathbb{Z}_n \mid j = 0, 1, \ldots\}$$

Eigenschaften: (i) C_1 ist die von q erzeugte Untergrupppe $\langle q \rangle$ in der primen Restklassengruppe $\bmod n$, $(\mathbb{Z}^\times, \cdot)$ mit $\mathbb{Z}_n^\times = \{x \in \mathbb{Z}_n \mid \mathrm{ggT}(x, n) = 1\}$ und es gilt $|C_1| = \mathrm{ord}(q \bmod n)$; weiters ist $|C_i|$ ein Teiler von $\mathrm{ord}(q \bmod n)$ für alle i, das ergibt sich aus folgenden Überlegungen: Sei $m = \mathrm{ord}(q \bmod n)$, dann ist $C_i = \{i, iq, \ldots, iq^{m-1}\}$; aus $iq^k \equiv iq^\ell$ mit $k < \ell$ folgt $i \equiv iq^{\ell-k}$; sei jetzt $r > 0$ minimal mit $iq^r \equiv i$, dann ist $|C_i| = r$; wir zeigen $r|m$: sei $m = rs + t$ (Division von m durch r mit Rest), $0 \le t < r$, dann gilt $i \equiv iq^m = iq^{rs}q^t = (iq^r)q^r \cdots q^r q^t \equiv iq^t$ und folglich $t = 0$, also $r|m$.

(ii) wenn n prim: $C_1 = \langle q \rangle$ ist die von q erzeugte Untergruppe der multiplikativen Gruppe $\mathbb{Z}_n^\times := \mathbb{Z}_n \setminus \{0\}$ und $C_i = i\langle q \rangle$ ist die Nebenklasse von i bzgl. $\langle q \rangle$; bekanntlich gilt dann $|C_i| = |C_1| = \mathrm{ord}(q \bmod n)$,

(iii) die Kreisteilungsklassen von q modulo n, $\{C_i \mid 0 \le i \le n\}$, bilden eine Partition von $\mathbb{Z}_n$: es gilt $i \in C_i$; wenn $j \in C_i$, dann ist $j \equiv iq^k$ folglich $jq^\ell \equiv iq^{k+\ell}$ also $C_j \subseteq C_i$, und wegen $i \equiv jq^{\mathrm{ord}(q)-k}$, also $i \in C_j$ folgt sofort $C_i = C_j$.

Beispiel: Kreisteilungsklassen von 2 modulo 15:
$C_0 = \{0\}$, $C_1 = \{1, 2, 4, 8\} = C_2 = C_4 = C_8$,
$C_3 = \{3, 6, 12, 9\} = C_6 = C_9 = C_{12}$, $C_5 = \{5, 10\} = C_{10}$,
$C_7 = \{7, 14, 13, 11\} = C_{11} = C_{13} = C_{14}$.

Lemma: Sei q eine Primzahlpotenz und $n \in \mathbb{N}$. Wenn $\mathrm{ggT}(n, q) = 1$ dann existiert (ein minimales) $m \in \mathbb{N}$ mit $n|(q^m - 1)$ und im Erweiterungskörper $\mathbb{F}_{q^m}$ von $\mathbb{F}_q$ existiert dann ein Element α mit multiplikativer Ordnung n, sodass $x^n - 1 = \prod_{i=1}^{n}(x - \alpha^i)$, d. h. $x^n - 1$ zerfällt in $\mathbb{F}_{q^m}$ in Linearfaktoren mit n verschiedenen Nullstellen. α heißt dann eine **primitive n-te Einheitswurzel** in $\mathbb{F}_{q^m}$.)

Beweis: Die Bedingung $n|(q^m - 1)$ ist äquivalent zu $q^m \equiv 1 \bmod n$. Man kann also m definieren als die Ordnung von q in der primen Restklassengruppe $(\mathbb{Z}_n^\times, \cdot)$. Ist γ ein primitives Element in $\mathbb{F}_{q^m}$, so ist offensichtlich $\alpha := \gamma^{(q^m-1)/n}$ eine primitive n-te Einheitswurzel. □

In der Folge werden wir zu vorgegebenem n und q immer dieses minimale $m = \operatorname{ord} q$ in $(\mathbb{Z}_n^\times, \cdot)$ betrachten.

Satz: Sei α eine primitive n-te Einheitswurzel in $\mathbb{F}_{q^m}$. Dann ist das Minimalpolynom $m^{(i)}(x)$ von α^i über $\mathbb{F}_q$ gegeben durch

$$m^{(i)}(x) = \prod_{j \in C_i} (x - \alpha^j),$$

wobei C_i die i-te Kreisteilungsklasse von $q \bmod n$ ist.

Beweis: 1) Da $i \in C_i$ ist $m^{(i)}(\alpha^i) = \prod_{j \in C_i}(\alpha^i - \alpha^j) = 0$.

2) $m^{(i)}(x)$ hat Koeffizienten aus $\mathbb{F}_q$: Sei $m^{(i)}(x) = a_0 + a_1 x + \ldots + a_d x^d$, $d = |C_i|$, dann gilt:
$(m^{(i)}(x))^q = (a_0 + a_1 x + \ldots + a_d x^d)^q = a_0^q + a_1^q x^q + \ldots + a_d^q x^{qd}$, aber auch
$(m^{(i)}(x))^q = \prod\limits_{j \in C_i} (x - \alpha^j)^q = \prod\limits_{j \in C_i} (x^q - \alpha^{jq}) \overset{qC_i = C_i}{=} \prod\limits_{j \in C_i} (x^q - \alpha^j) = m^{(i)}(x^q) = a_0 + a_1 x^q + \ldots a_d x^{qd}$.
Betrachten wir diese Identität im Polynomring $\mathbb{F}_{q^m}[x]$, so führt ein Koeffizientenvergleich auf $a_i^q = a_i$, also $a_i \in \mathbb{F}_q$.

3) Da α eine primitive n-te Einheitswurzel in $\mathbb{F}_{q^m}$ ist, sind alle Nullstellen von $m^{(i)}(x)$ verschieden. Sei jetzt $f(x) \in \mathbb{F}_q[x]$ ein Polynom mit $f(\alpha^i) = 0$. Wir müssen zeigen $m^{(i)}(x) | f(x)$. Sei $j \in C_i$, dann ist j von der Form $j = iq^l + kn$ mit ganzzahligen l und k. Daher gilt,

$$f(\alpha^j) = f(\alpha^{iq^l}) = (f(\alpha^i))^{q^l} = 0,$$

wobei bei der vorletzten Gleichheit benützt wurde, dass für die Koeffizienten f_i von $f(x) \in \mathbb{F}_q[x]$ gilt $f_i = f_i^{q^l}$ und folglich der Exponent q^l wieder aus der Summe herausgezogen werden kann. Wegen der Verschiedenheit der α^j, $j \in C_i$, folgt schließlich $m^{(i)}(x) = \prod_{j \in C_i}(x - \alpha^j) | f(x)$.

Aus 1)–3) folgt, dass $m^{(i)}(x)$ das Minimalpolynom von α^i über $\mathbb{F}_q$ ist. □

Beispiel: 1) Sei α ein primitives Element von $\mathbb{F}_4$, also $n = 3 = 2^2 - 1$, dann gilt für die Kreisteilungsklassen von 2 mod 3: $C_1 = \{1, 2\} = C_2$. Für α gilt $\alpha^3 = 1$, also $\alpha^3 - 1 = 0$, und weil $\alpha \neq 1$ folgt auch $\alpha^2 + \alpha + 1 = 0$, d. h. $\alpha^2 + \alpha = 1$. Wir erhalten

$$m^{(1)}(x) = m^{(2)}(x) = (x - \alpha)(x - \alpha^2) = x^2 + (\alpha^2 + \alpha)x + \alpha^3 = x^2 + x + 1.$$

2) Für $n = 9$ und $q = 2$ erhält man: $C_1 = \{1, 2, 4, 8, 7, 5\}$, also ist das minimale $m = 6$. Die weiteren Kreisteilungsklassen sind $C_0 = \{0\}$ und $C_3 = \{3, 6\}$. Sei α eine primitive 9-te Einheitswurzel in $\mathbb{F}_{2^6}$. Es gilt: $m^{(0)}(x) = x + 1$, $m^{(3)}(x)$ hat Grad 2 und $m^{(1)}(x)$ hat Grad 6. Für $\beta := \alpha^3$ und $\gamma := \alpha^6$ gilt $\beta^3 = \gamma^3 = 1$, also sind $\beta, \gamma \in \mathbb{F}_4 \setminus \{0, 1\}$, und daraus erhält man direkt $m^{(3)}(x) = x^2 + x + 1$. Oder alternativ dazu, weil $\beta + \gamma \neq \beta, \gamma, 0$, also $\beta + \gamma = \alpha^3 + \alpha^6 = 1$:

$$m^{(3)}(x) = (x - \alpha^3)(x - \alpha^6) = x^2 + (\alpha^3 + \alpha^6)x + \alpha^9 = x^2 + x + 1.$$

Daraus folgt $m^{(3)}(\alpha^3) = \alpha^6 + \alpha^3 + 1 = 0$, und weil $m^{(1)}(x)$ Grad 6 hat, gilt $m^{(1)}(x) = x^6 + x^3 + 1$. Insgesamt erhalten wir die Zerlegung in irreduzible Polynome über $\mathbb{F}_2$

$$x^9 + 1 = (x + 1)(x^2 + x + 1)(x^6 + x^3 + 1).$$

Definition: Sei α eine primitive n-te Einheitswurzel in $\mathbb{F}_{q^m}$ und $m^{(i)}(x)$ das Minimalpolynom von α^i über $\mathbb{F}_q$. Ein **BCH Code** (benannt nach Bose/Chaudhuri/Hocquenghem) über $\mathbb{F}_q$ der Länge n mit vorgegebenem Abstand δ ist ein zyklischer Code mit Generatorpolynom

$$g(x) = \text{kgV}\{m^{(a)}(x), m^{(a+1)}(x), \ldots, m^{(a+\delta-2)}(x)\}.$$

Da alle Potenzen von α Nullstellen von $x^n - 1$ sind, folgt $g(x)|(x^n - 1)$. Im Fall $n = q^m - 1$ heißt der BCH Code **primitiv** (α ist dann ein primitives Element von $\mathbb{F}_{q^m}$), wenn $a = 1$ heißt solch ein Code **BCH Code im engeren Sinn**.

Bemerkung: Im Fall $m = 1$ und $n = q - 1$ entsprechen BCH Codes genau den Reed-Solomon Codes: Es ist dann α ein primitives Element in $\mathbb{F}_q$ und daher $m^{(i)}(x) = x - \alpha^i$; weil die $m^{(i)}(x)$ teilerfremd sind, ist das kgV das Produkt der $x - \alpha^i$.

Satz: Die Dimension eines BCH Codes über $\mathbb{F}_q$ der Länge n mit vorgegebenem Abstand δ ist mindestens $n - m(\delta - 1)$ (m ist wie immer in diesem Zusamenhang minimal mit $n|(q^m - 1)$).

Beweis: Sei $S := \bigcup_{i=a}^{a+\delta-2} C_i$. Dann gilt $m^{(j)}(x) = \prod_{i \in C_j}(x - \alpha^i)$ und folglich $g(x) = \prod_{i \in S}(x - \alpha^i)$. Für die Dimension k eines zyklischen Codes mit Generatorpolynom $g(x)$ gilt $k = n - \operatorname{grad} g(x)$. Also erhalten wir

$$k = n - |S| = n - \left| \bigcup_{i=a}^{a+\delta-2} C_i \right| \geq n - \sum_{i=a}^{a+\delta-2} |C_i|.$$

Alle Elemente β in $\mathbb{F}_{q^m}$ haben ein Minimalpolynom $m_\beta(x)$ über $\mathbb{F}_q$ mit Grad $\leq m$, weil $\mathbb{F}_q \leq \mathbb{F}_q(\beta) \leq \mathbb{F}_{q^m}$ und $\dim_{\mathbb{F}_q} \mathbb{F}_q(\beta) = \operatorname{grad} m_\beta(x) \leq \dim_{\mathbb{F}_q} \mathbb{F}_{q^m} = m$ (mit dem Gradsatz $\dim_K M = \dim_K L \cdot \dim_L M$ für Körpererweiterungen $K \leq L \leq M$ folgt sogar $\operatorname{grad} m_\beta(x)|m$). Also erhalten wir $|C_i| = \operatorname{grad} m^{(i)}(x) \leq m$, woraus die behauptete Schranke für die Dimension folgt. □

Eigenschaften des oben definierten BCH Codes C:

- $c(x) \in C \Leftrightarrow c(\alpha^a) = c(\alpha^{a+1}) = \ldots = c(\alpha^{a+\delta-2}) = 0$
- man nennt die Matrix

$$H = \begin{pmatrix} 1 & \alpha^a & \ldots & \alpha^{(n-1)a} \\ 1 & \alpha^{a+1} & \ldots & \alpha^{(n-1)(a+1)} \\ \vdots & & & \vdots \\ 1 & \alpha^{a+\delta-2} & \ldots & \alpha^{(n-1)(a+\delta-2)} \end{pmatrix}$$

  eine **verallgemeinerte Kontrollmatrix** von C
- die Elemente von H liegen im Erweiterungskörper $\mathbb{F}_{q^m}$ (und i.A. nicht im Alphabet $\mathbb{F}_q$ des Codes)
- für ein Empfangswort v bzw. das zugehörige Polynom $v(x)$ ist

$$s_H(v) := v \cdot H^T = (v(\alpha^a), \ldots, v(\alpha^{a+\delta-2}))$$

  eine **Syndromfunktion** (charakterisiert Nebenklassen von C), denn es gilt:
  $s_H : \mathbb{F}_q^n \to (\mathbb{F}_{q^m})^{\delta-1}$, s_H ist $\mathbb{F}_q$-linear und $\ker s_H = C$:

$$s_H(v) = 0 \Leftrightarrow v(\alpha^a) = \ldots = v(\alpha^{a+\delta-2}) = 0 \Leftrightarrow v(x) \in C$$

- fixiert man eine Darstellung der Elemente aus $\mathbb{F}_{q^m}$ als m-Tupel über $\mathbb{F}_q$ und ersetzt die Einträge in H durch diese m-Tupel (als Spalten), so entsteht eine $m(\delta - 1) \times n$-Matrix $\bar{H}$ über $\mathbb{F}_q$; die Zeilen von $\bar{H}$ sind nicht notwendigerweise linear unabhängig (siehe obige Schranke für die Dimension eines BCH Codes), sie erzeugen aber den Dualcode $C^\perp$.

Satz: Ein BCH Code mit vorgebenem Abstand δ hat Minimalabstand $d \geq \delta$. Man nennt δ deshalb auch die **konstruierte Minimaldistanz** des BCH Codes.

Beweis: Angenommen die Minimaldistanz d des BCH Codes C wäre strikt kleiner als δ, also $d \leq \delta - 1$. Dann gäbe es ein Codewort $c(x) = c_0 + c_1 x + \ldots + c_{n-1}x^{n-1} \in C$ mit Gewicht d kleiner als δ. Ist $g(x) = \mathrm{kgV}\{m^{(a)}(x), \ldots, m^{(a+\delta-2)}(x)\}$ das Generatorpolynom von C, so existiert ein Polynom $b(x) \in \mathbb{F}_q[x]$ mit $c(x) = b(x) \cdot g(x)$. Folglich ist $c(\alpha^j) = b(\alpha^j) \cdot g(\alpha^j) = 0$ für $j = a, a+1, \ldots, a+\delta-2$. Seien $c_{l_1}, \ldots, c_{l_d}$ die von 0 verschiedenen Koeffizienten von $c(x)$, so schreibt sich das Gleichungssystem $c(\alpha^j) = 0$ für $j = a, a+1, \ldots, a+d-1$ (man beachte: $a + d - 1 \leq a + \delta - 2$) in der Form

$$\alpha^{jl_1} c_{l_1} + \ldots + \alpha^{jl_d} c_{l_d} = 0, \qquad j = a, \ldots, a+d-1.$$

Die Determinante der Koeffizientenmatrix $(\alpha^{jl_i})_{j=a,\ldots,a+d-1}^{i=1,\ldots,d}$ (Vandermonde Matrix) ist

$$\prod_{i=1}^{d} \alpha^{al_i} \prod_{\substack{r,r'=1 \\ r>r'}}^{d} (\alpha^{l_r} - \alpha^{l_{r'}}) \neq 0.$$

Daher hat das Gleichungssystem die eindeutige Lösung $c_{l_1} = \ldots = c_{l_d} = 0$, Widerspruch. □

Ein primitiver BCH Code über $\mathbb{F}_q$ hat also die Parameter $[q^m - 1, \geq q^m - 1 - m(\delta - 1), \geq \delta]$, wobei m und damit die Länge beliebig groß werden kann, und δ unabhängig von m gewählt werden kann. Für (nicht primitive) BCH Codes mit Länge $n | (q^m - 1)$ erhält man die Parameter $[n, \geq n - m(\delta - 1), \geq \delta]$.

Beispiel: Sei α eine Nullstelle des primitiven Polynoms $x^3 + x + 1 \in \mathbb{F}_2[x]$ und damit ein primitives Element in $\mathbb{F}_8$. Wir betrachten den primitiven BCH Code mit $a = 0$ und $\delta = 4$. Wegen $m = 3$ ist $n = 2^3 - 1 = 7$, und $a + \delta - 2 = 2$. Das Generatorpolynom ist gegeben durch $\mathrm{kgV}\{m^{(0)}(x), m^{(1)}(x), m^{(2)}(x)\}$: $m^{(0)}(x) = x + 1$, $m^{(1)}(x) = m^{(2)}(x) = x^3 + x + 1$ weil die Kreisteilungsklasse C_1 von 2 mod 7 gegeben ist durch $C_1 = \{1, 2, 4\}$. Daraus folgt

$$g(x) = (x+1)(x^3 + x + 1) = x^4 + x^3 + x^2 + 1.$$

Wir erhalten einen Binärcode mit Parametern $[7, 3, 4]$. Es handelt sich dabei um einen Simplexcode (siehe Übungen), den Dualcode des binären [7,4,3]-Hamming Codes.

Bemerkungen zum Beispiel: (1) Die allgemeine Abschätzung für die Dimension liefert hier nur $k \geq q^m - 1 - m(\delta - 1) = -1$, wegen $\mathrm{grad}\, m^{(0)} = 1$ und $m^{(1)} = m^{(2)}$ ist die tatsächliche Dimension größer. (2) Für $\delta = 3$ hätten wir den gleichen Code erhalten.

Allgemeine Bemerkungen zum Generatorpolynom $g(x) = \mathrm{kgV}\{m^{(a)}(x), \ldots, m^{(a+\delta-2)}(x)\}$:

- $m^{(i)}(x) = m^{(j)}(x) \;\Leftrightarrow\; C_i = C_j$, d. h. i und j liegen in derselben Kreisteilungsklasse von q mod n,
- $m^{(i)}(x) \neq m^{(j)}(x) \;\Rightarrow\; \mathrm{ggT}\{m^{(i)}(x), m^{(j)}(x)\} = 1$: $m^{(i)}(x), m^{(j)}(x) \in \mathbb{F}_q[x]$ also ist auch $\mathrm{ggT}\{m^{(i)}(x), m^{(j)}(x)\} \in F_q[x]$ (mit Hilfe des Euklidischen Algorithmus kann man Polynome $a(x), b(x) \in \mathbb{F}_q[x]$ finden, sodass $\mathrm{ggT}\{m^{(i)}(x), m^{(j)}(x)\} = a(x) \cdot m^{(i)}(x) + b(x) \cdot m^{(j)}(x)$); weil aber die Minimalpolynome $m^{(i)}(x)$ und $m^{(j)}(x)$ irreduzibel über $\mathbb{F}_q$ sind, ist ihr ggT daher gleich 1.
- Aus diesen beiden Eigenschaften folgt

  $$g(x) = \mathrm{kgV}\{m^{(a)}(x), m^{(a+1)}(x), \ldots, m^{(a+\delta-2)}(x)\} = \prod_{i \in L} m^{(i)}(x),$$

  wobei L ein System von Repräsentanten der Klassen $C_a, C_{a+1}, \ldots, C_{a+\delta-2}$, d. h. $\forall j \in \{a, \ldots, a+\delta-2\}$ $\exists$ genau ein $i \in L$ mit $j \in C_i$.

Binäre BCH Codes

Wenn $q = 2$ dann gilt $C_i = \{i, 2i, \ldots\}$, also ist $m^{(i)}(x) = m^{(2i)}(x)$ für alle i. Betrachtet man BCH Codes im engeren Sinn mit $\delta = 2t + 1$, also $a = 1$ und $a + \delta - 2 = 2t$, dann erhalten wir für das Generatorpolynom

$$\begin{aligned} g(x) &= \mathrm{kgV}\{m^{(1)}(x), \underbrace{m^{(2)}(x)}_{=m^{(1)}(x)}, m^{(3)}(x), \underbrace{m^{(4)}(x)}_{=m^{(1)}(x)} \ldots, \underbrace{m^{(2t)}(x)}_{=m^{(t)}(x)}\} \\ &\quad \mathrm{kgV}\{m^{(1)}(x), m^{(3)}(x), \ldots, m^{(2t-1)}(x)\}. \end{aligned}$$

Daraus folgt $\operatorname{grad} g(x) \leq mt$ und damit $k \geq n - mt$, und $d \geq \delta = 2t + 1$. Wir erhalten also einen $[n, \geq n - mt, \geq 2t + 1]$ Binärcode. Vergleicht man das mit den allgemeinen Schranken für die Parameter von BCH Codes $[n, \geq n - m(\delta - 1), \geq \delta]$, so ist bei gleicher Schranke für die Dimension die Schranke für die Minimaldistanz binärer BCH Codes doppelt so groß.

Auch die Kontrollmatrix kann für solche binäre BCH Codes angepasst werden: Da für ein Empfangswort $v(x)$ gilt $v(\alpha^{2i}) = v((\alpha^i)^2) = (v(\alpha^i))^2$, betrachtet man als Kontrollmatrix

$$H = \begin{pmatrix} 1 & \alpha^1 & \ldots & \alpha^{n-1} \\ 1 & \alpha^3 & \ldots & \alpha^{3(n-1)} \\ \vdots & & & \vdots \\ 1 & \alpha^{2t-1} & \ldots & \alpha^{(2t-1)(n-1)} \end{pmatrix}.$$

Beispiel: Wir bestimmen alle primitven binären BCH Codes im engeren Sinn der Länge $n = 15 = 2^4 - 1$, d. h. $m = 4$, $q = 2$.
Kreisteilungsklassen: $C_0 = \{0\}$, $C_1 = \{1, 2, 4, 8\}$, $C_3 = \{3, 6, 12, 9\}$, $C_5 = \{5, 10\}$, $C_7 = \{7, 14, 13, 11\}$.

$\delta = 2t + 1$	$g(x)$	k	d
1	1	15	1
3	$m^{(1)}$	11	3
5	$m^{(1)} \cdot m^{(3)}$	7	5
7	$m^{(1)} \cdot m^{(3)} \cdot m^{(5)}$	5	7
$9, 11, 13, 15$	$m^{(1)} \cdot m^{(3)} \cdot m^{(5)} \cdot m^{(7)}$ $= \frac{x^{15}-1}{x-1}$	1	15

Die angegebenen Werte für d verifiziert man wie folgt: Ist α ein primitives Element in $\mathbb{F}_{16}$, so kann $m^{(1)}(x)$ nur eines der beiden Polynome $x^4 + x + 1$ oder $x^4 + x^3 + 1$ jedes mit Gewicht 3 sein, also erhält man für $\delta = 3$ den Wert $d = 3$. O.b.d.A. werde im Folgenden $m^{(1)}(x) = x^4 + x + 1$ angenommen, im anderen Fall muss man jeweils nur zum reziproken Polynom übergehen. Weiters ist wegen $(\alpha^3)^5 - 1 = 0$ das Minimalpolynomom $m^{(3)}(x) = x^4 + x^3 + x^2 + x + 1$ und man erhält $m^{(1)}(x)m^{(3)}(x) = x^8 + x^7 + x^6 + x^4 + 1$ mit Gewicht 5, also ist $5 = \delta \leq d \leq 5$ folglich $d = 5$. Im nächsten Schritt gilt $m^{(5)}(x) = x^2 + x + 1$ und nach kurzer Rechnung $m^{(1)}(x)m^{(3)}(x)m^{(5)}(x) = x^{10} + x^8 + x^5 + x^4 + x^2 + x + 1$ mit Gewicht 7, sodass man $d = 7$ im Fall $\delta = 7$ erhält.

In diesem Beispiel stimmt die Minimaldistanz d immer mit dem maximal möglichen δ überein. Das muss nicht immer so sein.

Beispiel: Binäre BCH Codes der Länge $n = 23$: $C_1 = \{1, 2, 4, 8, 16, 9, 18, 13, 3, 6, 12\}$, also ist $m = |C_1| = 11$, α ist eine primitive 23te Einheitswurzel in $\mathbb{F}_{2^{11}}$. Weitere Klassen $C_0 = \{0\}$, $C_5 = \{5, 10, 20, 17, 11, 22, 21, 19, 15, 7, 14\}$. Daraus folgt $x^{23} - 1 = m^{(0)}(x) \cdot m^{(1)}(x) \cdot m^{(5)}(x)$ mit $m^{(0)}(x) = x - 1$. Eine längere Rechnung führt zu

$$\begin{aligned} m^{(1)}(x) &= x^{11} + x^9 + x^7 + x^6 + x^5 + x + 1, \\ m^{(5)}(x) &= x^{11} + x^{10} + x^6 + x^5 + x^4 + x^2 + 1 \end{aligned}$$

($m^{(5)}(x)$ ist das reziproke Polynom zu $m^{(1)}(x)$). Für $\delta = 5 = 2 \cdot 2 + 1$ erhält man für das Generatorpolynom des BCH Codes im engeren Sinn

$$g(x) = \text{kgV}\{m^{(1)}(x), m^{(2)}(x), m^{(3)}(x), m^{(4)}(x)\} = m^{(1)}(x).$$

Das liefert einen $[23, 12, \geq 5]$ Binärcode. Bei genauerer Untersuchung stellt sich heraus, dass es sich um einen $[23, 12, 7]$ Code handelt. Das ist der **Golay Code** G_{23} mit vielen interessanten Eigenschaften. Eine Kontrollmatrix ist gegeben durch $H = (1\,\alpha\,\alpha^2\,\ldots\,\alpha^{22})$, binär dargestellt hat H 11 Zeilen. Ein anderer Zugang zum Golay Code G_{23} erfolgt über Quadratische Reste Codes (späterer Abschnitt).

12 Decodierung von BCH Codes

Die bei der Decodierung von BCH Codes auftretenden Probleme und die zur Lösung verwendeten Methoden sollen zunächst an Hand eines einfachen Beispiels demonstriert werden.

Einführendes Beispiel

Sei C der primitive binäre BCH Code im engeren Sinn der Länge $n = 15$, der 2 Fehler korrigiert. D. h. wir wählen die konstruierte Minimaldistanz $\delta = 5 = 2 \cdot 2 + 1$. Da die Kreisteilungsklassen $C_1 = \{1, 2, 4, 8\}$ und $C_3 = \{3, 6, 12, 9\}$ verschieden sind und beide 4 Elemente enthalten, erhalten wir das Generatorpolynom

$$g(x) = \text{kgV}\{m^{(1)}(x), m^{(2)}(x), m^{(3)}(x), m^{(4)}(x)\} = \text{kgV}\{m^{(1)}(x), m^{(3)}(x)\} = m^{(1)}(x) \cdot m^{(3)}(x)$$

mit $\operatorname{grad} g(x) = 8$, und die Minimaldistanz d von C ist 5 (siehe Beispiel im vorigen Abschnitt). Also ist C ein $[15, 7, 5]$ Code. Bezeichnet α eine primitive 15te Einheitswurzel, ein primitives Element in $\mathbb{F}_{16}$, so ist eine Kontrollmatrix von C gegeben durch

$$H = \begin{pmatrix} 1 & \alpha^1 & \ldots & \alpha^{14} \\ 1 & \alpha^3 & \ldots & \alpha^{3\cdot 14} \end{pmatrix}.$$

Für ein Empfangswort $v(x)$ ist daher das Syndrom gegeben durch

$$s_H(v(x)) = (v(\alpha), v(\alpha^3)) =: (z_1, z_2)$$

mit $z_1, z_2 \in \mathbb{F}_{16}$.

Für die Fehlerkorrektur eines konkreten Empfangswortes $v(x)$ gehen wir alle möglichen Fälle schrittweise durch:

1) Kein Fehler: Das ist genau dann der Fall, wenn $s_H(v(x)) = \vec{0}$, d. h $z_1 = z_2 = 0$.

2) Einfachfehler an der Stelle j, $0 \leq j \leq 14$: Das zugehörige Fehlerwort $e(x)$ ist dann $e(x) = x^j$ und das Syndrom von $v(x)$ gegeben durch $z_1 = \alpha^j$ und $z_2 = \alpha^{3j} = z_1^3$. Da α^j für $0 \leq j \leq 14$ alle Elemente aus $\mathbb{F}_{16} \setminus \{0\}$ durchläuft, kann man Folgendes festhalten: Ein Einfachfehler liegt genau dann vor, wenn $z_1 \neq 0 \wedge z_2 = z_1^3$ und die Fehlerstelle j ist gegeben durch den diskreten Logarithmus $j = \log_\alpha(z_1)$.

3) Zweifachfehler an den Stellen i und j: Das Fehlerwort ist dann gegeben durch $e(x) = x^i + x^j$, und für das Syndrom ergibt sich $z_1 = \alpha^i + \alpha^j$ und

$$z_2 = \alpha^{3i} + \alpha^{3j} = (\alpha^i + \alpha^j)(\alpha^{2i} + \alpha^{i+j} + \alpha^{2j}) = z_1(z_1^2 + \alpha^{i+j}).$$

Aus der letzten Beziehung folgt $\alpha^{i+j} = z_2/z_1 + z_1^2$ (man beachte, dass $z_1 = \alpha^i + \alpha^j \neq 0$, weil $i \neq j$). Zusammen mit $z_1 = \alpha^i + \alpha^j$ sieht man, dass α^i und α^j die beiden verschiedenen Nullstellen des Polynoms

$$x^2 + z_1 x + (\frac{z_2}{z_1} + z_1^2) \tag{3}$$

sind. Da umgekehrt (α^i, α^j) für $0 \leq i, j \leq 14, i \neq j$, alle Elemente in $(\mathbb{F}_{16} \backslash \{0\})^2$ mit verschiedenen Komponenten durchlaufen, erhalten wir insgesamt: Ein Zweifachfehler liegt genau dann vor, wenn $z_1 \neq 0 \wedge z_2 \neq z_1^3$ und die quadratische Gleichung (3) zwei Lösungen $u_1, u_2 \in \mathbb{F}_{16}$ besitzt. Die Fehlerstellen i und j sind dann gegeben durch die diskreten Logarithmen $i = \log_\alpha(u_1)$ und $j = \log_\alpha(u_2)$. (Bemerkung: $u_1 \neq u_2$ folgt aus $z_1 \neq 0$ und $u_1, u_2 \neq 0$ ist eine Konsequenz von $z_2 \neq z_1^3$.)

Damit haben wir eine klare Vorgangsweise mit der wir bis zu 2 Fehler eindeutig an Hand des Syndroms (z_1, z_2) korrigieren können ohne eine Liste Syndrom/zugehöriges Fehlerwort verwenden zu müssen. Alle übrigen Fehler — auch wenn sie vom Prinzip her eindeutig rekonstruierbar wären — behandeln wir nicht. Das fassen wir im folgenden Punkt 4) zusammen.

4) Wenn die quadratische Gleichung (3) keine Lösung in $\mathbb{F}_{16}$ hat oder $z_1 = 0 \wedge z_2 \neq 0$, dann sind mindestens 3 Fehler aufgetreten und es erfolgt keine Fehlerkorrektur.

Das ergibt insgesamt ein unvollständiges Korrekturschema: Unser Code hat die Parameter $[15, 7, 5]$ und daher existieren $2^{n-k} = 2^8 = 256$ Nebenklassen. Wir korrigieren nur
• 0 Fehler – enspricht 1 Klasse,
• 1 Fehler – enspricht 15 Klassen,
• 2 Fehler – enspricht $\binom{15}{2} = 105$ Klassen,
das ergibt insgesamt 121 Klassen. Die restlichen 135 Klassen haben jeweils einen Anführer vom Gewicht ≥ 3. Fällt das Empfangswort in eine dieser 135 Klassen (mehr als die Hälfte der möglichen Fälle aber bei „realistischer“ Übertragungsfehlerwahrscheinlichkeit nicht sehr wahrscheinlich), wird nur ein ≥ 3-fach Fehler erkannt aber nicht korrigiert.

Einen Teilaspekt wollen wir noch genauer untersuchen: Wie löst man die quadratische Gleichung (3)? — Die gängige Lösungsformel für Lösungen in den komplexen Zahlen ist für Körper der Charakteristik 2 nicht anwendbar: wir können nicht durch 2 dividieren, bzw. das Ergänzen auf ein vollständiges Quadrat ist i. A. nicht möglich; abgesehen davon ist zu klären, wie man (effektiv!) die Quadratwurzel aus einem Element zieht.

Lösen quadratischer Gleichungen in endlichen Körpern der Charakteristik 2

Wir betrachten also die quadratische Gleichung $x^2 + bx + c = 0$ für gegebene Werte $b, c \in \mathbb{F}_{2^m}$ und suchen nach Lösungen $x \in \mathbb{F}_{2^m}$.

1. Fall $b = 0$: Hier haben wir $x^2 + c = 0 \Leftrightarrow x^2 = c$ mit der doppelten Lösung $x_1 = x_2 = c^{2^{m-1}}$, denn $(c^{2^{m-1}})^2 = c^{2^m} = c$ und $x^2 + c = x^2 + x_1^2 = (x + x_1)^2$. D. h. man kann die Lösung $x_1 = x_2$ durch $(m-1)$-maliges Quadrieren von c finden.

Dass in diesem Fall für jedes $c \in \mathbb{F}_{2^m}$ eine Lösung existiert, liegt daran, dass

$$\sigma : \left\{ \begin{array}{ccc} \mathbb{F}_{2^m} & \to & \mathbb{F}_{2^m} \\ \alpha & \mapsto & \alpha^2 \end{array} \right.$$

ein Körperautomorphismus ist: die Verträglichkeit von σ mit $+$ und $\cdot$ und die Injektivität folgt unmittelbar aus bereits bekannten Eigenschaften, und jede injektive Abbildung einer endlichen Menge auf sich selbst ist auch surjektiv und damit bijektiv.

Um den allgemeinen Fall der quadratischen Gleichung lösen zu können, brauchen wir noch einige Vorarbeiten.

Obige Überlegung funktioniert für beliebige Charakteristik p prim. Dazu betrachten wir jetzt allgemein die Körpererweiterung $\mathbb{F}_{q^m}/\mathbb{F}_q$, $q = p^r$. Sei $\mathrm{Aut}(\mathbb{F}_{q^m}) = \{\sigma : \mathbb{F}_{q^m} \to \mathbb{F}_{q^m}, \sigma \text{ ein bijektiver Homomorphismus }\}$ die Menge aller Automorphismen von $\mathbb{F}_{q^m}$. $\mathrm{Aut}(\mathbb{F}_{q^m})$ bildet mit der Komposition als Operation eine Gruppe. Ganz analog wie zuvor bei $\mathbb{F}_{2^m}$ kann man zeigen, dass die Abbildung

$$\sigma : \left\{ \begin{array}{rcl} \mathbb{F}_{q^m} & \to & \mathbb{F}_{q^m} \\ \alpha & \mapsto & \alpha^q \end{array} \right.$$

ein Automorphismus von $\mathbb{F}_{q^m}$ ist mit $\sigma(\alpha) = \alpha$ für alle $\alpha \in \mathbb{F}_q$. Dieser Automorphismus heißt **Frobenius Automorphismus** von $\mathbb{F}_{q^m}$ bezüglich $\mathbb{F}_q$.

Wir definieren jetzt die Automorphismengruppe der Körpererweiterung $\mathbb{F}_{q^m}/\mathbb{F}_q$ (diese Gruppe heißt auch Galoisgruppe von $\mathbb{F}_{q^m}/\mathbb{F}_q$):

$$\mathrm{Aut}(\mathbb{F}_{q^m}/\mathbb{F}_q) = \{\sigma \in \mathrm{Aut}(\mathbb{F}_{q^m}) \mid \sigma(\alpha) = \alpha \ \forall \alpha \in \mathbb{F}_q\}$$

Wir zeigen zunächst, dass der Frobenius Automorphismus alle anderen Automorphismen durch Iteration erzeugt.

Satz: Sei $\sigma(x) = x^q$ der Frobenius Automorphismus von $\mathbb{F}_{q^m}$. Dann ist

$$\mathrm{Aut}(\mathbb{F}_{q^m}/\mathbb{F}_q) = \{\sigma, \sigma^2, \ldots, \sigma^m = \mathrm{id}_{\mathbb{F}_{q^m}}\}.$$

D. h. die Automorphismengruppe der Körpererweiterung $\mathbb{F}_{q^m}/\mathbb{F}_q$ ist eine zyklische Gruppe der Ordnung m und wird vom Frobenius Automorphismus erzeugt.

Beweis: Dass $\sigma \in \mathrm{Aut}(\mathbb{F}_{q^m}/\mathbb{F}_q)$ haben wir oben schon erwähnt. Als erstes zeigen wir, dass die Iterierten σ^i, $1 \le i \le m$, alle verschieden sind. Es gilt $\sigma^2(x) = \sigma(\sigma(x)) = \sigma(x^q) = x^{q^2}$ und allgemein $\sigma^i(x) = x^{q^i}$. Sei γ ein primitives Element in $\mathbb{F}_{q^m}$ und $1 \le i < j \le m$, dann folgt

$$\sigma^i(\gamma) = \gamma^{q^i} \neq \gamma^{q^j} = \sigma^j(\gamma),$$

und daher $\sigma^i \neq \sigma^j$.

Nun zeigen wir, dass jedes $\tau \in \mathrm{Aut}(\mathbb{F}_{q^m}/\mathbb{F}_q)$ von der Gestalt $\tau = \sigma^i$ ist: Sei $m_\gamma(x) = x^m + a_{m-1}x^{m-1} + \ldots + a_0$ das Minimalpolynom von γ über $\mathbb{F}_q$. Dann erhalten wir

$$\begin{array}{rcl} m_\gamma(\tau(\gamma)) & = & \tau(\gamma)^m + a_{m-1}\tau(\gamma)^{m-1} + \ldots + a_0 \\ & = & \tau(\gamma^m) + \tau(a_{m-1}\gamma^{m-1}) + \ldots + \tau(a_0) \ = \ \tau(m_\gamma(\gamma)) = \tau(0) = 0. \end{array}$$

Weil $m_\gamma(x) = \prod_{j \in S_1}(x - \gamma^j)$ und die Kreisteilungsklasse $S_1 = \{1, q, q^2, \ldots, q^{m-1}\}$, existiert ein i mit $\tau(\gamma) = \gamma^{q^i}$. Für beliebiges $\alpha \in \mathbb{F}_{q^m}^\times$, $\alpha = \gamma^r$, folgt dann $\tau(\alpha) = \tau(\gamma^r) = (\tau(\gamma))^r = \gamma^{q^i r} = \alpha^{q^i}$, also $\tau = \sigma^i$. □

Definition: Für ein Element $\alpha \in \mathbb{F}_{q^m}$ ist die **Spur** von α bezüglich $\mathbb{F}_q$ definiert durch

$$\mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(\alpha) = \alpha + \alpha^q + \ldots + \alpha^{q^{m-1}}.$$

Äquivalent dazu könnte man auch definieren

$$\mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(\alpha) = \sum_{\tau \in \mathrm{Aut}(\mathbb{F}_{q^m}/\mathbb{F}_q)} \tau(\alpha) = \sum_{j \in C_1} \alpha^j.$$

Hier ist C_1 die 1-te Kreisteilungsklasse von q modulo $n = q^m - 1$.

Folgende Eigenschaften der Spur ergeben sich unmittelbar:

Lemma: 1) $\mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(\alpha^q) = \mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(\alpha) = \mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(\alpha)^q$.
2) $\mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q} : \mathbb{F}_{q^m} \to \mathbb{F}_q$ ist $\mathbb{F}_q$-linear und surjektiv.

Beweis: 1) folgt unmittelbar aus der Definition von $\mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}$, wenn man beachtet, dass $\alpha^{q^m} = \alpha$ und dass Potenzieren mit q ein Homomorphismus ist.

2) Dass $\mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(\alpha) \in \mathbb{F}_q$ folgt aus der zweiten Gleichung von 1), die $\mathbb{F}_q$-Linearität folgt wie üblich, und dass $\mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}$ surjektiv ist, folgt aus der $\mathbb{F}_q$-Linearität, sobald wir wissen, dass $\mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}$ nicht identisch 0 ist. Das kann aber nicht der Fall sein, weil das Polynom $x^{q^{m-1}} + \ldots + x^q + x$ höchstens q^{m-1} Nullstellen in $\mathbb{F}_{q^m}$ haben kann (und auch tatsächlich hat — man betrachte den Kern von $\mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}$), also existiert ein $\alpha \in \mathbb{F}_{q^m}$ mit $\mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(\alpha) \neq 0$. □

Folgerung: $x + x^q + \ldots + x^{q^{m-1}} \mid x^{q^m} - x$.

Zur Bezeichnung Spur: Man kann zeigen, dass $\mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(\alpha)$ die Spur der $\mathbb{F}_q$-linearen Abbildung $f_\alpha : \mathbb{F}_{q^m} \to \mathbb{F}_{q^m}$, $f_\alpha(x) = \alpha \cdot x$ ist: Sei $m_\alpha(x) = x^d + a_{d-1}x^{d-1} + \ldots + a_0$ das Minimalpolynom von $\alpha \in \mathbb{F}_{q^m}$ über $\mathbb{F}_q$, also $d = [\mathbb{F}_q(\alpha) : \mathbb{F}_q]$ der Erweiterungsgrad der Körpererweiterung $\mathbb{F}_q(\alpha)$ über $\mathbb{F}_q$ und $d \cdot s = m$. Für ein primitives Element β von $\mathbb{F}_{q^m}$ ist dann

$$\{1, \alpha, \ldots, \alpha^{d-1}, \beta, \beta\alpha, \ldots, \beta\alpha^{d-1}, \ldots, \beta^{s-1}, \beta^{s-1}\alpha, \ldots, \beta^{s-1}\alpha^{d-1}\}$$

eine Basis von $\mathbb{F}_{q^m}$ über $\mathbb{F}_q$, und die Matrix von f_α bezüglich dieser Basis besteht aus s Blöcken der Form

$$\begin{pmatrix} 0 & 0 & 0 & \cdots & 0 & -a_0 \\ 1 & 0 & 0 & \cdots & 0 & -a_1 \\ 0 & 1 & 0 & \cdots & 0 & -a_2 \\ \vdots & & \ddots & & & \vdots \\ 0 & & \cdots & & 1 & -a_{d-1} \end{pmatrix}$$

entlang der Diagonale aufgefädelt und sonst aus lauter Nullen. Daher ist die Spur von f_α — als Summe der Elemente in der Hauptdiagonale einer zugehörigen Matrix — gegeben durch $s \cdot (-a_{d-1})$. Weiters ist bekanntlich $m_\alpha(x) = (x - \alpha)(x - \alpha^q) \ldots (x - \alpha^{q^{d-1}})$ und daher $a_{d-1} = -(\alpha + \alpha^q + \ldots \alpha^{q^{d-1}})$, also ist die Spur von f_α gleich $s \cdot (\alpha + \alpha^q + \ldots \alpha^{q^{d-1}})$.

Andererseits ist $\alpha^{q^d} = \alpha$ wegen $\mathbb{F}_q(\alpha) \cong \mathbb{F}_{q^d}$ und daher

$$\mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(\alpha) = \alpha + \alpha^q + \ldots \alpha^{q^{d-1}} + \underbrace{\alpha^{q^d}}_{=\alpha} + \ldots + \underbrace{\alpha^{q^{m-1}}}_{=\alpha^{q^{d-1}}} = s \cdot (\alpha + \alpha^q + \ldots \alpha^{q^{d-1}}),$$

womit die Gleichheit der Spur von f_α mit $\mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(\alpha)$ gezeigt ist.

Artin-Lemma: Seien $\psi_1, \ldots, \psi_m$ paarweise verschiedene Homomorphismen von einer Gruppe G in die multiplikative Gruppe $(\mathbb{F}^\times, \cdot)$ eines Körpers $\mathbb{F}$. Dann gilt für alle $a_1, \ldots, a_m \in \mathbb{F}$: Wenn $a_1\psi_1(g) + a_2\psi_2(g) + \ldots + a_m\psi_m(g) = 0$ für alle $g \in G$, dann folgt $a_1 = \ldots = a_m = 0$.

Beweis: Durch vollständige Induktion nach m, Induktionsanfang $m = 1$: Für das neutrale Element $e \in G$ gilt $\psi_1(e) = 1 \in \mathbb{F}$ und aus $a_1\psi_1(e) = 0$ folgt daher $a_1 = 0$.

$m \to m + 1$: Nehmen wir an es gilt

$$a_1\psi_1(g) + \ldots + a_{m+1}\psi_{m+1}(g) = 0 \quad \forall g \in G. \tag{4}$$

Weil $\psi_1 \neq \psi_{m+1}$ existiert $h \in G$, sodass $\psi_1(h) \neq \psi_{m+1}(h)$. In (4) ersetzten wir g durch $g \cdot h$ und erhalten

$$a_1\psi_1(h)\psi_1(g) + \ldots + a_{m+1}\psi_{m+1}(h)\psi_{m+1}(g) = 0 \quad \forall g \in G.$$

Nach Division durch $\psi_{m+1}(h) \neq 0$ entsteht daraus

$$b_1\psi_1(g) + \ldots + b_m\psi_m(g) + a_{m+1}\psi_{m+1}(g) = 0 \quad \forall g \in G \tag{5}$$

mit $b_i = a_i\psi_i(h)\psi_{m+1}(h)^{-1}$. Subtrahiert man (5) von (4), so bekommt man eine Gleichung der Form

$$c_1\psi_1(g) + \ldots + c_m\psi_m(g) = 0 \quad \forall g \in G.$$

Nach Induktionsvoraussetzung folgt daraus insbesondere $c_1 = a_1 - b_1 = a_1(1 - \psi_1(h)\psi_{m+1}(h)^{-1}) = 0$ woraus wegen $\psi_1(h) \neq \psi_{m+1}(h)$ folgt, dass $a_1 = 0$. Jetzt kann man die Induktionsvoraussetzung auf (4) anwenden und erhält $a_2 = \ldots = a_{m+1} = 0$. □

Bemerkung: Die Menge $\mathrm{Hom}(G, \mathbb{F}^\times)$ aller Homomorphismen von G nach $\mathbb{F}^\times$ ist eine Teilmenge von $\mathbb{F}^G = \{f : G \to \mathbb{F}\}$ aller Funktionen von G nach $\mathbb{F}$. $\mathbb{F}^G$ bildet in natürlicher Weise einen $\mathbb{F}$-Vektorraum. Das Artin-Lemma sagt aus, dass $\mathrm{Hom}(G, \mathbb{F}^\times)$ eine linear unabhängige Teilmenge in diesem Vektorraum $\mathbb{F}^G$ ist.

Definition: Eine Basis von $\mathbb{F}_{q^m}$ über $\mathbb{F}_q$ der Gestalt $\{\alpha, \alpha^q, \ldots, \alpha^{q^{m-1}}\}$ heißt eine **normale Basis**.

Satz: Jede Körpererweiterung $\mathbb{F}_{q^m}/\mathbb{F}_q$ besitzt eine normale Basis.

Beweis: Sei $\sigma(x) = x^q$ der Frobenius Automorpismus von $\mathbb{F}_{q^m}/\mathbb{F}_q$. Wir haben bereits gezeigt: $\sigma^m = id_{\mathbb{F}_{q^m}}$ und $\sigma, \sigma^2, \ldots, \sigma^m$ sind paarweise verschiedene Homomorphismen von $G = \mathbb{F}_{q^m}^\times$ nach $\mathbb{F}_{q^m}^\times$. Wir fassen $\sigma : \mathbb{F}_{q^m} \to \mathbb{F}_{q^m}$ jetzt als $\mathbb{F}_q$-lineare Abbildung auf und bestimmen Minimalpolynom und charakteristisches Polynom von σ als $\mathbb{F}_q$-lineare Abbildung. Die Beziehung $\sigma^m = id_{\mathbb{F}_{q^m}}$ bedeutet, dass $x^m - 1$ die Abbildung σ annulliert. Aus dem Artin-Lemma, angewendet auf $\psi_1 = id_{\mathbb{F}_{q^m}^\times}$, $\psi_2 = \sigma$, $\ldots$, $\psi_m = \sigma^{m-1}$, folgt, dass es kein Polynom in $\mathbb{F}_q[x]$ vom Grad kleiner m gibt, welches σ annulliert. Also ist $x^m - 1$ das Minimalpolynom von σ. Das charakteristische Polynom $\chi_\sigma(x)$ von σ hat Grad $m = [\mathbb{F}_{q^m} : \mathbb{F}_q]$ und wird vom Minimalpolynom geteilt. Daraus folgt: charakteristisches Polynom ist gleich Minimalpolynom, $\chi_\sigma(x) = x^m - 1$.

Aus der linearen Algebra ist bekannt: Es gilt genau dann charakteristisches Polynom ist gleich Minimalpolynom, wenn es einen zyklischen Vektor gibt, d. h. es gibt ein $\gamma \in \mathbb{F}_{q^m}$ sodass die $\mathbb{F}_q$-lineare Hülle $\langle \gamma, \sigma(\gamma), \ldots, \sigma^{m-1}(\gamma)\rangle = \mathbb{F}_{q^m}$ ist. Das bedeutet, dass $\{\gamma, \gamma^q, \ldots, \gamma^{q^{m-1}}\}$ eine Basis von $\mathbb{F}_{q^m}$ über $\mathbb{F}_q$, also eine normale Basis ist. □

Beispiel: In $\mathbb{F}_9 = \mathbb{F}_{3^2}$ ist das Element α mit $\alpha^2 = \alpha + 1$ primitiv: $\alpha^3 = 2\alpha + 1$, $\alpha^4 = 2$, $\alpha^5 = 2\alpha$, $\alpha^6 = 2\alpha + 2$, $\alpha^7 = \alpha + 2$, $\alpha^8 = 1$.

α erzeugt eine normale Basis: $\{\alpha, \alpha^3\} = \{\alpha, 2\alpha + 1\}$ Basis von $\mathbb{F}_9/\mathbb{F}_3$.

α^2 erzeugt keine normale Basis: $\{\alpha^2, \alpha^6\} = \{\alpha + 1, , 2(\alpha + 1)\}$ keine Basis von $\mathbb{F}_9/\mathbb{F}_3$.

Zur Spurabbildung: Sei $\mathrm{Tr} = \mathrm{Tr}_{\mathbb{F}_9/\mathbb{F}_3}$, $\mathrm{Tr}(x) = x + x^3$.

$$\begin{array}{lll} \mathrm{Tr}(0) = 0, & \mathrm{Tr}(1) = 2, & \mathrm{Tr}(2) = 1, \\ \mathrm{Tr}(\alpha) = 1, & \mathrm{Tr}(\alpha + 1) = 0, & \mathrm{Tr}(\alpha + 2) = 2, \\ \mathrm{Tr}(2\alpha) = 2, & \mathrm{Tr}(2\alpha + 1) = 1, & \mathrm{Tr}(2\alpha + 2) = 0. \end{array}$$

Wir kommen zurück zum Lösen quadratischer Gleichungen in $\mathbb{F}_{2^m}$. Sei $\{\gamma, \gamma^2, \ldots, \gamma^{2^{m-1}}\}$ eine normale Basis von $\mathbb{F}_{2^m}$ über $\mathbb{F}_2$. Für alle $\beta \in \mathbb{F}_{2^m}$ existieren dann (eindeutig) Koeffizienten $b_0, \ldots, b_{m-1} \in \mathbb{F}_2$, sodass $\beta = b_0\gamma + b_1\gamma^2 + \ldots + b_{m-1}\gamma^{2^{m-1}}$.

Bezeichne jetzt $\mathrm{Tr} := \mathrm{Tr}_{\mathbb{F}_{2^m}/\mathbb{F}_2}$ die (sogenannte absolute) Spur von $\mathbb{F}_{2^m}$ auf den Primkörper $\mathbb{F}_2$. Wir behaupten $\mathrm{Tr}(\gamma) = 1$: Wäre $\mathrm{Tr}(\gamma) = 0$, so wäre auch $\mathrm{Tr}(\gamma^{2^i}) = 0$ für alle i und daher $\mathrm{Tr}(\beta) = b_0\mathrm{Tr}(\gamma) + \ldots + b_{m-1}\mathrm{Tr}(\gamma^{2^{m-1}}) = 0$ für alle $\beta \in \mathbb{F}_{2^m}$, was falsch ist (siehe Punkt 2 in Lemma auf Seite 48).

Aus $\mathrm{Tr}(\gamma) = 1$ folgt $\mathrm{Tr}(\gamma^{2^i}) = 1$ und daher $\mathrm{Tr}(\beta) = b_0 + b_1 + \ldots + b_{m-1}$.

Satz: Die quadratische Gleichung

$$x^2 + x + \beta = 0 \tag{6}$$

mit $\beta \in \mathbb{F}_{2^m}$ hat
- zwei Lösungen in $\mathbb{F}_{2^m}$, wenn $\mathrm{Tr}(\beta) = 0$,
- keine Lösung in $\mathbb{F}_{2^m}$, wenn $\mathrm{Tr}(\beta) = 1$.

Mit anderen Worten: $x^2 + x + \beta$ ist irreduzibel über $\mathbb{F}_{2^m}$ genau dann, wenn $\mathrm{Tr}(\beta) = 1$.

Beweis: Sei $x \in \mathbb{F}_{2^m}$ mit $x = x_0\gamma + x_1\gamma^2 + \ldots + x_{m-1}\gamma^{2^{m-1}}$. Dann folgt $x^2 = x_{m-1}\gamma + x_0\gamma^2 + \ldots + x_{m-2}\gamma^{2^{m-1}}$. Wenn x eine Lösung von (6) ist, dann erhalten wir durch Koeffizientenvergleich

$$x_0 + x_{m-1} = b_0, \quad x_1 + x_0 = b_1, \quad \ldots, \quad x_{m-1} + x_{m-2} = b_{m-1}.$$

Addiert man diese Gleichungen, so erhalten wir $0 = b_0 + b_1 + \ldots + b_{m-1} = \mathrm{Tr}(\beta)$, also ist $\mathrm{Tr}(\beta) = 0$ notwendig für die Existenz einer Lösung von (6) in $\mathbb{F}_{2^m}$.

$\mathrm{Tr}(\beta) = 0$ ist auch hinreichend für eine Lösung: Sei $\delta \in \{0, 1\}$. Wir definieren

$$x_0 := \delta,\ x_1 = \delta + b_1,\ x_2 = \delta + b_1 + b_2,\ \ldots,\ x_{m-1} = \delta + b_1 + \ldots + b_{m-1}$$

und behaupten, dass $\xi = \sum_{i=0}^{m-1} x_i\gamma^{2^i}$ eine Lösung von (6) ist. Dazu betrachten wir den Koeffizienten von γ^{2^i} in $\xi^2 + \xi + \beta$: Für $i = 1, 2, \ldots, m-1$ lautet dieser

$$\underbrace{\delta^2 + b_1^2 + b_2^2 + \ldots + b_{i-1}^2}_{\text{von } \xi^2} + \underbrace{\delta + b_1 + b_2 + \ldots + b_i}_{\text{von } \xi} + \underbrace{b_i}_{\text{von } \beta} = 0$$

($= 0$ ergibt sich, weil $b_j^2 = b_j$ für alle j und $\delta^2 = \delta$). Bei $i = 0$ lautet der Koeffizient

$$\underbrace{\delta^2 + b_1^2 + b_2^2 + \ldots + b_{m-1}^2}_{\text{von } \xi^2} + \underbrace{\delta}_{\text{von } \xi} + \underbrace{b_0}_{\text{von } \beta} = b_0 + b_1 + \ldots + b_{m-1} = 0$$

($= 0$ ergibt sich hier, weil nach Voraussetzung $\mathrm{Tr}(\beta) = 0$). Also sind alle Koeffizienten gleich 0 und daher $\xi^2 + \xi + \beta = 0$. (Die beiden Lösungen, die man durch die Wahl von $\delta = 0$ bzw. $\delta = 1$ bekommt, unterscheiden sich um 1, d. h. sie lauten $x_1 = \xi$, $x_2 = \xi + 1$.) □

Beispiel: Wir betrachten $\mathbb{F}_4 = \{0, 1, \alpha, \alpha + 1\}$ mit $\alpha^2 = \alpha + 1$. Hier ist $\{\alpha, \alpha^2\} = \{\alpha, \alpha + 1\}$ eine normale Basis. Für die Spur gilt: $\mathrm{Tr}(0) = \mathrm{Tr}(1) = 0$, $\mathrm{Tr}(\alpha) = \alpha + \alpha^2 = \mathrm{Tr}(\alpha + 1) = 1$. Daher sind die beiden Polynome $x^2 + x + \alpha$ und $x^2 + x + (\alpha + 1)$ irreduzibel über $\mathbb{F}_4$, während die Polynome $x^2 + x + 0$ und $x^2 + x + 1$ Nullstellen in $\mathbb{F}_4$ haben. „Berechnung“ der Nullstellen: 1) $x^2 + x + 0$: $\beta = 0 = 0 \cdot \alpha + 0 \cdot (\alpha + 1)$, d. h. $b_0 = b_1 = 0$, und wir erhalten die beiden Nullstellen

$$\xi_{1,2} = \delta\alpha + (\delta + b_1)(\alpha + 1) = \delta\alpha + \delta\alpha + \delta = \delta$$

mit $\delta = 0, 1$, also $\xi_1 = 0$, $\xi_2 = 1$.

2) $x^2 + x + 1$: $\beta = 1 = 1 \cdot \alpha + 1 \cdot (\alpha + 1)$, d. h. $b_0 = b_1 = 1$, und wir erhalten die beiden Nullstellen

$$\xi_{1,2} = \delta\alpha + (\delta + 1)(\alpha + 1) = \delta\alpha + \delta\alpha + \delta + \alpha + 1 = \delta + \alpha + 1$$

mit $\delta = 0, 1$, also $\xi_1 = \alpha + 1$, $\xi_2 = \alpha$.

Satz: 1) Gegeben $\beta \in \mathbb{F}_{2^m}$ mit $\mathrm{Tr}(\beta) = 1$. Dann kann jedes über $\mathbb{F}_{2^m}$ irreduzible quadratische Polynom durch eine lineare Transformation auf die Gestalt $\eta(x^2 + x + \beta)$ mit $\eta \in \mathbb{F}_{2^m}$ gebracht werden.
2) Gegeben $\beta \in \mathbb{F}_{2^m}$ mit $\mathrm{Tr}(\beta) = 0$. Dann kann jedes über $\mathbb{F}_{2^m}$ reduzible quadratische Polynom durch eine lineare Transformation auf die Gestalt $\eta(x^2 + x + \beta)$ mit $\eta \in \mathbb{F}_{2^m}$ gebracht werden.

Beweis: 1) Sei $ax^2 + bx + c$ ein irreduzibles quadratisches Polynom aus $\mathbb{F}_{2^m}[x]$. Dann sind alle Koeffizienten $a, b, c \neq 0$: für a und c ist das klar; $b = 0$ ist auch nicht möglich, weil $ax^2 + c$ Nullstellen in $\mathbb{F}_{2^m}$ hat ($x \mapsto x^2$ ist bijektiv in $\mathbb{F}_{2^m}$). Wir setzen jetzt $x = by/a$ und erhalten

$$ax^2 + bx + c = \frac{b^2}{a}(y^2 + y + d)$$

mit $d = ac/b^2$ und $\mathrm{Tr}(d) = 1$, weil $y^2 + y + d$ ebenfalls irreduzibel über $\mathbb{F}_{2^m}$ ist. Daraus folgt $\mathrm{Tr}(\beta + d) = 0$ und nach dem letzten Satz existiert ein $u \in \mathbb{F}_{2^m}$ mit $u^2 + u + (\beta + d) = 0$. Nun setzen wir $z = y + u$ und erhalten

$$\frac{b^2}{a}(y^2 + y + d) = \frac{b^2}{a}(z^2 + u^2 + z + u + d) = \frac{b^2}{a}(z^2 + z + \beta) = \eta(z^2 + z + \beta) = ax^2 + bx + c.$$

Bei 2) kann man analog vorgehen (Übungen). □

Damit ist das einführende Beispiel ($n = 15$, $\delta = 5$) zur Decodierung von binären BCH Codes abgeschlossen. Wir halten fest, dass die dabei verwendeten Methoden für beliebige Länge n anwendbar sind. Vergrößert man allerdings die konstruierte Minimaldistanz δ, so treten andere Probleme auf, denen wir uns jetzt widmen werden.

Allgemeiner Fall binärer BCH Codes

Sei jetzt C ein binärer BCH Code im engeren Sinn mit Parametern $[n, k, \geq \delta]$, $\delta = 2t + 1$ ungerade. Wir gehen davon aus, dass das Codewort $\vec{c} = c_0 \dots c_{n-1} \in C$ gesendet wurde und der Übertragungsfehler $\vec{e} = e_0 \dots e_{n-1}$ aufgetreten ist, d. h. $\vec{v} = v_0 \dots v_{n-1} = \vec{c} + \vec{e} = c_0 \dots c_{n-1} + e_0 \dots e_{n-1}$ beim Empfänger angekommen ist.

Die Decodierung des Empfangsworts $\vec{v}$ findet in 3 Schritten statt:

I) Syndromberechnung

II) Berechnung des Fehlerlokalisierungspolynoms $\sigma(z)$

III) Berechnung der Nullstellen von $\sigma(z)$

Ad I) Die Kontrollmatrix von C ist gegeben durch

$$H = \begin{pmatrix} 1 & \alpha^1 & \alpha^2 & \dots & \alpha^{n-1} \\ 1 & \alpha^3 & \alpha^{3\cdot 2} & \dots & \alpha^{3\cdot(n-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \alpha^{\delta-2} & \alpha^{(\delta-2)\cdot 2} & \dots & \alpha^{(\delta-2)\cdot(n-1)} \end{pmatrix},$$

wobei α wie immer in diesem Zusammenhang eine primitive n-te Einheitswurzel in $\mathbb{F}_{2^m}$ ist, $n|(2^m - 1)$. Das Syndrom von $\vec{v} = v(x)$ ist dann

$$s_H(\vec{v}) = \vec{v} \cdot H = (v(\alpha), v(\alpha^3), \dots, v(\alpha^{\delta-2})) = (e(\alpha), e(\alpha^3), \dots, e(\alpha^{\delta-2})).$$

Zur Abkürzung bezeichnen wir $A_i := v(\alpha^i)$, $i \in \mathbb{N}$, und beachten, dass gilt $A_{2i} = v(\alpha^{2i}) = (v(\alpha^i))^2 = A_i^2$.

Zur effektiven Berechnung der Syndromwerte A_i: Das Minimalpolynom $m^{(i)}(x)$ von α^i über $\mathbb{F}_q$ ist schon aus der Berechnung des Generatorpolynoms $g(x)$ bekannt, $g(x) = \mathrm{kgV}\{m^{(1)}(x), \dots, m^{(\delta-1)}(x)\}$. Sei $r^{(i)}(x) = v(x) - q(x)m^{(i)}(x)$ der Rest von $v(x)$ bei Division durch $m^{(i)}(x)$, dann ist $A_i = v(\alpha^i) = r^{(i)}(\alpha^i)$, $i = 1, 3, \dots, \delta - 2$, d. h. man muss zur Berechnung der A_i nur die Polynome $r^{(i)}(x)$ vom Grad $< m$ auswerten und nicht $v(x)$ vom Grad $< n$.

Mit $A_{2i} = (A_i)^2$ hat der Empfänger nach Schritt I) die Werte A_i, $i = 1, 2, \ldots, \delta - 1 = 2t$, berechnet.

Ad II) Wir gehen jetzt davon aus, dass das Empfangswort $\vec{v}$ genau w Fehler enthält und seien $i_1, i_2, \ldots, i_w$ die fehlerhaften Stellen in $\vec{v} = v_0 \ldots v_{n-1}$, d. h. $e_{i_1}, e_{i_2}, \ldots, e_{i_w} \neq 0$ und damit im binären Fall $e_{i_1}, e_{i_2}, \ldots, e_{i_w} = 1$. Wir definieren das **Fehlerlokalisierungspolynom**

$$\sigma(z) := \prod_{r=1}^{w}(1 - \alpha^{i_r} z) = \sum_{i=0}^{w} \sigma_i z^i.$$

Das Fehlerlokalisierungspolynom hat also genau die Werte $z = \alpha^{-i_r}$, $r = 1, \ldots, w$, als Nullstellen. Sei $x_r := \alpha^{i_r}$, $r = 1, \ldots, w$, dann gilt:

$$A_k = v(\alpha^k) = e(\alpha^k) = \sum_{r=1}^{w} e_{i_r} \alpha^{k i_r} = \sum_{r=1}^{w} e_{i_r} x_r^k.$$

Der Empfänger berechnet die Werte A_k nur für $k = 1, \ldots, \delta - 1$, aber grundsätzlich ist $A_k = v(\alpha^k)$ für alle $k \in \mathbb{N}$ definiert und wegen $\alpha^n = 1$ ist die Folge $(A_k)_{k \in \mathbb{N}}$ periodisch mit Periodenlänge n (n muss nicht die primitive (=kleinstmögliche) Periode sein!). Aus der Beziehung $\sigma(z) = \prod_{r=1}^{w}(1 - x_r z)$ erhalten wir für die Koeffizienten σ_i von $\sigma(z)$ folgende Gleichungen:

$$\begin{aligned}
\sigma_0 &= 1 \\
\sigma_1 &= -(x_1 + \ldots + x_w) \\
\sigma_2 &= x_1 x_2 + x_1 x_3 + \ldots + x_{w-1} x_w \\
&\vdots \\
\sigma_w &= (-1)^w x_1 x_2 \ldots x_w
\end{aligned}$$

Die wesentliche Aufgabe in Schritt II) besteht darin, $\sigma(z)$, d. h. die Koeffizienten $\sigma_1 \ldots, \sigma_w$, mit Hilfe der Syndromwerte $A_1, \ldots, A_{\delta-1}$ zu berechnen. Zu beachten ist insbesondere, dass der Grad w von $\sigma(z)$ (w = Anzahl der Fehler) vorerst nicht bekannt ist.

Satz: 1) Es gilt:

$$\sigma_0 A_{j+w} + \sigma_1 A_{j+w-1} + \ldots + \sigma_w A_j = 0, \qquad j = 1, 2, \ldots, \delta - w - 1, \tag{7}$$

d. h. die Folge $(A_1, \ldots, A_{\delta-1})$ genügt einer linearen Rekursion der Ordnung w, oder anders ausgedrückt: $(A_1, \ldots, A_{\delta-1})$ wird von einem **linear rückgekoppelten Schieberegister** (LFSR ... linear feedback shift register) der Länge w erzeugt. Als konstanten Faktoren in diesem Schieberegister treten die Koeffizienten des Fehlerlokalisierungspolynoms auf.
2) Wenn $w \leq (\delta - 1)/2 = t$, dann ist (7) das (bis auf einen konstanten Faktor $\neq 0$) eindeutig bestimmte kürzestmögliche LFSR mit Koeffizienten in $\mathbb{F}_{2^m}$, welches die Folge $(A_1, \ldots, A_{\delta-1})$ erzeugt.

Beweis: Ad 1) Wir gehen von der Beziehung

$$\sigma(z) = \prod_{r=1}^{w}(1 - \alpha^{i_r} z) = 1 + \sigma_1 z + \ldots + \sigma_w z^w$$

aus, setzen die Nullstelle $z = 1/x_s = 1/\alpha^{i_s}$ ein und multiplizieren den entstehenden Ausdruck mit $e_{i_s} x_s^{j+w}$, dann erhalten wir für alle $j \geq 1$:

$$\begin{aligned}
(1 + \sigma_1 \tfrac{1}{x_s} + \ldots + \sigma_w \tfrac{1}{x_s^w}) e_{i_s} x_s^{j+w} &= 0 \\
e_{i_s} x_s^{j+w} + \sigma_1 e_{i_s} x_s^{j+w-1} + \ldots + \sigma_w e_{i_s} x_s^j &= 0
\end{aligned}$$

Nun summieren wir diese Gleichung von $s = 1$ bis $s = w$, beachten dabei $A_k = \sum_{s=1}^{w} e_{i_s} x_s^k$ für alle $k \in \mathbb{N}$ und erhalten

$$\underbrace{\sigma_0}_{=1} \cdot A_{j+w} + \sigma_1 \cdot A_{j+w-1} + \ldots + \sigma_w \cdot A_j = 0, \qquad j = 1, 2, \ldots \delta - w - 1.$$

Damit haben wir (7) gezeigt.

Ad 2) Sei jetzt $w \leq (\delta - 1)/2 = t$, d. h. der Code kann w Fehler korrigieren. Wir behaupten, dass $\vec{x} = (\sigma_1, \ldots, \sigma_w)^t$ die einzige Lösung des linearen Gleichungssystems

$$\begin{pmatrix} A_w & \ldots & A_2 & A_1 \\ A_{w+1} & A_w & \ldots & A_2 \\ \vdots & \ddots & \ddots & \vdots \\ A_{2w-1} & A_{2w-2} & \ldots & A_w \end{pmatrix} \cdot \vec{x} = - \begin{pmatrix} A_{w+1} \\ \vdots \\ A_{2w} \end{pmatrix}. \tag{8}$$

ist.

Es gilt:

$$\begin{vmatrix} A_w & \ldots & A_2 & A_1 \\ A_{w+1} & A_w & \ldots & A_2 \\ \vdots & \ddots & \ddots & \vdots \\ A_{2w-1} & A_{2w-2} & \ldots & A_w \end{vmatrix} = (-1)^{\binom{w}{2}} \cdot \prod_{r=1}^{w} x_r \cdot \prod_{1 \leq s < r \leq w} (x_r - x_s)^2$$

Das sieht man wie folgt:

$$\begin{pmatrix} A_w & \ldots & A_2 & A_1 \\ A_{w+1} & A_w & \ldots & A_2 \\ \vdots & \ddots & \ddots & \vdots \\ A_{2w-1} & A_{2w-2} & \ldots & A_w \end{pmatrix} = \begin{pmatrix} \sum_{r=1}^{w} x_r^w & \ldots & \sum_{r=1}^{w} x_r^2 & \sum_{r=1}^{w} x_r \\ \sum_{r=1}^{w} x_r^{w+1} & \sum_{r=1}^{w} x_r^w & \ldots & \sum_{r=1}^{w} x_r^2 \\ \vdots & \ddots & \ddots & \vdots \\ \sum_{r=1}^{w} x_r^{2w-1} & \sum_{r=1}^{w} x_r^{2w-2} & \ldots & \sum_{r=1}^{w} x_r^w \end{pmatrix} =$$

$$\begin{pmatrix} 1 & \ldots & 1 \\ x_1 & \ldots & x_w \\ \vdots & \ddots & \vdots \\ x_1^{w-1} & \ldots & x_w^{w-1} \end{pmatrix} \cdot \begin{pmatrix} x_1^w & \ldots & x_1 \\ x_2^w & \ldots & x_2 \\ \vdots & \ddots & \vdots \\ x_w^w & \ldots & x_w \end{pmatrix}$$

Die obige Determinante ist also das Produkt der Determinanten der beiden Matrizen in der letzten Zeile. Die linke Matrix ist eine Vandermondesche Matrix mit Determinante

$$\prod_{1 \leq s < r \leq w} (x_r - x_s).$$

Aus der rechten Matrix kann man zeilenweise $x_1, \ldots, x_w$ herausheben, und diese dann durch $(w-1) + (w-2) + \ldots + 1 = \binom{w}{2}$ Spaltenvertauschungen und Transponieren in eine Vandermondesche Matrix umformen. Daher ist die Determinante der rechten Matrix gleich

$$(-1)^{\binom{w}{2}} \cdot \prod_{r=1}^{w} x_r \cdot \prod_{1 \leq s < r \leq w} (x_r - x_s),$$

und durch Multiplikation der beiden Determinanten erhält man die behauptete Formel.

Wegen $x_r = \alpha^{i_r}$ und $0 \leq i_1 < i_2 < \ldots < i_w \leq n - 1$ ist der Wert der Determinante $\neq 0$. Daraus folgt, dass die Lösung $\vec{x}$ von (8) existiert und eindeutig bestimmt ist, nach 1) gilt $\vec{x} = (\sigma_1, \ldots, \sigma_w)^t$.

Nimmt man an, dass es eine kürzere Rekursion als (7) für die Folge $(A_i)_{i=1}^{\delta-1}$ gibt, dann folgt daraus, dass die Zeilen der Koeffizientenmatrix im Gleichungssystem (8) linear abhängig sind und folglich die Determinante der Koeffizientenmatrix gleich 0, Widerspruch. Also folgt 2). □

Bemerkung: Man beachte, dass wegen (8) bereits die ersten $2w \leq \delta-1$ Folgenglieder $A_1, \ldots, A_{2w}$ ausreichen, um die kürzeste Rekursion der Länge w, der die Folge $A_1, \ldots, A_{\delta-1}$ genügt, eindeutig festzulegen.

Definition: Seien $(B_i)_{i=1}^{t+s}$ eine Folge in $\mathbb{F}_{2^m}$ und $\tau_0, \ldots, \tau_s \in \mathbb{F}_{2^m}$. Wenn $\tau_0 \cdot B_{j+s} + \tau_1 \cdot B_{j+s-1} + \ldots + \tau_s \cdot B_j = 0$, $j = 1, 2, \ldots, t$, die (bis auf einen multiplikativen Faktor $\neq 0$) eindeutig bestimmte kürzeste Rekursion ist, der die Folge $(B_i)_{i=1}^{t+s}$ genügt, dann nennt man das Polynom $\tau_0 + \tau_1 z + \ldots + \tau_s z^s$ das **Minimalpolynom** der Folge $(B_i)_{i=1}^{t+s}$ in $\mathbb{F}_{2^m}$.

Punkt 2) im letzten Satz sagt also aus, dass im Fall $w \leq (\delta-1)/2$ das Fehlerlokalisierungspolynom $\sigma(z)$ das Minimalpolynom der Syndromfolge $(A_i)_{i=1}^{\delta-1}$ ist.

Es bleibt die Frage: Wie findet man zu gegebener n-periodischer Folge $(A_1, A_2, \ldots)$ das Minimalpolynom $\sigma(z)$? — Die Antwort dazu liefert der **Berlekamp-Massey Algorithmus**:

Sei $(s_0, s_1, s_2, \ldots)$ eine Folge in $\mathbb{F}_q$ und $G(z) := \sum_{i=0}^{\infty} s_i z^i$ die erzeugende Funktion der Folge. Für $j = 0, 1, \ldots$ definieren wir rekursiv Polynome $g_j(z)$, $h_j(z)$, ganze Zahlen m_j und Elemente $b_j \in \mathbb{F}_q$ wie folgt:
Initialisierung: $g_0(z) = 1$, $h_0(z) = z$, $m_0 = 0$, und dann rekursiv

$$
\begin{array}{rcl}
b_j & \ldots & \text{der Koeffizient von } z^j \text{ in } g_j(z)G(z) \\
g_{j+1}(z) & = & g_j(z) - b_j h_j(z) \\
h_{j+1}(z) & = & \begin{cases} b_j^{-1} z g_j(z), & \text{wenn } b_j \neq 0 \ \wedge \ m_j \geq 0 \\ z h_j(z), & \text{sonst} \end{cases} \\
m_{j+1} & = & \begin{cases} -m_j, & \text{wenn } b_j \neq 0 \ \wedge \ m_j \geq 0 \\ m_j + 1, & \text{sonst} \end{cases}
\end{array}
$$

Dann gilt: Wenn $(s_i)_{i \in \mathbb{N}}$ das Minimalpolynom $\sigma(z)$ mit Grad w hat, dann ist $\sigma(z) = g_{2w}(z)$.

Beispiel: Sei $(s_i) = (1, 1, 0, 0, 1, 0, 1, 1, \ldots)$ eine Folge in $\mathbb{F}_2$. Die erzeugende Funktion lautet $G(z) = 1 + z + z^4 + z^6 + z^7 + \ldots$.

j	$g_j(z)$	$h_j(z)$	m_j	b_j
0	1	z	0	1
1	$1+z$	z	0	0
2	$1+z$	z^2	1	1
3	$1+z+z^2$	$z+z^2$	-1	1
4	1	z^2+z^3	0	1
5	$1+z^2+z^3$	z	0	0
6	$1+z^2+z^3$	z^2	1	0
7	$1+z^2+z^3$	z^3	2	0
8	$1+z^2+z^3$		3	

Daraus ersehen wir $g_{2k}(z) = g_8(z) = 1 + z^2 + z^3$ und man sieht, dass gilt: $s_{j+3} + s_{j+1} + s_j = 0$, $j = 0, 1, \ldots$.

Résumé nach Schritt II): Der Empfänger hat $\sigma(z) = \prod_{r=1}^{w}(1 - \alpha^{i_r} z)$ berechnet.

Ad III) Finden der Nullstellen von $\sigma(z)$: Bis $\operatorname{grad} \sigma(z) \leq 2$ existieren effektive Methoden, für höhere Grade werden die Potenzen von α (etwas vereinfacht gesprochen) einfach durchprobiert ($\leq n$-mal Auswerten von $\sigma(z)$). Für Details dazu siehe z.B. den Wikipedia-Beitrag zu „Chien

search". Die Kehrwerte der Nullstellen $z_r = \alpha^{-i_r}$ liefern die Fehlerstellen $i_r = -\log_\alpha(z_r)$, $r = 1, \ldots, w$.

Bemerkungen: 1) Der bei weitem aufwändigste der Schritte I)–III) ist der Schritt II).

2) Bei der Decodierung von nicht-binären BCH Codes geht man i. W. gleich vor. Zusätzlich müssen noch die Fehlerwerte $e_{i_1}, \ldots, e_{i_w}$ berechnet werden. Das ist mit einer Variante des Berlekamp-Massey Algorithmus' möglich.

13 Quadratische Reste Codes

Definition: Für eine ungerade Primzahl n heißt eine Zahl $r \in \mathbb{F}_n^\times$ quadratischer Rest modulo n, wenn die Gleichung $x^2 = r$ eine Lösung $x \in \mathbb{F}_n$ hat. Ist das nicht der Fall, heißt r quadratischer Nichtrest modulo n. Wir bezeichnen mit Q bzw. N die Menge der quadratischen Reste bzw. Nichtreste modulo n.

Ist β ein primitives Element in $\mathbb{F}_n$, so sind offensichtlich die Elemente $\beta^2, \beta^4, \ldots, \beta^{n-1} = 1$ quadratische Reste. Die ungeraden Potenzen von β sind quadratische Nichtreste: Angenommen $x^2 = \beta^{2i+1}$ hat eine Lösung $x = \beta^j \in \mathbb{F}_n$, dann erhalten wir $\beta^{2i+1} = \beta^{2j}$ und folglich $\mathrm{ord}(\beta) = \underbrace{n-1}_{\text{gerade}} \mid \underbrace{2(i-j)+1}_{\text{ungerade}}$, Widerspruch. So bekommen wir

$$Q = \{\beta^{2i} \mid i = 0, \ldots, (n-3)/2\}, \quad N = \{\beta^{2i+1} \mid i = 0, \ldots, (n-3)/2\}$$

und damit $|Q| = |N| = (n-1)/2$. Es gibt also immer gleich viele quadratische Reste wie Nichtreste. Für die Bestimmung von Q bzw. N benötigt man natürlich kein primitives Element β, man kann auch $Q = \{x^2 \mid x \in \mathbb{F}_n^\times\}$ und $N = \mathbb{F}_n^\times \setminus Q$ benützen.

Beispiel: Für $n = 11$ ist $Q = \{1, 4, 9, 5, 3\}$ und $N = \{2, 6, 7, 8, 10\}$.

Aus der obigen Darstellung von Q und N mit Hilfe der Potenzen von β erhält man auch unmittelbar: $Q \cdot Q = N \cdot N = Q$, $Q \cdot N = N$ und Q ist eine Untergruppe von $(\mathbb{F}_n^\times, \cdot)$ mit Index 2.

Sei jetzt p eine Primzahl, die quadratischer Rest modulo n ist und α eine primitive n-te Einheitswurzel in einem Erweiterungskörper $\mathbb{F}_{p^m}$ von $\mathbb{F}_p$. Wie bei den BCH Codes kann man $m = \mathrm{ord}(p) \bmod n$ und $\alpha := \gamma^{(p^m-1)/n}$ wählen, wobei γ ein primitives Element in $\mathbb{F}_{p^m}$ ist.

Satz: Die Polynome

$$g_Q(x) := \prod_{r \in Q} (x - \alpha^r) \quad \text{und} \quad g_N(x) := \prod_{r \in N} (x - \alpha^r)$$

haben Koeffizienten in $\mathbb{F}_p$ und es gilt

$$x^n - 1 = (x-1) \cdot g_Q(x) \cdot g_N(x).$$

Beweis: Sei $g_Q(x) = a_0 + a_1 x + \ldots + a_\ell x^\ell$ mit $a_i \in \mathbb{F}_{p^m}$ und $\ell = (n-1)/2$. Bei den folgenden Umformungen rechnen wir im Polynomring $\mathbb{F}_{p^m}[y] \supset \mathbb{F}_{p^m}[x]$ mit $x = y^p$:

$$\begin{aligned} a_0^p + a_1^p x + \ldots + a_\ell^p x^\ell &= a_0^p + a_1^p y^p + \ldots + a_\ell^p y^{\ell p} = (a_0 + a_1 y + \ldots + a_\ell y^\ell)^p \\ &= (\prod_{r \in Q} (y - \alpha^r))^p = \prod_{r \in Q} (y^p - \alpha^{rp}) = \prod_{r \in Q} (y^p - \alpha^r) = g_Q(y^p) = g_Q(x) \\ &= a_0 + a_1 x + \ldots + a_\ell x^\ell \end{aligned}$$

Koeffizientenvergleich liefert $a_i^p = a_i$, also $a_i \in \mathbb{F}_p$.

Aus $\mathbb{F}_n = \{0\} \dot\cup Q \dot\cup N$ und $x^n - 1 = \prod_{i=0}^{n-1}(x - \alpha^i)$ folgen die übrigen Behauptungen. □

Definition: Seien n, m und p wie zuvor gewählt. Die zyklischen Codes der Länge n über dem Alphabet $\mathbb{F}_p$ mit Generatorpolynom $g_Q(x)$ bzw. $g_N(x)$ heißen quadratische Reste Codes und werden mit C_Q bzw. C_N bezeichnet.

Weil $g_Q(x)$ und $g_N(x)$ Grad $(n-1)/2$ haben, ist die Dimension von C_Q und C_N gleich $(n+1)/2$.

Beispiel: Wegen $6^2 \equiv 2 \bmod 17$ ist $p = 2$ quadratischer Rest modulo $n = 17$. Daher existiert ein binärer quadratischer Reste Code der Länge 17. Es gilt hier $Q = \{1, 4, 9, 16, 8, 2, 15, 13\} = C_1$ die 1-te Kreisteilungsklasse von 2 modulo 17, daher ist $g_Q(x) = m^{(1)}(x)$ als Minimalpolynom von α irreduzibel und C_Q in diesem Fall ein spezieller BCH Code. Wie man durch Zerlegung von $x^{17} - 1$ in irreduzible Faktoren zeigen kann, gilt

$$g_Q(x) = x^8 + x^5 + x^4 + x^3 + 1, \quad g_N(x) = x^8 + x^7 + x^6 + x^4 + x^2 + x + 1$$

(oder mit vertauschten Rollen für $g_Q(x)$ und $g_N(x)$, je nachdem wie man das Minimalpolynom von α wählt). Weiters kann man zeigen, dass sowohl C_Q als auch C_N $[17, 9, 5]$ Codes sind.

Satz: Die quadratischen Reste Codes C_Q und C_N sind äquivalent, insbesondere gibt es eine Permutation der Stellen, die C_Q in C_N überführt.

Beweis: Sei $s \in N$ ein quadratischer Nichtrest modulo n. Wir betrachten die Abbildung

$$\psi : \begin{cases} \mathbb{F}_p[x]/(x^n - 1) & \to & \mathbb{F}_p[x]/(x^n - 1) \\ v(x) & \mapsto & v(x^s) \end{cases}$$

Dabei handelt es sich um eine $\mathbb{F}_p$-lineare Bijektion, die die Monome $x^i \mapsto x^{is \bmod n}$ permutiert: Die Abbildung $\varphi : \{0, 1, \ldots, n-1\} \to \{0, 1, \ldots, n-1\}$ mit $\varphi(i) = is \bmod n$ permutiert die Exponenten und hat wegen $s \in N$ die Eigenschaften $\varphi(Q) = N$ und $\varphi(N) = Q$. Dementsprechend gilt:

$$c(x) \in C_Q \;\Leftrightarrow\; \forall i \in Q : \; c(\alpha^i) = 0 \;\Leftrightarrow\; \forall i \in N : \; c(\alpha^{is}) = 0 \;\Leftrightarrow\; \psi(c(x)) = c(x^s) \in C_N$$

Also entspricht $\psi|_{C_Q} : C_Q \to C_N$ einer Permutation der Stellen, was zeigt, dass C_Q und C_N äquivalente Codes sind. □

Beispiel: Im obigen Beispiel mit $n = 17$ und $p = 2$ ist $3 \in N$ und wir erhalten

$$\psi(g_Q(x)) = x^{24} + x^{15} + x^{12} + x^9 + 1 \equiv x^{15} + x^{12} + x^9 + x^7 + 1 \bmod (x^{17} - 1).$$

Eine einfache Division zeigt $\psi(g_Q(x)) = g_N(x) \cdot (x^7 + x^6 + x^3 + x + 1) \in C_N$.

Satz: Ein Codewort $\vec{c} = c(x)$ eines quadratischen Reste Codes C_Q der Länge n mit $c(1) \neq 0$ hat Hamming-Gewicht $w(\vec{c}) > \sqrt{n}$. Ist $n \equiv 3 \bmod 4$, so gilt sogar $w(\vec{c})^2 - w(\vec{c}) + 1 \geq n$.

Beweis: Sei $s \in N$ ein quadratischer Nichtrest modulo n. Dann ist nach dem Beweis des vorigen Satzes $c(x^s) \bmod (x^n - 1) \in C_N$. Daher existieren Polynome $a(x), b(x) \in \mathbb{F}_p[x]$, sodass $c(x) = g_Q(x) \cdot a(x)$ und $c(x^s) \bmod (x^n - 1) = g_N(x) \cdot b(x)$. Daraus folgt

$$(c(x) \cdot c(x^s)) \bmod (x^n - 1) = (\underbrace{g_Q(x) \cdot g_N(x)}_{= \frac{x^n - 1}{x - 1}} \cdot a(x) \cdot b(x)) \bmod (x^n - 1).$$

Dividieren wir $a(x)b(x)$ durch $x - 1$, so erhält man $a(x)b(x) = q(x)(x-1) + r$, wobei $r \in \mathbb{F}_p$ mit $r = a(1)b(1) \neq 0$, weil aus der Voraussetzung $c(1) \neq 0$ folgt $a(1), b(1) \neq 0$. Setzen wir diese Darstellung für $a(x)b(x)$ oben ein, so erhalten wir

$$(c(x) \cdot c(x^s)) \bmod (x^n - 1) = \frac{x^n - 1}{x - 1} \cdot (q(x) \cdot (x-1) + r) \bmod (x^n - 1) = r \cdot (x^{n-1} + x^{n-2} + \ldots + 1).$$

Also hat $(c(x)\cdot c(x^s)) \bmod (x^n-1)$ Hamming-Gewicht n. Wenn nun $c(x)$ (und damit auch $c(x^s)$) das Gewicht w hat, so hat $(c(x)\cdot c(x^s)) \bmod (x^n-1)$ höchstens Gewicht w^2 und folglich ist $n \le w^2$. Da n eine Primzahl ist, folgt $w > \sqrt{n}$.

Ist zusätzlich $n \equiv 3 \bmod 4$, dann ist -1 ein quadratischer Nichtrest modulo n: Aus $n = 4\ell+3$ folgt $(-1)^{(n-1)/2} = (-1)^{2\ell+1} = -1$, also ist das Jacobi-Symbol $\left(\frac{-1}{n}\right)$ gleich -1, also -1 ein quadratischer Nichtrest mod n. Daher können wir in obigem Beweis $s=-1$ wählen und beachten, dass im Produkt $(c(x)\cdot c(x^{-1})) \bmod (x^n-1)$, wenn man jeden Summanden von $c(x)$ mit jedem Summanden von $c(x^{-1})$ multipliziert, in der entstehenden Summe mindestens w Summanden den Exponenten 0 haben, daher können wir schließen $w^2-(w-1) \ge n$, und die Behauptung ist gezeigt. □

Für einen binären (d. h. $p=2$) quadratischen Reste Code C kann man zeigen (ohne Beweis), dass die Minimaldistanz $d(C)$ immer ungerade ist. Daher gilt für ein Codewort $\vec{c} = c(x) \in C$ mit minimalem Gewicht $c(1) = w(\vec{c}) \bmod 2 \neq 0$ und daher mit dem letzten Satz $w(\vec{c}) > \sqrt{n}$.

Satz: Binäre quadratische Reste Codes der Länge n haben eine Minimaldistanz d mit $d > \sqrt{n}$. Ist $n \equiv 3 \bmod 4$, dann gilt sogar $d^2 - d + 1 \ge n$.

Golay Codes

Wir definieren jetzt den binären Golay Code G_{23} und den ternären Golay Code G_{11} als spezielle quadratische Reste Codes. Man kann zeigen (ohne Beweis), dass G_{23} und G_{11} (bis auf Äquivalenz) die einzigen nicht-trivialen perfekten Codes sind, die mehr als einen Fehler korrigieren können.

Binärer Golay Code G_{23}: Die Länge $n=23$, $p = 2 \equiv 5^2 \bmod 23 \in Q$ ist quadratischer Rest, also existiert ein binärer quadratischer Reste Code der Länge 23. Die Dimension dieses Codes ist $(n+1)/2 = 12$ und für die Minimaldistanz d erhalten wir wegen $23 \equiv 3 \bmod 4$ die Bedingung $d^2 - d + 1 \ge 23$, woraus $d \ge 6$ folgt. Im binären Fall ist d ungerade, weshalb $d \ge 7$ gelten muss. Jetzt kann man mit Hilfe der Hamming Schranke nachrechnen (Übungen), dass ein binärer $[23, 12, 7]$-Code 3-perfekt ist, daher kann d nicht größer als 7 sein. Die Kreisteilungsklassen von 2 mod 23 stimmen (abgesehen von $C_0 = \{0\}$) mit den quadratischen Resten/Nichtresten überein:

$$C_1 = \{1,2,4,8,16,9,18,13,3,6,12\} = Q, \quad C_5 = \{5,10,20,17,11,22,21,19,15,7,14\} = N,$$

daher sind die Generatorpolynome $g_Q(x)$ bzw. $g_N(x)$ irreduzibel und können durch Zerlegung von $x^{23}-1$ in irreduzible Faktoren erhalten werden:

$$x^{23}-1 = (x-1)(x^{11}+x^{10}+x^6+x^5+x^4+x^2+1)(x^{11}+x^9+x^7+x^6+x^5+x+1)$$

Durch Verlängerung kann man aus G_{23} den erweiterten Golay Code G_{24} erhalten:

$$G_{24} = \{x_1 \dots x_{24} \in \mathbb{F}_2^{24} \mid x_1 \dots x_{23} \in G_{23} \wedge x_1 + \dots + x_{24} = 0 \bmod 2\}$$

G_{24} hat die Parameter $[24, 12, 8]$, ist selbstdual und jedes Wort von G_{24} hat ein durch 4 teilbares Gewicht.

Ternärer Golay Code G_{11}: Die Länge $n=11$, $p = 3 \equiv 5^2 \bmod 11 \in Q$ ist quadratischer Rest, also existiert ein ternärer ($p=3$) quadratischer Reste Code der Länge 11. Die Dimension dieses Codes ist $(n+1)/2 = 6$ und für die Minimaldistanz d kann man zeigen, dass $d=5$. Wieder kann man mit Hilfe der Hamming Schranke nachrechnen, dass ein ternärer $[11, 6, 5]$-Code 2-perfekt ist. Die Kreisteilungsklassen von 3 mod 11 stimmen (wieder abgesehen von $C_0 = \{0\}$) mit den quadratischen Resten/Nichtresten überein:

$$C_1 = \{1,3,9,5,4\} = Q, \quad C_2 = \{2,6,7,10,8\} = N,$$

daher sind die Generatorpolynome $g_Q(x)$ bzw. $g_N(x)$ wieder irreduzibel:

$$x^{11} - 1 = (x-1)(x^5 + x^4 - x^3 + x^2 - 1)(x^5 - x^3 + x^2 - x - 1)$$

Analog wie im binären Fall kann man aus G_{11} den erweiterten ternären Golay Code G_{12} konsturieren.