Zur Dimension und zum Minimalabstand von BCH–Codes

Schriftliche Hausarbeit
im Rahmen der Ersten Staatsprüfung
für das Lehramt für die Sekundarstufe II

dem Staatlichen Prüfungsamt Dortmund
vorgelegt von

Stefan Wiegand

Ruhr–Universität Bochum, Oktober 2000

Themensteller: Herr Professor Dr. L. Gerritzen
Fachbereich: Mathematik

Vorwort

Die *Kanalcodierung* bzw. die *Theorie fehlerkorrigierender Codes* wurde Ende 1940 mit den Arbeiten von Golay, Hamming und Shannon begründet. Überall dort, wo es um die möglichst fehlerfreie Datenübermittlung geht, ist sie von großer praktischer Bedeutung. Als Beispiele ihrer Anwendung wären etwa das Auslesen zerkratzter *„Compact Discs“* (kurz: *CDs*) durch CD-Player oder die Übertragung von Satellitenaufnahmen zur Erde zu nennen.
Es ist möglich, zu sendende Nachrichten schon vor der Übertragung mathematisch so aufzuarbeiten, dass eine fehlerhafte Übertragung bis zu einem gewissem Grade erkannt wird. Darüber hinaus ist es erreichbar, ursprünglich gesendete Nachrichten aus fehlerhaft übertragenen Daten zu rekonstruieren. Seit der Begründung der Kanalcodierung wurden zahlreiche Möglichkeiten entwickelt, Nachrichten in diesem Sinne zu codieren.
Die nach ihren Entdeckern *Bose*, *Chaudhuri* und *Hocquenghem* benannte und zu den *zyklischen Codes* gehörige Klasse der *BCH-Codes* hat sich von großer praktischer Relevanz erwiesen. Nur wenige Parameter, wie die *Länge* n, die *Dimension* k und die *Minimaldistanz* d, sowie die Ordnung des zugrundeliegenden Körpers q, bestimmen einen BCH-Code vollständig. Bei der Konstruktion von BCH-Codes werden die Länge n, die Körperordnung q und eine untere Schranke für die Minimaldistanz vorgegeben, die als *Entwurfsdistanz* δ bezeichnet wird. Die Dimension k und die wahre Minimaldistanz d liegen mit diesen Eingangsparametern nur implizit vor. Durch die Vorgabe der Entwurfsdistanz δ bei der Konstruktion kann bei einem BCH-Code eine Mindestfehlerkorrekturfähigkeit vorausgesetzt werden. Dies ist nicht bei allen Codes der Fall. *E. R. Berlekamp* (vgl. [Berlekamp 1968], S. vii) betonte jedoch, dass die Wichtigkeit von BCH-Codes vor allem auf deren

strenger algebraischer Struktur begründet ist, die einfach zu implementierende Decodierungsalgorithmen von geringer Komplexität ermöglicht.
Die Berechnung der Dimension k ist kein tiefsinniges Problem. In der Literatur zu fehlerkorrigierenden Codes gibt es für wichtige Spezialfälle einige Vorschläge, die das Problem lösen (z.B. [Berlekamp 1968] und [MacWilliams et al. 1977]). Für den allgemeinen Fall von BCH-Codes sind jedoch zur Dimension nur wenige Angaben zu finden.
Im Fall der wahren Minimaldistanz d sind seit längerem weitere Schranken neben der Entwurfsdistanz δ bekannt und es wurden einige große Unterklassen von BCH-Codes entdeckt, bei denen die Minimaldistanz mit der Entwurfsdistanz übereinstimmt. Die konkrete Angabe der Minimaldistanz d gehört jedoch im Allgemeinen zu den alten, bisher noch ungelösten Problemen der Kanalcodierung (vgl. *P. Charpin* in [Pless et al. 1998], S. 990).

Der erste Teil dieser Arbeit bespricht die Theorie der zyklischen Codes, die eng mit der Theorie der endlichen Körper verbunden ist. In diesem Rahmen werden Berechnungshilfen für die *multiplikative Ordnung von* $q \bmod n$ entwickelt. Dies ist die kleinste positive ganze Zahl $m_n(q)$, für die das Polynom $X^n - 1$ über dem endlichen Körper mit $q^{m_n(q)}$ Elementen vollständig in Linearfaktoren zerfällt.
Im zweiten Teil wird ein im symbolischen Algebrasystem *MATHEMATICA* implementierter Algorithmus beschrieben, der die Dimension beliebiger BCH-Codes berechnet. Ferner wird der gegenwärtige Stand der Forschung zur Minimaldistanz von BCH-Codes erörtert und ein Verfahren zum Auffinden der Minimaldistanzen zyklischer Codes angegeben, dass auf die Arbeiten [Augot et al. 1991] und [Augot et al. 1992] zurückgeht. Im Kern dieses Verfahrens steht das Lösen bestimmter algebraischer Gleichungen, die *Newton Identitäten* genannt werden. Newton Identitäten beschreiben den Zusammenhang zwischen den Koeffizienten des *Mattson-Solomon Polynoms* und des *Lokator Polynoms* eines Codeworts, was das Auffinden der Positionen von Null verschiedener Komponenten des Codeworts ermöglicht. Als Beispiel für das Verfahren wird die Minimaldistanz des binären BCH-Codes im engeren Sinne, der Länge $n = 17$ und Entwurfsdistanz $\delta = 3$ berechnet.
Der Abschluss dieser Arbeit stellt einen Abriss über die Geschichte und die Anwendungsmöglichkeiten einer wichtigen Klasse von BCH-Codes dar, den *Reed-Solomon Codes.*

Inhaltsverzeichnis

1. Grundlagen

Die *Codierungstheorie* ist eine noch junge Teildisziplin der diskreten Mathematik. Unter dem Begriff Codierung versteht man im Allgemeinen eine Zuordnung einer Nachrichtenmenge zu einer Menge von Symbolen oder Zeichen. Es können unterschiedliche Ziele damit verfolgt werden, z.B. das Verschlüsseln von Daten gegen unerwünschte Einsicht anderer (*Kryptographie*) oder die Komprimierung einer Nachrichtenmenge, um die Datenlast zu verkleinern (*Quellencodierung*). Eine weitere Motivation stellt das Codieren einer Nachrichtenmenge dar, um Fehler durch Störungen bei der Übertragung durch einen Kanal erkennen und korrigieren zu können, man spricht in diesem Rahmen von der *Kanalcodierung* oder der *Theorie fehlerkorrigierender Codes*. BCH-Codes sind fehlerkorrigierende Codes. In diesem Kapitel sollen die Grundlagen erarbeitet werden, die zu den BCH-Codes führen.

Im ersten Teil werden *zyklische Codes* als Unterklasse der *linearen Codes* eingeführt und einige ihrer wichtigsten Eigenschaften besprochen. Lineare Codes zählen ihrerseits zu der allgemeineren Klasse der *Blockcodes*. Das sind Codes, in denen alle Codewörter[1] voneinander unabhängig sind, über die gleiche Wortlänge verfügen und im Gegensatz zu sogenannten *Faltungscodes* ein konstantes Verhältnis zwischen obligatorischen *Informationszeichen* und redundanten *Prüfzeichen* besteht. Die Argumentation führt zur Betrachtung von Polynomen im Zusammenhang mit endlichen Körpern im zweiten Abschnitt. Im dritten Teil dieses Kapitels werden BCH-Codes definiert und ihre grundlegenden Eigenschaften dargestellt.

[1] Codewörter sind die Elemente eines Codes.

1.1. Einführung in die Theorie zyklischer Codes

BCH-Codes werden zunächst als lineare Codes betrachtet. Dazu werden *lineare Codes* als Vektorräume mit Hammingmetrik eingeführt. In diesem und in den folgenden Abschnitten sei $GF(q)$ stets ein beliebiger endlicher Körper[2] und k, n positive ganze Zahlen.

1.1.1. Lineare Codes

Definition 1.1
Eine *Nachricht* u ist ein k-dimensionaler Zeilenvektor $(u_0, u_1, \ldots, u_{k-1}) \in GF(q)^k$. Der Vektorraum $GF(q)^k$ wird *Nachrichtenraum* über $GF(q)$ genannt. Die Komponenten $u_i \in GF(q)$, $0 \leq i \leq k-1$, von Nachrichten heißen *Nachrichtensymbole.*

Für $n \geq k$ werden Nachrichten der Länge k durch Codierung in n-stellige *Codewörter* $(c_0, c_1, \ldots, c_{n-1}) \in GF(q)^n$ überführt. Beispielsweise fügt man bei der systematischen Codierung (siehe Bemerkung 1.12) an die k-stelligen Nachrichten $n-k$ *Prüfsymbole* an. Wie die Prüfsymbole gewählt werden und was ein Codewort von „gewöhnlichen" Vektoren $y \in GF(q)^n$ unterscheidet, hängt von der besonderen Art des Codierens ab. Die sogenannte *lineare Codierung* soll in der folgenden Definition präzisiert werden:

Definition 1.2
Es sei $GF(q)^n$ ein endlichdimensionaler Zeilenvektorraum über dem endlichen Körper $GF(q)$. Ein *linearer* (n,k)*-Code* $\mathcal{C}$ über $GF(q)$ ist der Kern[3] einer surjektiven linearen Abbildung $\theta : GF(q)^n \longrightarrow GF(q)^{n-k}$, der *Kontrollabbildung.* Die Elemente $c \in \mathcal{C}$ heißen *Codewörter.* Der Parameter k wird die *Dimension* und der Parameter n die *Länge* des Codes $\mathcal{C}$ genannt.

Nach obiger Definition ist ein linearer (n,k)-Code ein k-dimensionaler Untervektorraum des $GF(q)$-Vektorraums $GF(q)^n$, weil der Kern einer linearen Abbildung

[2] Zu endlichen Körpern (*Galois Felder*) vergleiche Abschnitt 1.2. Der Parameter q bezeichnet die Anzahl seiner Elemente, sie ist eine Primzahlpotenz und bestimmt den Körper im wesentlichen vollständig.
[3] Der Kern einer Abbildung bezeichnet das Urbild der 0.

stets ein Untervektorraum des Urbildes ist (vgl. z.B. [Fischer et al. 1986]).

Bemerkung 1.3
Bei der Übertragung von Codewörtern treten gelegentlich *Fehler* durch Kanalstörungen auf. Fehler bedeuteten in diesem Zusammenhang, dass einige Komponenten des empfangenen Vektors andere Elemente aus $GF(q)$, als ursprünglich gesendet, besitzen.
Durch Auswahl von Basen zu $GF(q)^{n-k}$ und $GF(q)^n$ ist der Kontrollabbildung θ eine Abbildugsmatrix Θ zugeordnet. Solch eine $(n-k) \times n$ Matrix Θ wird als die *Kontrollmatrix* von $\mathcal{C}$ bezeichnet, denn: fehlerfrei übertragene Codewörter $c \in \mathcal{C}$ werden als *transponierte Vektoren*[4] c^{tr} durch Linksmultiplikation mit Θ auf die Null abgebildet, d.h. $\Theta c^{tr} = 0$. Im Gegensatz dazu werden fehlerhaft übertragene Vektoren $y \notin \mathcal{C}$, die kein anderes Codewort $c \neq c' \in \mathcal{C}$ darstellen, auf von Null verschiedene Bilder von θ abgebildet, also $\Theta y^{tr} \neq 0$.

Satz 1.4 (und Definition)
Es seien $w = (w_0, w_1, \ldots, w_{n-1})$ *und* $w' = (w'_0, w'_1, \ldots, w'_{n-1}) \in GF(q)^n$. *Die Abbildung* $\mathrm{dist} : GF(q)^n \times GF(q)^n \longrightarrow \mathbb{N}$ *definiert durch*[5]

$$\mathrm{dist}(w, w') := \left|\{0 \leq i \leq n-1 \ : \ w_i \neq w'_i\}\right| ,$$

ist eine Metrik auf $GF(q)^n$, *die als die* Hammingmetrik *bezeichnet wird. Das Paar* $(GF(q)^n, \mathrm{dist})$ *ist ein metrischer Raum. Die nichtnegative ganze Zahl* $\mathrm{dist}(w, w')$ *heißt* Hammingabstand zwischen w und $w' \in GF(q)^n$.

Beweis. [Betten et al. 1998], (S. 6) ∎

Bemerkung 1.5
Neben Länge n und Dimension k eines (n,k)-Codes $\mathcal{C}$ ist ein weiterer charakterisierender Parameter für den Code von Bedeutung, die *Minimaldistanz*

$$d := \min_{c,c' \in \mathcal{C}, c \neq c'} \mathrm{dist}(c, c') ,$$

die auch als *Minimalabstand* bezeichnet wird. Welche Bedeutung ihr zukommt, zeigt der folgende Satz.

[4] Der zum Zeilenvektor $w = (w_0, w_1, \ldots, w_{n-1})$ transponierte Vektor ist der Spaltenvektor w^{tr} mit den gleichen Komponenten $w_0, w_1, \ldots, w_{n-1} \in GF(q)$.
[5] $|.|$ meint hier die Ordnung, d.h. die Anzahl der Elemente einer Menge.

Satz 1.6
Ein Code $\mathcal{C}$ mit Minimalabstand d kann $\lfloor \frac{1}{2}(d-1) \rfloor$ Fehler[6] korrigieren. Ist d eine gerade ganze Zahl, dann kann der Code darüber hinaus $\frac{d}{2}$ Fehler erkennen.

Beweis. [MacWilliams et al. 1977], (S. 10) ■

Viele Verfahren zur Fehlererkennung und -korrektur stützen sich, z.B. im Falle $d = 2t+1$, auf die Tatsache, dass fehlerhaft übertragene Vektoren bzgl. des Hammingabstands einem nächstliegenden Codewort zugeordnet werden können. Sind bis zu t Fehler in einem Codewort aufgetreten, kann der empfangene Vektor dem ursprünglich gesendeten zugeordnet werden. Derartige Übertragungsfehler sind also korrigierbar.
Im Fall $d = 2t$ kann ein übertragener Vektor mit t Fehlern zwei Codewörtern bzgl. einem minimalen Hammingabstand zugeordnet werden. Solche Fehler sind nicht verlässlich korrigierbar, aber erkennbar.
Sind bei der Übertragung von einem Codewort $c \in \mathcal{C}$ in beiden Fällen mehr als $t+1$ Fehler aufgetreten, dann hat der empfangene Vektor zu einem anderen Codewort als dem ursprünglich gesendeten eine minimale Hammingdistanz. Solche Fehler sind erkennbar, wenn nicht das ursprünglich gesendete Codewort $c \in C$ bei der Übertragung in ein anderes Codewort $c \neq c' \in \mathcal{C}$ überführt wurde. Solche Übertragungsfehler sind aber generell nicht korrigierbar.
Die Wahrscheinlichkeit dafür, dass durch die Übertragung viele Fehler in einem Codewort auftreten ist kleiner als das nur wenige Fehler auftreten, daher bezeichnet man dieses Konzept als *Maximum-Likelihood-Decodierung.*

Bemerkung 1.7
Ein Linearer (n,k)-Code $\mathcal{C}$ mit Minimaldistanz d wird zukünftig als (n,k,d)-Code bezeichnet.

Definition 1.8
Das *Hamminggewicht* eines Vektors $w = (w_0, \ldots, w_{n-1}) \in GF(q)^n$ ist definiert durch

$$\mathrm{wt}(w) := \mathrm{dist}(w, 0) \ .$$

Das sich anschließende Lemma und der darauf folgende Satz werden durch die Vektorraumeigenschaft eines Codes impliziert. Die beiden Aussagen geben Aus-

[6] $\lfloor x \rfloor$ die größte ganze Zahl $\leq x$, $\lceil x \rceil$ bezeichnet die kleinste ganze Zahl $\geq x$.

kunft darüber, wie bei linearen Codes die Bestimmung der Minimaldistanz vereinfacht werden kann.

Lemma 1.9
Es seien $w, v \in GF(q)^n$. Dann gilt: $\mathrm{dist}(w, v) = \mathrm{wt}(w - v)$ *und damit*

$$d = \min_{c \in \mathcal{C}, c \neq 0} \mathrm{wt}(c) \ ,$$

d.h. der Minimalabstand eines Codes $\mathcal{C}$ stimmt mit dem kleinsten Gewicht eines von Null verschiedenen Codeworts überein.

Beweis. [MacWilliams et al. 1977], (S. 8) ∎

Satz 1.10
Ein (n, k)-Code $\mathcal{C}$ hat genau dann die Minimaldistanz d, wenn in jeder Kontrollmatrix Θ von $\mathcal{C}$ je $d-1$ Spalten linear unabhängig sind und es d linear abhängige Spalten gibt.

Beweis. [Betten et al. 1998], (S. 11) ∎

1.1.2. Generatormatrizen linearer Codes

Es sei $u = (u_0, \ldots, u_{k-1}) \in GF(q)^k$ eine Nachricht, wie erhält man das zu u korrespondierende Codewort $c = (c_0, \ldots, c_{n-1}) \in \mathcal{C}$?

Lemma 1.11
Es sei $GF(q)^k$ der Nachrichtenraum mit Standardbasis[7]*, $u = (u_0, \ldots, u_{k-1})$ eine Nachricht und $\mathbf{L} = \{l_0, \ldots, l_{k-1}\} \subseteq \mathcal{C}$ Basis des linearen (n, k)-Codes $\mathcal{C}$ mit Kontrollabbildung θ. Dann ist $\gamma : GF(q)^k \longrightarrow GF(q)^n$ mit Bild γ = Kern θ vermöge $\gamma(u) := \sum_{i=0}^{k-1} u_i l_i$ eine injektive $GF(q)$-lineare Abbildung.*

Beweis. Die Linearität von γ sieht man sofort. Es sei $\gamma(u) = \gamma(u')$. Dann ist $\sum_{i=0}^{k-1}(u_i - u'_i)l_i = 0$. Da $l_i \in \mathbf{L}$ linear unabhängig ist, folgt $u_i - u'_i = 0$ für alle $i = 0, \ldots, k-1$ und damit $u = u'$. Die Abbildung ist also injektiv. ∎

[7] Die Standardbasis von $GF(q)^k$ besteht aus den Einheitsvektoren $e_1, \ldots, e_k \in GF(q)^k$.

Bemerkung 1.12

(i). Ist die Basis $\mathbf{L}$ so gewählt, dass jedes Codewort c zu einer Nachricht u an den ersten k-Stellen aus den Nachrichtensymbolen, d.h.

$$c_0 = u_0, \quad c_1 = u_1, \quad \ldots, \quad c_{k-1} = u_{k-1} \ ,$$

gefolgt von $n-k$ redundanten *Prüfsymbolen*

$$c_k, \ldots, c_{n-1} \ ,$$

besteht, so spricht man von *systematischer Codierung.*

(ii). Es seien γ und $l_i = (l_{i,0}, \ldots, l_{i,n-1})$ wie in Lemma 1.11. Aus der Definition von γ kann abgeleitet werden, dass

$$\begin{aligned}
\gamma(u) &= \sum_{i=0}^{k-1} u_i l_i \\
&= u_0(l_{0,0}, \ldots, l_{0,n-1}) + \cdots + u_{k-1}(l_{k-1,0}, \ldots, l_{k-1,n-1}) \\
&= (u_0, \ldots, u_{k-1}) \begin{pmatrix} l_{0,0} & \ldots & l_{0,n-1} \\ \vdots & \ddots & \vdots \\ l_{k-1,0} & \ldots & l_{k-1,n-1} \end{pmatrix} \\
&=: u\,\Gamma \ .
\end{aligned}$$

Die $k \times n$ Matrix Γ heißt die *Generatormatrix* von $\mathcal{C}$ und ist bis auf Austausch der Basis $\mathbf{L}$ und der Basis des Nachrichtenraums eindeutig bestimmt.

(iii). Ist Γ der Form $(E_k|A)$, wobei E_k die $k \times k$ Einheitsmatrix über GF(q) bezeichnet und A eine $k \times (n-k)$ Matrix über GF(q) ist, so ist $\mathcal{C}$ systematisch codiert und Γ heißt *systematische Generatormatrix.* Es gibt zu jedem linearen (n,k)-Code $\mathcal{C}$ einen *äquivalenten* (n,k)-Code $\mathcal{C}'$ mit einer systematischen Generatormatrix (vgl. [Betten et al. 1998], S. 26f). Äquivalenz heißt dabei, dass die Generatormatrix von $\mathcal{C}$ durch eine invertierbare *Isometrie*[8] bzgl. der Hammingmetrik in die Generatormatrix von $\mathcal{C}'$ überführt werden kann.

[8]Mit Isometrien sind hier bijektive lineare Abbildungen gemeint, unter denen das Hamminggewicht invariant ist.

Für die Minimaldistanz linearer Codes existiert die folgende Schranke, auf die in Abschnitt 3.1 zurückgegriffen wird.

Satz 1.13 (Singleton-Schranke)
Für jeden linearen (n, k, d)-Code $\mathcal{C}$ gilt

$$d \leq n - k + 1 \ .$$

Beweis. Sei $(E_k|A)$ eine systematische $k \times n$ Generatormatrix (vgl. Bemerkung 1.12), die einen zu $\mathcal{C}$ äquivalenten Code generiert, dann hat für jeden Zeileneinheitsvektor $e_i \in GF(q)^k$ das Codewort $e_i \cdot (E_k|A) = c$ höchstens ein Gewicht von $\mathrm{wt}(c) \leq n - (k-1)$. ■

Es seien wieder $w, w' \in GF(q)^n$ und $\langle w, w' \rangle := \sum_i w_i w'_i$ die Standardbilinearform, das *Skalarprodukt*. Gilt $\langle w, w' \rangle = 0$ nennt man zwei Vektoren w, w' *zueinander orthogonal*. Mit Hilfe der Orthogonalität von Vektoren lässt sich aus einem (n, k)-Code $\mathcal{C}$ unmittelbar ein neuer gewinnen:

Definition 1.14
Es sei $\mathcal{C}$ ein linearer (n, k)-Code über $GF(q)$. Der zu $\mathcal{C}$ *duale* oder *orthogonale Code* $\mathcal{C}^\perp$ ist die Menge der Vektoren, die orthogonal zu allen Codewörtern aus $\mathcal{C}$ sind:

$$\mathcal{C}^\perp = \{v \in GF(q)^n | \langle v, c \rangle = 0 \ \forall c \in \mathcal{C}\} \ .$$

Lemma 1.15
Für die Kontroll- und Generatormatrizen Θ und Γ eines linearen (n, k)-Codes $\mathcal{C}$ über $GF(q)$ gelten:

$$\Theta\Gamma^{tr} = 0 \text{ und } \Gamma\Theta^{tr} = 0 \ ,$$

wobei '0' die Nullmatrix meint. D.h. jede Zeile von Γ steht orthogonal auf allen Zeilen von Θ und jede Zeile von Θ steht orthogonal auf allen Zeilen von Γ.

Beweis. Es seien Θ und Γ wie in Bemerkungen 1.3 und 1.12. Es folgt:

$$\Theta\Gamma^{tr} = \Theta \cdot (l_0^{tr}, \ldots, l_{k-1}^{tr}) = (\Theta \cdot l_0^{tr}, \ldots, \Theta \cdot l_{k-1}^{tr}) = 0 \ ,$$

da $l_i \in \mathcal{C} \ \forall \ i = 0, \ldots, k-1$. Umgekehrt folgt wegen $(\Theta\Gamma^{tr})^{tr} = \Gamma\Theta^{tr}$ und $0^{tr} = 0$ die Behauptung. ■

Aus dem in Lemma 1.15 genannten Zusammenhang folgt unmittelbar:

Folgerung 1.16
Die Generatormatrizen (Kontrollmatrizen) von $\mathcal{C}$ sind genau die Kontrollmatrizen (Generatormatrizen) von $\mathcal{C}^{\perp}$.

1.1.3. Zyklische Codes

Die am besten untersuchte Klasse der linearen Codes sind die *zyklischen Codes*, zu denen die BCH-Codes zählen.

Definition 1.17
Ein linearer Code $\mathcal{C}$ der Länge n heißt zyklisch, wenn jede zyklische Verschiebung eines Codeworts c auch zum Code gehört. D.h., ist $c = (c_0, \ldots, c_{n-1}) \in \mathcal{C}$, so auch $c' = (c_{n-1}, c_0, \ldots, c_{n-2}) \in \mathcal{C}$.

Um zyklische Codes zu beschreiben, erweist es als sich zweckmäßig statt Vektorschreibweise eine alternative Darstellung der Codewörter zu wählen, die *polynomiale Repräsentation*. Dazu wird einem Codevektor $c = (c_0, \ldots, c_{n-1})$ das *formale Polynom* $c(X) = c_0 + c_1 X + \cdots + c_{n-1} X^{n-1} \in GF(q)[X]$ in der Unbestimmten[9] X zugeordnet. Multiplikation von $c(X)$ mit X ergibt

$$Xc(X) = c_0 X + c_1 X^2 + \cdots + c_{n-1} X^n$$

und anschließende Division durch $X^n - 1$ liefert den Rest

$$c_{n-1} + c_0 X + c_1 X^2 + \cdots + c_{n-2} X^{n-1} \ ,$$

der dem zyklisch verschobenen Vektor $c' = (c_{n-1}, c_0, \ldots, c_{n-2})$ entspricht. Die Multiplikation mit anschließender Restbildung entspricht der Multiplikation in der Restklassenalgebra[10] $R_n := GF(q)[X]/I(X^n - 1)$, wobei $I(X^n - 1)$ das von

[9] Die Notation $GF(q)[X]$ bezeichnet die Ringadjunktion einer Unbestimmten X. GF(q)[X] ist der Ring der formalen Polynome mit Koeffizienten aus $GF(q)$. Multiplikation und Addition auf $GF(q)[X]$ seien standardmäßig wie bei [Reiffen et al. 1984] erklärt.

[10] Die Begriffe Algebra, Ring, Restklasse, Ideal, Grad, Ordnung, normiert, zyklisch usw., sind die Standardbezeichnungen für Objekte der Algebra. Vergleiche dazu z.B. [Reiffen et al. 1984]. Multiplikation, Addition und Skalarmultiplikation der Polynomrestklassenalgebra seien wie üblich erklärt.

$X^n - 1$ erzeugte Ideal

$$I(X^n - 1) := \{(X^n - 1) \cdot f \mid f \in GF(q)[X]\}$$

bezeichnet. In jeder Restklasse $f + I(X^n - 1) \in R_n$ befindet sich aufgrund des Divisionstheorems[11] ein eindeutig bestimmtes Polynom vom Grad $< n$. Es wird der *kanonische Repräsentant* der Restklasse $f + I(X^n - 1)$ genannt, so dass durch

$$R_n = \{f + I(X^n - 1) | \operatorname{grad} f \leq n - 1\} .$$

der Polynomrestklassenring dargestellt werden kann. Mit dieser Darstellung von R_n ist die Abbildung $GF(q)^n \longrightarrow R_n$ definiert durch

$$(c_0, \ldots, c_{n-1}) \mapsto c_0 + c_1 X + \cdots + c_{n-1} X^{n-1} + I(X^n - 1)$$

ein $GF(q)$-Isomorphismus (die *polynomiale Repräsentation*). Jeder Vektor von $GF(q)^n$ und damit auch jedes Codewort $c \in \mathcal{C}$ kann damit eineindeutig mit dem kanonischen Repräsentanten einer Restklasse aus R_n identifiziert werden.

Es sei $c_0 + c_1 X + \ldots + c_{n-1} X^{n-1} + I(X^n - 1)$ eine Restklasse von R_n, die ein Codewort eines beliebigen zyklischen Codes $\mathcal{C}$ repräsentiert. Ihre Multiplikation mit $X + I(X^n - 1)$ ergibt, wie schon beschrieben, die Restklasse $c_{n-1} + c_0 X + c_1 X^2 + \ldots + c_{n-2} X^{n-1} + I(X^n - 1)$, welche ebenfalls ein Codewort aus $\mathcal{C}$ repräsentiert. Es ist daher einsichtig, dass an einen zyklischen Code die Abgeschlossenheit unter der Multiplikation mit sich nicht im Code befindlichen Restklassen von R_n gefordert werden muss. Die Aussage des folgenden Lemmas konkretisiert dieses.

Lemma 1.18
Ein linearer Code $\mathcal{C}$ der Länge n über $GF(q)$ ist genau dann zyklisch, wenn $\mathcal{C}$ vermöge seiner polynomialen Repräsentation ein Ideal im Restklassenring R_n bildet.

Beweis. [Betten et al. 1998], (S. 78) ■

Wegen Lemma 1.18 sollte man die Idealstruktur auf R_n kennen. Alle Ideale $I(g(X)) := I(g)$ von $GF(q)[X]$ sind *Hauptideale*, da $GF(q)$ ein Körper ist (vgl.

[11] Eindeutigkeit der Division mit Rest für Polynome, siehe z.B. [Fischer et al. 1986], S. 222.

[Reiffen et al. 1984], S. 156), d.h. sie können von einem einzigen Element $g(X) \in GF(q)[X]$ erzeugt werden. Ein solches erzeugendes Element $g(X)$ heißt Erzeuger und hat minimalen Grad in $I(g)$. Unter allen Erzeugern von $I(g)$ gibt es genau ein normiertes Polynom; in der Notation von $I(g)$ soll $g(X)$, sofern nicht anders angegeben, genau diesen Erzeuger bezeichnen.

Ein Ideal $I(g) \subseteq GF(q)[X]$ enthält ein Ideal $I(h)$ genau dann, wenn das Polynom $g(X)$ das Polynom $h(X)$ teilt. Es seien daher

$$\begin{aligned} I(g) &:= \{f(X) \cdot g(X) \in GF(q)[X] \mid g(X) \text{ teilt } X^n - 1\} , \\ J &:= \{I(g) \mid g(X) \in GF(q)[X]\}, \end{aligned}$$

d.h. $I(g)$ ein beliebiges Ideal von $GF(q)[X]$, das $I(X^n - 1)$ enthält, und J die Menge aller derartigen Ideale von $GF(q)[X]$. Es sei außerdem

$$\tilde{J} := \{I(\tilde{r}) \mid I(\tilde{r}) := \{\tilde{r}(X) \cdot \tilde{s}(X) \mid \tilde{s}(X) \in R_n\} , \ \tilde{r}(X) \in R_n\}$$

die Menge aller Ideale von R_n. Die natürliche Projektion

$$\pi : GF(q)[X] \longrightarrow R_n \text{ mit } f(X) \mapsto f(X) + I(X^n - 1) := \tilde{f}(X) ,$$

die jedes Polynom aus $GF(q)[X]$ auf seine Restklasse Modulo $X^n - 1$ abbildet, induziert eine Bijektion[12] $\tilde{\pi} : J \longrightarrow \tilde{J}$ mittels der Zuordnung

$$I(g) \mapsto \tilde{\pi}(I(g)) := I(\pi(g)) .$$

Also sind alle Ideale von R_n der Form

$$I(\tilde{g}) = I(g)/I(X^n - 1) \subseteq R_n ,$$

wobei $g(X)$ das Polynom $X^n - 1$ teilt. Aus dem Gesagten folgt unmittelbar:

Folgerung 1.19
Ein linearer Code $\mathcal{C}$ der Länge n über $GF(q)$ ist genau dann zyklisch, wenn es einen normierten Teiler $g(X) \in GF(q)[X]$ von $X^n - 1$ gibt, der $\mathcal{C}$ als Ideal von R_n erzeugt, d.h. $\mathcal{C} = I(g)/I(X^n - 1)$.

[12] Einen Beweis dazu wird in Anhang A gegeben.

Bemerkung 1.20

(i). Ein Teiler $g(X)$ von $X^n - 1 \in GF(q)[X]$, der einen Code $\mathcal{C}$ als ein Ideal von R_n erzeugt, wird in der Sprache der Codierungstheorie das *Generatorpolynom* des zyklischen Codes $\mathcal{C}$ genannt.

(ii). Ist das Generatorpolynom $g(X) \in GF(q)[X]$ vom Grad $n-k$, erzeugt es durch Multiplikation mit allen Vektoren des Nachrichtenraums $u \in GF(q)^k$ mittels deren polynomialer Repräsentation[13] eine Teilmenge $C \subseteq GF(q)[X]$. Die Menge C ist kanonisch isomorph zum Code $\mathcal{C} = I(g)/I(X^n - 1) \subseteq R_n$, denn sie enthält genau die kanonischen Repräsentanten der Restklassen in $\mathcal{C}$.

(iii). Der Quotient $h(X) := X^n - 1/g(X) \in GF(q)[X]$ mit $\operatorname{grad} h(x) = k$ heißt *Kontrollpolynom* des zyklischen Codes $\mathcal{C}$, denn für ein beliebiges Codewort $g(X) \cdot u(X) = c(X) \in \mathcal{C}$ ist

$$h(X) \cdot c(X) = h(X) \cdot g(X) \cdot u(X) \equiv 0 \bmod X^n - 1 \ .$$

1.2. Polynome und endliche Körper

Wie man im Abschnitt 1.1.3 sehen konnte, ist die Kenntnis der Teiler von $X^n - 1$ über $GF(q)$ für die Konstruktion zyklischer Codes von Bedeutung.
Ein Polynom f vom Grad ≥ 1 heißt *irreduzibel* über einem endlichen Körper $GF(q)$, wenn es höchstens durch Ausklammern eines Zahlfaktors in ein Produkt von Polynomen zerlegbar ist. Die Zerlegung von Polynomen aus $GF(q)[X]$ in irreduzible Faktoren kann für Polynome großen Grades sehr aufwendig werden. Es gibt zur Lösung dieser Aufgabe einige auf dem *Berlekamp-Faktorisierungs-Algorithmus* (z.B. [Geddes et al. 1992], S. 347f) basierende Algorithmen, die in Algebrasystemen wie *MATHEMATICA*[14] oder *MuPAD*[15] implementiert sind. *MuPAD* verwendet z.B. den *Cantor-Zassenhaus-Algorithmus.*

[13] Die polynomiale Repräsentation von u wird hier als $u(X) \in GF(q)[X]$ mit $\operatorname{grad} u(X) \leq k-1$ aufgefasst.
[14] http://www.wolfram-research.com/
[15] http://www.mupad.de/

Polynome $f(X) \in GF(q)[X]$, die über $GF(q)$ in ein Produkt irreduzibler Polynome vom Grad ≥ 1 zerlegbar sind, zerfallen über bestimmten *Erweiterungskörpern* $\mathbb{K}$, die $GF(q)$ enthalten, vollständig in ein Produkt von *Linearfaktoren*[16]. Ist $(X - \beta)$ ein Linearfaktor von $f(X)$ über $\mathbb{K}$, so nennt man $\beta \in \mathbb{K}$ eine *Nullstelle* von $f(X)$.
Die Nullstellen des Polynoms $X^n - 1$ über einem Erweiterungskörper von $GF(q)$ spielen, wie im folgenden noch dargestellt wird, eine wichtige Rolle für BCH-Codes. In diesem Abschnitt seien wieder n eine positive ganze Zahl und $q = p^r$ eine Primzahlpotenz.

1.2.1. Die multiplikative Ordnung von q mod n

In diesem Abschnitt bezeichnen p und $\mathfrak{p}$ Primzahlen. In der Notation wird p für eine Primzahl verwendet, die die Zahl q teilt, und $\mathfrak{p}$ für eine Primzahl, die ein Teiler der Zahl n ist.

Satz 1.21
Das Polynom $X^{q^m} - X \in GF(q)[X]$ ist das Produkt aller normierten, über $GF(q)$ irreduziblen Polynome, deren Grad m teilt.

Beweis. [Betten et al. 1998], (S. 18) ∎

Lemma 1.22
Es seien $X^s - 1, X^r - 1 \in GF(q)[X]$ und r, s positive ganze Zahlen. Es gilt[17]*:*
$X^s - 1 | X^r - 1$ genau dann, wenn $s | r$.

Beweis. Es gelte $X^s - 1 | X^r - 1$ mit $s \neq r$, also $X^r - 1 = (X^s - 1) \cdot f(X)$ mit $f(X) \in GF(q)[X]$, wobei $\operatorname{grad} f := l \geq 1$ (der Fall $s = r$ ist trivial). Division mit Rest liefert $r = a \cdot s + b$ mit $a \in \mathbb{N}$ und $0 \leq b < s$. Dann gilt sicherlich $s \cdot l = as + b \Leftrightarrow s(l - a) = b$. Dies ist aber ein Widerspruch zu $0 \leq b < s$, wenn $b \neq 0$, also folgt $r = s \cdot l = s \cdot a$.
Andererseits teile s die Zahl r. Dann gibt es ein $a \in \mathbb{N}$ mit $r = s \cdot a$, so dass $X^r - 1 = X^{sa} - 1 = (X^s - 1) \cdot (X^{s(a-1)} + X^{s(a-2)} + \ldots + X^s + 1)$. Also teilt in diesem Fall das Polynom $X^s - 1$ das Polynom $X^r - 1$. ∎

[16] Linearfaktoren sind Polynome $P(X)$ mit $\operatorname{grad} P(X) = 1$.
[17] '|' bedeutet 'teilt'.

Wegen Lemma 1.22 teilt also $X^n - 1$ das Polynom $X^{q^m-1} - 1$ genau dann, wenn n die Zahl $q^m - 1$ teilt. Teilt n die Zahl $q^m - 1$, so zerfällt $X^n - 1$ wegen Satz 1.21 ebenfalls in normierte, irreduzible Polynome vom Grad $\leq m$ über $GF(q)$.

Definition 1.23
Für[18] $\mathrm{ggT}(q, n) = 1$ ist die *multiplikative Ordnung von* q mod n die kleinste Zahl $m_n(q) := m$, für die n die Zahl $q^m - 1$ teilt.

Bemerkung 1.24
Die Eulersche φ-Funktion ordnet jeder positiven ganzen Zahl n die Anzahl der zu ihr teilerfremden[19] Zahlen zu, die kleiner als sie selbst sind. Wegen des Satzes von *Euler-Fermat* gilt (z.B. [Remmert et al. 1995], S. 186):

$$a^{\varphi(n)} \equiv 1 \bmod n \text{ , für alle } 0 < a \in \mathbb{N} \text{ mit } \mathrm{ggT}(a, n) = 1 \text{ .}$$

Die multiplikative Ordnung $m_n(q)$ ist daher in Definition 1.23 wegen der φ-Funktion wohldefiniert, d.h. sie existiert und es gilt $m_n(q) > 0$, wenn die Körperordnung von $GF(q)$ teilerfremd zum Grad n des Polynoms $X^n - 1$ ist.
Man kann die Zahl q mit einer Restklasse der multiplikativen Restklassengruppe $((\mathbb{Z}/n\mathbb{Z})^*, \cdot\,)$, d.h. als Element der Einheitengruppe des Rings $(\mathbb{Z}/n\mathbb{Z}, +\,, \cdot\,)$ identifizieren. Dann ist aber $m_n(q)$ identisch mit der Elementordnung $\mathrm{ord}\, q$ von q in $(\mathbb{Z}/n\mathbb{Z})^*$, der Anzahl der Elemente der von q erzeugten zyklischen Untergruppe. Jede natürliche Zahl l mit $q^l \equiv 1 \bmod n$ ist daher ein Vielfaches von $\mathrm{ord}\, q = m_n(q)$. Aus diesem Grund ist auch $\varphi(n)$ ein Vielfaches von $m_n(q)$.

Der restliche Teil dieses Abschnitts wird sich mit der Kalkulation von $m_n(q)$ befassen. Die positive ganze Zahl $\varphi(n)$ lässt sich vergleichsweise schnell berechnen (vgl. Beweis von 1.28). Daher kann man sich den Zusammenhang $m_n(q) | \varphi(n)$ bei der Berechnung von $m_n(q)$ zu Nutze machen, was den Rechenaufwand gegenüber dem naiven Ausprobieren aller Zahlen $1 \leq l \leq n - 1$ in der Kongruenzgleichung $q^l - 1 \equiv 0 \bmod n$ erheblich vermindert. Man vergleiche dazu die Implementierung der Berechnung von $m_n(q)$ im Algebrasystem *MATHEMATICA*, dargestellt in Abschnitt 2.2.

[18] Die Bezeichnung $\mathrm{ggT}(a, b)$ meint den *größten gemeinsamen Teiler.* zweier Zahlen $a, b \in \mathbb{Z}$
[19] Zwei positive ganze Zahlen a und b heißen teilerfremd, wenn $\mathrm{ggT}(a, b) = 1$.

Für positives n und Primzahlpotenzen $q = p^r$ mit $1 < r \in \mathbb{Z}$ kann man die Kalkulation von $m_n(q)$ darüberhinaus vereinfachen, wie die folgenden Aussagen zeigen werden.

Lemma 1.25
Es sei $\mathrm{ggT}(n,p) = 1$ *und* p *eine Primzahl. Dann gilt:*

$$m_n(p^r) = \frac{m_n(p)}{\mathrm{ggT}(m_n(p), r)} \ ,$$

wobei $0 < r \in \mathbb{Z}$.

Beweis. Da $\mathrm{ggT}(n,p) = 1$ kann man annehmen, dass $p \in (\mathbb{Z}/n\mathbb{Z})^*$, wobei $((\mathbb{Z}/n\mathbb{Z})^*, \cdot\,)$ die Einheitengruppe des Restklassenrings $\mathbb{Z}/n\mathbb{Z}$ bezeichnet. Dann haben p und p^r die Elementordnungen $m_n(p)$ und $m_n(p^r)$ in $(\mathbb{Z}/n\mathbb{Z})^*$, d.h.

$$p^{m_n(p)} \equiv 1 \bmod n \quad \text{und} \quad (p^r)^{m_n(p^r)} \equiv 1 \bmod n \ .$$

Es ist leicht einzusehen, dass $m_n(p) | m_n(p^r) \cdot r$. Definiere $T := \mathrm{ggT}(m_n(p), r)$. Folglich gilt auch

$$m_n(p) | m_n(p^r) \cdot T \ . \tag{1.1}$$

Andererseits ist $(p^r)^{m_n(p)} = (p^{m_n(p)})^r \equiv 1 \bmod n$, weshalb $m_n(p^r) | m_n(p)$. Definiere $r \cdot \frac{1}{T} := T' \in \mathbb{N}$. Dann gilt sogar:

$$(p^r)^{m_n(p)\cdot\frac{1}{T}} = (p^{m_n(p)})^{r\cdot\frac{1}{T}} = (p^{m_n(p)})^{T'} \equiv 1 \bmod n \ ,$$

weswegen $m_n(p^r) | (m_n(p) \cdot \frac{1}{T}) \Leftrightarrow (m_n(p^r) \cdot T) | m_n(p)$. Mit (1.1) folgt dann abschließend die Behauptung. ∎

Satz 1.26
Sei $n = \mathfrak{p}_1^{l_1} \cdot \ldots \cdot \mathfrak{p}_h^{l_h}$, $1 \leq h, l_1, \ldots, l_h \in \mathbb{Z}$, *die Primfaktorzerlegung von* n *und* $q = p^r$ *eine Primzahlpotenz für eine positive ganze Zahl* r *mit* $\mathfrak{p}_i \neq p$ *für alle* $1 \leq i \leq h$. *Dann gilt*[20]

$$m_n(q) = \mathrm{kgV}\left(\frac{m_{\mathfrak{p}_1^{l_1}}(p)}{\mathrm{ggT}(m_{\mathfrak{p}_1^{l_1}}(p), r)}, \ldots, \frac{m_{\mathfrak{p}_h^{l_h}}(p)}{\mathrm{ggT}(m_{\mathfrak{p}_h^{l_h}}(p), r)} \right) \ .$$

[20] $\mathrm{kgV}(a,b)$ meint das kleinste gemeinsame Vielfache zweier ganzer Zahlen a und b.

Beweis. Da $n = \mathfrak{p}_1^{l_1} \cdot \ldots \cdot \mathfrak{p}_h^{l_h}$ die Primfaktorzerlegung von n ist, sind $\mathfrak{p}_i$ und $\mathfrak{p}_j$ paarweise teilerfremd für alle $i \neq j$. Man betrachte die multiplikativen Gruppen $(\mathbb{Z}/n\mathbb{Z})^*$ und $\left(\mathbb{Z}/\mathfrak{p}_i^{l_i}\mathbb{Z}\right)^*$ der Restklassenringe $(\mathbb{Z}/n\mathbb{Z}, \cdot\,, +\,)$ und $\left(\mathbb{Z}/\mathfrak{p}_i^{l_i}\mathbb{Z}, \cdot\,, +\,\right)$ für alle $1 \leq i \leq h$. Dann gibt es eine natürliche Gruppenisomorphie (vgl. [Reiffen et al. 1984], S. 121)

$$(\mathbb{Z}/n\mathbb{Z})^* \longrightarrow \left(\mathbb{Z}/\mathfrak{p}_1^{l_1}\mathbb{Z}\right)^* \times \ldots \times \left(\mathbb{Z}/\mathfrak{p}_h^{l_h}\mathbb{Z}\right)^*$$

mit $a \bmod n \mapsto (a \bmod \mathfrak{p}_1^{l_1}, \ldots, a \bmod \mathfrak{p}_h^{l_h})$ zwischen $(\mathbb{Z}/n\mathbb{Z})^*$ und dem *direkten Produkt* der Gruppen $\left(\mathbb{Z}/\mathfrak{p}_i^{l_i}\mathbb{Z}\right)^*$ mit komponentenweiser Multiplikation. Es gilt:

$$1 \bmod n \mapsto (1 \bmod \mathfrak{p}_1^{l_1}, \ldots, 1 \bmod \mathfrak{p}_h^{l_h}) \ .$$

Da $\mathrm{ggT}(q, \mathfrak{p}_i) = 1$, ist $q \in (\mathbb{Z}/n\mathbb{Z})^*$ und $q \in \left(\mathbb{Z}/\mathfrak{p}_i^{l_i}\mathbb{Z}\right)^*$ für alle $1 \leq i \leq h$. Definitionsgemäß sind aber $q^{m_n(q)} \equiv 1 \bmod n$ und für $m_i := m_{\mathfrak{p}_i^{l_i}}(q)$ die Kongruenzen $q^{m_i} \equiv 1 \bmod \mathfrak{p}_i^{l_i}$ gegeben. Die multiplikativen Ordnungen $m_n(q)$ und $m_{\mathfrak{p}_i^{l_i}}(q)$ sind minimal mit dieser Eigenschaft, weshalb die Identität

$$m_n(q) = \mathrm{kgV}\left\{m_{\mathfrak{p}_1^{l_1}}(q), \ldots, m_{\mathfrak{p}_h^{l_h}}(q)\right\} \ .$$

gilt. Der Rest des Beweises folgt aus Lemma 1.25. ∎

Für die Anwendung sind vor allem BCH-Codes über Körper der Charakteristik[21] „2“, von Bedeutung [Peterson et al. 1972], (S. 272). Deshalb stellt der folgende wird nun eine Berechnungshilfe für die multiplikative Ordnung von Körperordnungen der Charakteristik „2“ zu vorgegebenen n entwickelt.

Definition 1.27
Eine ungerade Primzahl $\mathfrak{p}$ wird *Wieferiche Primzahl* genannt, wenn

$$2^{\mathfrak{p}-1} \equiv 1 \bmod \mathfrak{p}^2 \ .$$

Crandall et al. zeigen in ihrem Artikel [Crandall et al. 1997], dass es nur zwei *Wieferiche Primzahlen* $\mathfrak{p} < 4 \cdot 10^{12}$ gibt, nämlich $\mathfrak{p} = 1093$ und $\mathfrak{p} = 3511$. Es wird jedoch vermutet, dass es unendlich viele Wieferiche Primzahlen gibt.

[21] Die Charakteristik eines endlichen Körpers ist die Anzahl der Summationen des Einselements, die nötig sind, um $\sum 1 = 0$ zu erreichen.

Satz 1.28
Es sei $n = \mathfrak{p}^l$, $2 \leq l \in \mathbb{Z}$, *eine Primzahlpotenz mit* $\mathrm{ggT}(\mathfrak{p}, 2) = 1$ *und* $\mathfrak{p}$ *keine* Wiefericheе Primzahl. *Dann ist*

$$m_{\mathfrak{p}^l}(2) = m_{\mathfrak{p}}(2) \cdot \mathfrak{p}^{l-1} \ .$$

Beweis. Es sei $\mathfrak{p}$ eine beliebige Primzahl. Definiere $m_l := m_{\mathfrak{p}^l}(2)$. Definitionsgemäß gilt:

$$2^{m_l} \equiv 1 \bmod \mathfrak{p}^l \ \forall \ 1 \leq l \tag{1.2}$$

und damit $\mathfrak{p}^l | 2^{m_l} - 1$. Wegen $\mathfrak{p}^{l-1} | \mathfrak{p}^l$ gilt dann $\mathfrak{p}^{l-1} | 2^{m_l} - 1$, also
$2^{m_l} \equiv 1 \bmod p^{l-1}$, weshalb nach Bemerkung 1.24

$$m_{l-1} | m_l \ \forall \ 2 \leq l. \tag{1.3}$$

Für jede Primzahl $\mathfrak{p}$ gilt die Identität $\varphi(\mathfrak{p}) = \mathfrak{p} - 1$. Eine Primzahl ist also eine Wiefericheе, wenn $2^{\varphi(\mathfrak{p})} \equiv 1 \bmod \mathfrak{p}^2$. Außerdem ist nach den Überlegungen von Bemerkung 1.24 $\varphi(\mathfrak{p}) = m_1 \cdot b$, wobei $1 \leq b < \mathfrak{p}$; wäre $b \geq \mathfrak{p}$ würde folgen, dass $\varphi(\mathfrak{p}) \geq \mathfrak{p}$, da $m_1 > 0$. Dies wäre ein Widerspruch zur Definition von $\varphi(n)$. Deshalb gilt:

$$2^{\varphi(\mathfrak{p})} - 1 = 2^{m_1 \cdot b} - 1 = (2^{m_1} - 1) \cdot \sum_{i=1}^{b} 2^{m_1(b-i)} \ ,$$

mit $\sum_{i=1}^{b} 2^{m_1(b-i)} \equiv b \bmod \mathfrak{p}$, wegen (1.2). Das bedeutet aber, dass $\mathfrak{p} \nmid \sum_{i=1}^{b} 2^{m_1(b-i)}$, denn $b < \mathfrak{p}$. Folglich teilt $\mathfrak{p}^2 | (2^{m_1} - 1)$ und mithin: $m_2 | m_1$. Für Wiefericheе Primzahlen folgt letztendlich mit (1.3): $m_2 = m_1$.

Für die *Eulersche* φ-Funktion gilt ([Remmert et al. 1995], S.85):

$\varphi(a) = a \prod_{\mathfrak{p}|a} \frac{\mathfrak{p}-1}{\mathfrak{p}}$, wobei $\mathfrak{p}$ alle Primzahlen durchläuft, die $0 < a \in \mathbb{Z}$ teilen.

Für alle Primzahlpotenzen $\mathfrak{p}^{l-1}$ ist demzufolge die Zuordnung durch die φ-Funktion identisch mit $\varphi(\mathfrak{p}^{l-1}) = \varphi(\mathfrak{p}) \cdot \mathfrak{p}^{l-2}$. Aus diesem Grund gilt die Gleichheit

$$2^{\varphi(\mathfrak{p}^{l-1})} - 1 = 2^{\varphi(\mathfrak{p})\mathfrak{p}^{l-2}} - 1 = (2^{\varphi(\mathfrak{p})} - 1) \cdot \sum_{i=1}^{\mathfrak{p}^{l-2}} 2^{\varphi(\mathfrak{p})(\mathfrak{p}^{l-2}-i)} \tag{1.4}$$

mit $2^{\varphi(\mathfrak{p})(\mathfrak{p}^{l-2}-i)} \equiv 1 \bmod \mathfrak{p} \Leftrightarrow 2^{\varphi(\mathfrak{p})(\mathfrak{p}^{l-2}-i)} = \mathfrak{p} \cdot a_i + 1$ für alle $1 \leq i \leq \mathfrak{p}^{l-2}$ und $0 < a_i \in \mathbb{Z}$. Es gibt also eine Darstellung der Summe durch

$$\sum_{i=1}^{\mathfrak{p}^{l-2}} 2^{\varphi(\mathfrak{p})(\mathfrak{p}^{l-2}-i)} = \sum_{i=1}^{\mathfrak{p}^{l-2}} (\mathfrak{p} \cdot a_i + 1) = \sum_{i=1}^{\mathfrak{p}^{l-2}} \mathfrak{p} \cdot a_i + \mathfrak{p}^{l-2} \ ,$$

die zeigt, dass $\mathfrak{p}^{l-1} \nmid \sum_{i=1}^{\mathfrak{p}^{l-2}} 2^{\varphi(\mathfrak{p})(\mathfrak{p}^{l-2}-i)}$. Im folgenden sei $\mathfrak{p}$ eine nicht Wieferiche Primzahl, also $\mathfrak{p}^2 \nmid 2^{\varphi(\mathfrak{p})} - 1$. Dann gilt zusammengenommen wegen Gleichung (1.4) $\mathfrak{p}^l \nmid 2^{\varphi(\mathfrak{p}^{l-1})} - 1$ und schließlich $\mathfrak{p}^l \nmid 2^{m_{l-1}} - 1$. Mit dieser Tatsache und (1.3):

$$m_l > m_{l-1} \ \forall l \geq 2. \tag{1.5}$$

Man betrachte abschließend die Identität

$$2^{m_{l-1}\mathfrak{p}} - 1 = (2^{m_{l-1}} - 1) \cdot \sum_{i=1}^{\mathfrak{p}} 2^{m_1(\mathfrak{p}-i)} \ .$$

Weil $\sum_{i=1}^{\mathfrak{p}} 2^{m_1(\mathfrak{p}-i)} \equiv \sum_{i=1}^{\mathfrak{p}} (1 \bmod \mathfrak{p}) \equiv 0 \bmod \mathfrak{p}$, teilt $\mathfrak{p}$ diese Summe. Nach (1.2) gilt sowieso $\mathfrak{p}^{l-1} | (2^{m_{l-1}} - 1)$. Zusammengenommen teilt somit $\mathfrak{p}^l$ die Zahl $2^{m_{l-1}\mathfrak{p}} - 1$. Das aber zeigt, dass $m_l | (m_{l-1} \cdot \mathfrak{p})$. Wegen (1.3) und (1.5) ist es dann notwendig, dass $m_l = m_{l-1} \cdot \mathfrak{p}$ für alle $l \geq 2$, wenn $\mathfrak{p}$ keine Wieferiche Primzahl ist.
Induktion von $l-1$ nach l liefert dann die Behauptung: $m_l = m_{l-1} \cdot \mathfrak{p} = m_1 \cdot \mathfrak{p}^{l-1}$ für alle $2 \leq l$. ■

Das folgende Beispiel fasst die Ergebnisse dieses Abschnitts zusammen.

Beispiel 1.29
Betrachte die Zahlen $n = 23625 = 3^3 \cdot 5^2 \cdot 7$ und $q = 2^6$. Es sind

$$\begin{aligned} m_3(2) = 2 \quad &\Rightarrow \quad m_{3^3}(2) = 2 \cdot 3^2 \quad \text{und ggT}\,(6, m_3(2)) = 6 \ , \\ m_5(2) = 4 \quad &\Rightarrow \quad m_{5^3}(2) = 4 \cdot 5^2 \quad \text{und ggT}\,(6, m_5(2)) = 2 \ , \\ m_7(2) = 3 \quad & \qquad\qquad\quad\ \ \text{und ggT}\,(6, m_7(2)) = 3 \ . \end{aligned}$$

Zusammengenommen ist also $m_{23625}(64) = \text{kgV}\left\{\frac{18}{6}, \frac{100}{2}, \frac{3}{3}\right\} = 150$.

1.2.2. Zerfällungskörper und Minimalpolynome

Dieser Abschnitt befasst sich mit der tiefer gehenden Bedeutung der Zahl $m_n(q)$ für das Polynom $X^n - 1$ und mit der Konkretisierung von irreduziblen Polynomen über $GF(q)$, die durch Satz 1.21 bestimmt sind. Im folgenden gelte stets $ggT(n, q) = 1$.

Satz 1.30 (und Definition)
*Zu jeder Primzahl p und jeder ganzen Zahl $r \geq 1$ gibt es bis auf Isomorphie genau einen Körper mit p^r Elementen. Dieser wird in der Notation mit $GF(p^r)$ (*Galois Feld*) bezeichnet.*

Beweis. [Reiffen et al. 1984], (S. 202) ∎

Bemerkung 1.31
Ist $q = p^r$, $0 < m' \in \mathbb{Z}$, so sind die endlichen Körper $GF(q^{m'}) \cong GF(p^{m' \cdot r})$ im wesentlichen gleich.

Satz 1.32
Die Einheitengruppe $GF(q^{m'})^*$ ist zyklisch von Ordnung $q^{m'} - 1$, d.h. sie wird von einem einzigen Element $\alpha \in GF(q^{m'})^*$ erzeugt.

Beweis. [Reiffen et al. 1984], (S. 172) ∎

Folgerung 1.33
Jede Untergruppe $U \subseteq GF(q^{m'})^*$ ist zyklisch. $GF(q^{m'})^*$ besitzt genau dann eine Untergruppe von Ordnung n, wenn $n | q^{m'} - 1$.

Beweis. Angenommen $U \subseteq GF(q^{m'})^*$ ist Untergruppe und wird von zwei Elementen $\alpha^i \neq \alpha^j$ erzeugt, wobei α ein Erzeuger von $GF(q^{m'})^*$ ist und j, i minimal gewählt sind. Ohne Einschränkung der Allgemeinheit seien $0 < i < j < q^{m'} - 1$, denn wäre eine der beiden Zahlen, z.B. j, Null oder $q^{m'} - 1$, so würde folgen, dass $\{\alpha^{q^{m'}-1} = \alpha^0 = 1\} = U$ zyklisch ist. Division mit Rest liefert $j = a \cdot i + b$, wobei $a, b \in \mathbb{N}$ und $0 \leq b \leq i - 1$. Die Produkte der Potenzen in $l, l' \in \mathbb{N}$ $(\alpha^i)^l \cdot (\alpha^j)^{l'} = \alpha^{i(l+l'a)} \cdot \alpha^{l'b}$ erzeugen U, wobei durch α^b ein Widerspruch zur Minimalität von j entsteht, wenn $b \neq 0$. Also ist auch α^j eine Potenz von α^i und deshalb jede Untergruppe U von $GF(q^{m'})^*$ zyklisch.
Es sei nun U eine Untergruppe von $(GF(q^{m'}))^*$ mit $\operatorname{ord} U = n$. Nach dem *Satz*

von Lagrange (z.B. [Reiffen et al. 1984]) teilt die Ordnung jeder Untergruppe die Gruppenordnung, folglich $n|q^{m'}-1$. Andererseits gelte $n|q^{m'}-1 \Leftrightarrow q^{m'}-1 = i \cdot n$ mit $1 \leq i \leq q^{m'}-1$. Dann gilt für jedes erzeugende Element α von $GF(q^{m'})^*$, dass $(\alpha^i)^n = 1$, und α^i erzeugt eine zyklische Untergruppe U von Ordnung n in $(GF(q^{m'}))^*$. ■

Lemma 1.34
Das Polynom $X^{q^{m'}} - X$ zerfällt über $GF(q^{m'})$ vollständig in Linearfaktoren und alle Nullstellen des Polynoms sind paarweise verschieden.

Beweis. Es sei $\beta \in GF(q^{m'})$. Ist $\beta = 0$, dann ist auch $\beta^{q^{m'}} - \beta = 0$. Für $\beta \neq 0$ gilt $\beta^{q^{m'}} = \beta$, weil β eine Einheit ist. Also sind alle Elemente des Körpers $GF(q^{m'})$ Nullstellen des Polynoms $X^{q^{m'}} - X$. Die Körperelemente sind paarweise verschieden und da $X^{q^{m'}} - X$ höchstens $q^{m'}$ Nullstellen besitzen kann, ist die Aussage bewiesen. ■

Definition 1.35
Der kleinste endliche Körper bzgl. der Anzahl seiner Elemente über dem ein Polynom $X^n - 1 \in GF(q)[X]$ vollständig in Linearfaktoren zerfällt, wird *Zerfällungskörper von $X^n - 1$* genannt.

Satz 1.36
Angenommen das Polynom $P(X)$ ist irreduzibel über $GF(q)$ und vom Grad m'. Dann ist der Polynomrestklassenring $GF(q)[X]/I(P(X))$ ein endlicher Körper der Ordnung $q^{m'}$.

Beweis. [MacWilliams et al. 1977], (S. 94) ■

Bemerkung 1.37
Nach dem vorangegangenen Abschnitt (1.2.1) existiert für jedes n mit $\mathrm{ggT}(q,n) = 1$ die multiplikative Ordnung $m := m_n(q)$, so dass

$$X^n - 1 | X^{q^m - 1} - 1 \ .$$

Zusammengenommen mit Lemma 1.34 besitzt $X^n - 1$ daher ebenfalls nur paarweise verschiedene Nullstellen in $GF(q^m)$ und durch die Minimalität der multiplikativen Ordnung m ist $GF(q^m)$ der Zerfällungskörper von $X^n - 1$.
Wegen der Sätze 1.30 und 1.36 ist $GF(q^m) \cong GF(q)[X]/I(P(X))$. Die Nullstellen

$\beta \in (GF(q^m))^*$ von $X^n - 1$ lassen sich als (Spalten-)m-Tupel mit Einträgen in $GF(q)$ auffassen, wobei die Einträge den Koeffizienten kanonischer Restklassenrepräsentanten von $GF(q)[X]/I(P(X))$ entsprechen.
Da für jede Nullstelle β von $X^n - 1$ die Identität $\beta^n = 1$ gilt, nennt man β eine *n-te Einheitswurzel*. Die *Menge der n-ten Einheitswurzeln* E_n bildet wegen Folgerung 1.33 eine zyklische Untergruppe von $GF(q^m)^*$. Es gibt also immer eine n-te Einheitswurzel α, die E_n in $(GF(q^m))^*$ erzeugt. So ein Element $\alpha \in E_n$ wird *primitive n-te Einheitswurzel über* $GF(q)$ genannt.

Definition 1.38
Das *Minimalpolynom* $M(X) \in GF(q)[X]$ eines Elements $\beta \in E_n \subseteq GF(q^m)$ ist das eindeutig bestimmte ([Reiffen et al. 1984], S. 189) normierte Polynom kleinsten Grades, so dass[22]

$$M(\beta) = 0 \; .$$

über $GF(q^m)$. Ist $\alpha \in E_n$ primitiv, dann wird das Minimalpolynom einer n-ten Einheitswurzel $\alpha^i \in E_n$ im folgenden mit $M^{(i)}(X)$ bezeichnet.

Satz 1.39
Für Minimalpolynome $M^{(i)}(X)$ *von n-ten Einheitswurzeln* α^i *über* $GF(q)$ *gilt:*

(i). $M^{(i)}(X)$*ist irreduzibel über* $GF(q)$.

(ii). *Für* $f(X) \in GF(q)[X]$ *und* $f(\alpha^i) = 0$ *folgt* $M^{(i)}(X) | f(X)$.

(iii). $M^{(i)}(X) | X^{q^m} - X$, $m = m_n(q)$.

(iv). $\operatorname{grad} M^{(i)}(X) | m$.

Beweis. Angenommen $M^{(i)}(X) = M_1^{(i)}(X) M_2^{(i)}(X)$ mit $\operatorname{grad} M_1^{(i)}(X)$ und $\operatorname{grad} M_2^{(i)}(X) > 0$. Wegen $M^{(i)}(\alpha^i) = M_1^{(i)}(\alpha^i) M_2^{(i)}(\alpha^i) = 0$ folgt, dass entweder $M_1^{(i)}(\alpha^i)$ oder $M_2^{(i)}(\alpha^i) = 0$. Dies widerspricht jedoch der Definition $M^{(i)}(X)$, wonach das Minimalpolynom den kleinsten Grad unter den Polynomen mit dieser Eigenschaft hat. Also ist (i) bewiesen.
Division mit Rest liefert $f(X) = M^{(i)}(X) \cdot a(X) + r(X)$, wobei $a(X), r(X) \in$

[22] Es ist zu beachten, dass hier das Polynom $M(X)$ als Abbildung $M : GF(q^m) \to GF(q^m)$ mit $X \mapsto M(X)$ aufgefasst wird. Da aus dem jeweiligen Zusammenhang stets klar wird, ob ein Polynom die Abbildung oder das formale Element eines Polynomrings meint, wird dies im folgenden in der Notation nicht unterschieden.

$GF(q)[X]$ und $\operatorname{grad} r(X) < \operatorname{grad} M^{(i)}(X)$. Mit $f(\alpha^i) = M^{(i)}(\alpha^i) \cdot a(\alpha^i) + r(\alpha^i) = 0$ folgt, dass auch $r(\alpha^i) = 0$. Anderenfalls würde aus $r(X) \neq 0$ ein Widerspruch zur Minimalität des Grades von $M^{(i)}(X)$ folgen. Deshalb gilt (ii).
Die n-te Einheitswurzel $\alpha^i \in GF(q^m)$ ist eine Nullstelle von $X^{q^m} - X$ nach Lemma 1.34. Deshalb folgt (iii) aus (ii).
Die Aussage (iv) folgt unmittelbar aus (i), (iii) und Satz 1.21. ■

Bemerkung 1.40
Die Minimalpolynome über $GF(q)$ von Elementen $\beta \in GF(q^m)$ sind also genau die irreduziblen, normierten Polynome von Satz 1.21.
Ist $\alpha \in E_n$ eine primitive n-te Einheitswurzel und $M^{(1)}(X)$ das Minimalpolynom von α, dann erzeugt α die zyklische Gruppe E_n vermöge $M^{(1)}(\alpha) = 0$. Erzeugt α die ganze multiplikative Gruppe $(GF(q^m))^*$, dann wird α *primitives Element* von $(GF(q^m))^*$ genannt und das Minimalpolynom von α *primitives Minimalpolynom.*

Beispiel 1.41
$(GF(2^3))^*$ kann durch die primitiven Nullstellen der normierten, irreduziblen Polynome $X^3 + X + 1$ und $X^3 + X^2 + 1$ über $GF(2)$ erzeugt werden:

$GF(2^3)$ *definiert durch* $\alpha^3 + \alpha + 1 = 0$		$GF(2^3)$ *definiert durch* $\alpha^3 + \alpha^2 + 1 = 0$	
3-Tupel	Potenzen in α	3-Tupel	Potenzen in α
000	$0 := \alpha^{-\infty}$	000	$0 := \alpha^{-\infty}$
001	α^0	001	α^0
010	α	010	α
100	α^2	100	α^2
011	$\alpha^3 = \alpha + 1$	101	$\alpha^3 = \alpha^2 + 1$
110	$\alpha^4 = \alpha^2 + \alpha$	111	$\alpha^4 = \alpha^2 + \alpha + 1$
111	$\alpha^5 = \alpha^2 + \alpha + 1$	011	$\alpha^5 = \alpha + 1$
101	$\alpha^6 = \alpha^2 + 1$	110	$\alpha^6 = \alpha^2 + \alpha$

Es ist bei der Wahl eines irreduziblen Polynoms Vorsicht geboten, denn nicht jedes irreduzible, normierte Polynom von Grad m ist primitiv. Beispielsweise kann $(GF(2^4))^*$ durch die primitiven Nullstellen der Polynome $X^4 + X + 1$ und $X^4 + X^3 + 1$ erzeugt werden, aber nicht durch diejenigen des normierten, über $GF(2)$ irreduziblen Polynoms $X^4 + X^3 + X^2 + X + 1$. Die Nullstellen dieses Polynoms erzeugen lediglich die Untergruppe $E_5 \subseteq (GF(2^4))^*$ und daher ist das Polynom nicht primitiv.

Bemerkung 1.42
Die Abbildung $f : (\mathbb{Z}/n\mathbb{Z}, +) \to (E_n, \cdot\ )$ ist ein Isomorphismus von Gruppen vermöge der Zuordnung $i \mapsto \alpha^i$, wenn α eine primitive n-te Einheitswurzel bezeichnet.

Lemma 1.43
Es seien $\beta_1, \beta_2 \in E_n \subseteq GF(q^m)$ und $GF(q^m)$ von Charakteristik p. Dann gilt:

$$(\beta_1 + \beta_2)^p = \beta_1^p + \beta_2^p \ .$$

Beweis. Wegen des Binomialtheorems[23] (z.B. [Reiffen et al. 1984]) ist

$$(\beta_1 + \beta_2)^p = \sum_{j=0}^{p} \binom{p}{j} \beta_1^{p-j} \beta_2^j \ .$$

Da $\binom{p}{l} = \frac{p \cdot (p-1) \cdot \ldots \cdot (p-l+1)}{1 \cdot 2 \cdot \ldots \cdot l} \equiv 0 \bmod p$ für $1 < l < p$ und $\binom{p}{0} = \binom{p}{p} \equiv 1 \bmod p$ und p die Charakteristik von $GF(q^m)$ ist, folgt die Behauptung. ∎

Folgerung 1.44
Sei $\alpha \in E_n$ primitive n-te Einheitswurzel und $M^{(i)}(X)$ das Minimalpolynom von α^i über $GF(q)$. Dann gilt: $M^{(i)}(\alpha^j) = 0 \Leftrightarrow iq^r \equiv j \bmod n$ für ein $r \geq 0$.

Beweis. Sei $m_i := \operatorname{grad} M^{(i)}(X) \leq m_n(q)$. Wegen den Sätzen 1.21 und 1.39 gilt: $M^{(i)}(X) | X^{q^{m_i}} - X$. Setze $\beta := \alpha^i$, dann ist $\beta \in (GF(q^{m_i}))^*$ und $\operatorname{ord} \beta | q^{m_i} - 1$, die Ordnung der Einheitengruppe von $GF(q^m)$.
Angenommen es existiert ein $0 < l < m_i$, so dass $\operatorname{ord} \beta | q^l - 1$. Dann wäre β eine Nullstelle von $X^{q^l} - X$, aber $M^{(i)}(X) \nmid X^{q^l} - X$, weil $m_i > l$. Das wäre ein Widerspruch zu Satz 1.39. Also ist m_i minimal mit der Eigenschaft

$$\operatorname{ord} \beta | q^{m_i} - 1 \ . \tag{1.6}$$

Wegen Lemma 1.43 sind $\beta, \beta^q, \ldots, \beta^{q^{m_i - 1}}$ Nullstellen von $M^{(i)}(X)$. Die m_i n-ten Einheitswurzeln sind paarweise verschieden. Angenommen sie seien es nicht und es sei $\beta^{q^r} = \beta^{q^{r'}}$ mit $1 \leq r < r' \leq m_{i-1}$. Dann gilt:

$$\begin{aligned} \beta &= \beta^{q^{m_i}} = (\beta^{q^{r'}})^{m_i - r'} = (\beta^{q^r})^{q^{m_i - r'}} = \beta^{q^{m_i - r' + r}}; \\ 1 &= \beta^{q^{m_i - r' + r} - 1}, \end{aligned}$$

[23] $(a+b)^n = \sum_{j=0}^{n} \binom{n}{j} a^{n-j} b^j$ für a, b Elemente eines kommutativen Rings und $n \in \mathbb{N}$.

weshalb ord $\beta | q^{m_i - r' + r} - 1$. Da $m_i - r' + r < m_i$ ist dies ein Widerspruch zu (1.6). Also sind $\beta, \beta^q, \ldots, \beta^{q^{m_i - 1}}$ paarweise verschieden. Wegen grad $M^{(i)}(X) = m_i$ kann $M^{(i)}(X)$ höchstens m_i Nullstellen besitzen und da $\beta^{q^r} = \alpha^{iq^r} = \alpha^j$ genau dann, wenn $iq^r \equiv j \bmod n$, ist die Behauptung bewiesen. ∎

Lemma 1.45 (und Definition)
Sei $\alpha \in GF(q^m)$ primitive n-te Einheitswurzel. Die Relation auf den Exponenten der $\alpha^i, \alpha^j \in E_n$

$$i \sim j \iff M^{(i)}(\alpha^j) = 0$$

ist eine Äquivalenzrelation und ihre Äquivalenzklassen werden zyklotomische Nebenklassen von $q \bmod n$ *genannt. (Im folgenden kurz:* ZNK, *wenn eindeutig ist, dass von $q \bmod n$ gemeint ist.)*
Jede ZNK kann durch jedes Elemente s, das sie enthält, eindeutig repräsentiert werden. Daher sei im Folgenden C_s die ZNK, die das Element s enthält. Der kanonische Repräsentant ist das kleinste Element $s' \in C_{s'}$ mit $0 \leq s' < n$.

Beweis. Die Reflexivität $i \sim i$ ist unmittelbar klar. Wegen Folgerung 1.44 ist die Symmetrie „$i \sim j \Rightarrow j \sim i$“ gegeben durch $i \cdot q^r \equiv j \bmod n \Leftrightarrow j \cdot q^{m_n(q) - r} \equiv i \bmod n$, da q^r Einheit in $\mathbb{Z}/n\mathbb{Z}$ ist. Seien nun $i \sim j$ und $j \sim j'$, d.h. $iq^r \equiv j \bmod n$ und $jq^{r'} \equiv j' \bmod n$, $0 \leq r, r' < m_n(q)$, dann folgt wegen Lemma 1.43 aus

$$M^{(i)}(\alpha^{j'}) = M^{(i)}(\alpha^{jq^{r'}}) = (M^{(i)}(\alpha^i))^{q^{r'} q^r} = 0$$

die Transitivität: $i \sim j'$. ∎

Bemerkung 1.46
Die ZNK zur n-ten Einheitswurzel α^s, α primitiv, ist demnach der Form

$$C_s = \{s \bmod n, sq \bmod n, sq^2 \bmod n, \ldots, sq^{m_s - 1} \bmod n\} ,$$

wobei $m_s | m_n(q)$ (vgl. Satz 1.39), und enthält alle Exponenten der Nullstellen von $M^{(s)}(X)$. Ihre Elemente nennt man zueinander *konjugiert*. Die ZNK von sind durch Bemerkung 1.42 und Folgerung 1.44 unabhängig von primitiven n-ten Einheitswurzeln und deren zugehörigen Minimalpolynomen. D.h. sie sind vollständig durch die Struktur bestimmt, die q auf $\mathbb{Z}/n\mathbb{Z}$ erzeugt. In Anhang B findet man einen in *MATHEMATICA* implementierten Algorithmus zur Bestimmung der ZNK von $q \bmod n$.

Der abschließende Satz dieses Abschnitts fasst die genannten Ergebnisse zusammen:

Satz 1.47
Sei $\alpha \in E_n \subseteq GF(q^m)$ primitiv und C_s die ZNK des kanonischen Repräsentanten s. Dann gilt:

$$M^{(s)}(X) = \prod_{j \in C_s} (X - \alpha^j)$$

und

$$X^n - 1 = \prod_s M^{(s)}(X) \ ,$$

wobei $0 \leq s \leq n-1$ die kanonischen Repräsentanten der ZNK durchläuft.

Beweis. [MacWilliams et al. 1977], (S. 105) ■

1.3. BCH-Codes

In der jüngeren Literatur werden *mehrfach-Wurzel* von *einfach-Wurzel zyklischen Codes* über $GF(q)$ unterschieden. Hat $GF(q)$ die Charakteristik p und ist die Codelänge identisch mit $n = p^l \cdot \rho$ für $l, \rho \in \mathbb{N}$ mit $\mathrm{ggT}(p, \rho) = 1$, dann spricht man von mehrfach-Wurzel zyklischen Codes. Gilt für die Codelänge $\mathrm{ggT}(q, n) = 1$, dann ist der Code ein einfach-Wurzel zyklischer Code (vgl. *P. Charpin* in [Pless et al. 1998], S. 966f). Bei *mehrfach-Wurzel zyklischen Codes* ist das Polynom $X^n - 1$ darstellbar durch $(X^\rho - 1)^{p^l}$ und zerfällt wegen der in Abschnitt 1.2 geschilderten Zusammenhänge in nur ρ verschiedene Linearfaktoren mit Vielfachheit über seinem Zerfällungskörper.
BCH-Codes werden in der „klassischen“ Literatur als *einfach-Wurzel zyklische Codes* behandelt, wovon hier nicht abgewichen wird. Daher soll in folgenden immer $\mathrm{ggT}(n, q) = 1$ vorausgesetzt werden und wenn von zyklischen Codes gesprochen wird seien stets *einfach-Wurzel zyklische Codes* gemeint.
Der erste Teil dieses Abschnitts leitet einen Typ von Kontrollmatrizen zu zyklischen Codes her, der für BCH-Codes eine besondere Relevanz besitzt. Im zweiten Teil werden BCH-Codes als eine Unterklasse von zyklischen Codes motiviert und definiert. Anstelle von $GF(q^{m_n(q)})$ wird im folgenden immer die Notation $GF(q^m)$, also $m_n(q) = m$, verwendet.

1.3.1. Eine Kontrollmatrix

Satz 1.48 (und Definition)
Es sei $g(X)$ das Generatorpolynom eines zyklischen Codes $\mathcal{C}$ der Länge n über dem endlichen Körper $GF(q)$. Die Menge aller Nullstellen von $g(X)$

$$V_g := \{\beta \in GF(q^m) | g(\beta) = 0\} \subseteq E_n$$

über dem Zerfällungskörper $GF(q^m)$ wird die Varietät von g über $GF(q)$ *genannt. Die* Varietät *des zugehörigen* Codes $\mathcal{C}$

$$V_{\mathcal{C}} := \{\beta \in GF(q^m) | c(\beta) = 0 \ \forall \ c(X) \in \mathcal{C}\}$$

stimmt mit der Varietät V_g überein, d.h. $V_g = V_{\mathcal{C}}$.

Beweis. Es gilt $V_g \subseteq V_{\mathcal{C}}$, denn für alle $c(X) \in \mathcal{C}$ gibt es eine Nachricht $u(X)$ des Nachrichtenraums mit $c(X) = u(X) \cdot g(X)$. Andererseits ist $g(X) \in \mathcal{C}$, weshalb $\beta \in V_{\mathcal{C}} \Leftrightarrow g(\beta) = 0$. Daher $V_{\mathcal{C}} \subseteq V_g$. ∎

Es sei V_g die Varietät des zyklischen (n,k)-Codes $\mathcal{C}$ über $GF(q)$, d.h.

$$V_g := \{\beta_0, \beta_1, \ldots, \beta_{n-k-1}\} \ .$$

Dann ist $c(\beta_0) = c(\beta_1) = \cdots = c(\beta_{n-k-1}) = 0 \ \forall \ c(X) \in \mathcal{C}$. Folglich ist die Matrix

$$\Theta = \begin{pmatrix} 1 & \beta_0 & \beta_0^2 & \cdots & \cdots & \beta_0^{n-1} \\ 1 & \beta_1 & \beta_1^2 & \cdots & \cdots & \beta_1^{n-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \beta_{n-k-1} & \beta_{n-k-1}^2 & \cdots & \cdots & \beta_{n-k-1}^{n-1} \end{pmatrix} , \tag{1.7}$$

eine Kontrollmatrix von $\mathcal{C}$ über $GF(q^m)$, denn:

$$\Theta \cdot c^{tr} = \begin{pmatrix} c_0 + c_1 \cdot \beta_0 + \cdots + c_{n-1} \cdot \beta_0^{n-1} \\ c_0 + c_1 \cdot \beta_1 + \cdots + c_{n-1} \cdot \beta_1^{n-1} \\ \\ \vdots \\ c_0 + c_1 \cdot \beta_{n-k-1} + \cdots + c_{n-1} \cdot \beta_{n-k-1}^{n-1} \end{pmatrix} = \begin{pmatrix} c(\beta_0) \\ c(\beta_1) \\ \vdots \\ c(\beta_{n-k-1}) \end{pmatrix} = 0 \ .$$

Im folgende Lemma wird ein oft verwendetes Werkzeug zur Berechnung von Minimaldistanzen vorgestellt.

Lemma 1.49 (Vandermonde Matrix)
Sei $\mathbb{K}$ ein beliebiger endlicher Körper, $a_i \in \mathbb{K}$, $0 \leq i \leq r-1$ und i, r positive ganze Zahlen. Die Matrix

$$A = \begin{pmatrix} 1 & a_0 & a_0^2 & \cdots & \cdots & a_0^{r-1} \\ 1 & a_1 & a_1^2 & \cdots & \cdots & a_1^{r-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & a_{r-1} & a_{r-1}^2 & \cdots & \cdots & a_{r-1}^{r-1} \end{pmatrix}$$

wird Vandermonde Matrix *genannt. Ihre Determinante*[24] *ist*

$$\det A = \prod_{j=1}^{r-2} \prod_{i=j+1}^{r-1} (a_i - a_j) \ .$$

Sie ist verschieden von Null, wenn a_i und a_j paarweise verschieden sind.

Beweis. [MacWilliams et al. 1977], (S. 116) ∎

1.3.2. Die BCH-Schranke

Der folgende Satz ist eine zentrale Aussage über zyklische Codes.

Satz 1.50 (BCH-Schranke)
Es sei $\mathcal{C}$ ein zyklischer Code der Länge n über $GF(q)$, $\alpha \in E_n$ primitive n-te Einheitswurzel und $g(X)$ Generatorpolynom von $\mathcal{C}$ derart, dass für ganze Zahlen $b \geq 0$ und $\delta \geq 2$,

$$g(\alpha^b) = g(\alpha^{b+1}) = \ldots = g(\alpha^{b+\delta-2}) = 0 \ ,$$

d.h. der Code $\delta - 1$ aufeinander folgende Potenzen von α als Nullstellen hat. Dann ist die Minimaldistanz d von $\mathcal{C}$ mindestens δ, also $d \geq \delta$.

[24] Eine Einführung in die Determinantentheorie bietet [Fischer et al. 1986], S. 130ff.

Beweis. Es sei $c \in C$ ein von Null verschiedenes Codewort. Nach dem vorangegangenen Abschnitt sind

$$c(\alpha^b) = c(\alpha^{b+1}) = \cdots = c(\alpha^{b+\delta-2}) = 0 \ ,$$

so dass es die Matrix Θ' mit der folgenden Eigenschaft gibt:

$$\Theta' \cdot c^{tr} = \begin{pmatrix} 1 & \alpha^b & \alpha^{2b} & \ldots & \alpha^{b(n-1)} \\ 1 & \alpha^{b+1} & \alpha^{2(b+1)} & \ldots & \alpha^{(b+1)(n-1)} \\ 1 & \vdots & \vdots & \vdots & \vdots \\ 1 & \alpha^{b+\delta-2} & \alpha^{2(b+\delta-2)} & \ldots & \alpha^{(b+\delta-2)(n-1)} \end{pmatrix} \cdot \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-1} \end{pmatrix} = 0 \ . \qquad (1.8)$$

Θ' ist nicht notwendig die komplette Kontrollmatrix Θ von $\mathcal{C}$ über $GF(q^m)$ wie in (1.7). Satz 1.10 sagte, dass ein linearer Code die Minimaldistanz d besitzt, wenn in jeder Kontrollmatrix je $d-1$ Spalten linear unabhängig sind und es d linear abhängige Spalten gibt. Es wird nun gezeigt, dass schon in Θ' je $\delta - 1$ oder weniger Spalten linear unabhängig sind. Das reicht, denn wenn Θ' diese Eigenschaft besitzt, dann auch die Kontrollmatrix Θ. Angenommen es gibt in $\mathcal{C}$ ein Codewort c von Gewicht $\mathrm{wt}(c) = \omega \leq \delta - 1$ mit von Null verschiedenen Komponenten $c_{a_1}, \ldots, c_{a_\omega}$. Dann impliziert (1.8)

$$\begin{pmatrix} \alpha^{a_1 b} & \alpha^{a_2 b} & \ldots & \alpha^{a_\omega b} \\ \alpha^{a_1 (b+1)} & \alpha^{a_2 (b+1)} & \ldots & \alpha^{a_\omega (b+1)} \\ \vdots & \vdots & \vdots & \vdots \\ \alpha^{a_1 (b+\omega-1)} & \alpha^{a_2 (b+\omega-1)} & \ldots & \alpha^{a_\omega (b+\omega-1)} \end{pmatrix} \cdot \begin{pmatrix} c_{a_1} \\ c_{a_2} \\ \vdots \\ c_{a_\omega} \end{pmatrix} = 0 \ . \qquad (1.9)$$

Aus diesem Grund muss die Determinante der Matrix auf der linken Seite von Gleichung (1.9) identisch mit Null sein. Elementare Umformungen der Determinante[25] dieser Matrix liefern aber

$$\det \begin{pmatrix} \alpha^{a_1 b} & \ldots & \alpha^{a_\omega b} \\ \alpha^{a_1 (b+1)} & \ldots & \alpha^{a_\omega (b+1)} \\ \vdots & \vdots & \vdots \\ \alpha^{a_1 (b+\omega-1)} & \ldots & \alpha^{a_\omega (b+\omega-1)} \end{pmatrix} = \alpha^{(a_1 + \ldots + a_\omega) b} \cdot \det \begin{pmatrix} 1 & \ldots & 1 \\ \alpha^{a_1} & \ldots & \alpha^{a_\omega} \\ \vdots & \vdots & \vdots \\ \alpha^{a_1 (\omega-1)} & \ldots & \alpha^{a_\omega (\omega-1)} \end{pmatrix} \qquad (1.10)$$

[25] Vergleiche dazu z.B. [Fischer et al. 1986].

Die Determinante der rechten Seite ist die einer Vandermonde Matrix. Da die Einträge $\alpha^{a_1}, \ldots, \alpha^{a_\omega}$ paarweise verschieden sind, ist sie wegen Lemma 1.49 verschieden von Null. Das ist ein Widerspruch zu Gleichung (1.9). Also gibt es kein Codewort $c \in \mathcal{C}$ mit $\mathrm{wt}(c) \leq \delta - 1$. ■

Der Satz über die BCH-Schranke zeigt, dass eine erwartete Mindestfehlerkorrekturfähigkeit bei der Konstruktion von zyklischen Codes vorausgesetzt werden kann. Diese Tatsache motiviert die folgende Definition:

Definition 1.51
Ein zyklischer Code $\mathcal{C}$ der Länge n über $GF(q)$ heißt *BCH-Code der Entwurfsdistanz* δ, wenn für eine positive ganze Zahl $b \geq 0$ gilt

$$g(X) = \mathrm{kgV}\{M^{(b)}(X), M^{(b+1)}(X), \ldots, M^{(b+\delta-2)}(X)\} \ .$$

Bemerkung 1.52

(i). Die Kontrollmatrix Θ eines BCH-Codes $\mathcal{C}$ der Länge n und Entwurfsdistanz δ über $GF(q)$ ist nach Abschnitt 1.3.1 von der Form

$$\Theta = \begin{pmatrix} 1 & \alpha^{b} & \alpha^{2b} & \ldots & \ldots & \alpha^{(n-1)b} \\ 1 & \alpha^{2b} & \alpha^{4b} & \ldots & \ldots & \alpha^{(n-1)2b} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \alpha^{b+\delta-2} & \alpha^{2(b+\delta-2)} & \ldots & \ldots & \alpha^{(n-1)(b+\delta-2)} \end{pmatrix}$$

wobei jeder Eintrag durch sein zugehöriges m-Tupel mit Einträgen in $GF(q)$ ersetzt wird.

(ii). Ist $b = 1$ so wird vom *BCH-Code im engeren Sinne („Narrow Sense“)* gesprochen. Im Fall $b \neq 1$ wird $\mathcal{C}$ ein *BCH-Code im weiteren Sinne („Wide Sense“)* genannt.

(iii). Ist $n = q^m - 1$, dann heißt $\mathcal{C}$ *primitiver BCH-Code*, weil in diesem Fall eine primitive n-te Einheitswurzel α existiert, die die gesamte multiplikative Gruppe von $GF(q^m)$ erzeugt.

(iv). BCH-Codes mit $n = q - 1$ sind ein Spezialfall von primitiven BCH-Codes und heißen nach ihren Entdeckern benannt *Reed-Solomon Codes*. Natürlich gilt bei Reed-Solomon Codes $q = p^r$ mit $r > 1$.

Beispiel 1.53

(i). Weil die durch b und δ festgelegten Minimalpolynome in der Definition eines BCH-Codes $\mathcal{C}$ redundant sein können, kann es passieren, dass für festes $b > 0$ die Entwurfsdistanzen $\delta' > \delta$ das gleiche Generatorpolynom festlegen. Die maximalen Entwurfsdistanzen aus größtmöglichen, um die ganze Zahl „1“ wachsenden Folgen von Entwurfsdistanzen $\delta_0 < \delta_1 < \cdots < \delta_{Bose}$ für die das Generatorpolynom identisch aber von dem durch δ_{Bose+1} definierten verschieden ist, werden *Bose-Distanzen* genannt und mit δ_{Bose} bezeichnet. Beispielsweise sind die ZNK von 2 mod 15:

$$\begin{aligned} C_0 &= \{0\} \\ C_1 &= \{1,2,4,8\} \\ C_3 &= \{3,6,9,12\} \\ C_5 &= \{5,10\} \\ C_7 &= \{7,11,13,14\}\ . \end{aligned}$$

Die Entwurfsdistanzen $\delta_1 = 4$ und $\delta_2 = \delta_{Bose} = 5$ definieren bei festem $b = 1$ das gleiche Generatorpolynom $g(X) = M^{(1)}(X) \cdot M^{(3)}(X)$. Die Entwurfsdistanz $\delta_3 = 6$ legt jedoch unter sonst gleichen Voraussetzungen das Generatorpolynom $g(X) = M^{(1)}(X) \cdot M^{(3)}(X) \cdot M^{(5)}(X)$ fest. Bei BCH-Codes im engeren Sinne sind die Bose-Entwurfsdistanzen genau die kanonischen Repräsentanten der zyklotomischen Nebenklassen (unter Beachtung von $n \equiv 0 \mod n$).

(ii). Für $n - b + 2 \leq \delta$ ist das Generatorpolynom stets von der Form $g(X) = X^n - 1$, da in diesem Fall die Varietät $V_{\mathcal{C}}$ identisch mit E_n ist. Das Kontrollpolynom ist dann von der Form $h(X) = X^n - 1/X^n - 1 = 1$ und das einzige Codewort des Codes ist das Nullcodewort.
Ein durch solche Parameter δ und b definierter BCH-Code $\mathcal{C}$ hat die Dimension $k = 0$, denn nach Bemerkung 1.20 ist die Dimension eines zyklischen Codes identisch mit dem Grad des Kontrollpolynoms. Die wahre Minimaldistanz von $\mathcal{C}$ ist $d = \infty$, da zu jeder beliebigen Entwurfsdistanz $\delta' \geq n - b + 2$ die Entwurfsdistanz $\delta' + 1$ existiert, die das gleiche Generatorpolynom $g(X) = X^n - 1$ festlegt.
Das Generatorpolynom $g(X) = X^n - 1$ kann aber auch in manchen Fällen schon durch $\delta \leq n - b + 2$ definiert sein. Ist z.B. $n = 15$, $q = 2$, $b = 0$ ist es bereits festgelegt durch $\delta = 8 < 15 + 2$ (vgl. oben stehende Tabelle der ZNK von 2 mod 15).

(iii). Es existiert kein BCH-Code mit Entwurfsdistanz $\delta < 2$, weil das Generatorpolynom definitionsgemäß keine Nullstellen in $GF(q^m)$ hat. Anders interpretiert hat ein zyklischer $(n, k = n)$-Code ein Generatorpolynom der Form $g(X) = 1$. Also wäre so ein Code identisch mit der Nachrichtenmenge, und weil der Code Codewörter c von Gewicht $wt(c) = 1$ enthält (Standardbasisvektoren von $GF(q)^k$) besitzt er die Minimaldistanz $d = 1$.

(iv). Für jeden BCH-Code im engeren Sinne der Länge n über $GF(q)$ ist der Fall $\delta = n$ trivial und wird *n-facher Wiederholungscode* genannt. Die einzige n-te Einheitswurzel, die nicht Nullstelle des Generatorpolynoms ist, ist $\alpha^0 = 1$. Also hat das Generatorpolynom den Grad $\operatorname{grad} g(X) = n - 1$ und ist von der Form

$$g(X) = X^n - 1/X - 1 = 1 + X + \ldots + X^{n-1} .$$

Wegen Bemerkung 1.20 ist die Dimension des Codes $k = 1$ und der Nachrichtenraum enthält die Elemente $0, 1, \ldots, q-1$, wobei diese Elemente zugehörige r-Tupel aus $GF(q)$ mit Einträgen in $GF(p)$ repräsentieren, wenn $q = p^r$ eine Primzahlpotenz zur Basis p ist. Daher sind die einzigen Codewörter des Codes $(0, \ldots, 0), (1, \ldots, 1), \ldots, (q-1, \ldots, q-1)$ und die Minimaldistanz ist identisch mit der Entwurfsdistanz, also $d = \delta = n$.

(v). Der Fall $b = 0$ und $\delta = 2$ eines BCH-Codes der Länge n über $GF(q)$ ist ebenso trivial. Das Generatorpolynom hat nur eine Nullstelle, nämlich $\alpha^0 = 1$ und ist deshalb stets von der Form

$$g(X) = X - 1 .$$

Da das Generatorpolynom selbst ein Codewort repräsentiert, ist offensichtlich $d = \delta = 2$ und die Dimension des Codes beträgt wieder wegen Bemerkung 1.20 $k = n - 1$, da $\operatorname{grad} g(X) = 1$. Dieser Code wird *Paritätscode* genannt. Die Codewörter des Codes sind alle Vektoren aus $GF(q)^n$, die ein gerades Hamminggewicht besitzen, denn die Multiplikation jeder Nachricht $u \in GF(q)^{n-1}$ in ihrer polynomialen Repräsentation mit $g(X)$ erzeugt immer ein Codewort von geradem Gewicht.

BCH-Codes über $GF(2)$, auch binäre BCH-Codes genannt, sind in Anwendungen der Datenübermittlung von großer Bedeutung. Deshalb lohnt es sich Besonderheiten solcher Codes zu betrachten.

Bemerkung 1.54

Für Minimalpolynome über $GF(2)$ gilt: $M^{(i)}(X) = M^{(2i)}(X)$ (vgl. Satz 1.44). Ist $b = 1$, so haben binäre Codes mit Entwurfsdistanz $2t$ oder $2t + 1$ gleiche Generatorpolynome (vgl. auch Beispiel 1.53):

$$g(X) = \text{kgV}\{M^{(1)}(X), M^{(3)}(X), \ldots, M^{(2t-1)}(X)\} \ .$$

Die Kontrollmatrix Θ kann durch die lineare Abhängigkeit von je einer ungeraden mit einer geraden Zeile somit auf die Form

$$\Theta = \begin{pmatrix} 1 & \alpha & \alpha^2 & \ldots & \ldots & \alpha^{(n-1)} \\ 1 & \alpha^3 & \alpha^6 & \ldots & \ldots & \alpha^{(n-1)3} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \alpha^{2t-1} & \alpha^{2(b+\delta-2)} & \ldots & \ldots & \alpha^{(n-1)(2t-1)} \end{pmatrix}$$

reduziert werden.

2. Minimaldistanz und Dimension von BCH-Codes

Die ersten beiden Abschnitte diese Kapitels befassen sich mit der Dimension von BCH-Codes. Ausgehend von Berechnungsvorschlägen für Spezialfälle von BCH-Codes, die in der einschlägigen Literatur zu finden sind, wird ein Algorithmus entworfen, der die Dimension beliebiger BCH-Codes berechnet.
Die anderen beiden Abschnitte des Kapitels sind der Minimaldistanz von BCH-Codes gewidmet. Zunächst wird der gegenwärtige Stand der Forschung zusammengefasst, um anschließend eine Methode zu beschreiben, die die Minimaldistanz von zyklischen Codes liefert. Die Theorie dessen wird am Beispiel eines binären BCH-Codes im engeren Sinne veranschaulicht.

2.1. Bemerkungen zur Dimension

Die Dimension eines Codes entspricht nach Kapitel 1 der Länge der Nachrichten, die codiert werden sollen. BCH-Codes sind im Gegensatz zu vielen anderen Codes darauf zugeschnitten, eine erwartete Mindestfehlerkorrekturfähigkeit, impliziert durch die Entwurfsdistanz, auf Kosten der Kenntnis der Dimension voraussetzen zu können. Dies ist in gewisser Hinsicht ein Vorteil gegenüber anderen Codeklassen, bei denen die Dimension vorausgesetzt wird und die Minimaldistanz völlig im Verborgenen liegt. Es ist bei großen Codelängen wesentlich leichter, eine implizit

vorliegende Dimension, als eine implizit vorliegende Minimaldistanz, zu bestimmen. Nach Bemerkung 1.20 ist die Dimension eines zyklischen Codes $\mathcal{C}$ der Länge n identisch mit

$$k = n - \operatorname{grad} g(X) = \operatorname{grad} h(X) \quad , \tag{2.1}$$

wobei $g(X)$ das Generatorpolynom und $h(X)$ das Kontrollpolynom von $\mathcal{C}$ bezeichnet. Gleichung (2.1) zeigt, dass bei zyklischen Codes die Dimension unmittelbar durch den Grad des Generatorpolynoms gegeben ist. Das Generatorpolynom von BCH-Codes ist das kleinste gemeinsame Vielfache der Minimalpolynome von $\delta - 1$ aufeinander folgenden n-ten Einheitswurzeln α^i. Wie schon in Beispiel 1.53 erwähnt, ist es nicht deren Produkt, weil mehrere der durch die Größe b und der Entwurfsdistanz δ festgelegten $\alpha^i \in E_n$ das gleiche Minimalpolynom besitzen können.

Das nachfolgende Lemma liefert für die Dimension eines BCH-Codes $\mathcal{C}$ eine Schranke in Abhängigkeit von der Entwurfsdistanz δ. Im Wesentlichen ist es eine Folgerung aus Gleichung (2.1) und den Bemerkungen 1.46 und 1.54.

Lemma 2.1
Ein BCH-Code der Länge n und Entwurfsdistanz δ über $GF(q)$ besitzt die Dimension $k \geq n - m_n(q) \cdot (\delta - 1)$. Ist $\mathcal{C}$ ein binärer BCH-Code der Länge n und Entwurfsdistanz $\delta = 2t+1$ oder $\delta = 2t$, dann ist die Dimension von $\mathcal{C}$ nach unten beschränkt durch $k \geq n - m_n(2) \cdot t$.

Beweis. [MacWilliams et al. 1977], (S. 203) ∎

Die Größe $\delta - 1$ in der ersten Ungleichung von Lemma 2.1 ist eine obere Schranke für die Anzahl der Minimalpolynome, die das Generatorpolynom bilden. Daneben taucht in dieser Ungleichung eine weitere „Ungewissheit“ auf. Satz 1.39 und Beispiel 1.53 zeigten, dass Minimalpolynome über $GF(q)$ zu Elementen aus $GF(q^m)$ von verschiedenem Grad sein können, nämlich den Teilern der multiplikativen Ordnung $m_n(q)$. Dies ist der zweite Grund dafür, dass das Lemma nur obere Schranken und keine Gleichungen für die Dimension liefert.

Der elementare Zugang zur Dimension k wäre mit Gleichung (2.1), die ZNK derjenigen Minimalpolynome, die nicht zum Generatorpolynom gehören, explizit aus-

zurechnen und deren Elemente abzuzählen. In der Literatur, z.B. [Berlekamp 1968] (S. 273ff) und [MacWilliams et al. 1977] (S. 262ff), existieren für Spezialfälle von BCH-Codes im engeren Sinne, d.h. $b = 1$, eine Reihe von Abstraktionen dieser Vorgehensweise. Das folgende Lemma (nach [Berlekamp 1968], S. 274) ist grundlegend für eine Vielzahl solcher Aussagen:

Lemma 2.2
Es sei $\mathcal{C}$ ein BCH-Codes im engeren Sinne der Länge n und Entwurfsdistanz δ über $GF(q)$. Die Zahl $[i]$ sei definiert durch $i \equiv [i] \bmod n$, wobei $1 \leq [i] \leq n$. Dann gilt für die Dimension k von $\mathcal{C}$:

$$k = \left|\left\{1 \leq j \leq n : \ [j \cdot q^l] \geq \delta \ \forall \ l \in \mathbb{N}\right\}\right| \ ,$$

d.h. k ist die Anzahl der Zahlen im Bereich von $1 \leq j \leq n$ deren Konjugierte ausnahmslos größer bzw. gleich δ sind.

Beweis. $\alpha^j \in E_n$ ist eine Nullstelle des Generatorpolynoms von $\mathcal{C}$ genau dann, wenn ein $0 \leq l_j \leq m_n(q) - 1$ existiert, so dass $[j \cdot q^{l_j}] < \delta$. (Die Schranken von l_j stellen keine Einschränkung der Allgemeinheit dar, weil die Ordnung jeder ZNK durch $m_n(q)$ beschränkt ist). Kontradiktorisch ist dies gleichbedeutend mit der Aussage, dass α^j keine Nullstelle des Generatorpolynoms von $\mathcal{C}$ ist, wenn $[j \cdot q^l] \geq \delta$ für alle $0 \leq l \leq m_n(q) - 1$. Mit Gleichung (2.1) folgt dann die Behauptung. ■

Beispiel 2.3
Sei $\mathcal{C}$ der binäre BCH-Code im weiteren Sinne mit $b = 11$, der Länge $n = 15$ und Entwurfsdistanz $\delta = 7$. Die multiplikative Ordnung von 2 mod 15 beträgt $m_{15}(2) = 4$. Mit der in Beispiel 1.53 gegebenen Tabelle der ZNK von 2 mod 15 ist es offensichtlich, dass die Dimension des Codes $k = 2$ beträgt. Mit der gleichen Tabelle ist leicht zu überprüfen, dass

$$\left|\left\{1 \leq j \leq 15 : \ [j \cdot 2^l] \geq 7 \text{ für } 0 \leq l \leq 3\right\}\right| = 5 \neq k \ .$$

Dies zeigt, dass Lemma 2.2 nicht für BCH-Codes im weiteren Sinne, d.h. $b \neq 1$, anwendbar ist. Eine etwas aufwendigere Verallgemeinerung zu Lemma 2.2 liefert aber die folgende Aussage:

Satz 2.4

Es sei $\mathcal{C}$ ein beliebiger BCH-Codes der Länge n und Entwurfsdistanz δ über $GF(q)$. Die Zahl $\overline{i}$ sei gegeben durch $i \equiv \overline{i} \bmod n$ und folgende Bedingungen an positive ganze Zahlen j und l seien definiert durch

$$\begin{aligned}
B_1^{(j,l)} &:= \quad \overline{b}+\delta-2 \geq n \quad \vee \quad \delta-1 \geq n \quad \vee \quad \overline{b} > \overline{j \cdot q^l} \quad \vee \quad \overline{j \cdot q^l} > \overline{b}+\delta-2 \\
B_2^{(j,l)} &:= \quad \overline{b}+\delta-2 < n \quad \vee \quad \delta-1 \geq n \quad \vee \quad \left(\overline{b} > \overline{j \cdot q^l} \quad \wedge \quad \overline{j \cdot q^l} > \overline{b+\delta-2}\right) \\
B_3^{(j,l)} &:= \quad \delta-1 < n \ ,
\end{aligned}$$

wobei $\vee$ das logische 'oder' und $\wedge$ das logische 'und' bezeichnet. Dann gilt für die Dimension k von $\mathcal{C}$:

$$k = \left| \left\{ 0 \leq j \leq n-1 : B_1^{(j,l)} \wedge B_2^{(j,l)} \wedge B_3^{(j,l)}, \ \forall \ \ 0 \leq l \leq m_n(q)-1 \right\} \right| \ .$$

Beweis. Analog zum Beweis von Lemma 2.2 ist im Fall $\overline{b}+\delta-2 < n$ und $\delta-1 < n$ die n-te Einheitswurzel α^j, $0 \leq j \leq n-1$, genau dann eine Nullstelle des Generatorpolynoms $g(X)$ von $\mathcal{C}$, wenn ein $0 \leq l_j \leq m_n(q)-1$ existiert, so dass

$$\left(\overline{b} \leq \overline{j \cdot q^{l_j}}\right) \ \wedge \ \left(\overline{j \cdot q^{l_j}} \leq \overline{b}+\delta-2\right) \tag{2.2}$$

wahr ist. Im Fall $\delta-1 < n$ und $\overline{b}+\delta-2 \geq n$ ist α^j eine Nullstelle von g(X), genau dann, wenn es ein l_j gibt, dass der Bedingung

$$\left(\overline{b} \leq \overline{j \cdot q^{l_j}}\right) \ \vee \ \left(\overline{j \cdot q^{l_j}} \leq \overline{b+\delta-2}\right)$$

genügt. Für $\delta-1 \geq n$ ist jedes $\alpha^i \in E_n$ Nullstelle von $g(X)$. Es sei definiert:

$$\begin{aligned}
\rceil B_1^{(j,l)} &:= \quad \overline{b}+\delta-2 < n \quad \wedge \quad \delta-1 < n \quad \wedge \quad \overline{b} \leq \overline{j \cdot q^l} \quad \wedge \quad \overline{j \cdot q^l} \leq \overline{b}+\delta-2 \\
\rceil B_2^{(j,l)} &:= \quad \overline{b}+\delta-2 \geq n \quad \wedge \quad \delta-1 < n \quad \wedge \quad \left(\overline{b} \leq \overline{j \cdot q^l} \quad \vee \quad \overline{j \cdot q^l} \leq \overline{b+\delta-2}\right) \\
\rceil B_3^{(j,l)} &:= \quad \delta-1 \geq n \ .
\end{aligned}$$

Zusammengenommen ist $\alpha^j \in E_n$ also eine Nullstelle von $g(X)$ genau dann, wenn ein $0 \leq l_j \leq m_n(q)-1$ existiert, so dass die Bedingung

$$\rceil B_1^{(j,l_j)} \ \vee \ \rceil B_2^{(j,l_j)} \ \vee \ \rceil B_3^{(j,l_j)} \tag{2.3}$$

wahr ist. Die kontradiktorischen Aussagen zu $\rceil B_1^{(j,l)}$, $\rceil B_2^{(j,l)}$ und $\rceil B_3^{(j,l)}$ sind $B_1^{(j,l)}$, $B_2^{(j,l)}$ bzw. $B_3^{(j,l)}$, so dass Überführung von (2.3) in die entgegengesetzte Aussage mit Gleichung (2.1) die Behauptung liefert. ■

2.2. Ein Algorithmus zur Berechnung der Dimension

Auf die im Satz 2.4 vorgestellte „Brute-Force"-Methode stützt sich der im folgenden dargestellte, im Algebrasystem *MATHEMATICA* implementierte Algorithmus zur Berechnung der Dimension k von beliebigen BCH-Codes über endlichen Körpern $GF(q)$.

Die Dimension eines beliebigen BCH-Codes der Länge n und Entwurfsdistanz δ über GF(q)

```
q = 2    (* beliebige Primzahlpotenz, Ordnung des Grundkörpers *)
n = 15   (* ggT(n,q)=1, Länge des Codes *)
b = 1    (* ≥0, b = 1 'narrow sense', b ≠ 1 'wide sense'*)
δ = 5    (* ≥2, Entwurfsdistanz *)
```

```
Bezeichner
ϕ , Eulersche Phi - Funktion
DIV , Menge der Teiler von ϕ (n)
QMODN , Menge der Reste von 2^l mod n, l ∈ DIV
```

```
DEFN = Catch[
          If[GCD[n, q] ≠ 1,
             Throw["ggT(n,q)≠1: Eingabe überprüfen!"],
             {ϕ = EulerPhi[n],
              DIV = Divisors[ϕ] ,
              QMODN = PowerMod[q, DIV, n]
             }
            ]
          ]
```

```
Do[
   {m = Infinity,
    If[QMODN[[i]] == 1 , {m = DIV[[i]], Break[]}, Continue[]]
   }, {i, Count[QMODN, _Integer]}
  ]
```

Abb.2.1 a: „Brute-Force" Algorithmus zur Berechnung der Dimension eines beliebigen BCH-Codes implementiert in *MATHEMATICA* (Fortsetzung auf der nächsten Seite).

```
TableForm[ {{"m = ", m}}, TableSpacing -> {0, 0}]   (* Ausgabe von m_n (q) *)
```

```
Bezeichner
WAHR , enthält NST[j] zu jedem 1 ≤ j ≤ n
NST[j] , Elementmenge mit Wahrheitswerten zu  j und 0 ≤ l ≤ m_n (q) - 1 = m - 1,  in Bezug zu Satz (2.4)
```

```
WAHR = Catch[
          If[b < 0  ||  δ < 2,
             Throw["Dieser BCH-Code existiert nicht :  Eingabe überprüfen!"],
             {Table[NST[j]
                    = Union[
                        Table[
                            And[
           Mod[b, n] + δ - 2 ≥ n || Mod[b, n] > Mod[j*q^l, n] || Mod[j*q^l, n] > Mod[b, n] + δ - 2,
           Mod[b, n] + δ - 2 < n || Mod[b, n] > Mod[j*q^l, n] && Mod[j*q^l, n] > Mod[b + δ - 2, n],
           δ - 1 < n
                                ] , {l, 0, m - 1}
                              ]
                            ], {j, 0, n - 1}
                    ]
              }
             ]
            ]
```

```
Bezeichner
NOTVAR , Menge der j bzgl . α^j ,  die Satz (2.4) genügen und damit nicht zur Varietät des Codes gehören
```

```
NOTVAR =  DeleteCases[
                Table[
                    If[And[First[NST[j]], Last[NST[j]]],
                       Mod[j, n]
                      ], {j, 0, n - 1}
                    ], Null
                  ]
```

```
If[GCD[n, q] ≠ 1,
   TableForm[{DEFN}],
   If[(b < 0  ||  δ < 2),
      TableForm[{WAHR}],
      TableForm[ {{"k = ", Count[NOTVAR, _Integer]}}, TableSpacing -> {0, 0}]
     ]
  ]
```

Abb.2.1 b: (Fortsetzung) „Brute-Force“ Algorithmus zur Berechnung der Dimension eines beliebigen BCH- Codes implementiert in *MATHEMATICA*.

2.2.1. Beschreibung des Algorithmus

Der Algorithmus benötigt n, q, b und δ als Eingabeparameter, wobei die Teilerfremdheit von n und q, $b \geq 0$ und $\delta \geq 2$ beachtet werden müssen (vgl. Bemerkung 1.24, Definition 1.51 und Beispiel 1.53).
Im ersten Schritt berechnet der Algorithmus die multiplikative Ordnung $m_n(q)$. Dazu bestimmt *MATHEMATICA* zunächst den Wert von n bezüglich der Eulerschen φ-Funktion. Die multiplikative Ordnung $m_n(q)$ ist ein Teiler von $\varphi(n)$, deshalb bildet der Algorithmus die Menge aller Teiler $T \in DIV$ von $\varphi(n)$. Aus den Elementen $T \in DIV$ wird eine neue Menge *QMODN* generiert, die die Werte von $T \in DIV$ bezüglich $q^T \bmod n$ enthält. Die nachfolgende Programmschleife überprüft mit Hilfe von *QMODN*, welcher kleinste Teiler $T_0 \in DIV$ der Kongruenz $q^{T_0} \equiv 1 \bmod n$ genügt. Der Wert T_0 entspricht der multiplikativen Ordnung $m_n(q)$ und wird daher der Variablen m zugewiesen. Wurde bei der Eingabe der Parameter n und q die Bedingung $\mathrm{ggT}(n, q) = 1$ missachtet, bricht der Algorithmus schon vor Berechnung von $\varphi(n)$ ab und gibt eine Fehlermeldung aus.

Die Berechnung von m ist gemäß Satz 2.4 als obere Schranke von l, für die Überprüfung des Wahrheitsgehalts der Bedingungen $B_1^{(j,l)}$, $B_2^{(j,l)}$ und $B_3^{(j,l)}$ von Bedeutung.
Der auf die Berechnung von m folgende Programmblock generiert die Menge *WAHR*, die zu jeder Zahl im Bereich von $0 \leq j \leq n-1$ eine Elementmenge *NST[j]* enthält. Die Elementmengen *NST[j]* enthalten zu jedem j die Wahrheitswerte bzgl. der Bedingung $B_1^{(j,l)} \wedge B_2^{(j,l)} \wedge B_3^{(j,l)}$ für alle $0 \leq l \leq m_n(q) - 1$. Durch die Mengenoperation der Vereinigung sind sie jeweils von einer der Arten {*True*}, {*False*} oder {*True,False*}, je nach dem, ob die Bedingung für alle, keine oder nur einige l erfüllt ist. Wurden bei der Eingabe die Voraussetzungen $b \geq 0$ und $\delta \geq 2$ missachtet, wirft der Programmblock eine Fehlermeldung auf und es wird die Ausgabe veranlasst, dass ein BCH-Code mit diesen Eingabeparametern nicht existiert.

Der nächste Programmblock erzeugt die Menge *NOTVAR*, die die Exponenten der n-ten Einheitswurzeln enthält, die nicht zur Varietät $V_{\mathcal{C}}$ des Codes $\mathcal{C}$ gehören. Dazu überprüft der Algorithmus welche *NST[j]* der Art {*True*} sind, das gleich-

bedeutend damit ist, dass j die Bedingung aus Satz 2.4 für alle $0 \leq l \leq m-1$ hält. Entsprechende j werden in die Menge $NOTVAR$ geschrieben.

Die Anzahl der Elemente in $NOTVAR$ entspricht der Dimension des Codes, daher wird im letzten Schritt die Anzahl der Elemente von $NOTVAR$ ermittelt und die Dimension k ausgegeben.

2.2.2. Leistungsfähigkeit des Algorithmus

Satz 2.4 verallgemeinert die bekannte Aussage von Lemma 2.2 für beliebige BCH-Codes. Seine Implementation in *MATHEMATICA (Version 3.0)* ermöglicht eine schnelle Berechnung der Dimension für große Codelängen n über großen Körperordnungen von $GF(q)$ beliebiger Charakteristik.
Der Computer, mit dem Testberechnungen erstellt wurden, enthielt einen *Pentium II* 233 *Mhz* Prozessor und 32 *Mb RAM*. Der Algorithmus kann auf diesem System bei kleinem n, z.B. $n = 32$, in wenigen Sekunden die Dimension für Körperordnungen bis mindestens $q = 252.097.800.623$ (das ist die 10. Milliardste Primzahl) berechnen. Die Grenze für die Berechnung der Dimension in Bezug zu Codelängen n liegt auf diesem System etwa bei $n < 2^{25}$. Die Rechenzeit bei Eingabeparametern $q = 2,\ b = 1,\ \delta = 3$ und $n = 2^{18} - 1$ lag bei etwa 50 Minuten. Für $n < 2^{13}$ bedurfte die Berechnung weniger als eine Minute.

In der Literatur zu fehlerkorrigierenden Codes sind Tabellen zu den Parametern n, k und δ bzw. der wahren Minimaldistanz d von binären, primitiven BCH-Codes im engeren Sinne zu finden (vgl. [MacWilliams et al. 1977], S.267: $n = 2^3 - 1, \ldots, 2^8 - 1$; *P. Charpin* in [Pless et al. 1998], S.1013: $n = 2^9 - 1$).
A.E. Brouwer veröffentlichte 1998 in [Pless et al. 1998], (S. 319–449), eine Aktualisierung seiner Tabelle über Schranken linearer Codes für die Körperordnungen $q = 2, 3, 4, 5, 7, 8, 9$ mit Beschränkungen $n \leq 256, 243, 128, 100, 50, 85, 121$, aber ohne jeden einzelnen in der Tabelle enthaltenen BCH-Code kenntlich zu machen. Das symbolische Algebrasystem *MAGMA*[1] erlaubt die Berechnung der Dimension beliebiger BCH-Codes. Jedoch erfordert *MAGMA* stets die Erzeugung der

[1] http://www.maths.usyd.edu.au:8000/u/magma/

Basis des als Vektorraum aufgefassten BCH-Codes. Deswegen ist der Berechnungsrahmen auf dem Testsystem stark eingeschränkt gewesen. Schon bei den Eingabeparametern $q = 2,\ \delta = 3,\ b = 1$ und $n \geq 2^{13} - 1$ war der verfügbare Speicher ausgeschöpft, so dass Ergebnisse ausblieben.

Der hier dargestellte Algorithmus stellt eine Alternative zur Berechnung der Dimension von beliebigen BCH-Codes durch *MAGMA* dar und erweitert die Angaben zur Dimension von BCH-Codes in der Literatur beträchtlich. Er berechnet die Dimension eines Codes unabhängig von dessen Basis, so dass weit weniger Speicherkapazität des Systems bei gleichen Eingabeparametern verlangt wird.

Der Rechenaufwand gegenüber Lemma 2.4 lässt sich noch verbessern, indem die Abstraktionsebene erhöht wird und die q-nären Darstellungen der Zahlen $j \in \mathbb{Z}/n\mathbb{Z}$ betrachtet werden. Für BCH-Codes im engeren Sinne kann man entsprechende Aussagen bei [MacWilliams et al. 1977] (S. 262ff) und [Berlekamp 1968] (S. 273ff) finden.

2.3. Bemerkungen zur Minimaldistanz

Die mit der BCH-Schranke verbundene Entwurfsdistanz eines BCH-Codes ist nur eine untere Schranke der Minimaldistanz. *Augot et al.* [Augot et al. 1992] stellten kürzlich wiederholt heraus, dass es ein schwieriges Problem ist, die wahre Minimaldistanz eines allgemeinen BCH-Codes großer Länge zu finden.
Es verwundert daher nicht, dass sich die Forschung der letzten 35 Jahre auf Spezialfälle beschränkte. Besondere Beachtung erhielt vor allem der wichtige Fall der binären, primitiven BCH-Codes im engeren Sinne. Schon früh war für Codes dieser Art bekannt, dass bei Codelängen $n = 2^m - 1,\ m = 2, 3, 4, 5, 6$, *Bose-Entwurfsdistanzen* (vgl. Beispiel 1.53) identisch mit den wahren Minimaldistanzen sind. *Peterson* warf daher in [Peterson et al. 1967] die Frage auf, ob diese Tatsache nicht für beliebige primitive Codelängen $n = 2^m - 1$ richtig wäre. *Kasami* und *Tokura* vermochten eine negative Antwort auf diese Frage zu geben, indem sie für $m > 6$ und $m \neq 8, 12$ die Existenz von Unterklassen des binären, primitiven Falls im engeren Sinne nachwiesen, in denen die wahre Minimaldistanz die Entwurfsdistanz knapp überstieg (vgl. [Kasami et al. 1969]).

Das kleinste Beispiel dafür, dass die Minimaldistanz d über der Entwurfsdistanz δ liegt, ist der Fall $n = 2^7 - 1$ mit $\delta = 29$ und $d = \delta + 2$. Erst vor wenigen Jahren konnte gezeigt werden, dass im Fall $n = 2^8 - 1$ für zwei Bose-Entwurfsdistanzen die wahre Minimaldistanz $d = \delta + 2$ die BCH-Schranke übersteigt (vgl. [Augot et al. 1992] und Tab. 2.1). Für $n = 2^{12} - 1$ ist die Frage noch ungeklärt, ob es einen Code gibt, dessen Minimaldistanz echt größer als seine Entwurfsdistanz ist.

Die Vielzahl der BCH-Code Klassen, bei denen das Verhältnis von Entwurfsdistanz zur wahren Minimaldistanz bekannt ist, wurde bereits in den 60er Jahren entdeckt. Neben *Peterson, Kasami* und *Tokura* haben sich *Berlekamp, Lin, Weldon* und *Chen*, um nur einige weitere Wissenschaftler aus dieser Zeit zu nennen, mit Schlüsselsätzen zur Minimaldistanz von speziellen BCH-Codes hervorgetan. Die oft verwendete Methode um die Minimaldistanz von Codes aufzuspüren, ist die Existenz eines Codeworts im Code mit einem bestimmten Gewicht nachzuweisen oder auszuschließen. Die Kenntnis von Schranken für die Minimaldistanz, wie die BCH-Schranke, leisten das Übrige. Die wichtigsten „klassischen“ Aussagen über die Minimaldistanzen zyklischer Codes und insbesondere von BCH-Codes lassen sich in [Peterson et al. 1972], (S. 278ff), und [MacWilliams et al. 1977], (S. 259ff), finden. Auch in der letzten Dekade des 20. Jahrhunderts erschienen noch einige Arbeiten zur Minimaldistanz von speziellen Unterklassen von BCH-Codes (z.B. [Charpin 1990]). Neuere Methoden zur Berechnung der Minimaldistanz sind von *P. Charpin* in [Pless et al. 1998] zusammengefasst worden. Einen Überblick über untere Schranken für die Minimaldistanzen zyklischer Codes bietet [van Lint et al. 1986].

Noch bis Anfang der 80er Jahre waren für $n = 2^7 - 1$ nicht alle Minimaldistanzen für gegebene Bose-Entwurfsdistanz δ bei binären, primitiven BCH-Codes im engeren Sinne aufgeklärt. Durch die Entwicklung von sogenannten „Brute-Force“-Angriffen, konnten aber einige Lücken geschlossen werden. Heute sind die Minimaldistanzen zu allen Bose-Entwurfsdistanzen bis einschließlich der Länge $n = 2^8 - 1$ bekannt (siehe Tab. 2.1). Für $n = 2^9 - 1$ bleibt die Frage für die Bose-Entwurfsdistanzen $\delta = 59, 61, 75, 77, 85$ und 107 weiterhin unbeantwortet. Dies zeigt, dass das Problem der wahren Minimaldistanz selbst im Spezialfall der binären, primitiven BCH-Codes im engeren Sinne nicht zufriendenstellend gelöst ist.

n	k	δ	d	n	k	δ	d	n	k	δ	d	n	k	δ	d
127	120	3			171	23		511	466	11		511	229	77	$d \geq 77$
	113	5			163	25			457	13			220	79	
	106	7			155	27			448	15			211	83	
	99	9			147	29			439	17			202	85	$d \geq 85$
	92	11			139	31			430	19			193	87	
	85	13			131	37			421	21			184	91	
	78	15			123	39			412	23			175	93	#
	71	19			115	43			403	25			166	95	
	64	21			107	45			394	27			157	103	
	57	23			99	47			385	29			148	107	$d \geq 107$
	50	27			91	51			376	31			139	109	#
	43	29	#		87	53			367	35			130	111	
	36	31			79	55			358	37			121	117	#
	29	43			71	59	#		349	39			112	119	
	22	47			63	61	#		340	41			103	123	##
	15	55			55	63			331	43			94	125	#
	8	63			47	85			322	45			85	127	
255	247	3			45	87			313	47			76	171	
	239	5			37	91			304	51			67	175	
	231	7			29	95			295	53			58	183	
	223	9			21	111			286	55			49	187	
	215	11			13	119			277	57			40	191	
	207	13			9	127			268	59	$d \geq 59$		31	219	
	199	15		511	502	3			259	61	$d \geq 61$		28	223	
	191	17			493	5			250	63			19	239	
	187	19			484	7			241	73			10	255	
	179	21			475	9			238	75	$d \geq 75$				

Tab. 2.1: Parameter von primitiven BCH-Codes im engeren Sinne über $GF(2)$: Codelänge n , Dimension k, Bose-Entwurfsdistanz δ und wahre Minimaldistanz d. Ist der Spalteneintrag unter d leer, dann stimmt δ mit d überein. Die Symbole # und ## bezeichnen die Fälle $d = \delta + 2$ bzw. $d = \delta + 4$. Alle Einträge nach [MacWilliams et al. 1977],[Augot et al. 1992],[Wambach 1993],[Augot et al. 1994] und *P. Charpin* in [Pless et al. 1998], (S. 1013). Triviale Fälle sind nicht eingetragen.

Das Algebrasystem *MAGMA*, das von der „*Computational Algebra Group*“ des Instituts für Mathematik und Statistik der Universität Sydney (Australien) entwickelt wurde, beinhaltet eine umfassende Bibliothek von Befehlen rund um fehlerkorrigierende Codes. Es ist mit *MAGMA* möglich, Codes vieler bekannter Klassen der linearen Codes zu konstruieren. Dazu gehört neben der Konstruktion von *Hamming, Reed-Muller, Golay, Alternant* und *allgemeinen zyklischen Codes* die Erzeugung beliebiger BCH-Codes, bis auf Kapazitätsbeschränkung des Systems.

MAGMA ist in der Lage die systematischen Generatormatrizen von BCH-Codes auszugeben und nachträgliche Modifikationen an ihnen vorzunehmen, wie sie bei [MacWilliams et al. 1977], (S. 27ff), beschrieben sind. Neben weiteren interessanten Möglichkeiten, die dieses Algebrasystem bietet, ist es sogar möglich, Nachrichtenmengen zu codieren, die Übertragung durch einen gestörten Kanal zu simulieren und die fiktiv empfangenen Vektoren zu decodieren.
Allen Steel[2] implementierte zur Berechnung der Minimaldistanzen linearer Codes Algorithmen in *MAGMA* nach Ideen von *A. E. Brouwer*. Damit lassen sich die Minimaldistanzen von beliebigen BCH-Codes auf dem gleichen Computersystem, wie in Abschnitt 2.2.2 schon beschrieben, bis zu einer Länge von $n \leq 2^7 - 1$ in wenigen Sekunden berechnen. Schon ab einer Länge von $n \geq 2^8 - 1$ braucht *MAGMA* für die gleiche Aufgabe bis zu mehreren Tagen.

Gewöhnlich wird im nichttrivialen, binären, primitiven Fall im engeren Sinne vermutet, dass die Diskrepanz zwischen d und δ die ganze Zahl 4 nicht überschreitet (vgl. *P. Charpin* in [Pless et al. 1998], S. 1011). Wie leicht zu findende Beispiele von nicht primitiven BCH-Codes zeigen, kann im Allgemeinen der Unterschied größer sein. Beispielsweise berechnet *MAGMA* für den binären BCH-Code im engeren Sinne, der Länge $n = 43$ und Entwurfsdistanz $\delta = 5$ die Minimaldistanz $d = 13$.

Mit der Entwicklung ständig leistungsfähigerer Computersysteme haben „Brute-Force"-Methoden zum Auffinden der Minimaldistanzen von BCH-Codes eine größere Bedeutung gewonnen. „Brute-Force" meint die Erzeugung von Codewörtern eines Codes durch einem Algorithmus in systematischer Weise in Verbindung mit der Feststellung ihres Hamminggewichts. Wird ein Codewort, dessen Hamminggewicht mit einer unteren Schranke für die Minimaldistanz dieses Codes übereinstimmt, auf diese Weise gefunden, dann ist sein Gewicht mit der Minimaldistanz des Codes identisch.
„Brute-Force"-Methoden sind aus mathematischer Sicht in erster Linie nicht deshalb interessant, weil sie Wissenslücken in konkreten Fällen schließen, sondern weil man sich aufgrund der neu gewonnenen Beispiele bislang verschlossene Zusammenhänge aufzudecken erhofft.

[2] J. J. Cannon und C. Playoust: „An Introduction to Algebraic Programming with Magma (Draft)", Online-Handbuch (intro.dvi) S. 826, auch erschienen im Springer Verlag, Berlin 1997.

Da ein binärer, primitiver BCH-Code einer Länge $n \geq 2^9 - 1$ und Dimension $k \geq 148$ schon mehr als $3,5 \cdot 10^{44}$ Codewörter enthält, wird klar, warum „Brute-Force"-Algorithmen recht schnell an ihre Grenzen stoßen können. Das Problem bei der Entwicklung effektiver Algorithmen ist, die Systematik bei der Erzeugung von Codewörtern vermeintlich minimalem Gewichts darauf zu optimieren, möglichst wenige Codewörter erzeugen zu müssen.
Erst kürzlich wurde ein Algorithmus für die Minimaldistanzen linearer Codes vorgestellt [Canteaut et al. 1998], der von speziellen Strukturen der zu untersuchenden Codes unabhängig operiert und sich wahrscheinlichkeitstheoretischer Grundsätze bedient.
Im folgenden Abschnitt soll eine Möglichkeit zum Auffinden der Minimaldistanzen von BCH-Codes im engeren Sinne vorgestellt werden, die auf die Arbeiten [Augot et al. 1991] und [Augot et al. 1992] zurückgeht und Kenntnisse über die Struktur des betreffenden Codes und des ihm zugrundeliegenden Körpers verwendet. Der Kernstück dieser Methode ist das Lösen bestimmter algebraischer Gleichungen, die man *Newton Identitäten* nennt. Auf diese Weise lassen sich Codewörter jedes im Code existierenden Gewichts finden.

2.4. Finden minimalgewichtiger Codewörter durch Newton Identitäten

2.4.1. Zur Theorie

Definition 2.5
Es sei $v = (v_0, \ldots, v_{n-1}) \in GF(q)^n$ ein Vektor von Gewicht $\mathrm{wt}(v) = \omega$ und den von Null verschiedenen Komponenten

$$v_{l_1}, v_{l_2}, \ldots, v_{l_\omega}$$

Die *Lokatoren* $X_1, \ldots, X_\omega$ von v seien definiert durch

$$X_1 := \alpha^{l_1}, X_2 := \alpha^{l_2}, \ldots, X_\omega := \alpha^{l_\omega} \ ,$$

wobei $\alpha \in GF(q^m)$ eine primitive n-te Einheitswurzel bezeichne.

Bemerkung 2.6
Ein Vektor $v \in GF(q)^n$ wird also eindeutig beschrieben durch die Paare

$$(X_1, v_{l_1}), (X_2, v_{l_2}), \ldots, (X_\omega, v_{l_\omega}) \ .$$

Im binären Fall $q = 2$ sind offensichtlich die $v_{l_i} = 1$ für alle $1 \leq i \leq \omega$, weshalb ein binärer Vektor allein durch seine Lokatoren $X_i \in GF(2^m)$ eindeutig bestimmt ist.

Definition 2.7
Das *Lokator-Polynom* eines Vektors $v \in GF(q)^n$ vom Gewicht $\mathrm{wt}(v) = \omega$ sei definiert durch

$$\sigma_v(Z) := \prod_{i=1}^{\omega} (1 - X_i Z) \ .$$

Bemerkung 2.8
Die Nullstellen des Lokator-Polynoms $\sigma_v(Z)$ sind die zu den Lokatoren $X_i = \alpha^{l_i} \in E_n \subseteq GF(q^m)$ reziproken n-ten Einheitswurzeln $1/X_i = \alpha^{n-l_i} \in E_n$.

Lemma 2.9
Für das Lokator-Polynom eines Vektors $v \in GF(q)^n$ gilt:

$$\sigma_v(Z) = \sum_{j=0}^{\omega} \sigma_j Z^j \ ,$$

wobei $\sigma_0 := 1$ und für $1 \leq j \leq \omega$ die Koeffizienten σ_j die elementar-symmetrischen Funktionen der Lokatoren $X_1, \ldots, X_\omega$ sind:

$$\sigma_j := (-1)^j \sum_{1 \leq i_1 < \ldots < i_j \leq \omega} X_{i_1} \cdot \ldots \cdot X_{i_j} \ .$$

Beweis. (Durch vollständige Induktion von ω nach $\omega + 1$.) Sei $\omega = 1$, dann ist $\sigma_v(Z) = 1 - X_1 Z$ mit $\sigma_0 = 1$ und $\sigma_1 = -X_1$. Angenommen die Aussage gelte für ω und es seien für $1 \leq i \leq \omega$

$$\sigma_j' = (-1)^j \sum_{1 \leq i_1 < \ldots < i_j \leq \omega} X_{i_1} \cdot \ldots \cdot X_{i_j}$$

und $\sigma_0' = 1$ die elementar-symmetrischen Funktionen der $X_1, \ldots, X_\omega$. Definiere

für $0 \leq j \leq \omega$

$$-\sigma_j^{''} := -X_{\omega+1} \cdot \sigma_j^{'} = (-1)^{j+1} \sum_{1 \leq i_1 < \ldots < i_j \leq \omega} X_{i_1} \cdot \ldots \cdot X_{i_j} \cdot X_{\omega+1} \ .$$

Dann ist für $0 \leq j \leq \omega - 1$

$$\sigma_{j+1} := \sigma_{j+1}^{'} - \sigma_j^{''} = (-1)^{j+1} \sum_{1 \leq i_1 < \ldots < i_{j+1} \leq \omega+1} X_{i_1} \cdot \ldots \cdot X_{i_{j+1}} \ ,$$

und es wird definiert $\sigma_0 := \sigma_0^{'} = 1$ und $\sigma_{\omega+1} := -\sigma_\omega^{''} = (-1)^{\omega+1} X_1 \cdot \ldots \cdot X_{\omega+1}$. Es folgt für $\omega + 1$:

$$\begin{aligned}
\prod_{i=1}^{\omega+1}(1 - X_i Z) &= (1 - X_{\omega+1}Z) \cdot \prod_{i=1}^{\omega}(1 - X_i Z) \\
&= \sigma_0^{'} + \sum_{j=1}^{\omega} \sigma_j^{'} Z^j - X_{\omega+1} Z \sum_{j=0}^{\omega} \sigma_j^{'} Z^j \\
&= \sigma_0^{'} + \sum_{j=0}^{\omega-1} \sigma_{j+1}^{'} Z^{j+1} - \sum_{j=0}^{\omega-1} \sigma_j^{''} Z^{j+1} - \sigma_\omega^{''} Z^{\omega+1} \\
&= \sigma_0^{'} + \sum_{j=0}^{\omega-1} (\sigma_{j+1}^{'} - \sigma_j^{''}) Z^{j+1} - \sigma_\omega^{''} Z^{\omega+1} \\
&= \sigma_0 + \sum_{j=1}^{\omega} \sigma_j Z^j + \sigma_{\omega+1} Z^{j+1} = \sum_{j=0}^{\omega+1} \sigma_j Z^j \ ,
\end{aligned}$$

wobei $\sigma_0, \ldots, \sigma_{\omega+1}$ die elementar-symmetrischen Funktionen der $X_1, \ldots, X_{\omega+1}$ darstellen. ■

Definition 2.10

Das *Mattson-Solomon Polynom* (kurz: *MS-Polynom*) eines Vektors $v = (v_0, \ldots, v_{n-1}) \in GF(q)^n$ ist definiert als das Polynom $A(Z) \in GF(q^m)[Z]$:

$$\begin{aligned}
A(Z) &:= \sum_{j=1}^{n} A_j Z^{n-j} \ , \text{ wobei} \\
A_j &:= \sum_{i=0}^{n-1} v_i \alpha^{ij} \ , \ j \in \{0, 1, 2, \ldots\} \ .
\end{aligned}$$

Bemerkung 2.11
Die Koeffizienten des MS-Polynoms zu einem Vektor v sind gegeben durch Multiplikation von v mit der n-ten *Fouriermatrix* über $GF(q^m)$:

$$\begin{pmatrix} A_0 \\ A_1 \\ A_2 \\ \vdots \\ A_{n-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \alpha & \alpha^2 & \dots & \alpha^{n-1} \\ 1 & \alpha^2 & \alpha^4 & \dots & \alpha^{(n-1)2} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & \alpha^{n-1} & \alpha^{2(n-1)} & \dots & \alpha^{(n-1)^2} \end{pmatrix} \begin{pmatrix} v_0 \\ v_1 \\ v_2 \\ \vdots \\ v_{n-1} \end{pmatrix} .$$

Deswegen wird das MS-Polynom zu einem Vektor v auch die *diskrete Fouriertransformation* von v genannt. Zwischen den Koeffizienten des MS-Polynoms des Vektors $v \in GF(q)^n$ mit Null verschiedenen Komponenten $v_{l_1}, \dots, v_{l_\omega}$ und den Lokatoren von v gibt es einen Zusammenhang, der durch die polynomiale Repräsentation von v gegeben ist:

$$v(\alpha^j) = A_j = \sum_{i=1}^{\omega} v_{l_i} X_i^j \ ,$$

insbesondere im binären Fall: $A_j = \sum_{i=1}^{\omega} X_i^j$.

Lemma 2.12

(i). Es sei $\mathcal{C}$ ein beliebiger BCH-Code im engeren Sinne, der Länge n und Entwurfsdistanz δ und $A(Z)$ das MS-Polynom zu einem beliebigen Codewort $c \in \mathcal{C}$. Dann gilt: $A_j = 0 \ \forall \ 1 \leq j \leq \delta - 1 \Leftrightarrow v \in C$ und wenn $\delta = \delta_{Bose}$ eine Bose-Entwurfsdistanz ist: $A_{\delta_{Bose}} \neq 0$.

(ii). Es sei $\alpha \in E_n$ primitive n-te Einheitswurzel und $\mathcal{C}$ ein BCH-Code der Länge n über $GF(q)$. Für jedes Codewort $c \in \mathcal{C}$ mit $\mathrm{wt}(c) = \omega$ und allen $0 \leq j \leq n-1$ gilt: $A_{qj+n} = A_{j \bmod n}^q$.

Beweis. Da die Standard-Kontrollmatrix von $\mathcal{C}$ (siehe Bemerkung 1.52) den auf die erste Zeile der n-ten Fouriermatrix folgenden $\delta - 1$ Zeilen (von n-ten Einheitswurzeln) entspricht, folgt unmittelbar $A_0, \dots, A_{\delta-1} = 0$ genau dann, wenn c ein Codewort von $\mathcal{C}$ ist. Angenommen $\mathcal{C}$ ist der BCH-Code im engeren Sinne zur Bose-Entwurfsdistanz δ_{Bose} und es gilt $A_{\delta_{Bose}} = 0$. Dann kann man der zu $\mathcal{C}$ zugehörigen Kontrollmatrix die „δ_{Bose}"-Zeile der n-ten Fouriermatrix hinzufügen, so dass die Matrix weiterhin Kontrollmatrix des Codes bleibt. Dann existiert aber

ein $\delta' \geq \delta_{Bose}+1$, dass den gleichen BCH-Code definiert. Dies ist aber ein Widerspruch dazu, dass δ_{Bose} die größte Entwurfsdistanz einer konsekutiven Folge von Entwurfsdistanzen $\delta_1 < \delta_2 < \ldots < \delta_{Bose}$ ist, die den gleichen Code definieren. Also folgt (i). Wegen Lemma 1.43 und $\alpha \in E_n$ folgt unmittelbar (ii):

$$\begin{aligned} A_{jq+n} &= \sum_{i=0}^{\omega} v_i(\alpha^i)^{qj+n} = \sum_{i=0}^{\omega} v_i(\alpha^i)^{qj}(1)^i \\ &= \left(\sum_{i=0}^{\omega} v_i(\alpha^i)^j\right)^q = A_{j \bmod n}^q \ . \end{aligned}$$

■

Satz 2.13 (verallgemeinerte Newton-Identitäten)
Für alle r genügen die Koeffizienten des MS- und des Lokator-Polynoms zu einem Vektor $v \in GF(q)^n$ mit Gewicht $\mathrm{wt}(v) = \omega$ der Gleichung

$$A_{r+\omega} + \sigma_1 A_{r+\omega-1} + \ldots + \sigma_\omega A_r = 0 \ .$$

Beweis. Nach Bemerkung 2.8 und Lemma 2.9 gilt die Gleichung

$$\sigma(1/X_i) = \sum_{j=0}^{\omega} \sigma_j \cdot 1/(X_i)^j = 0 \quad \forall\ 1 \leq i \leq \omega \ . \tag{2.4}$$

Multiplizieren von Gleichung (2.4) mit $v_{l_i} X_i^{r+\omega}$ liefert:

$$\sum_{j=0}^{\omega} \sigma_j \cdot v_{l_i} X_i^{\omega+r-j} = 0 \ . \tag{2.5}$$

Anschließende Summation der linken Seite von Gleichung (2.5) über $i = 1, ..., \omega$ ergibt:

$$\begin{aligned} 0 = \sum_{i=1}^{\omega}\sum_{j=0}^{\omega} \sigma_j \cdot v_{l_i} X_i^{\omega+r-j} &= \sum_{j=0}^{\omega}\left(\sigma_j \cdot \sum_{i=1}^{\omega} v_{l_i} X_i^{\omega+r-j}\right) \\ &= \sum_{j=0}^{\omega} \sigma_j \cdot A_{\omega+r-j} \ , \end{aligned}$$

die Behauptung. ■

Die Aussage von Satz 2.13 wird z.B. im binären Fall „handlicher“, wenn man folgende Fallunterscheidung trifft:

Lemma 2.14
Es sei $v \in GF(q)^n$ ein Vektor von Gewicht $\mathrm{wt}(v) = \omega$, *dessen Komponenten ausschließlich „0“ oder „1“ sind, und $A(Z)$ sowie $\sigma_v(Z)$ das MS- bzw. Lokatorpolynom von v. Dann definiert*

$$I_r : \begin{cases} A_r + \sum_{i=1}^{r-1} A_{r-i}\sigma_i + r\sigma_r = 0, & \text{für } r \leq \omega \\ A_r + \sum_{i=1}^{\omega} A_{r-i}\sigma_i = 0, & \text{für } r > \omega \end{cases}$$

die r-te gewöhnliche Newton-Identität.

Beweis. Es ist $A_j = \sum_{r=1}^{\omega} X_r^j \in GF(q^m)$. Definiere zum beliebigen Vektor v die formale Potenzreihe $P(Z) := \sum_{j=1}^{\infty} A_j Z^j \in (GF(q^m)[[Z]],\ +,\ \cdot)$ des Rings der formalen Potenzreihen[3] in der Unbestimmten Z mit Addition

$$\sum_{j=0}^{\infty} a_j Z^j + \sum_{j=0}^{\infty} b_j Z^j := \sum_{j=1}^{\infty} (a_j + b_j) Z^j$$

und Multiplikation

$$\sum_{s=0}^{\infty} a_s Z^s \cdot \sum_{t=0}^{\infty} b_t Z^t := \sum_{j=0}^{\infty} \left(\sum_{s+t=j} a_s b_t \right) Z^j \ .$$

Es gilt für $1 \leq r \leq \omega$ und $1 - X_r Z$, aufgefasst als formale Reihe $\sum_{s=0}^{\infty} a_s Z^s$ in Z mit Koeffizienten $a_0 = 1, a_1 = -X_r$ und $a_2, a_3, \ldots = 0$:

$$\begin{aligned} (1 - X_r Z) \sum_{j=1}^{\infty} (X_r Z)^j &= \sum_{j=1}^{\infty} (X_r Z)^j - X_r Z \sum_{j=1}^{\infty} (X_r Z)^j \\ &= X_r Z + \sum_{j=2}^{\infty} (X_r Z)^j - \sum_{j=1}^{\infty} (X_r Z)^{j+1} \end{aligned}$$

[3] Der Ring der formalen Potenzreihen $GF(q^m)[[Z]]$ mit Koeffizienten in $GF(q^m)$ ist eine Verallgemeinerung des Polynomrings $GF(q^m)[Z]$. Eine wichtige Eigenschaft ist, dass $\sum_{s=0}^{\infty} a_s Z^s$ eine Einheit in $GF(q^m)[[Z]]$ genau dann ist, wenn s_0 eine Einheit von $GF(q^m)$ ist. Vergleiche dazu [Bosch 1993], S. 31.

$$\begin{aligned} &= X_r Z + \sum_{j=1}^{\infty} (X_r Z)^{j+1} - \sum_{j=1}^{\infty} (X_r Z)^{j+1} \\ &= X_r Z \end{aligned}$$

$$\iff \qquad \sum_{j=1}^{\infty} (X_r Z)^j = \frac{X_r Z}{1 - X_r Z} \ , \tag{2.6}$$

weil $1 - X_r Z$ eine Einheit in $GF(q^m)[[Z]]$ ist. Deshalb folgt mit Gleichung (2.6) für $P(Z)$:

$$\begin{aligned} P(Z) = \sum_{j=1}^{\infty} A_j Z^j &= \sum_{j=1}^{\infty} \left(\sum_{r=1}^{\omega} X_r^j \right) Z^j \\ &= \sum_{r=1}^{\omega} \sum_{j=1}^{\infty} (X_r Z)^j \\ &= \sum_{r=1}^{\omega} \frac{X_r Z}{1 - X_r Z} \ . \end{aligned} \tag{2.7}$$

Nach Definition ist $\sigma_v(Z) = \prod_{i=1}^{\omega} (1 - X_i Z)$, weshalb die hier als bekannt vorausgesetzte *Ableitung* $(\sigma_v(Z))'$ von $\sigma_v(Z)$ *nach* Z zusammen mit der *Produktregel* die folgende Gleichung erzeugt:

$$Z \cdot (\sigma_v(Z))' = - \sum_{r=1}^{\omega} \left(X_r Z \cdot \prod_{i=1, i \neq r}^{\omega} (1 - X_i Z) \right) \ . \tag{2.8}$$

Die Definition von $\sigma_v(Z)$ und (2.7) liefern:

$$\sigma_v(Z) \cdot P(Z) = \sum_{r=1}^{\omega} \left(X_r Z \cdot \prod_{i=1, i \neq r}^{\omega} (1 - X_i Z) \right) \ . \tag{2.9}$$

Aus den Gleichungen (2.8) und (2.9) folgt sodann

$$\sigma_v(Z) \cdot P(Z) + Z \cdot (\sigma_v(Z))' = 0 \ . \tag{2.10}$$

Umordnen und Zusammenfassen der linken Seite von (2.10) liefert dann die Behauptung, in dem man für $\sigma_v(Z)$ die polynomiale Schreibweise von Lemma 2.9

(als Reihe aufgefasst mit $\sigma_i = 0 \ \forall \ i > \omega$) und die Reihendarstellung von $P(Z)$ verwendet:

$$\begin{aligned}
0 &= \sigma_v(Z) \cdot P(Z) + Z \cdot (\sigma_v(Z))' \\
&= \sum_{j=0}^{\omega} \sigma_j Z^j \cdot \sum_{j=1}^{\infty} A_j Z^j + \sum_{j=1}^{\omega} j\sigma_j Z^j \\
&= \sum_{j=1}^{\infty} A_j Z^j + \sigma_1 Z \cdot \sum_{j=1}^{\infty} A_j Z^j + \ldots + \sigma_\omega Z^\omega \cdot \sum_{j=1}^{\infty} A_j Z^j + \sum_{j=1}^{\omega} j\sigma_j Z^j \\
&= \sum_{j=1}^{\infty} A_j Z^j + \sigma_1 \cdot \sum_{j=1}^{\infty} A_j Z^{j+1} + \ldots + \sigma_\omega \cdot \sum_{j=1}^{\infty} A_j Z^{j+\omega} + \sum_{j=1}^{\omega} j\sigma_j Z^j \\
&= \sum_{j=1}^{\infty} A_j Z^j + \sigma_1 \cdot \sum_{j=2}^{\infty} A_{j-1} Z^j + \ldots + \sigma_\omega \cdot \sum_{j=\omega+1}^{\infty} A_{j-\omega} Z^j + \sum_{j=1}^{\omega} j\sigma_j Z^j \\
&= \sum_{j=1}^{\omega} \left(A_j + \sum_{i=1}^{j-1} A_{j-i}\sigma_i + j\sigma_j \right) Z^j + \sum_{j=\omega+1}^{\infty} \left(A_j + \sum_{i=1}^{j-1} A_{j-i}\sigma_i \right) Z^j \ .
\end{aligned}$$

■

Folgerung 2.15
Es sei $GF(q^m)$ ein Körper der Charakteristik p und s fest gewählt. Dann gilt für die Koeffizienten von MS- und Lokator-Polynom zu einem Vektor v von Gewicht $\mathrm{wt}(v) = \omega$ dessen Einträge nur „0“ oder „1“ sind:

$$A_l = 0 \ \forall \ 1 \leq l \leq s \Leftrightarrow \sigma_l = 0 \text{ für alle } 1 \leq l \leq s \text{ und } p \nmid l \ .$$

Beweis. Ohne Einschränkung der Allgemeinheit kann man wegen der Definition von $\sigma_v(Z)$ annehmen, dass $s \leq \omega$. Es folgt wegen Lemma 2.14 für alle $1 \leq j \leq s$ mit $A_j = 0$, dass $j\sigma_j = 0$. Dies ist aber genau dann der Fall, wenn $j \equiv 0 \bmod p$ oder $\sigma_j = 0$. Mit der gleichen Argumentation wie in Satz 1.43 ist für $p \nmid j$ jedoch $j \bmod p \neq 0$ und daher $\sigma_j = 0$. Umgekehrt folgt wegen Lemma 2.14 mit $\sigma_1 = 0$, dass $A_1 = 0$. Dann ist aber auch wegen $A_2 + 2\sigma_2 = 0$ der Koeffizient $A_2 = 0$, denn entweder ist $GF(q^m)$ ein Körper der Charakteristik 2 oder $p \nmid 2$, so dass in jedem der beiden Fälle $2\sigma_2 = 0$. Genauso folgt sukzessive: $A_3, \ldots, A_s = 0$. ■

Folgerung 2.16
Es sei $\sigma_c(Z) = \sum_{j=0}^{\omega} \sigma_j Z^j$ ein beliebiges Polynom über $GF(q^m)$. Dann ist $\sigma_c(Z)$ das Lokator-Polynom eines Codeworts c, dessen Komponenten ausschließlich „0“ oder „1“ sind, eines BCH-Codes $\mathcal{C}$ im engeren Sinne der Entwurfsdistanz δ über $GF(q)$ genau dann, wenn die folgenden beiden Bedingungen gehalten werden:

(i). Die Nullstellen von $\sigma_v(Z)$ sind paarweise verschiedene n-te Einheitswurzeln.

(ii). Es gilt $\sigma_i = 0$ für alle i im Bereich von $1 \leq i \leq \delta - 1$, die nicht durch die Charakteristik von $GF(q^m)$ geteilt werden.

Beweis. Wenn $\sigma_c(Z)$ das Lokator-Polynom eines Codeworts $c \in \mathcal{C}$ ist, dann gilt für die Koeffizienten des zu c gehörigen MS-Polynoms im Bereich von $1 \leq j \leq \delta - 1$, dass $A_j = 0$ (vgl. Bemerkung 2.12). Bedingung (ii) ist dann eine Folge von Folgerung 2.15 und Bedingung (i) entspricht Bemerkung 2.8.
Umgekehrt sollen die Bedingungen (i) und (ii) gelten und es seien $X_1, \ldots, X_\omega$ die reziproken n-ten Einheitswurzeln zu den Nullstellen von $\sigma_c(Z)$. Dann gilt für den Vektor c mit den Einträgen „1“ in diesen Positionen wieder wegen Folgerung 2.15 $A_j = 0$ für alle $1 \leq j \leq \delta - 1$, weshalb c mit Bemerkung 2.12 ein Codewort eines BCH-Codes im engeren Sinne der Entwurfsdistanz δ über $GF(q)$ ist. ■

2.4.2. Zum Verfahren

Im folgenden sei $\mathcal{C}$ ein BCH-Code im engeren Sinne der Länge n und Entwurfsdistanz δ über GF(2) und $c \in C$ ein Codewort vom Gewicht ω. Vom vorhergehenden Abschnitt können folgende Tatsachen zusammengefasst werden:

> Es seien $X_1, \ldots, X_\omega$ die Lokatoren von c und $A_0, \ldots, A_{n-1}$ die Koeffizienten des MS-Polynoms von c. Dann ist $A_j = 0 \; \forall \; 1 \leq j \leq \delta - 1$.
>
> Das Lokator-Polynom $\sigma_c(Z)$ von c zerfällt über $GF(2^m)$ vollständig in paarweise verschiedene Linearfaktoren.
>
> Die Koeffizienten des MS-Polynoms und des Lokatorpolynoms von c stehen durch die gewöhnlichen Newton Identitäten zueinander in Beziehung.

Um herauszufinden, ob ein binärer BCH-Code $\mathcal{C}$ ein Codewort vom Gewicht ω besitzt, wird wie folgt vorgegangen:

(i). Es wird angenommen es gäbe ein Codewort $c \in \mathcal{C}$ mit $\mathrm{wt}(c) = \omega$ und die gewöhnlichen Newton Identitäten werden für die A_i und σ_i unter Verwendung der in Lemmata 2.12 und 2.16 bewiesenen Eigenschaften aufgestellt.

(ii). Die gewonnenen Gleichungen werden in Termen nach σ_i vereinfacht.

(iii). Sind die Gleichungen von (ii) widerspruchsfrei, dann wird das Lokatorpolynom des vermeintlichen Codeworts konstruiert. Abschließend wird geprüft, ob dieses über $GF(2^m)$ vollständig in paarweise verschiedene Linearfaktoren zerfällt und seine Nullstellen in E_n liegen.

Auf diese Weise sind zwei exklusive Ergebnisse möglich. Entweder es wird ein Lokator-Polynom gefunden, das alle Schritte widerspruchsfrei hält, dann gibt es in $\mathcal{C}$ ein Codewort von Gewicht $\mathrm{wt}(c) = \omega$, das sich aus seinem Lokatorpolynom rekonstruieren lässt. Oder es tritt zu einem der genannten Schritte ein Widerspruch auf, dann gibt es kein Codewort vom Gewicht $\mathrm{wt}(c) = \omega$ in $\mathcal{C}$.
Um die wahre Minimaldistanz eines BCH-Codes zu bestimmen, wird mit der Entwurfsdistanz des Codes oder einer anderen Schranke für zyklische Codes (siehe [van Lint et al. 1986]) als „Startgewicht“ begonnen. Die Schritte werden iterativ durch das Erhöhen des Gewichts solange durchgeführt, bis eine Lösung gefunden wird.

Beispiel 2.17
Der binäre BCH-Code $\mathcal{C}$ im engeren Sinne der Länge $n = 17$ und der Entwurfsdistanz $\delta = 3$ besitzt die wahre Minimaldistanz $d = 5$. Der Beweis dieser Aussage erfolgt mit Hilfe der gewöhnlichen Newton-Identitäten.

Angenommen es gäbe ein Codewort $c \in \mathcal{C}$ vom Gewicht $\mathrm{wt}(c) = \delta = 3$. Die zyklotomischen Nebenklassen von 2 mod 17 sind:

$$\begin{aligned} C_0 &= \{0\} \\ C_1 &= \{1, 2, 4, 8, 16, 15, 13, 9\} \\ C_3 &= \{3, 6, 12, 7, 14, 11, 5, 10\} \, . \end{aligned}$$

Also ist $\delta = 3$ eine Bose-Distanz, da es keine größere Entwurfsdistanz gibt, die lediglich die zyklotomische Nebenklasse C_1 für die Varietät des Generatorpolynoms definiert. Dann kann man für die Koeffizienten des MS-Polynoms von c nach Lemma 2.12 annehmen, dass $A_1, A_2, A_4, A_8, A_{16}, A_{15}, A_{13}, A_9 = 0$ und $A_3 \neq 0$. Außerdem ist gewiss, dass

$$\begin{array}{llllllll} A_6 = & A_3^2 & A_{12} = & A_3^4 & A_7 = & A_3^8 & A_{14} = & A_3^{16} \\ A_{11} = & A_3^{32} & A_5 = & A_3^{64} & A_{10} = & A_3^{128} & & \end{array}$$

und $\sigma_1 = 0$. Die Newton-Identitäten $r = 3$, $r = 5$ und $r = 11$ für c lauten:

$$\begin{array}{lcrcrcl} I_3 & : & A_3 + \sigma_3 & = 0 & \Longleftrightarrow & \sigma_3 & = A_3 \\ I_5 & : & A_5 + A_3\sigma_2 & = 0 & \Longleftrightarrow & \sigma_2 & = A_3^{63} \\ I_{11} & : & A_{11} & = 0 & \Longleftrightarrow & 0 & = A_3^{32} \quad . \end{array}$$

Die Newton-Identität I_{11} liefert einen Widerspruch zur Annahme, dass $A_3 \neq 0$. Daher kann es in $\mathcal{C}$ kein Codewort vom Gewicht $\mathrm{wt}(c) = 3$ geben.

Angenommen, es gäbe ein Codewort $c \in \mathcal{C}$ mit $\mathrm{wt}(c) = 4$. Die Annahmen von gerade bleiben die Selben, weil sie im Wesentlichen von δ abhängen und nicht von ω. Zusätzlich wird berücksichtigt, dass $A_0 = 1+1+1+1 = 4 \equiv 0 \bmod 2$ ist. Dann liefert die 17. Newton-Identität:

$$I_{17}: \quad A_0 + A_{14}\sigma_3 \quad = 0 \quad \Longleftrightarrow \quad 0 \quad = A_3^{16} \ ,$$

denn $A_0 = \omega = 4$ weil wir über $GF(2^8)$ rechnen, so dass $4 \bmod 2 \equiv 0$. Demnach gibt es in $\mathcal{C}$ auch kein Codewort c von Gewicht $\mathrm{wt}(c) = 4$.

Es wird nun angenommen es existiert in $\mathcal{C}$ ein Codewort $c \in C$ von Gewicht $\mathrm{wt}(c) = 5$. Die Newton-Identitäten für $r = 3$, $r = 13$, $r = 5$ und $r = 7$ zeigen unter den gleichen Annahmen wie sonst:

$$\begin{array}{lcrcrcl} I_3 & : & A_3 + \sigma_3 & = 0 & \Longleftrightarrow & \sigma_3 & = A_3 \\ I_{13} & : & A_{11}\sigma_2 + A_{10}\sigma_3 & = 0 & \Longleftrightarrow & \sigma_2 & = A_3^{97} \\ I_5 & : & A_5 + A_3\sigma_2 + \sigma_5 & = 0 & \Longleftrightarrow & \sigma_5 & = A_3^{64} + A_3^{98} \\ I_7 & : & A_7 + A_5\sigma_2 + A_3\sigma_4 & = 0 & \Longleftrightarrow & \sigma_4 & = A_3^{7} + A_3^{160} \quad . \end{array}$$

Der MS-Koeffizient A_3 und das Lokator-Polynom $\sigma_c(Z)$ sind dann von der Form:

$$A_3 = X_1^3 + X_2^3 + X_3^3 + X_4^3 + X_5^3 \tag{2.11}$$

$$\sigma_c(Z) = 1 + A_3^{97}Z^2 + A_3Z^3 + (A_3^7 + A_3^{160})Z^4 + (A_3^7 + A_3^{160})Z^5 \tag{2.12}$$

Gesucht werden nun Nullstellen $1/X_1, 1/X_2, 1/X_3, 1/X_4, 1/X_5 \in E_{17}$ für die Gleichung $\sigma_c(Z) = 0$, denn nach Folgerung 2.16 zerfällt das Lokatorpolynom eines Codeworts in ω verschiedene Linearfaktoren über dem Zerfällungskörper von $Z^n - 1$. Die Lokatoren eines Codeworts von Gewicht $\mathrm{wt}(c) = 5$ sind, nach Bemerkung 2.8, die zu den Nullstellen reziproken Elemente $X_1, X_2, X_3, X_4, X_5 \in E_{17}$.
Ein bequemer Weg die Nullstellen von $\sigma_c(Z)$ zu finden, ist die „Exhaustive Search“ mit einem symbolischen Algebrasystem, das in der Lage ist mit Polynomen über endlichen Körpern von Primzahlpotenzordnung zu rechnen. Damit ist die umfassende Suche nach allen fünfelementigen Mengen $\{\alpha^{j_1}, \alpha^{j_2}, \alpha^{j_3}, \alpha^{j_4}, \alpha^{j_5}\} \subseteq E_{17}$ gemeint, mit primitiven Element α und $0 \leq j_l \leq n-1$, für die $\sigma_c(Z)$ durch Einsetzen der α^{j_l} in (2.11) und (2.12) vollständig über $GF(2^8)$ in Linearfaktoren zerfällt.
E_{17} besitzt $\binom{17}{5} = 6188$ verschiedene fünfelementige Teilmengen, die nacheinander zur Lösung abgearbeitet werden müssen. Für diese Aufgabe wurde das Algebrasystem *MAGMA* verwendet (vgl. Anhang C). Der Zerfällungskörper $GF(2^8)$ wurde von *MAGMA* durch das primitive Polynom $P(X) = X^8 + X^4 + X^3 + X^2 + 1$ mit primitiver Nullstelle $P(\beta) = 0$ erzeugt. Die multiplikative Gruppe der 17. Einheitswurzeln E_{17} wurde durch das primitive Element $\alpha = \beta^{15}$ generiert. Innerhalb weniger Sekunden fand die „Exhaustive Search“-Implementierung in *MAGMA* insgesamt 34 verschiedene Lokatoren-5-Tupel, die den genannten Forderungen genügen (Fortsetzung auf der nächsten Seite):

$(\alpha^5, \alpha^6, \alpha^9, \alpha^{10}, \alpha^{16})$	$(\alpha^0, \alpha^7, \alpha^9, \alpha^{12}, \alpha^{15})$	$(\alpha^1, \alpha^3, \alpha^{10}, \alpha^{12}, \alpha^{15})$
$(\alpha^6, \alpha^8, \alpha^{11}, \alpha^{14}, \alpha^{16})$	$(\alpha^0, \alpha^2, \alpha^5, \alpha^8, \alpha^{10})$	$(\alpha^1, \alpha^4, \alpha^6, \alpha^{13}, \alpha^{15})$
$(\alpha^3, \alpha^5, \alpha^8, \alpha^{11}, \alpha^{13})$	$(\alpha^3, \alpha^9, \alpha^{10}, \alpha^{13}, \alpha^{14})$	$(\alpha^0, \alpha^3, \alpha^4, \alpha^{10}, \alpha^{16})$
$(\alpha^1, \alpha^2, \alpha^5, \alpha^6, \alpha^{12})$	$(\alpha^0, \alpha^1, \alpha^7, \alpha^{13}, \alpha^{14})$	$(\alpha^1, \alpha^4, \alpha^7, \alpha^9, \alpha^{16})$
$(\alpha^5, \alpha^{11}, \alpha^{12}, \alpha^{15}, \alpha^{16})$	$(\alpha^0, \alpha^6, \alpha^{12}, \alpha^{13}, \alpha^{16})$	$(\alpha^2, \alpha^3, \alpha^9, \alpha^{15}, \alpha^{16})$
$(\alpha^1, \alpha^7, \alpha^8, \alpha^{11}, \alpha^{12})$	$(\alpha^2, \alpha^5, \alpha^7, \alpha^{14}, \alpha^{16})$	$(\alpha^1, \alpha^6, \alpha^7, \alpha^{10}, \alpha^{12})$
$(\alpha^0, \alpha^3, \alpha^5, \alpha^{12}, \alpha^{14})$	$(\alpha^2, \alpha^3, \alpha^6, \alpha^7, \alpha^{13})$	$(\alpha^2, \alpha^4, \alpha^7, \alpha^{10}, \alpha^{12})$
$(\alpha^0, \alpha^3, \alpha^6, \alpha^8, \alpha^{15})$	$(\alpha^3, \alpha^4, \alpha^7, \alpha^8, \alpha^{14})$	$(\alpha^4, \alpha^6, \alpha^9, \alpha^{12}, \alpha^{14})$

$(\alpha^5, \alpha^7, \alpha^{10}, \alpha^{13}, \alpha^{15})$ $(\alpha^4, \alpha^{10}, \alpha^{11}, \alpha^{14}, \alpha^{15})$ $(\alpha^2, \alpha^4, \alpha^{11}, \alpha^{13}, \alpha^{16})$
$(\alpha^1, \alpha^8, \alpha^{10}, \alpha^{13}, \alpha^{16})$ $(\alpha^2, \alpha^8, \alpha^9, \alpha^{12}, \alpha^{13})$ $(\alpha^1, \alpha^2, \alpha^8, \alpha^{14}, \alpha^{15})$
$(\alpha^4, \alpha^5, \alpha^8, \alpha^9, \alpha^{15})$ $(\alpha^0, \alpha^2, \alpha^9, \alpha^{11}, \alpha^{14})$ $(\alpha^1, \alpha^3, \alpha^6, \alpha^9, \alpha^{11})$
$(\alpha^0, \alpha^1, \alpha^4, \alpha^5, \alpha^{11})$

Jedes 5-Tupel entspricht einem Codewort $c \in \mathcal{C}$. Beispielsweise repräsentiert das Tupel $(\alpha^5, \alpha^6, \alpha^9, \alpha^{10}, \alpha^{16})$ das Codewort $(0\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 1) \in \mathcal{C}$. Also besitzt der BCH-Code $\mathcal{C}$ im engeren Sinne, der Länge $n = 17$ und Entwurfsdistanz $\delta = 3$ insgesamt 34 verschiedene Codewörter c vom Gewicht $\mathrm{wt}(c) = 5$ und keine Codewörter von geringeren Gewicht. Die Minimaldistanz von $\mathcal{C}$ ist also $d = 5$.

Das Verfahren kann in einem Algebrasystem so implementiert werden, dass der Algorithmus das Aufstellen und Vereinfachen der gewöhnlichen Newton-Identitäten mit übernimmt. Jedoch bemerkten die Autoren von [Augot et al. 1992], dass es schwierig sei einen effektiven Algorithmus ohne die Notwendigkeit eines Benutzereingriffs zu implementieren.
Die Suche der Minimaldistanz eines Codes kann noch wesentlich effektiver als hier dargestellt gestaltet werden, wenn man nach *idempotenten Erzeugern* eines BCH-Codes bzw. zyklischen Codes sucht. Man vergleiche dazu [Betten et al. 1998], (S. 96f), und [Augot et al. 1994].

3. Anwendungen von BCH-Codes

Im Folgenden sollen einige Anwendungen von *Reed-Solomon Codes* (kurz: RS-Codes), einem wichtigen Spezialfall von BCH-Codes, vorgestellt werden. Wie schon in Abschnitt 1.3.2 erwähnt, sind RS-Codes nichts anderes als BCH-Codes mit Codelängen $n = q - 1$ über $GF(q)$.

3.1. Reed-Solomon Codes

Die Geschichte der RS-Codes ist eng mit der Entwicklung der Kanalcodierung und der Block-Codes verbunden. 1960 stellten *R. Reed* und *G. Solomon* in ihrer Arbeit *„Polynomial Codes over certain finite fields“* [Reed & Solomon 1960] eine Vorform von dem vor, was heute unter RS-Codes verstanden wird. Sie beschrieben RS-Codes als Mengen algebraischer Kurven, definiert durch Polynome beschränkten Grades. Der wesentliche Unterschied zum heutigen Verständnis der RS-Codes bestand vor allem darin, dass sie nicht im Kontext der zyklischen Codes beschrieben wurden und Codelängen von $n = q$ vorgesehen waren. Noch 1960 redefinierte *W. W. Peterson* RS-Codes als zyklische Codes, was den Vorteil verschaffte, dass nun auch jeder RS-Code vollständig durch ein einziges Generatorpolynom beschrieben werden konnte. Außerdem brachte *Peterson* RS-Codes mit den von *A. Hocquenghem* [Hocquenghem 1959] und von *R. C. Bose* und *D. K. Chaudhuri* [Bose & Chaudhuri 1960] unabhängig voneinander entdeckten BCH-Codes in Verbindung. Seither verfügen RS-Codes über Codelängen von $n = q - 1$. Seit Ende der sechziger Jahre werden RS-Codes erfolgreich in zahlreichen Anwendungen eingesetzt, z.B. in der unbemannten Raumfahrt.

RS-Codes besitzen als spezielle BCH-Codes Generatorpolynome von der Gestalt

$$g(X) = \prod_{j=b}^{2t+b-1} (X - \alpha^j) \ , \ \alpha \text{ primitive } n\text{-te Einheitswurzel von } (GF(q))^* \ .$$

Die Nullstellen von $g(X)$ sind stets $2t$ aufeinander folgende Potenzen von α, denn der Grundkörper $GF(q)$ ist für RS-Codes immer auch der Zerfällungskörper des Polynoms $X^n - 1$. Die ZNK auf $GF(q)$ sind in diesem Fall stets einelementig, wie sich leicht nachrechnen lässt. Wegen des Satzes 1.50 über die BCH-Schranke gilt für die Minimaldistanz d von RS-Codes $d \geq \delta = 2t + 1 = \operatorname{grad} g(X) + 1$. Die Singleton-Schranke (Satz 1.13) liefert aber für jeden linearen (n, k, d)-Code die Beziehung $d \leq n - k + 1 = \operatorname{grad} g(X) + 1$. Daher ist für RS-Codes die Entwurfsdistanz identisch mit der wahren Minimaldistanz und letztere ist wegen der Singleton-Schranke (Satz 1.13) dazu von maximaler Größe. Codes, die diese Eigenschaft erfüllen, werden *MDS-Codes* (*'maximum distance separable'*) genannt.

Da RS-Codes über Körpern der Ordnung $q = p^m - 1$ definiert sind, p ist eine Primzahl, sind Nachrichten- und Prüfsymbole der Codewörter m-Tupel bzw. Spaltenvektoren mit m Einträgen in $GF(p)$. Je nach Anwendung kann jedoch die Codelänge $n = q - 1$ für eine große benötigte Minimaldistanz zu unhandlich werden. In solchen Fällen ist es möglich, die Codelänge so zu verkürzen, dass Linearität, Zyklizität, Varietät, Minimaldistanz und die Dimension der Symbole des 'Muttercodes' auf den verkürzten Code vererbt werden (*verkürzte BCH-Codes*; [Bossert 1992], S. 100f und [Betten et al. 1998] S. 50f, 71f). Sei z.B. $\mathcal{C}$ ein $(n = q-1, k = q-d, d)$ RS-Code über $GF(q)$ mit Generatorpolynom $g(X)$. Dann ist

$$\mathcal{C} = \{u(X)g(X) | \operatorname{grad} u(X) < k\} \ .$$

Werden nur die Nachrichten $u(X) \in GF(q)[X]$ mit $\operatorname{grad} u < k^* < k$ benutzt, so wird der Code als *verkürzter RS-Code* der Länge $n^* = n - (k - k^*)$, der Dimension k^* und der gleichen Minimaldistanz d bezeichnet. Anders ausgedrückt: es werden nur diejenigen Codewörter des Codes $\mathcal{C}$ verwendet, die an den ersten $k - k^*$ Symbolstellen gleich Null sind. Die ersten $k - k^*$ Symbolstellen solcher Codewörter werden gestrichen. Auch der verkürzte RS-Code ist ein MDS-Code. Zum Beispiel basiert das für die Fehlerkorrektur in Compact Disc Systemen einge-

setzte Korrekturverfahren auf einer Verschachtelung zweier verkürzter RS-Codes mit den Parametern $(32, 28, 5)$ und $(28, 24, 5)$, die aus dem $(255, 251, 5)$ RS-Code abgeleitet sind [Hoeve et al. 1982].

3.2. Compact Disc Systeme

CD-Player und CDs haben wegen der guten Wiedergabequalität von Audioaufnahmen und ihrer Unempfindlichkeit gegenüber Abnutzung, im Vergleich zu analogen Tonträgern, die HiFi-Technik revolutioniert. Die Kanalcodierung machte die Entwicklung der *Compact Disc Systeme* für den praktischen Gebrauch überhaupt erst möglich. Kratzer, Staub oder Fingerabdrücke auf der CD-Oberfläche können den ursprünglichen Datenstrom bei der Abtastung der digitalen Information durch die empfindliche Laseroptik in Form von großen Fehlerbündeln („*Error Bursts*“) verfälschen. Im Rahmen der Massenproduktion musste außerdem zur Kostenreduzierung eine große Toleranz gegenüber dem Auftreten zufälliger (Einzel)-Fehler („*Random-Errors*“) vorausgesetzt werden können. Die beste bekannte Wahl, sowohl mit Fehlerbündeln als auch mit zufälligen Fehlern fertig zu werden, sind RS-Codes (vgl. *K. A. Schouhamer Immink* in [Wicker et al. 1994], S. 41). Zur Fehlerkorrektur werden zwei kreuzweise ineinander verschachtelte („*Cross Interleaved*“), verkürzte RS-Codes $\mathcal{C}_1$ und $\mathcal{C}_2$ über $GF(2^8)$ benutzt. Der $\mathcal{C}_1$-Code hat die Parameter $(32, 28, 5)$ und der $\mathcal{C}_2$-Code $(28, 24, 5)$. Beide Codes sind nach Satz 1.6 zweifehlerkorrigierend und vierfehlererkennend. Zusammen bilden sie ein komplexes Codierungsystem, das als *CIRC* („*Cross Interleaved Reed-Solomon Code*“) bezeichnet wird. Neben den bereits erläuterten Prinzipien bei der Fehlerkorrektur spielen bei CIRC weitere Konzepte eine tragende Rolle:

- Nachrichten- und Prüfsymbole der Codewörter sind m-Tupel. Deshalb werden Fehlerbündel im Gegensatz zu binären BCH-Codes entsprechender Länge innerhalb relativ weniger Symbole gefangen.
- Das Cross-Interleaving führt dazu, dass Codewörter von CIRC in der zweischrittigen Codierung nicht aus einem zusammengehörigen Datenwort gebildet werden, sondern aus einzelnen Nachrichtensymbolen vieler solcher

Datenwörter. Dies ermöglicht eine Aufspreizung von Fehlerbündeln nach der $\mathcal{C}_1$-Decodierung zu wenigen Fehlern in vielen Codewörtern für die $\mathcal{C}_2$-Decodierung.

- Nicht korrigierbare Fehler werden durch Interpolieren mit Hilfe vorausgegangener und nachfolgender Information oder sogar durch Auslöschung *maskiert* („*Concealment*“).

- Hinzufügen von „*Erasure Flags*“ zu den als fehlerhaft vermuteten Symbolen der $\mathcal{C}_1$-Decodierung zeigt dem $\mathcal{C}_2$-Decodierer die Fehlerpositionen in $\mathcal{C}_2$-Codewörtern an und vereinfacht die Korrektur und Maskierung von Symbolfehlern erheblich.

Auf die letzten beiden Konzepte soll hier nicht weiter eingegangen werden, da sie vor allem von der Decodierung selbst und den nachfolgenden Verarbeitungsprozessen abhängen (vgl. [Hoeve et al. 1982]). Stattdessen wird im Folgenden das Cross-Interleaving eingehend besprochen, das die Fehlerkorrekturfähigkeit von RS-Codes verbessert. Dazu ist es nötig, einige Schritte der Modifikation des Audiosignals von der Quelle zum speicherbaren Code zu erwähnen.

Die CD als Datenträger im „Compact Disc System“ hat einen Durchmesser von 120 mm. Sie besteht aus einer Kunststoffscheibe, auf deren Innenseite entlang einer bis zu $5,6$ km langen spiralförmigen Spur („*Track*“) der Information entsprechend Gruben („*Pits*“) eingeprägt sind. Die Spur besteht daher neben den Pits auch aus Anhöhen, den „*Lands*“. Jede Kante eines Lands entspricht einer binären „1“ und etwa alle 0.3μm eines Pits oder Lands einer binären „0“. Die Spur repräsentiert somit eine lange Folge binärer Ziffern, den *Bits*.
Bei digitalen Neuaufnahmen wird als erstes eine „*Masterdisc*“ gebrannt, von der später Kopien gepresst werden. Das zunächst analoge Audiosignal wird dazu an das *codierende System* übertragen. Am Anfang dieses Systems erfolgt die Analog-Digital Wandlung mittels „*Pulse Code Modulation*“ (*PCM*), bei Stereoaufnahmen getrennt nach linkem und rechtem Kanal (vgl. *K. A. Schouhamer Immink* in [Wicker et al. 1994], S. 41–59). Dabei wird das Audiosignal, aufgrund des Abtast-Theorems von Shannon, bei 44.1 kHz abgetastet, wobei der Wert einer Abtastung als „*Sample*“ bezeichnet wird. Nach der Quantisierung besteht ein digitales Sample aus 32 Bit, wovon je 16 Bit dem linken bzw. dem rechten Ste-

reokanal entsprechen. Jeweils sechs Samples $S_{6n}, S_{6n+1}, \ldots, S_{6n+5}$ fasst man zu einem „*Frame*“ F_n zusammen, das wiederum in vierundzwanzig Symbole zu 8 Bit, den *Nachrichtensymbolen*, zerlegt wird. Die 8 Bit breiten Nachrichtensymbole eines Frames F_n werden im Folgenden den Samples S_{6n+b}, denen sie angehören, entsprechend mit $u_{12n+2b;L1}$ bezeichnet, wenn sie den ersten 8 Bit Teil des linken Audiokanals repräsentieren. Ein zugehöriges Nachrichtensymbol wird durch $u_{12n+(2b+1);R2}$ dargestellt, wenn es den zweiten Teil des rechten Audiokanals codiert. Entsprechendes gilt für die Nachrichtensymbole $u_{12n+2b;L2}$ und $u_{12n+2b;R1}$ (vgl. Abb. 3.1). Die Frames $F_n, F_{n+1}, \ldots$ werden in Form eines ununterbrochenen *Bitstroms* dem CIRC-Codierer zugeleitet und in der 8-Bit Symbolstruktur für die Codierung durch den $\mathcal{C}_2$-Codierer von CIRC vorbereitet.

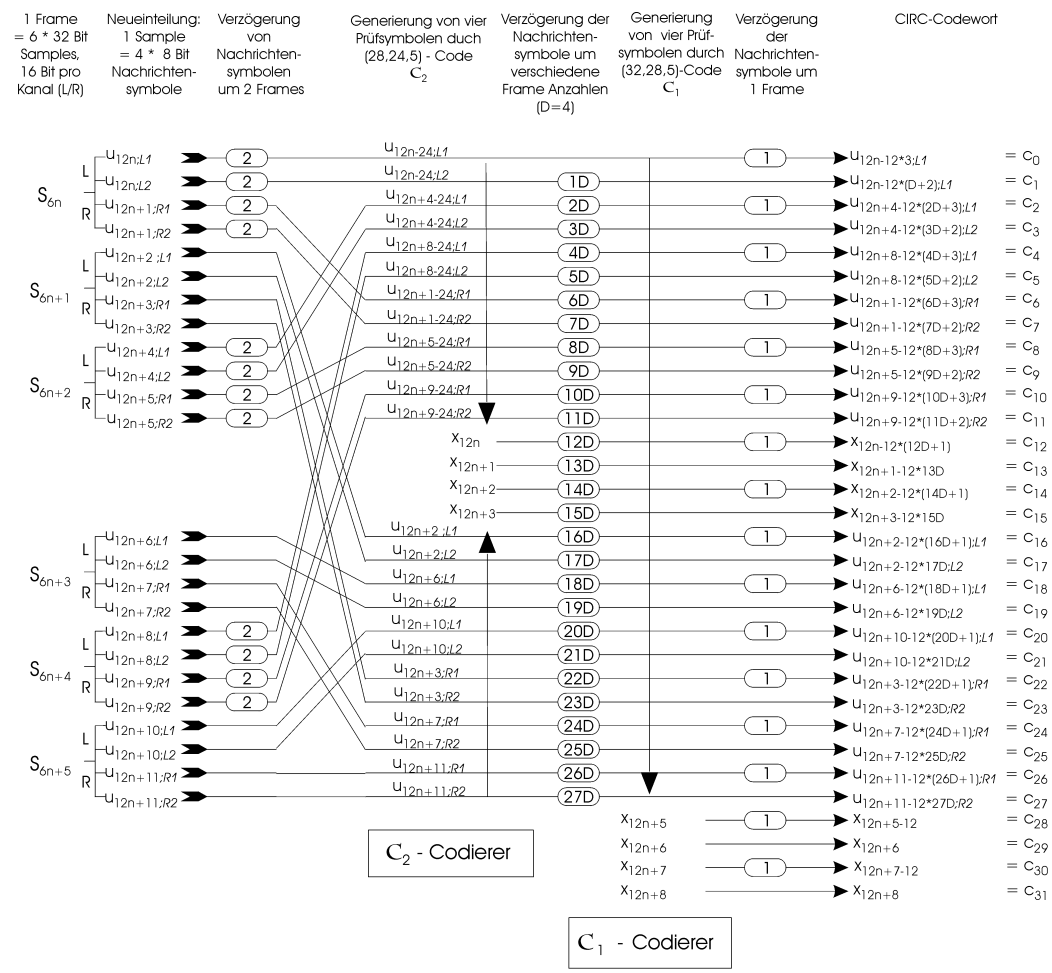

Abb. 3.1: Blockdiagramm des CIRC Codierers nach [McEliece et al. 1994]

Nachrichtensymbole von Samples mit geradzahligem Index werden vor dem Codierereingang um die doppelte Zeit verzögert, die der Durchlauf eines Frames benötigt. Das hat Bedeutung für die Maskierung unkorrigierbarer Fehler. Durch kreuzweises Vertauschen werden außerdem Nachrichtensymbole von Samples mit geradem Index auf die ersten zwölf Symbolpositionen, dabei nach linkem und rechtem Stereokanal geordnet, an den $\mathcal{C}_2$-Codierereingang geleitet. Die Nachrichtensymbole ungerader Samples werden damit in die letzten zwölf Symbolpositionen sortiert und dabei ebenfalls nach linkem und rechtem Stereoeingang getrennt. Samples mit ungeradem Index erreichen den Codierereingang ohne eine *Verzögerung*. Dies hat zur Folge, dass eine Nachricht des $\mathcal{C}_2$-Codes aus den Nachrichtensymbolen gerader Samples eines Frames F_n und den Nachrichtensymbolen ungerader Samples des zweiten nachfolgenden Frames F_{n+2} besteht. Der $\mathcal{C}_2$-Codierer fügt dann bei der Codierung zwischen den Nachrichtensymbolen gerader und ungerader Samples vier Prüfsymbole ein.
Es folgt der eigentliche Vorgang des kreuzweisen Verschachtelns, das Cross Interleaving oder auch „*Convolutional Interleaving*“ genannt wird. Die Bildung der Nachrichten für den $\mathcal{C}_1$-Code erfolgt ebenfalls durch frameweise Verzögerungen, jedoch mit dem Unterschied, dass jedes Codesymbol von $\mathcal{C}_2$-Codewörtern aufsteigend von $0D - 27D$ Frame-Verzögerungen (Verzögerungsoperator bei CIRC: $D = 4$) gepuffert wird (vgl. Abb. 3.1). Eine Nachricht des $\mathcal{C}_1$-Codes besteht somit aus den Codesymbolen von 28 verschiedenen Codewörtern von C_2, die aus jeder vierten von 109 $\mathcal{C}_2$-Codierungen stammen. Nach Verlassen des $\mathcal{C}_1$-Codierers wird jedes zweite Codesymbol um ein weiteres Frame verzögert, um zwei aufeinander folgende Symbolfehler durch kleinere Fehlerbündel im C_1-Code zu trennen.
Im Anschluss an die Codierung durch CIRC werden jedem Frame $C\&D$-Informationen („*Control & Display*“) in Form von 8 Bit hinzugefügt. (Der Vollständigkeit halber sei erwähnt, dass einige Symbole nach Verlassen des $\mathcal{C}_1$-Codierers noch logisch negiert werden. Für die Betrachtung hier spielt das keine Rolle). Der codierte Audiobitstrom wird dann *Databitstrom* genannt und hat eine Länge von 264 Bit.
Es folgen weitere Modifikationen, wie Konvertierung der 8 Bit Codesymbole auf 17 Bit Symbole durch *EFM* und „*Merging*“ sowie das Hinzufügen von 27 Synchronisationsbits zu jedem modifizierten Codewort, so dass es letztendlich aus 588 Bit („*Channelbits*“) besteht. Diese Modifikationen haben nichts mit der ei-

gentlichen Fehlerkorrektur zu tun und haben auch keine nennenswerten Auswirkungen auf diese. Die Konvertierung hat ausschließlich technische Gründe (vgl. [Carasso et al. 1982] oder *K. A. Schouhamer Immink* in [Wicker et al. 1994]). Der aus allen Modifikationen resultierende Channelbitstrom wird am Ende durch eine entsprechende Laseroptik auf die Masterdisc gebrannt.

Der wesentliche Vorteil der Verschachtelung beider verkürzter RS-Codes liegt darin, dass durch das Hinzufügen von Erasure Flags zu fehlerhaften Nachrichtensymbolen die Fehlerkorrektur zugleich wesentlich effizienter und drastisch vereinfacht wird. Durch einen Kratzer auf der CD-Oberfläche werden z.B. über eine Länge von einigen 100 Channelbits falsche Information abgetastet (Fehlerbündel). Die Laseroptik des Players liest an dieser Stelle die „Information“ aus und leitet sie dem internen System als Channelbitstrom zu. Bevor der CIRC-Decoder erreicht wird, werden die o.g. Modifikationen rückgängig gemacht, bis der Databitstrom vorliegt. Aus diesem werden dann die C&D Symbole entfernt und, sofern möglich, getrennt weiterverarbeitet. Vom verbleibenden Bitstrom wird dann die Symbolstruktur vor dem Eingang des $\mathcal{C}_1$-Decodierers erstellt.
Entdeckt der $\mathcal{C}_1$-Decodierer nur einen Fehler, korrigiert er diesen. Findet er jedoch mehr als einen Fehler im empfangenen Wort, so markiert er alle Codesymbole mit Erasure Flags und leitet sie sonst unverändert nach Entfernen der $\mathcal{C}_1$ Prüfsymbole an den $\mathcal{C}_2$-Decodierer weiter. Durch die reziproke Verzögerung zum Codieren werden aus Sicht des $\mathcal{C}_2$-Decodierers mögliche Fehlerbündel auf wenige Symbolfehler in vielen Codewörtern verteilt, deren Positionen im Codewort zugleich durch die Erasure Flags bekannt gegeben werden. Der $\mathcal{C}_2$-Decodierer kann dadurch bis zu 4 Fehler korrigieren [Hoeve et al. 1982]. Insgesamt lassen sich auf diese Weise um 4000 aufeinander folgende Fehler im Databitstrom korrigieren, was einer Kratzerlänge von etwa 2.5 mm auf der Oberfläche der CD in tangentialer Richtung der Spur entspricht. Treten mehr als vier Fehler in einem $\mathcal{C}_2$-Codewort auf, so wird versucht jene zu maskieren.
Auf den ersten Blick erscheint es merkwürdig, dass $\mathcal{C}_1$ nur zur einfachen und nicht zur zweifachen Fehlerkorrektur benutzt wird. Die Wahrscheinlichkeit, dass der $\mathcal{C}_1$-Decodierer vierfache oder höherwertige Fehler nicht erkennt, liegt bei 2^{-19}. Eine zweifache Fehlerkorrektur durch den $\mathcal{C}_1$-Decodierer erhöht diese Wahrscheinlichkeit [Driessen et al. 1982].

3.3. Unbemannte Raumfahrt

Die Erforschung unseres Sonnensystems durch unbemannte Raumfahrt ist eine vielbeachtete Errungenschaft des 20. Jahrhunderts. Fotos und Infrarotaufnahmen seiner Planeten, die durch Raumsonden wie *Mariner, Voyager* und *Galileo* getätigt und über Milliarden von Kilometern zur Erde gesandt wurden, haben die Astronomie weiter vorangebracht als es durch den Einsatz von stationären Teleskopen möglich gewesen wäre. Erst spät nach ihrer Entdeckung sind RS-Codes zu einem wichtigen Bestandteil der Kommunikationssysteme solcher Raumsonden geworden.

Für die Kommunikation zwischen Raumsonde im All und Empfänger auf der Erde liegt näherungsweise ein leistungsbeschränkter, breitbandiger, additiver weißer Gauß'scher Kanal vor. Der additive Gauß'sche Kanal ist gedächtnislos, d.h. er beschreibt einen Kanal, in dem momentane Störungen unabhängig von denen der Vergangenheit auftreten und somit keine Fehlerbündel hervorrufen (vgl. [Bossert 1992], S. 121ff).
Die von den Systemen der Raumsonde aufbereitete digitale Information (*Databitstrom*) moduliert für die Übertragung ein analoges Signal, die Trägerfrequenz mit den *Kanalsymbolen* (bei der Pulse-Amplituden-Modulation sind verschiedene Amplituden der Trägerfrequenz aus einer diskreten Menge die Kanalsymbole). Störungen im additiven Gaußkanal treten als *additives weißes Gauß'sches Rauschen (AWGN)* auf, das fast immer ein gleichverteiltes Spektrum analoger Signale darstellt, deren Amplituden annähernd gleich sind. Daher muss man davon ausgehen, das jede beliebige Trägerfrequenz im AWGN enthalten ist.
Die Wahrscheinlichkeit für das Auftreten von Störungen unterschiedlicher Stärke ist gaußverteilt, d.h. schwache Störungen aus einem Intervall nahe bei Null sind sehr wahrscheinlich, starke Störungen aus einem Intervall entfernt von Null sind weniger wahrscheinlich und kommen daher selten vor. Die aus einem Intervall stammenden Amplituden der im AWGN enthaltenen *Trägerfrequenz* überlagern im Kanal die Amplituden der ursprünglich gesendeten Trägerfrequenz. Deshalb stammen zwar die gesendeten Kanalsymbole aus einer diskreten Menge, aber das, was auf der Erde nach passieren des Kanals ankommt, ist ebenfalls aus einem Intervall.

Die Betrachtung des Signal-Rauschverhältnis (Leistungsverhältnis E_b/N_0 in dB [Bossert 1992]) pro empfangenem Nachrichtenbit ist ein Konzept, das die Beschreibung des Rausch-Einflusses auf die Bitfehlerwahrscheinlichkeit nach der Decodierung („*Bit Error Rate*“, *BER*) ermöglicht. Ist das Signal-Rauschverhältnis zu einem Zeitpunkt klein, so ist die Wahrscheinlichkeit groß, dass beim Empfang ein anderes als das ursprünglich gesendete Kanalsymbol demoduliert wurde. Ist das Verhältnis zu einem anderen Zeitpunkt groß, so ist auch die Wahrscheinlichkeit, das korrekte Symbol demoduliert zu haben, groß. Der Grad des Rauschens ist also eine Zuverlässigkeitsinformation beim Empfang eines jeden Symbols über den Zustand des Kanals. Dies führt zu zwei übergeordneten Decodierungsstrategien. Bei der einen wird die Zuverlässigkeitsinformation nicht genutzt („*Hard Decision Decoding*“, *HDD*), wohingegen bei der anderen („*Soft Decision Decoding*“, *SDD*) die Zuverlässigkeitsinformation zur Demodulation des Signals und zur Fehlerkorrektur verwendet wird. Die SDD-Strategie benötigt ein annähernd 2 dB geringeres durchschnittliches Signal-Rauschverhältnis auf dem additiven Gauß'schen Kanal als die HDD, um aus den empfangenen Kanalsymbolen die ursprünglich gesendeten Bits zu detektieren (vgl. *A. B. Cooper* in [Wicker et al. 1994], S. 108–124).

RS-Codes sind zur Übermittlung von Daten durch den additiven Gauß'schen Kanal nicht direkt geeignet. Zum einen deshalb, weil die eigentliche Stärke, Fehlerbündel zu korrigieren, nicht verlangt wird, zum anderen weil bisher keine praktikablen Methoden bekannt sind, RS-Codes mit SDD-Strategien zu verbinden und in den meisten Anwendungen der 2 dB Verlust durch HDD-Strategien nicht hinnehmbar ist (vgl. *R. J. McEliece et al.* in [Wicker et al. 1994], S. 25–40). *Faltungscodes* hingegen, im Englischen „*Convolutional Codes*“, erlauben die Verwendung von Zuverlässigkeitsinformationen bei der Decodierung in natürlicher Weise (vgl. [Bossert 1992], S. 146).
Faltungscodes sind neben den Blockcodes (vgl. Kapitel 1) die zweite große Gruppe fehlerkorrigierender Codes. Im Gegensatz zu den Blockcodes hängen ihre N Codesymbole nicht nur von den Nachrichten der Länge K ab, sondern zusätzlich von den letzten $M \cdot K$ Nachrichtensymbolen vorangegangener Nachrichten. Die Größe $M \cdot K$ wird als das *Gedächtnis* des Codierers bezeichnet. Die Redundanz wird damit nicht allein durch das Anfügen zusätzlicher Symbole an die Nachrichten erreicht, sondern auch durch die eingeschränkten Möglichkeiten der Abfolgen

von Codewörtern (*Pfade*). Decodiert werden Faltungscodes in der Praxis häufig durch den *Viterbi Algorithmus*, der durch Erweiterung („*Soft Output Viterbi Algorithm*“, *SOVA*) Zuverlässigkeitsinformationen über den Kanal verwerten kann.

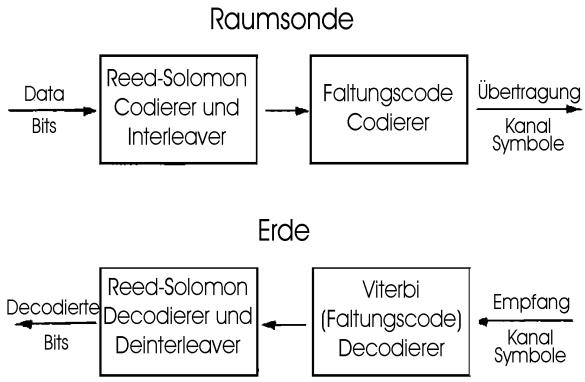

Abb. 3.2: Ein Odenwalder - Viterbi verknüpftes Codierungssystem für den Übertragungskanal vom Weltraum zur Erde nach [McEliece et al. 1994].

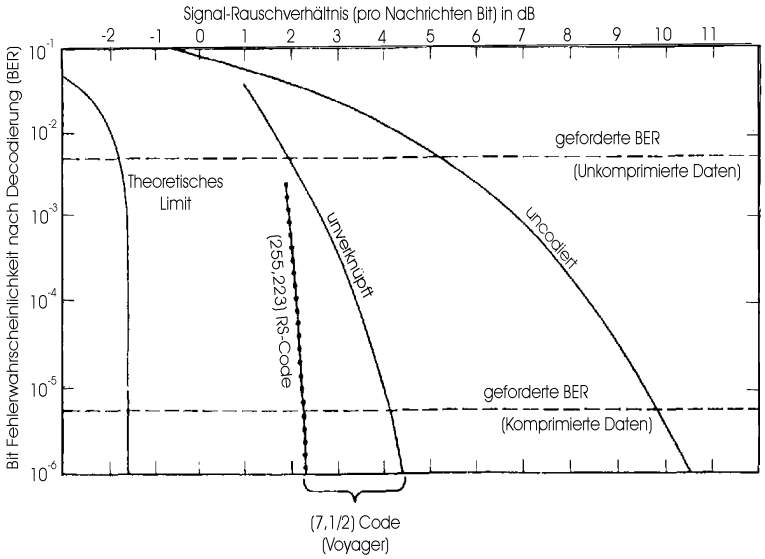

Abb. 3.3: Typische Leistungskurven für verknüpfte und unverknüpfte Codierungssysteme für den Übertragungskanal vom Weltraum zur Erde nach [McEliece et al. 1994].

Durch die Konstruktion von Faltungscodes ergibt es sich beinahe zwangsläufig, dass die Abhängigkeiten zwischen den Codewörtern große Fehlerbündel verursachen, wenn genügend Fehler im gesendeten Code auftreten (*katastrophale Fehler*). Für besondere Daten und Umstände sind außerdem sehr niedrige BER nach der Decodierung nötig, um die Qualität für eine wissenschaftliche Verwendbarkeit der

Daten zu gewährleisten. Unter diesen beiden Gesichtspunkten wird die Anwendung von RS-Codes in der unbemannten Raumfahrt interessant.
Odenwalder war 1970 der erste, der ein *verknüpftes* („*Concatenated*“) *System* aus Faltungs- und RS-Codes für die Übertragung von Daten aus dem All vorschlug. Dabei fungieren RS-Codes als *äußere* und Faltungscodes als *innere Codes* in einem hintereinander geschalteten Codierungssystem (vgl. Abb. 3.2).

Abbildung 3.3 zeigt anschaulich, warum RS-Codes erst spät Einzug in die Kommunikation mit Raumsonden erhielten. Die Übertragung unkomprimierter Bilder benötigt lediglich eine BER von etwa $5 \cdot 10^{-3}$, die unter annähernd gleichem Signal-Rauschverhältnis pro empfangenem Nachrichtenbit allein durch die Faltungscodes gewährleistet werden kann. Die demgegenüber leichte Verbesserung durch verknüpfte Codes bei einer hohen Mindest-BER, rechtfertigte damit nicht die verbundene höhere Komplexität des Systems. Mit Einzug effektiver Komprimierungstechniken Anfang der siebziger Jahre sank die geforderte BER jedoch beträchtlich auf etwa $5 \cdot 10^{-6}$, was in idealer Weise nur mit verknüpften RS- und Faltungscodes erreicht werden kann.
Um Nachrichten mit Fehlerbündeln effektiv korrekt zu decodieren, werden in der Raumfahrt ebenfalls Techniken des „Interleavings“ bei RS-Codes benutzt. Allerdings unterscheidet sich die verwendete Methode von der des CD-Players. Hierbei werden eine bestimmte Anzahl von Codewörtern (*Interleaving-Tiefe*) zeilenweise in eine Matrix geschrieben, deren Spalten für die Übertragung weiter aufgearbeitet werden. Dieses sehr einfache Interleaving-Verfahren bezeichnet man als *Blockinterleaving*.

Die nachfolgenden Beschreibungen der Involvierung von RS-Codes in Raumfahrtmissionen erfolgen nach *R. J. McEliece et al.* in [Wicker et al. 1994].

3.3.1. Die Mariner Mars Orbiter Mission

Die *Mariner Mars Orbiter Mission* der NASA (1971) war, unter etwas anderen Umständen als zuvor geschildert, die erste Anwendung von RS-Codes in der Raumfahrt.

Als Hauptübertragungscode für Oberflächenaufnahmen des Mars, die den Großteil der übertragenen Daten ausmachten, wurde ein 7-fehlerkorrigierender $(32, 6)$ *Reed-Muller Code erster Ordnung* verwendet (vgl. [Betten et al. 1998], S. 62f), bei dem ein durchschnittliches Signal-Rauschverhältnis für eine BER von $5 \cdot 10^{-3}$ gering genug war. Die Raumsonde übertrug aber auch noch einen kleinen Anteil von Daten aus einem anderen Experiment. Die Aufnahmen des *Infrarot Interferometer Spektrometers* (*IRIS*) benötigten zur wissenschaftlichen Verwertbarkeit eine BER von mindestens $5 \cdot 10^{-5}$. Die nahe liegende Lösung, den gesamten Datenstrom unter größerem Energieaufwand auf ein entsprechend besseres Signal-Rauschverhältnis zu bringen, war eine Verschwendung knapper Ressourcen. *Dorsch* und *Miller* schlugen daher nur für die Übersendung der IRIS Daten ein verknüpftes Codierungsystem des Basiscodes mit dem 1-fehlerkorrigierenden $(6, 4, 3)$ RS-Code vor, den sie als generalsierten Hammingcode bezeichneten. Im nachhinein stellte sich heraus, dass dieser gleichzeitig ein Spezialfall von RS-Codes war.

3.3.2. Die Voyager Mission

Erst ab 1977 wurden mit der Voyager Mission zu Jupiter, Saturn, Uranus und Neptun mehrfach-fehlerkorrigierende RS-Codes eingesetzt. Bei früheren Missionen und auch bei den Voyager-Aufnahmen von Jupiter und Saturn bestand keine Notwendigkeit, verknüpfte Codierungssysteme mit RS-Codes zu verwenden, da Daten stets unkomprimiert versandt wurden und die Leistungsfähigkeit der eingesetzten Faltungscodes völlig ausreichte. Wie in Abb. 3.3 zu sehen, ist ein verknüpftes Codierungssystem nach Odenwalder in diesen Größenordnungen gegenüber dem $K = 7, R = 1/2$ Faltungscode (kurz $(7, 1/2)$-Faltungscode: R bezeichnet die Coderate, das Verhältnis zwischen Information und Information + Redundanz) bzgl. des Signal-Rauschverhältnisses nur unwesentlich besser.

Eine unkomprimierte Farbaufnahme der beanspruchten Größe und Qualität benötigt nach der Digitalisierung rund $15,5$ Millionen Bit. Zwar war schon seit den sechziger Jahren bekannt, dass solche planetarischen Bilder über eine große Redundanz verfügten, so dass weniger als 15 Millionen Bit für eine vergleichbare

Qualität ausreichten, doch waren die Techniken, die Redundanz zu vermindern, zu komplex, um in die Systeme einer Raumsonde implementiert zu werden. Erst Anfang der siebziger Jahre gelang es, einen ausreichend einfachen Algorithmus (*Rice-Algorithmus*) bereitzustellen, der den Datenaufwand um Faktor $2,5$ reduzierte, ohne einen Qualitätsverlust herbeizuführen. Es bestand aber das Risiko, die Aufnahmen durch Störungen bei der Übertragung zur Unkenntlichkeit zu verstümmeln, da der *Rice-Dekomprimierungsalgorithmus*, wie die meisten Dekomprimierungsalgorithmen, sehr fehleranfällig ist. Die benötigte BER liegt für die Übertragung Rice-komprimierter Daten bei etwa 10^{-6}, einem Wert bei der verknüpfte Codierungssysteme Faltungscodes als alleinige Übertragungscodes deutlich überlegen sind, wie Abb 3.3 zeigt.
Obwohl die Benutzung des Rice-Komprimierungsalgorithmus in Verbindung mit dem Odenwalder Codierungssystem eine 78% höhere Datenrate gegenüber der Übertragung unkomprimierter Bilder mit dem Faltungscode allein versprachen, entschieden sich die verantwortlichen Ingenieure auch bei der Voyager Mission zu Saturn und Jupiter gegen ihren umfassenden Einsatz. Jedoch implementierten sie das Odenwalder Codierungssystem für den Notfall, dass die Hauptübertragungskanäle versagen würden. Bei den Zusatzmissionen zu Uranus und Neptun sollte das verknüpfte System aus einem RS-Code und dem $(7, 1/2)$-Faltungscode zum vollen Einsatz kommen, weil bei diesen Missionen das Risiko eines Versagens des Basissystems das Risiko durch den Rice-Algorithmus aufhob. Es wurde daher der 16-fehlerkorrigierende $(255, 223, 33)$ RS-Code im engeren Sinne über $GF(2^8)$ (also mit Generatorpolynom $g(X) = \prod_{i=1}^{32}(X - \alpha^i)$, α primitive 255. Einheitswurzel) verwendet mit Interleavingtiefe 4. Der zugrunde liegende endliche Körper $GF(2^8)$ wurde durch das Polynom $X^8 + X^7 + X^2 + X + 1$ erzeugt.

3.3.3. Die Galileo Mission

Die Galileo Mission, die wegen der Challenger Katastrophe erst 1989 und nicht 1986, zum Jupiter startete, sollte ursprünglich das gleiche verknüpfte Codierungssystem der Voyager benutzen, diesmal jedoch im vollen Umfang. Allerdings wurde aus unvermeidbaren technischen Gründen eine Interleavingtiefe des RS-Codes von 2 vorgesehen. Der verspätete Start und der durch die Erfahrung der Challenger Katastrophe beschränkte Antrieb mündete in einer verspäteten Zielankunft 1995, was eine weniger günstige Planetenkonstellation zur Folge hatte. Deshalb

tauschten die Ingenieure noch vor dem Start den $(7, 1/2)$-Faltungscode gegen den bzgl. der Fehlerkorrektureigenschaften besseren $(15, 1/4)$-Faltungscode aus, ohne jedoch den RS-Code zu verändern.
Nach dem Start von Galileo im Jahre 1991 ereignete sich ein weiterer folgenschwerer Zwischenfall, der sämtliche Planungen umwarf. Die Hochleistungsantenne von Galileo konnte sich nicht entfalten und fiel komplett aus. Daher mussten sich die Beteiligten auf die Niederleistungsantenne Galileos, auch *S-Band Antenne* genannt, verlassen, was die Datenrate, durch den Verlust von 40 dB Signalleistung, ohne große Umwälzungen von 100.000 bit/s auf 10 bit/s verminderte.
Die größte Verbesserung auf der Datenverarbeitungsebene war die Umstellung auf ein 15:1 Bildkompressionsverfahren, was eine BER-Verbesserung, unter der Voraussetzung jedes Zehntel des Signal-Rauschverhältnis einzusparen, auf 10^{-7} mit sich ziehen musste. Neben anderen, weniger tief greifenden Änderungen, war der Schlüssel zur Lösung des Problems, empfangene Daten mehrfach zu decodieren.
Außer technisch bedingten Änderungen des Faltungscodes zu einem $(14, 1/4)$-Code wurde die Blockinterleavingtiefe für die RS-Codierung auf 8 gesteigert, wobei die 8 RS-Codewörter eines Blocks aus verschiedenen RS-Codes der Länge 255 bestanden. Die Anzahl der Prüfsymbole der 8 Codewörter waren $100, 10, 32, 10, 60, 10, 32, 10$, was einer durchschnittliche Redundanz von 33 Prüfsymbolen pro Codewort entsprach. Codewörter mit hoher Redundanz werden als *starke Codewörter*, die mit niedriger Redundanz als *schwache Codewörter* bezeichnet. In Gegenwart eines durch den innen liegenden Viterbi-Decoder verursachten langen Fehlerbündels im ersten Durchlauf ist die Wahrscheinlichkeit noch hoch, dass das stärkste RS-Codewort korrekt decodiert werden kann, auch wenn dies bei allen anderen Codewörtern des Blocks nicht gelingt. Mit Hilfe der daraus gewonnenen korrigierten Bits wird ein zweiter Viterbi-Decoderdurchlauf vorgenommen, in der Hoffnung, dass weniger in Betracht kommende Faltungscodewörter-Pfade durch korrigierte Bits festgelegt werden können. Hat sich die Hoffnung erfüllt, ist es möglich, das nächst stärkste Codewort im RS-Decodierer zu korrigieren und damit die Möglichkeiten der Pfade weiter einzuschränken. Nach insgesamt vier Durchläufen werden die Wiederholungen beendet. Diese Decodierungsstrategie erfordert ein lediglich 0.53 dB größeres Signal-Rauschverhältnis als das gewöhnliche Odenwalder-Systemschema und steigert die BER auf $2 \cdot 10^{-7}$.

Inzwischen ist die Verwendung von RS-Codes in der Raumfahrt zur Routine geworden, so dass sich ein Komitee internationaler Raumfahrtbehörden auf einen gemeinsamen Standard verständigt hat. Dieser wurde schon bei zahlreichen Missionen angewandt (NASA: Observer 1992, Cassini 1997; joint venture NASA/ESA: Ulysses 1990; ESA: Giotto 1985, Huigens 1997, Cluster, Soho). Der CCSDD-Standard ist zweigleisig: er empfiehlt sowohl ein Faltungscode-System ohne Verknüpfung als auch ein verknüpftes RS-Code/Faltungscodesystem. Das unverknüpfte System benutzt einen $(7, 1/2)$ Faltungscode wie bei Voyager. Das verknüpfte System involviert zusätzlich einen $(255, 223, 33)$-RS-Code über $GF(2^8)$ mit möglichen Interleavingtiefen 1,2,3,4 und 5.
Anders als bei der Voyager Mission wird hier $GF(2^8)$ durch $X^8+X^4+X^3+X^2+1$ erzeugt. Es wird auch kein RS-Code im engeren Sinne benutzt, sondern ein modifizierter RS-Code nach einem Vorschlag von *E. R. Berlekamp* und *M. Perlman* mit Generatorpolynom

$$g(X) = \prod_{i=112}^{143} (X - \alpha^{11i}) \ , \ \alpha \text{ primitive 255. Einheitswurzel.}$$

Dies dient neben weiteren mathematischen Vorkehrungen der Minimierung des Hardwareaufwandes des Codierers in Bezug zum eingesetzten Decodierer.

3.4. Andere Anwendungen

Wicker et al. berichten in [Wicker et al. 1994], dass RS-Codes neben den ausführlicher beschriebenen Anwendungen in „*Frequency-Hopping/Spread Spectrum*"-Systemen (*FH/SS*) für höchstzuverlässige militärische Kommunikationssysteme eingesetzt wurden. Auch in fortgeschrittenen Fehler-Kontroll-Systemen von integrierten Schaltkreisen in Speicherbausteinen schneller Computer wurden sie verwendet. Außerdem diskutierte man über den Einsatz in Mobilfunk Anwendungen (bei „*Direct-Sequence/Spread Spectrum*"-Systemen, *DS/SS*), mobilen Datenübertragungssystemen und der Entwicklung auf Glasfaser basierenden Hochgeschwindigkeitsprozessoren.
Überall dort, wo Fehlerbündel im Übertragungskanal oder durch das Decodierungskonzept entstehen, ist die Anwendung von RS-Codes die alleinige und beste Lösung.

Anhang

A. Beweis der Bijektivität der von der natürlichen Projektion induzierten Abbildung

Satz.
Es seien J und $\tilde{J}$ definiert wie in Abschnitt 1.1.3. Die natürliche Projektion

$$\pi : GF(q)[X] \longrightarrow R_n \text{ mit } f(X) \mapsto f(X) + I(X^n - 1) := \tilde{f}(X) \ ,$$

die jedes Polynom auf seine Restklasse bzgl. $I(X^n-1)$ abbildet, induziert vermöge der Zuordnung $\tilde{\pi}(I(g)) := I(\pi(g))$ die Bijektion $\tilde{\pi} : J \longrightarrow \tilde{J}$.

Beweis. Sei $I(g) \in J$, d.h. $g(X) \in GF(q)[X]$ ein normierter Teiler von $X^n - 1$. Es ist $\tilde{\pi}(I(g)) = I(\pi(g)) = I(\tilde{g})$ das Ideal in R_n, das von $\pi(g(X)) = \tilde{g} \in R_n$ erzeugt wird. Sei $I(f) \in J$ ein weiteres Ideal mit $I(g) = I(f)$ und $g(X) \neq f(X)$, d.h. $f(X)$ ist nicht der kanonische Repräsentant von $I(g)$. Dann gibt es Polynome $f'(X), g'(X) \in GF(q)[X]$, so dass

$$g(X) = g'(X) \cdot f(X) \text{ und } f(X) = f'(X) \cdot g(X) \ .$$

Da die natürliche Projektion π ein Homomorphismus von Ringen ist, folgt

$$\pi(g(X)) = \pi(g'(X)) \cdot \pi(f(X)) \text{ und } \pi(f(X)) = \pi(f'(X)) \cdot \pi(g(X)) \ ,$$

weshalb $\pi(g(X)) \in I(\pi(f))$ und $\pi(f(X)) \in I(\pi(g))$. Also ist

$$\tilde{\pi}(I(f)) = I(\tilde{f}) = I(\tilde{g}) = \tilde{\pi}(I(g))$$

und daher $\tilde{\pi}$ wohldefiniert.

Seien nun $I(\tilde{r}) = I(\tilde{s}) \in \tilde{\pi}(J) \subseteq \tilde{J}$, d.h. es existieren $\pi(a(X)) := \tilde{a}(X)$ und $\pi(b(X)) := \tilde{b}(X) \in R_n$, so dass

$$\begin{aligned} \tilde{r}(X) &= \tilde{a}(X) \cdot \tilde{s}(X) \ , \\ \tilde{s}(X) &= \tilde{b}(X) \cdot \tilde{r}(X) \ . \end{aligned}$$

Ohne Einschränkung der Allgemeinheit kann angenommen werden, dass grad $a(X)$ und grad $b(X) > 0$, andernfalls ist die Aussage trivial. Es existieren dann Polynome $g(X) \neq f(X) \in GF(q)[X]$, beide Teiler von $X^n - 1$, so dass $\pi(g(X)) = \tilde{r}(X)$ und $\pi(f(X)) = \tilde{s}(X)$ und

$$\begin{aligned} \pi(g(X)) &= \pi(a(X)) \cdot \pi(f(X)) = \pi(a(X) \cdot f(X)) \text{ und} \\ \pi(f(X)) &= \pi(b(X) \cdot g(X)) \ . \end{aligned}$$

Das bedeutet aber, dass

$$\begin{aligned} X^n - 1 \quad &| \quad g(X) - a(X)f(X) \text{ und} \\ X^n - 1 \quad &| \quad f(X) - b(X)g(X) \text{ in } GF(q)[X] \ . \end{aligned}$$

Zusammengenommen mit $g(X)|X^n-1$ und $f(X)|X^n-1$ gilt also für $l(X), l'(X) \in GF(q)[X]$:

$$\begin{aligned} g(X) &= \underbrace{\left(a(X) + l(X)\frac{X^n - 1}{f(X)}\right)}_{\in GF(q)[X]} \cdot f(X) \\ f(X) &= \underbrace{\left(b(X) + l'(X)\frac{X^n - 1}{g(X)}\right)}_{\in GF(q)[X]} \cdot g(X) \end{aligned}$$

was zeigt, dass $I(g) = I(f)$. Also ist $\tilde{\pi}$ injektiv.

$\tilde{\pi}$ ist auch surjektiv. Betrachte den Fall, dass $\tilde{g}(X)$ eine Einheit in R_n ist. Dann ist das Einselement $\tilde{e}_{R_n}(X) = 1 + I(X^n - 1)$ von R_n ein Element des Ideals $I(\tilde{g})$, weshalb mit $\tilde{g}(X) \in R_n = I(\tilde{e}_{R_n})$ die Ideale $I(\tilde{e}_{R_n}) = I(\tilde{g}) \in \tilde{J}$ identisch sind. Die natürliche Projektion π ist ein Ringhomomorphismus, deshalb wird das Einselement $e_{GF(q)[X]}(X) = 1$ von $GF(q)[x]$ auf $\pi(e_{GF(q)[X]}(X)) = \tilde{e}_{R_n}(X)$ abgebildet. Das Ideal $I(e_{GF(q)[X]}) = GF(q)[X]$ ist offensichtlich ein Element von J, so dass gilt:

$$\tilde{\pi}(I(e_{GF(q)[X]})) = I(\tilde{g}) \ .$$

Sei andererseits $I(\tilde{g}) \in \tilde{J}$ ein beliebiges Ideal mit Nullteiler $\tilde{g}(X)$ in R_n. Dann

gibt es ein $\tilde{f}(X) \in R_n$, verschieden von der Restklasse des Nullpolynoms mit

$$\tilde{g}(X) \cdot \tilde{f}(X) = 0 + I(X^n - 1) \ .$$

Deswegen $X^n - 1 | g(X) \cdot f(X) \in GF(q)[X]$. Sei $X^n - 1 = g_1(X) \cdot g_2(X) \cdot \ldots \cdot g_l(X)$ die Faktorisierung in irreduzible Faktoren über $GF(q)$. Dann gilt auch

$$g_1(X) \cdot g_2(X) \cdot \ldots \cdot g_l(X) | g(X) \cdot f(X) \ ,$$

so dass für eine Teilmenge $T \subseteq \{1, \ldots, l\}$ und $f'(X) := \prod_{t \in \{1,\ldots,l\} \setminus T} g_t(X)$

$$g'(X) := \prod_{t \in T} g_t(X) \mid g(X) \text{ und } \mathrm{ggT}\,(f'(X), g(X)) = 1$$

weshalb $I(g) \subseteq I(g')$ und sicher auch $I(\tilde{g}) \subseteq I(\tilde{g'})$. Sei $m(X) := g(X)/g'(X) \in GF(q)[X]$. Die Restklasse $\widetilde{m}(X) := m(X) + I(X^n - 1)$ ist eine Einheit in R_n, da $\mathrm{ggT}(m(X), X^n - 1) = 1$. Daher existiert $\widetilde{m}'(X) \in R_n$, so dass

$$\tilde{g'}(X) = \tilde{g}(X) \cdot \widetilde{m}'(X)$$

und daher $I(\tilde{g'}) \subseteq I(\tilde{g})$. Zusammengenommen also $I(\tilde{g'}) = I(\tilde{g})$. Nach Konstruktion von $g'(X)$ ist dieses Polynom ein Teiler von $X^n - 1$, weshalb $I(g') \in J$. Das liefert die Behauptung. ■

B. Algorithmus zur Berechnung der zyklotomischen Nebenklassen von q mod n mit *MATHEMATICA*

```
q = 2  (* Primzahlpotenz *)
n = 63   (* zwingend Teilerfremd zu q, Länge des Codes *)
```

```
Variablen
ϕ  : Eulersche Phi - Funktion
DIV  : Menge der Teiler von ϕ (n)
QMODN  : Menge der Reste von 2^l mod n, l ∈ DIV
m  :  multiplikative Ordnung von q mod n
```

```
ERROR = Catch[
          If[GCD[n, q] != 1,
                Throw["ggT(n,q)≠1: Eingabe überprüfen!"],
                {
                  ϕ = EulerPhi[n],
                  DIV = Divisors[ϕ] ,
                  QMODN = PowerMod[q, DIV, n]
                 }
             ]
              ]
```

```
Do[{
   m = Infinity,
   If[
    QMODN[[i]] == 1 , {m = DIV[[i]], Break[]}, Continue[]
      ]
    }, {i, Count[QMODN, _Integer]}
   ]
```

```
Variablen
ZNK  : Menge der zyklotomischen Nebenklassen
KONJUGIERTE[j]  :  zyklotomische Nebenklasse,  die das Element j enthält
```

```
ZNK = Union[
        Table[
          KONJUGIERTE[j] = Union[
                            Table[Mod[j * q^i, n] , {i, 0, m - 1}]
                                ], {j, 0, n - 1}
             ]
            ]
```

Abb. B1: Berechnung der ZNK mit *MATHEMATICA* am Beispiel von $q = 2$ und $n = 63$.

(Fortsetzung auf der folgenden Seite)

```
Variablen
DMENGE : Nach Größe geordnete Menge der kanonischen Repräsentanten
REPRÄSENTANT[j]  : kleinstes Element der j - ten zyklotomischen Nebenklasse
```

```
RMENGE = Union[
          Table[REPRÄSENTANT[j] = Min[KONJUGIERTE[j]], {j, 0, n - 1}]
            ]
```

```
If[GCD[n, q] != 1,
   TableForm[{ERROR}],
   Do[Print[Subscript["C", RMENGE[[j]]],
            " = ",
            KONJUGIERTE[RMENGE[[j]]]]
      , {j, 1, Count[RMENGE, _Integer]}
     ]
  ]
```

```
C0 = {0}
C1 = {1, 2, 4, 8, 16, 32}
C3 = {3, 6, 12, 24, 33, 48}
C5 = {5, 10, 17, 20, 34, 40}
C7 = {7, 14, 28, 35, 49, 56}
C9 = {9, 18, 36}
C11 = {11, 22, 25, 37, 44, 50}
C13 = {13, 19, 26, 38, 41, 52}
C15 = {15, 30, 39, 51, 57, 60}
C21 = {21, 42}
C23 = {23, 29, 43, 46, 53, 58}
C27 = {27, 45, 54}
C31 = {31, 47, 55, 59, 61, 62}
```

Abb. B1: (Fortsetzung) Berechnung der ZNK mit *MATHEMATICA* am Beispiel von $q = 2$ und $n = 63$.

Wie in Abschnitt 1.2.2 besprochen, werden als Eingabeparameter für die Berechnung der zyklotomischen Nebenklassen die Körperordnung von $GF(q)$ und der Grad des Polynoms $X^n - 1$ gebraucht.
Der Algorithmus berechnet zunächst genauso wie in Abschnitt 2.2.1 die multiplikative Ordnung von $q \bmod n$, da jede ZNK höchstens $m_n(q)$ Elemente besitzen kann (vgl. Bemerkung 1.46). Sodann wird zu jedem $0 \leq j \leq n-1$ die zugehörige ZNK durch explizites Ausrechnen aller möglichen Elemente $j \cdot q^i \bmod n$ für

$0 \leq i \leq m_n(q) - 1$ aufgestellt. Da sowohl einige Elemente in einigen ZNK als auch viele ZNK in der Menge der ZNK mehrfach auftreten, werden durch den Befehl „*Union*“ nur die verschiedenen Elemente von Mengen gespeichert.

Im nächsten Programmblock werden die kanonischen Repräsentanten zu jeder ZNK berechnet und dem Bezeichner *RMENGE* zugewiesen. Der letzte Programmblock bereitet die Daten für eine übersichtliche Ausgabe auf. Wurde bei der Eingabe $\mathrm{ggT}(n, q) = 1$ missachtet, gibt das in *MATHEMATICA* implementierte Programm eine Fehlermeldung aus.

C. „Exhaustive Search“ mit *MAGMA*

```
G2:= GF(2);
alpha:=RootOfUnity(17,G2);
G256< beta >:= Parent(alpha);
df< x >:=DefiningPolynomial(G256);
E17:= { alpha^i : i in [0..16]} ;
FelTmE17:= SetToIndexedSet(Subsets(E17,5));
t:=func< i | SetToIndexedSet(FelTmE17[i]) >;
Loes:={@ @};
P < Z >:=PolynomialAlgebra(G256);

print “17.Einheitswurzeln liegen in: “, G256;
print “Primitives Polynom von GF(256) mit primitiver Nullstelle beta: “;
print “pi(X)=“, df;
print “Primitive 17. Einheitswurzel: “ , “alpha = “, alpha;
print “E17 = “ , E17;
print “Anzahl der 5 elementigen Teilmengen von E17: “, #FelTmE17;

for k in [1..#FelTmE17] do
        A:=func< j | &+ [t(j)[i]^3 :  i  in [1..5]] >;
        sigma := 1 + A(k)^97 * Z^2 + A(k) * Z^3 + (A(k)^7 + A(k)^160) * Z^4
                    + (A(k)^64 + A(k)^98) * Z^5;
        NST:=Roots(sigma);
        if #NST eq 5 then
              if &and[NST[j][1] in E17 : j in [1..5]] then
                 Include(~Loes, { @ NST[1][1], NST[2][1], NST[3][1],
                                          NST[4][1], NST[5][1] @});
              end if;
        end if;
end for;
print “Menge der Lösungs 5-Tupel der Gleichung sigma(Z)= 0 mit fünf
          paarweise verschiedenen Nullstellen: “;

print Loes;
print “Anzahl der Loesungen: “ , #Loes;
```

Abb. C1: Suche nach den Lokatoren von Codewörtern c mit $\mathrm{wt}(c) = 5$ in einem BCH-Code mit Parametern $n = 17$, $b = 1$ und $\delta = 3$ über $GF(2)$.

```
Lok:={@ @};

for k in [1..#Loes] do
        Include(~Lok, {Log(1/Loes[k][1])/Log(alpha), Log(1/Loes[k][2])/Log(alpha),
                       Log(1/Loes[k][3])/Log(alpha), Log(1/Loes[k][4])/Log(alpha),
                       Log(1/Loes[k][5])/Log(alpha)});
end for;

print "Menge der Lokatoren 5-Tupel als Exponenten der primitiven
        n-ten Einheitswurzel alpha";
Lok;
```

Abb. C1: (Fortsetzung) Suche nach den Lokatoren von Codewörtern c mit $\mathrm{wt}(c) = 5$ in einem BCH-Code mit Parametern $n = 17$, $b = 1$ und $\delta = 3$ über $GF(2)$.

Das in *Magma* implementierte Programm ist nur in der Lage die Lokatoren aller Codewörter vom Gewicht $\mathrm{wt}(c) = 5$ des BCH-Codes im engeren Sinne, der Länge $n = 17$ und Entwurfsdistanz $\delta = 3$ über $GF(2)$ zu berechnen. Es ist aber leicht den Algorithmus für beliebige Fälle zu verallgemeinern.

Der Anfang des Algorithmus besteht aus den grundlegenden Definitionen. *G2* bezeichnet den endlichen Körper mit zwei Elementen und *alpha* eine primitive 17. Einheitswurzel über *G2*. *G256* definiert den Zerfällungskörper von $X^{17} - 1 \in GF(2)[X]$ und *df<x>* sein primitives Polynom. Mit *E17* ist die Menge der 17. Einheitswurzeln gemeint, die durch die Potenzieren von *alpha* erzeugt wird. Berechnungen mit *alpha* werden immer in seiner Darstellung als Potenz des primitiven Elements *beta* von *G256* durchgeführt ($alpha = beta^{15}$). *FeTmE17* ist die Menge aller fünfelementigen Teilmengen von *E17*. Die Variable t ist als eine Funktion definiert, die den Zugriff auf alle Elementmengen von *FeTmE17* ermöglicht. *Loes* ist ein Bezeichner einer leeren Menge, in die bei Programmablauf fünfelementige Elementmengen geschrieben werden sollen. Die Elemente dieser Elementmengen sind die jeweils paarweise verschiedenen Nullstellen der Gleichung

$$\sigma_c(Z) = 1 + A_3^{97} Z^2 + A_3 Z^3 + (A_3^7 + A_3^{160}) Z^4 + (A_3^7 + A_3^{160}) Z^5 = 0 \ . \qquad \text{(a)}$$

Die Definition *P<Z>* bezeichnet die Polynomalgebra mit Koeffizienten aus *G256* in der Unbestimmten Z. Sie ist essentiell, um das Lokator-Polynom $\sigma_c(Z)$ über *G256* zu implementieren.

Der anschließende Block veranlasst *MAGMA* einige Informationen über die Definitionen auszugeben.
Darauf folgt der Kernteil des Algorithmus. Es ist eine Schleife, die alle 6188 Elementmengen von *FeTmE17* nacheinander in (a) einsetzt und überprüft, ob mit diesen 17. Einheitswurzeln $\sigma_c(Z)$ über *G256* fünf paarweise verschiedene Nullstellen in *E17* besitzt. Ist dies der Fall, wird durch die folgenden Befehle die Menge der Nullstellen von $\sigma_c(Z)$ generiert und in *Loes* gespeichert.
Der letzte Programmblock bildet zu den einzelnen Nullstellen einer Elementmenge von *Loes* die reziproken 17. Einheitswurzeln, die bisher als Potenzen des primitiven Elements *beta* von *G256* behandelt wurden. Von diesen werden nun durch den Befehl „Log()“ die Exponenten isoliert und durch „Log(*alpha*)“ geteilt. „Log(*alpha*)“ ist der Exponent von *alpha* in der Darstellung durch *beta*.
Die Ergebnisse werden in neuen fünfelementigen Mengen gespeichert, welche während des Programmablaufes in der Menge *Lok* gesammelt werden. Die Elementmengen von *Lok* enthalten die Exponenten der Lokatoren von gesuchten Codewörtern, die die Positionen an denen ein Codewort vom Gewicht $\mathrm{wt}(c) = 5$ den Eintrag „1“ innehat, anzeigt. Zuletzt wird die Ausgabe der Menge *Lok* veranlasst.

Die Berechnung der Lösungen benötigte auf dem in Abschnitt 2.2.2 beschriebenen Computersystem nur wenige Sekunden. Die Ergebnisse sind auf den Seiten 61 und 62 abgedruckt.

Literaturverzeichnis

[Augot et al. 1991] D. Augot, P. Charpin and N. Sendrier, *The minimum distance of some binary codes via the Newton's identities*, in *Eurocode '90 (Udine 1990)*, 65–73, Lecture Notes in Comput. Sci. 514, Springer, Berlin, 1991

[Augot et al. 1992] D. Augot, P. Charpin und N. Sendrier, *Studying the locator polynomials of minimum weight codewords of BCH codes*, IEEE Trans. Inform. Theory **38** (1992), Nr. 3, 960–973

[Augot et al. 1994] D. Augot und N. Sendrier, *Idempotents and the BCH-Bound*, IEEE Trans. Inform. Theory **40** (1994), Nr. 1, 204–207

[Berlekamp 1968] E. R. Berlekamp, *Algebraic Coding Theory*, McGraw-Hill, New York 1968

[Betten et al. 1998] A. Betten, H. Fripertinger, A. Kerber, A. Wassermann und K. - H. Zimmermann, *Codierungstheorie – Konstruktion und Anwendung linearer Codes*, Springer, Berlin 1998

[Bosch 1993] S. Bosch, *Algebra*, Springer, Berlin 1993

[Bose & Chaudhuri 1960] R. C. Bose und D. K. Ray-Chaudhuri, *On a class of Error Correcting Binary Group Codes*, Information and Control **3** (März 1960), 68–79

[Bossert 1992] M. Bossert, *Kanalcodierung*, Teubner, Stuttgart 1992

[Canteaut et al. 1998] A. Canteaut and F. Chabaud, *A new algorithm for finding minimum-weight words in a linear code*, IEEE Trans. Inform. Theory **44** (1998), Nr. 1, 367–378

[Carasso et al. 1982] M. G. Carasso, J. B. H. Peek und J. P. Sinjou, *The Compact Disc Digital Audio system*, Philips Technical Review **40** (1982), Nr. 6, 151–156

[Charpin 1990] P. Charpin, *On a class of primitive BCH-codes*, IEEE Trans. Inform. Theory **36** (1990), Nr.. 1, 222–228

[Crandall et al. 1997] R. Crandall, K. Dilcher und C. Pomerance, *A Search For Wieferich And Wilson Primes*, Math. Comp. **66** (1997), Nr. 217, 433–449

[Driessen et al. 1982] L. M. H. Driessen und L. B. Vries, *Performance calculations of the Compact Disc error correcting code on a memoryless channel*, in *4th Int. Conf. on Video and data recording (Southhampton 1982)*, IERE Conf. Proc. **54**, 385–395

[Fischer et al. 1986] G. Fischer, *Lineare Algebra*, Vieweg, Braunschweig 1986

[Geddes et al. 1992] K.O. Geddes, S.R. Czapor und G. Labahn, *Algorithms for Computeralgebra*, Kluwer Academic Publishers, Dordrecht 1992

[Hocquenghem 1959] A. Hocquenghem, *Codes correcteurs d'erreurs*, Chiffres **2** (1959), 147–156

[Hoeve et al. 1982] H. Hoeve, J. Timmermans und L. B. Vries, *Error correction ans concealment in the Compact Disc system*, Philips Technical Review **40** (1982), Nr. 6, 166–173

[Kasami et al. 1969] T. Kasami, N. Tokura, *Some remarks on BCH bounds and minimum weights of binary primitive BCH codes*, IEEE Trans. Inform. Theory **15** (1969), Nr. 3, 408–413

[van Lint et al. 1986] J. H. van Lint and R. M. Wilson, *On the minimum distance of cyclic codes*, IEEE Trans. Inform. Theory **32** (1986), Nr. 1, 23–40

[MacWilliams et al. 1977] F. J. MacWilliams und N. J. A. Sloane, *The theory of error-correcting codes. I*, North-Holland, Amsterdam 1977

[Peterson et al. 1967] W. W. Peterson, *On the weight structure and symmetry of BCH Codes*, J. IECE Japan **50** (1967), 1183–1190

[Peterson et al. 1972] W. W. Peterson und E. J. Weldon Jr., *Error-Correcting Codes*, The MIT Press, London 1972

[Pless et al. 1998] V. S. Pless, W. C. Huffman R. A. Brualdi (Hrsg.), *Handbook of Coding Theory, Volume I*, Elsevier Science, Amsterdam 1998

[Reed & Solomon 1960] I. S. Reed und G. Solomon, *Polynomial Codes over certain finite fields*, SIAM Journal of Applied Mathematics **8** (1960), 300–304

[Reiffen et al. 1984] H.-J. Reiffen, G. Scheja und U. Vetter, *Algebra*, Bibliographisches Inst., Mannheim 1984

[Remmert et al. 1995] R. Remmert und P. Ullrich, *Elementare Zahlentheorie*, Birkhäuser, Berlin 1995

[Wambach 1993] G. Wambach, *The True Minimum Distance of Some Narrow-Sense BCH-Codes of Length 255*, Fakultät für angewandte Mathematik und Informatik der Universität Köln, Report Nr. zpr93.144 (1993), Stand Oktober 2000: http://www.zpr.uni-koeln.de/~ paper /paper.php3?paper=144

[Wicker et al. 1994] S. B. Wicker und V. K. Bhargava (Hrsg.), *Reed-Solomon Codes and Their Applications*, IEEE Press, Piscataway 1994