Majority-Logic-Decodierung für Euklidische-Geometrie-Codes

Dissertation

der Mathematisch-Naturwissenschaftlichen Fakultät
der Eberhard Karls Universität Tübingen
zur Erlangung des Grades eines
Doktors der Naturwissenschaften
(Dr. rer. nat.)

vorgelegt von
Frau Dipl.-Math. Juliane Bertram
aus Magdeburg

Tübingen
2018

Gedruckt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät
der Eberhard Karls Universität Tübingen.

Tag der mündlichen Qualifikation:	27. Juli 2018
Dekan:	Prof. Dr. Wolfgang Rosenstiel
1. Berichterstatter:	Prof. Dr. Peter Hauck
2. Berichterstatter:	Prof. Dr. Michael Huber

Danksagung

Ich danke meinem Doktorvater, Herrn Prof. Dr. Peter Hauck, für seine gerechte, sachliche Art, seine scharfsinnige und konstruktive Kritik, sein offenes Ohr während all der Zeit, was mir immer eine große Unterstützung war. Ich danke ihm herzlichst.

Abstract

Diese Arbeit befasst sich mit Majority-Logic-Decodieralgorithmen für Euklidische-Geometrie-Codes. Diese Verfahren zeichnen sich dadurch aus, auf Hardwareebene in Echtzeit unter Verteilung des Rechenaufwands auf mehrere Prozessoren decodieren zu können. Das Ziel der vorliegenden Dissertation ist es, die bestehenden Majority-Logic-Decodierverfahren, insbesondere den Reed-Algorithmus, hinsichtlich der Performanz zu verbessern beziehungsweise neue, effizientere Verfahren zu entwickeln. Wir werden zwei neue Algorithmen vorstellen, bei denen die Anzahl der auszuführenden Mehrheitsentscheidungen signifikant reduziert ist. Einer der beiden Algorithmen basiert wie jener von Reed einzig auf Mehrheitsentscheidungen. Der andere Algorithmus verwendet zusätzlich Additionen bzw. Subtraktionen, so dass weniger Mehrheitsentscheidungen als bei den anderen beiden Algorithmen getroffen werden müssen. Darüber hinaus haben wir eine neue Abstufung konstruiert, mit der wir – unabhängig vom verwendeten Decodierverfahren – mindestens die gleichen oder bessere Ergebnisse als Chen und Reed erzielen, so dass diese aus Gründen der Performanz stets vorzuziehen ist.

Die vorliegende Dissertation enthält zudem eine genaue Analyse des Aufwands der Majority-Logic-Decodierverfahren, einschließlich des Reed-Algorithmus, angewandt auf verschiedene Codeklassen wie Hamming-Codes, Reed-Muller-Codes, Euklidische-Geometrie-Codes sowie zweifache Euklidische-Geometrie-Codes. Darauf basierend sprechen wir Empfehlungen aus, welche Codes mit welcher Parameterwahl (bei gleichen Fehlerkorrektureigenschaften) die höchste Performanz bieten.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Liste der Algorithmen

Symbolverzeichnis

$[n, k, d]_{q_C}$-Code ... ein linearer Code über $\mathbb{F}_{q_C}$ der Länge n, der Dimension k und mit Minimaldistanz d, Seite 34

$\langle U \rangle$ ..... Vektorraumerzeugnis von U, Seite 15

$|S|$ ..... Kardinalität einer Menge S, Seite 14

$|x|$ ..... Absolutbetrag von x, Seite 14

$\arg\max_{s \in S} f(s)$ .. Menge der Argumente, die die Funktion f maximieren, Seite 18

$\lceil x \rceil$ ..... kleinste ganze Zahl größer oder gleich x, Seite 14

χ_V ..... Inzidenzvektor zu $V \subseteq \mathbb{F}_q^m$, Seite 47

k ..... Dimension des Codes $\mathcal{C}$, Seite 35

$\mathtt{C}$ ..... Codewort, Seite 35

δ ..... Hamming-Metrik, Seite 16

$\dim(U)$ ..... Dimension des Untervektorraums U, Seite 15

d ..... Minimaldistanz des Codes $\mathcal{C}$, Seite 35

$\mathbf{e}_i$ ..... kanonischer i-ter Einheitsvektor des $\mathbb{F}_q^m$, Seite 16

$\mathtt{E}$ ..... Fehlerwort, Seite 36

$\mathtt{E} \circ \chi_V$ ..... Fehlersumme von $V \subseteq \mathbb{F}_q^m$, Seite 47

$\mathtt{E}_i$ ..... i-tes Fehlersymbol, Seite 36

$\mathbb{F}_q$ ..... Körper mit q Elementen, Seite 15

$\mathbb{F}_q^*$ ..... Menge der Einheiten in $\mathbb{F}_q$, also $\mathbb{F}_q \setminus \{0\}$, Seite 15

$\mathbb{F}_q^m$ ..... $\mathbb{F}_q$-Vektorraum der Dimension m, Seite 15

$\lfloor x \rfloor$ ..... größte ganze Zahl kleiner oder gleich x, Seite 14

$\mathtt{G}$ ..... Erzeugermatix von $\mathcal{C}$, Seite 35

$\mathrm{ggT}\,(a, b)$ ..... größter gemeinsamer Teiler von a und b, Seite 14

$\mathtt{I}$ ..... Information, Seite 35

n ..... Länge des Codes $\mathcal{C}$, Seite 35

$\mathrm{Mat}(m, n, \mathbb{F}_q)$ .... Menge der $(m \times n)$-Matrizen mit Elementen aus $\mathbb{F}_q$, Seite 17

$\mathcal{A}_{D,m,q}$ ..... Menge der affinen D-dimensionalen Unterräume des $\mathbb{F}_q^m$, Seite 17

$\mathcal{A}^*_{D,m,q}$ ..... Menge der affinen D-dimensionalen Unterräume des $\mathbb{F}_q^m$, die nicht die Null enthalten, Seite 17

$\mathcal{C}$ ..... $[n, k, d]_{q_\mathcal{C}}$-Code, Seite 35

$\mathcal{P}(S)$ ..... Potenzmenge von S, Seite 14

$\mathbb{1}_T$ ..... Wahrheitswert der Aussage T, Seite 18

$a \bmod b$ ..... ganzzahliger Rest der Ganzzahldivision a durch b, Seite 14

μ^Γ ..... Majoritätsfunktion, Seite 19

$\mu_{(s_i)_{i=0}^{\eta-1}}$ ..... Menge der Mehrheiten bzgl. $(s_i)_{i=0}^{\eta-1}$, Seite 18

$\mathbb{N}$ ..... Menge der natürlichen Zahlen, Seite 14

$\mathbb{N}_0$ ..... Menge der nichtnegativen ganzen Zahlen, Seite 14

$\omega(\mathbf{v})$ ..... Gewicht von $\mathbf{v}$, Seite 16

$\omega_b(a)$ ..... Gewicht von a bzgl. b, Seite 14

$\mathbb{R}$ ..... Menge der reellen Zahlen, Seite 14

$\mathbb{R}^+$ ..... Menge der nichtnegativen reellen Zahlen, Seite 14

$\operatorname{sgn} x$ ..... Signum von x, Seite 14

$\operatorname{supp} \mathbf{v}$ ..... Träger von $\mathbf{v}$, Seite 16

τ ..... Fehleranzahl, Seite 36

$\tau_\mathbf{v}$ ..... Anzahl der Fehler in $\mathbf{v} \in \mathbb{F}_{q_\mathcal{C}}^n$, Seite 38

$\Theta(g)$ ..... Θ-Notation, Seite 19

$\mathbf{v} \circ \mathbf{w}$ ..... symmetrische Bilinearform, Seite 16

$\mathtt{z}$ ..... Wort, Seite 36

$\mathtt{z} \circ \chi_V$ ..... Checksumme von $V \subseteq \mathbb{F}_q^m$, Seite 47

$\mathbb{Z}$ ..... Menge der ganzen Zahlen, Seite 14

$\mathbb{Z}_N$ ..... Menge $\{0, 1, \ldots, N-1\} \subset \mathbb{Z}$, Seite 15

$a \mid b$ ..... a teilt b in $\mathbb{Z}$, Seite 14

$a' \equiv a \bmod b$ ..... $(a - a')$ teilt b in $\mathbb{Z}$, Seite 14

c_i ..... i-tes Codewortsymbol, Seite 35

$O(g)$ ..... O-Notation, Seite 19

$U \leq \mathbb{F}_q^m$ ..... U Untervektorraum des $\mathbb{F}_q^m$, Seite 15

z_i ..... i-tes Wortsymbol, Seite 36
$\mathcal{C}^\perp$ ..... Dualcode von $\mathcal{C}$, Seite 34

Kapitel 1

Einführung

1.1 Einleitung und Zielsetzung

Beim Übertragen von Daten über einen gestörten Kanal ist nicht auszuschließen, dass der Empfänger inkorrekte Daten erhält. Die Codierungstheorie beschäftigt sich damit, auftretende Fehler zu erkennen und gegebenenfalls zu korrigieren. Eine Strategie, dieses Problem zu lösen, ist die sogenannte *Vorwärtsfehlerkorrektur*. Bei dieser ergänzt der Sender die zu übertragenden Informationswörter nach einer Codierungsvorschrift zu Codewörtern. Das aus einem Informationswort generierte Codewort wird an den Empfänger gesendet. Der Empfänger kann das erhaltene Wort nun mittels eines geeigneten Algorithmus decodieren, um das ursprünglich übertragene Codewort oder die darin enthaltene Information zu rekonstruieren.

Der Vorwärtsfehlerkorrektur liegt folgende Überlegung zugrunde. Jedes Codewort enthält neben der eigentlichen Information eine gewisse Redundanz. Als Folge unterscheiden sich zwei Codewörter eines Codes an einer durch die Codierungsvorschrift vorgegebenen Anzahl von Positionen, *Minimaldistanz* d genannt, oder an mehr Positionen voneinander. Je größer die Minimaldistanz d, desto mehr Fehler können korrigiert werden. Denn beträgt die Anzahl der Fehler weniger als die Hälfte der Minimaldistanz, so kann das ursprünglich übertragene Codewort eindeutig rekonstruiert werden, indem gemäß dem Prinzip

der *Hamming-Decodierung* das zu $\mathtt{z}$ nächstgelegene Codewort bestimmt wird. Die durch den Code definierte Fehlerkorrekturgrenze liegt daher bei

$$\tau_{\max} \coloneqq \left\lfloor \frac{d-1}{2} \right\rfloor .$$

Die Redundanz ist somit eine Voraussetzung für die Vorwärtsfehlerkorrektur. Die eigentliche Korrektur leistet der Decodieralgorithmus, der auf den Code zugeschnitten sein muss. Die Anzahl der korrigierbaren Fehler hängt daher sowohl vom Code als auch vom Decodieralgorithmus ab. Der Zielkonflikt besteht darin, möglichst wenig Redundanz hinzuzufügen, aber dabei möglichst viele Fehler bei möglichst geringem Decodieraufwand zu korrigieren.

In dieser Arbeit werden wir uns auf *Majority-Logic-Decodieralgorithmen* für Euklidische-Geometrie-Codes fokussieren. Diese basieren auf dem Prinzip der Mehrheitsentscheidung. Eine Mehrheitsentscheidung kann mittels eines Majoritätsgatters (*majority gate*) oder allgemeiner eines Schwellen(wert)gatters (*threshold gate*) in Hardware realisiert werden. Bei Schwellengattern hängt das Ausgangssignal davon ab, ob die (gewichtete) Summe der Eingangssignale einen gegebenen Schwellenwert erreicht. Majoritätsgatter sind spezielle Schwellengatter, bei denen der Wert ausgegeben wird, der der Majorität der Eingangssignale entspricht. (Bei einer Parität wird in der Regel 0 ausgegeben.)

Unser Ziel ist es, die bestehenden Majority-Logic-Decodierverfahren zu verbessern beziehungsweise neue, performantere zu entwickeln. Es gibt andere Decodierverfahren mit teilweise besseren Fehlerkorrektureigenschaften, die unterhalb der durch den Code vorgegebenen Fehlerkorrekturgrenze in Höhe der halben Minimaldistanz stets korrekt arbeiten. Der Viterbi-Algorithmus [36] ist ein Verfahren, bei dem die Restfehlerwahrscheinlichkeit[1] zulasten der Komplexität minimiert werden kann. Er basiert darauf, den Code mit Hilfe von Trellis-Diagrammen darzustellen [22, Chapter 9]. Als Hard-Decision-Decoder (ohne Zuverlässigkeitsinformation) [6, §7.3.4] über einen binären, symmetrischen Kanal oder als Soft-Decision-Decoder (mit Zuverlässigkeitsinformation) [22, Chapter 12.1] implementiert, setzt der Viterbi-Algorithmus in bei-

[1] Die Restfehlerwahrscheinlichkeit ist die Wahrscheinlichkeit, dass ein decodiertes Codewort nicht dem ursprünglich übertragenen Codewort entspricht [6, Definition 1.15].

den Fällen das Prinzip der Maximum-Likelihood-Decodierung[2] um. Der DA-Algorithmus [6, S.186 ff.] ist ein Decodierverfahren für binäre lineare Codes, der unter gewissen Bedingungen auch dann korrekt decodiert, wenn die Anzahl der Fehler mehr als die halbe Minimaldistanz beträgt. In der Wissenschaft kehrte man sich für einige Jahre von den Majority-Logic-Decodierverfahren ab. Allerdings zeichnen sich die Majority-Logic-Decodierverfahren dadurch aus, auf Hardwareebene in Echtzeit unter Verteilung des Rechenaufwands auf mehrere Prozessoren decodieren zu können. Diese drei Themengebiete, Echtzeitberechnung, parallele Verarbeitung und hardwarenahe Realisierung, gewinnen in den letzten Jahren durch die zunehmende Vernetzung im Sinne der digitalen Transformation immer mehr an Bedeutung. Als Beispiele sind in der Automobilindustrie das vernetzte Fahrzeug (*connected car*) und Fahrerassistenzsysteme sowie im Finanzwesen Bitcoin Mining ASICs [35] zu nennen.

Die Majority-Logic-Decodierung wird in der Literatur vornehmlich theoretisch betrachtet: Es wird für einzelne Codeklassen bewiesen, dass man an jeder beliebigen Position mit Hilfe von Mehrheitsentscheidungen decodieren kann. Eine Aufwandsabschätzung findet in der Regel nicht statt. Die verschiedenen Codeklassen werden bestenfalls sehr allgemein hinsichtlich der Fehlerkorrektureigenschaften miteinander verglichen. Ein Vergleich derselben bezüglich der Decodierkomplexität bleibt aus. Dies nehmen wir zum Anlass, für jedes der von uns betrachteten Verfahren die Anzahl der auszuführenden Operationen konkret anzugeben – zum einen allgemein für alle Codeklassen übergreifend, zum anderen explizit für jede einzelne Codeklasse formuliert. Diese detaillierte Analyse der Komplexität erlaubt es uns Empfehlungen auszusprechen, welche Codes mit welchen Parametern (bei gleichen Fehlerkorrektureigenschaften) gewählt werden sollten, um die Performanz zu erhöhen.

Die Ergebnisse der Komplexitätsuntersuchung zeigen unmittelbar auf, dass eine Vielzahl von Mehrheitsentscheidungen bei der klassischen Majority-Logic-Decodierung benötigt werden. Daher ist ein Ziel dieser Arbeit, die Decodierung dahingehend zu verbessern, den Bedarf an Mehrheitsentscheidungen zu sen-

[2] Bei der Maximum-Likelihood-Decodierung wird jenes Codewort c des Codes ausgegeben, bei dem die Wahrscheinlichkeit maximal ist, das empfangene Wort $\mathtt{z}$ unter der Bedingung, c wurde zuvor gesendet, zu erhalten [15, §1.11.2].

ken. Tatsächlich stellen wir in dieser Arbeit neben dem klassischen Verfahren zwei neu entwickelte Algorithmen vor, bei denen die Anzahl der auszuführenden Mehrheitsentscheidungen signifikant reduziert ist. Einer der beiden neuen Algorithmen basiert wie das klassische Verfahren nur auf Mehrheitsentscheidungen. Der andere Algorithmus verwendet zusätzlich Additionen bzw. Subtraktionen, so dass noch weniger Mehrheitsentscheidungen getroffen werden müssen.

Alle drei Algorithmen eint, dass eine Abstufung vorgegeben werden muss, welche zu durchlaufen ist. Wir zeigen, dass diese Abstufung einen weiteren Ansatzpunkt bietet, um zu optimieren. Drei Abstufungen sind aus der Literatur bekannt: jene nach Reed, jene nach Chen und eine von Peterson und Weldon in [29, §10.4, S. 337 f.] präsentierte. Einen neuen Typ werden wir in dieser Arbeit vorstellen. Wir werden formal beweisen und graphisch veranschaulichen, dass jede der drei bekannten Abstufungen je nach Wahl der Parameter weniger oder bestenfalls gleich performant ist wie die von uns entwickelte *invertierte Abstufung*. Die Fehlerkorrektureigenschaften und die parallele Laufzeit jedoch sind bei allen Abstufungen identisch.

Die Arbeit ist folgendermaßen aufgebaut. In Kapitel 2 führen wir die Notationen und Definitionen ein, die wir im Folgenden verwenden werden. Da die Majoritätsfunktion einen großen Stellenwert bei der Majority-Logic-Decodierung einnimmt, widmen wir ihr ein eigenes Kapitel, Kapitel 3, in welchem wir Eigenschaften und Implementierung beleuchten. Anschließend legen wir in Kapitel 4 die Grundlagen der Codierungstheorie und das Prinzip der Majority-Logic-Decodierung dar. Dieses Kapitel schließen wir damit ab, dass wir in Abschnitt 4.4 verschiedene Möglichkeiten beschreiben, wie die Majority-Logic-Decodierung optimiert werden kann.

In Kapitel 5 wird das in Abschnitt 4.3 beschriebene Prinzip dazu genutzt, drei Decodierverfahren für Codes auszuformulieren, die sich über Euklidische Geometrien definieren: die klassische (Abschnitt 5.2), verbesserte (Abschnitt 5.3) und hybride (Abschnitt 5.4) Decodierung. Ersteres lehnt sich auf einfache Weise an das vorgestellte Prinzip an. Das zweite Verfahren, die verbesserte Decodierung, basiert auf der Idee, Nebenklassen von Unterräumen zu betrachten

und für jede nur jeweils eine einzige Mehrheitsentscheidung zu treffen. Das letzte Verfahren baut auf die verbesserte Decodierung auf und nutzt weitere Struktureigenschaften von affinen Räumen aus. In diesem Verfahren werden neben Mehrheitsentscheidungen auch Additionen/Subtraktionen ausgeführt, weshalb wir es als das hybride Decodierverfahren bezeichnen. Wie bereits erwähnt, ist allen drei Verfahren gemein, dass sie auf einer vorgegebenen Abstufung basieren. Diese spezifizieren wir erst im folgenden Kapitel näher.

In Kapitel 6 stellen wir drei mögliche Abstufungen vor, jene nach Reed, jene nach Chen und die von uns entwickelte. Abhängig von der gewählten Abstufung konkretisieren wir die Anzahl der korrigierbaren Fehler und den Decodieraufwand der Algorithmen aus Kapitel 5.

Wir zeigen in Kapitel 7 für verschiedene aus der Literatur bekannte Codeklassen auf, wie die in Kapitel 5 präsentierten Decodierverfahren eingesetzt werden können. Dazu repetieren wir für jede Codeklasse die formale Definition einschließlich einiger Eigenschaften wie Länge, Informationsrate, Minimaldistanz. Darüber hinaus zeigen wir auf, wie viele Fehler unter welchem Aufwand mit den Algorithmen aus Kapitel 5 korrigiert werden können. Diese detaillierte Aufschlüsselung der Fehlerkorrektureigenschaften und des Aufwands erlaubt uns, eine Empfehlung abzugeben, welcher konkrete Code bei gegebenen Eigenschaften zu wählen ist, um möglichst performant zu decodieren. Die Arbeit schließen wir mit einer Zusammenfassung und einem Ausblick auf offene Fragen in Kapitel 8 ab.

1.2 Euklidische-Geometrie-Codes und Majority-Logic-Decodierung in der wissenschaftlichen Literatur

Majority-Logic-Decodieralgorithmen können auf eine Reihe von Codes angewandt werden. In den 50er Jahren des zwanzigsten Jahrhunderts wurden die ersten fehlerkorrigierenden linearen Blockcodes betrachtet. Zum einen war das 1950 die Klasse der Hamming-Codes [12], die nach ihrem Entwickler Richard

W. Hamming benannt ist. Etwas später, im Jahr 1954, entwarf David E. Muller eine neue Klasse von Codes, zunächst mit dem Ziel, Fehler zu detektieren [26]. Irwin S. Reed zeigte im gleichen Jahr auf, wie die Klasse dieser Codes zur Fehlerkorrektur eingesetzt werden kann, indem er ein Decodierverfahren basierend auf Majority-Logic entwickelte [30]. Konsequenterweise wird diese Codeklasse als Reed-Muller-Codes bezeichnet und das dazugehörige Decodierverfahren als Reed-Algorithmus. Beide Codeklassen, die Hamming-Codes und die Reed-Muller-Codes, betrachten wir in Kapitel 7 näher. Auf den Reed-Algorithmus gehen wir in Abschnitt 6.1.1 ein.

Der originäre Reed-Algorithmus für Reed-Muller-Codes funktioniert in aller Kürze wie folgt. Der Algorithmus beginnt damit, Aussagen zu treffen hinsichtlich der Anzahl der Positionen, an denen ein Codewort $\mathtt{c}^\perp$ des Dualcodes eine eins aufweist und an denen beim Übertragen des ursprünglichen Codeworts ein Fehler aufgetreten ist. Diese Anzahl von Positionen bezeichnen wir auch als Fehleranzahl von $\mathtt{c}^\perp$, $\tau_{\mathtt{c}^\perp}$. Mit Hilfe des empfangenen Worts $\mathtt{z}$ wird eine sogenannte *Checksumme* von $\mathtt{c}^\perp$ berechnet. Wenn der Wert der Checksumme null ist, so ist $\tau_{\mathtt{c}^\perp}$, gerade; wenn der Wert der Checksumme eins ist, ist $\tau_{\mathtt{c}^\perp}$ ungerade.

Angenommen, S sei eine Teilmenge des Dualcodes mit der Eigenschaft der *Orthogonalität*: An jeder Position weist höchstens ein Wort oder weisen alle Wörter aus S eine eins auf (siehe auch Definition auf S. 16). Mit anderen Worten, der paarweise Schnitt der Träger von Wörtern aus S ist fest. Es sei $\mathbf{v}$ jenes Wort, das nur an den Positionen dieses (paarweisen) Schnitts der Träger eine eins besitzt. Der Reed-Algorithmus schätzt mittels einer Mehrheitsentscheidung die Fehleranzahl von $\mathbf{v}$, $\tau_{\mathbf{v}}$, genau dann als gerade ein, wenn mindestens die Hälfte der $\tau_{\mathtt{c}^\perp}$, $\mathtt{c}^\perp \in S$ gerade ist (nähere Erläuterungen in Proposition 4.3.1).

Enthält der Vektor $\mathbf{v}$ mehr als einen Eintrag mit eins, so wird obiger Schritt vom Reed-Algorithmus für weitere orthogonale Teilmengen des Dualcodes wiederholt. In diesem Fall liegt für verschiedene Wörter vor, ob deren Fehleranzahl gerade oder ungerade ist. Sind diese Wörter orthogonal zueinander, so kann erneut eine Mehrheitsentscheidung ausgeführt werden. Sukzessive werden die

Träger der Vektoren kleiner, deren Fehleranzahl bestimmt wird. Weist der Vektor $\mathbf{v}$ nur an einer Position eine eins auf, ist bekannt, ob an dieser Position ein Fehler aufgetreten ist oder nicht. Die Rekonstruktion des Codeworts an dieser Position kann erfolgen.

Der Reed-Algorithmus bestimmt also in mehreren Stufen mittels Mehrheitsentscheidungen, ob die Fehleranzahl bestimmter Wörter gerade oder ungerade ist. In jeder Stufe wird der Träger dieser Wörter kleiner. Der Algorithmus terminiert, sobald der Träger nur noch aus einer Position besteht. Speziell beim Reed-Muller Code RM(r, m) ist jeder Träger mit einem affinen Raum assoziiert. Die Größe des Trägers halbiert sich in jeder Stufe, beginnend bei 2^{r+1}, so dass insgesamt $r + 1$ Majority-Logic-Stufen benötigt werden.

Reed-Muller-Codes sind einfach zu konstruieren und weisen eine mathematische Struktur auf, die systematisch und vielfältig untersucht ist.Praktische Relevanz gewann der Reed-Muller-Code in den Jahren 1969 bis 1976 durch den Einsatz bei den Mariner- und Viking-Expeditionen. Um Bilder vom Mars zur Erde zu senden, wurde der $[32, 6, 16]$-RM$(1, 5)$-Code eingesetzt. Die an Bord aufgenommenen Bilder wurden gerastert. Von jeder Rasterzelle wurde der Schwarzgrad auf einer Skala von 0 bis 63 gemessen. Jede Graustufe stellte ein Codewort des RM$(1, 5)$ dar.

Welche Gründe sprachen für den Einsatz eines fehlerkorrigierenden Codes, insbesondere des RM$(1, 5)$? Die guten Fehlerkorrektureigenschaften (bis zu sieben Fehler gegenüber zwei bei einem fünffachen Wiederholungscode), die schnelle Decodierung und die Möglichkeit, die 64 Codewörter zu berechnen statt 64 32-Bit-Wörter abzuspeichern. Die Decodierung basierte jedoch nicht auf Majority-Logic, stattdessen wurde ein Soft-Decision-Maximum-Likelihood-Decodierer eingesetzt. Aus Gründen der Performanz setzte man bei späteren Expeditionen auf den [24,12,8]-Golay-Code und auf eine Verkettung eines [255,223,33]-Reed-Solomon-Codes mit einem Faltungscode kurzer Länge in Verbindung mit einem Soft-Decision-Viterbi-Algorithmus [14, S. 9 f.], [40, S. 22], [41, S. 68], [31, S. 25 f.], [22, S. 99].

Ausgehend von den Reed-Muller-Codes wurden in den vergangenen Jahrzehnten eine Vielzahl von neuen (häufig zyklischen) Codeklassen konstruiert. Ta-

dao Kasami, Shu Lin und W. Wesley Peterson bewiesen 1968, dass aus jedem Reed-Muller-Code ein zyklischer Code konstruiert werden kann, indem man den Reed-Muller-Code punktiert und die Positionen der Codewortes umordnet [17]. Der Reed-Muller-Code ist ein Code über $\mathbb{F}_2$, der Länge 2^m, bei dem die Positionen eines Codeworts mit den Punkten der Euklidischen Geometrie[3] $\mathrm{EG}(m, 2)$ assoziiert sind. Diese Strukturen lassen sich verallgemeinern, indem man p prim, nicht zwangsläufig zwei, $q_\mathcal{C}$ als eine Potenz von p und $q = {q_\mathcal{C}}^{m_2}$ annimmt und dann Codes über $\mathbb{F}_{q_\mathcal{C}}$ der Länge q^m (oder $q^m - 1$ nach Punktieren des Codes) über den Euklidischen Geometrien $\mathrm{EG}(m, q)$, $\mathrm{EG}(m \cdot m_2, q_\mathcal{C})$ betrachtet. Zu diesen verallgemeinerten Codes gehören die Euklidischen-Geometrie-Codes ($q_\mathcal{C} = p$ prim) [29, §10.2], die verallgemeinerten Euklidischen-Geometrie-Codes ($q_\mathcal{C} \geq p$) [18, S. 812 f.] und die verallgemeinerten Reed-Muller-Codes ($q = q_\mathcal{C}$) [17], [29, §10.5], [18, S. 811], auf die wir in Kapitel 7 näher eingehen werden. All diese Codes sind sogenannte *Polynomcodes* [29, §10.6], [18], [16].

Daneben wurden weitere Codes entwickelt, die sich mit Majority-Logic decodieren lassen. Luther D. Rudolph konstruierte 1967 eine Klasse von Codes basierend auf endlichen Geometrien mit speziellen Kontrollmatrizen und überlegte sich für diese ein geeignetes Majority-Logic-Verfahren [32]. Zu diesen Codes gehören auch jene, deren Kontrollmatrix eine Inzidenzmatrix ist, die die Punkte und die Unterräume von projektiven Geometrien in Beziehung setzt. Jean-Marie Goethals und Philippe Delsarte entwickelten Rudolphs Ergebnisse weiter und entwickelten Codes, deren Struktur sich über Blockdesigns beschreiben lassen [11]. Im Besonderen gehören die zyklischen Reed-Muller-Codes zu dieser Codeklasse. Für die Decodierung verweisen die Autoren auf Masseys Decodierschema [25]. Delsarte verallgemeinerte das Konzept der Euklidischen-Geometrie-Codes und konstruierte 1969 die „primitiven“ und „nichtprimitiven“ geometrischen Codes [9]. Shu Lin stellte 1973 die gefalteten EG-Codes (*multifold euclidean geometry codes*) vor [21]. Zusammen mit Kai-Ping Yiu ent-

[3] In der Mathematik handelt es sich bei einer *Euklidischen Geometrie* um einen reellen Vektorraum mit Skalarprodukt, mit Hilfe dessen sich Längen, Orthogonalität und Winkel definieren lassen. Dies hebt sich von der Bedeutung des Begriffs in der Codierungstheorie ab. Die Codierungstheorie versteht unter dem Begriff der Euklidischen Geometrie einen affinen Raum, der mit der *Hamming-Metrik* versehen ist. Wir behalten die Ausdrucksweise der Codierungstheorie in dieser Arbeit bei.

wickelte Lin aufbauend auf den gefalteten Codes eine verbesserte Klasse von gefalteten Codes (*improved multifold codes*). Diese Codes stellen insofern eine Verbesserung dar, weil ihr Dualcode als Erzeugnis bestimmter Inzidenzvektoren (statt als ein Oberraum dessen) beschrieben werden kann. In Kapitel 7 werden wir zwei Subklassen von gefalteten (und gleichzeitig verbesserten gefalteten) Codes näher beleuchten, die EG-Codes und die zweifach gefalteten EG-Codes. Die zweifach gefalteten Codes bilden gleichzeitig eine Subklasse der geometrischen Codes nach Delsarte [21, S. 542].

Gleichzeitig wurde der Reed-Algorithmus angepasst und/oder weiterentwickelt. Ein übergreifendes (für Blockcodes mit Symbolen aus $\mathbb{F}_{q_C}$, q_C eine Primzahlpotenz) Majority-Logic-Decodierschema stellte James L. Massey im Jahr 1963 vor [25]. Chin-Long Chen zeigt in [7, 8], dass Majority-Logic-Stufen häufig ausgelassen werden können, ohne die Fehlerkorrekturfähigkeit zu mindern. Wir gehen detailliert auf Chens Abstufung in Abschnitt 6.1.2 ein.

Bei Peterson und Weldon werden in [29, §10.4] mehrere Ideen zur Verbesserung angerissen. Zwei der Ideen beinhalten, dass man gegebenenfalls mehr Fehler als beim Reed-Algorithmus vorgesehen korrigieren kann, indem man den Decodieralgorithmus wiederholt hintereinander ausführt. Zum einen, indem man nicht an allen Positionen gleichzeitig decodiert sondern das erhaltene Wort nur an einer Position korrigiert, um die Decodierung anschließend mit diesem korrigierten Wort zu wiederholen („The Use of Feedback“ [29, S. 333f.]). Zum anderen, indem man anstelle von Majoritätsgattern die allgemeineren Schwellengatter einsetzt und die Schwellenwerte für jeden Durchlauf anpasst(“Variable Threshold Decoding Townsend and Weldon (1967)“ [29, S. 334]). In der dritten Idee wird aufgezeigt, dass sich für jeden linearen Code ein Majority-Logic-Decodierverfahren finden lässt. Allerdings ist das Verfahren so allgemein formuliert, dass es laut der Autoren für fast alle Codes ungefähr genauso komplex wie eine Versuch-und-Irrtum-Suche (*trial-and-error search*) und damit nicht praxistauglich ist.

Die anderen beiden Ideen beinhalten eine Reduzierung der Anzahl der Majority-Logic-Stufen beim Reed-Algorithmus, um die Performanz zu erhöhen. Um zu vermeiden, dass sich die Fehlerkorrekturfähigkeit verringert, kann man bei-

spielsweise sogenannte *non-orthogonale* Checksummen beim Decodieren mit einbeziehen. Wie oben erwähnt werden bei orthogonalen Checksummen Wörter betrachtet, deren Träger sich *paarweise* in einer gegebenen Menge von Positionen schneiden. Bei non-orthogonalen Checksummen hingegen wird auf die Eigenschaft „paarweise“ verzichtet und nur gefordert, dass der Schnitt der betrachteten Träger gleich bleibt. Tatsächlich ist dieser Unterschied für den originären Reed-Muller-Code irrelevant, da die betrachteten Wörter Inzidenzvektoren zu affinen Räumen sind, so dass jede Menge von non-orthogonalen Checksummen auch die Eigenschaft der Orthogonalität erfüllt (Eine Menge von affinen Räumen der Dimension D, die alle den gleichen affinen Raum der Dimension $D-1$ enthalten, schneiden sich unmittelbar paarweise in diesem.). Gleichzeitig bedeutet dies auch, dass die Fehlerkorrekturfähigkeit unter Berücksichtigung *non-orthogonaler* Checksummen diejenige des Reed-Algorithmus nicht übersteigt. Erst durch Weglassen mindestens einer Stufe, ist zwischen Orthogonalität und Non-Orthogonalität zu unterscheiden. Beispielhaft erklären wir den Unterschied anhand des binären [7,4,3]-Hamming-Codes. Der Reed-Algorithmus ist in der Lage mittels zwei Majority-Logic-Stufen einen Fehler zu korrigieren. Im ersten und zweiten Schritt gibt es drei bzw. sieben orthogonale Summen, die in jeder Mehrheitsentscheidung herangezogen werden können. Werden stattdessen auch *non-orthogonale* Checksummen mit herangezogen, so stehen bei nur einer Majority-Logic-Stufe sieben Checksummen zur Verfügung, die es ermöglichen ebenso einen Fehler zu korrigieren.

Die zweite Idee zur Reduzierung der Anzahl der Majority-Logic-Stufen bietet keinen Vorteil gegenüber der von Chen vorgestellten Abstufung [7, 8], was wir am Ende von Abschnitt 6.3 auf Seite 104 bewiesen haben werden.

Einen anderen Ansatz verfolgen Hauck et al [13]. Statt das gesamte Codewort zu ermitteln, werden bei systematischer Codierung nur die Codewortsymbole an Informationspositionen, nicht jedoch an Redundanzpositionen rekonstruiert, um weniger Mehrheitsentscheidungen treffen zu müssen.

Den Reed-Algorithmus, seine Weiterentwicklungen und Modifikationen eint, dass sie sich leicht implementieren lassen, für eine Vielzahl von Codes anwendbar sind und unter geringem Rechenaufwand mit wenig Speicherbedarf deco-

dieren [22, S. 99]. Aufgrund der systematischen Struktur der Codes und der Tatsache, dass die Eigenschaften des Kanals, wie die Wahrscheinlichkeitsverteilung der Fehlerbits, bei der Majority-Logic-Decodierung nicht berücksichtigt werden, ist der Decodieraufwand in der Regel fix und vor dem eigentlichen Decodieren bekannt. Der Nachteil ist jedoch die gegenüber anderen Verfahren häufig geringere Fehlerkorrekturfähigkeit. Die Anzahl der Fehler τ_{MLG}, die bei einem linearen Code $\mathcal{C}$ der Länge n mit dem Reed-Algorithmus korrigiert werden können, ist nach oben beschränkt durch

$$-\frac{1}{2}+\begin{cases}\frac{n}{d^{\perp}} & d^{\perp} \text{ gerade},\\ \frac{n+1}{d^{\perp}+1} & d^{\perp} \text{ ungerade},\end{cases},$$

wobei $d^{\perp}$ die Minimaldistanz des Dualcodes von $\mathcal{C}$ ist [29, Theorem 10.3]. Zwei Ausnahmen hinsichtlich der geringeren Fehlerkorrekturfähigkeit stellen der Reed-Muller-Code und die zweifachen Euklidischen-Geometrie-Codes dar, die die zyklischen Reed-Muller-Codes sowie die binären Hamming-Codes als Subklasse beinhalten. Bei all diesen Codes mit Minimaldistanz d gilt, dass die durch den Code definierte Fehlerkorrekturgrenze von $\tau_{\max}$ gleich der vom Majority-Logic-Decodierverfahren definierten Fehlerkorrekturgrenze τ_{MLG} ist. Aus diesem Grund räumen wir diesen Codes eine große Bedeutung ein und widmen uns ihnen in Kapitel 7.

Die Autorin dieser Arbeit wirkte als Co-Autorin in drei der zitierten Publikationen mit, [13,2,3]. Die Resultate aus [13] werden in dieser Arbeit nur insofern aufgegriffen, dass sie im Ausblick in Abschnitt 8.2.1 mit den Ergebnissen dieser Arbeit verglichen werden. Die Arbeiten [2, 3] beschränken sich darauf, die Klasse der Reed-Muller-Codes zu betrachten. Der in diesen beiden Schriften vorgestellte Decodieralgorithmus für gewisse Reed-Muller-Codes wurde von der Autorin dieser Arbeit entwickelt. Eine Verallgemeinerung des Algorithmus findet sich in Kapitel 5. Jeder der zitierten Passagen aus [2, 3] wurde für die jeweilige Publikation von der Autorin dieser Arbeit eigenständig verfasst.

Kapitel 2

Allgemeine Notationen und Definitionen

2.1 Mengen und Zahlbereiche

Es sei $\mathbb{Z}$ die *Menge der ganzen Zahlen* und $\mathbb{R}$ die *Menge der reellen Zahlen.* Es bezeichne $\mathbb{R}^+$ die Menge der nichtnegativen reellen Zahlen. Die *Menge der natürlichen Zahlen* $\mathbb{N}$ sei gegeben durch die Menge der positiven ganzen Zahlen, $\mathbb{N} \coloneqq \{z \in \mathbb{Z} \mid z > 0\}$. Die Menge der nichtnegativen ganzen Zahlen werde repräsentiert durch $\mathbb{N}_0 \coloneqq \mathbb{N} \cup \{0\}$. In der Regel verwenden wir eine nullbasierte Nummerierung.

Das Gewicht einer Zahl $a \in \mathbb{N}_0$ bzgl. $b \in \mathbb{N}$, $b \geq 2$, ist gegeben durch

$$\omega_b(a) \coloneqq \sum_{i \in \mathbb{N}_0} a_i \in \mathbb{N}_0,$$

wobei

$$a = \sum_{i \in \mathbb{N}_0} a_i b^i$$

mit $a_i \in \mathbb{N}_0$, $a_i < b$ für alle $i \in \mathbb{N}_0$.

Gegeben eine reelle Zahl x. Dann ist

$$\operatorname{sgn}(x) := \begin{cases} 1 & x \geq 0, \\ -1 & x < 0 \end{cases}$$

und

$$|x| := \operatorname{sgn}(x) \cdot x = \begin{cases} x & x \geq 0, \\ -x & x < 0 \end{cases}.$$

Weiterhin sei $\lfloor x \rfloor$ die größte ganze Zahl kleiner oder gleich x und $\lceil x \rceil$ sei die kleinste ganze Zahl größer oder gleich x,

$$\begin{aligned} \lfloor x \rfloor &:= \max \{ z \in \mathbb{Z} \mid z \leq x \}, \\ \lceil x \rceil &:= \min \{ z \in \mathbb{Z} \mid z \geq x \}. \end{aligned}$$

Für zwei ganze Zahlen $a, b \in \mathbb{Z}$ sagen wir a *teilt* b und schreiben $a \,|\, b$, falls es ein $c \in \mathbb{Z}$ gibt, so dass $b = a \cdot c$. Ebenfalls für beliebige ganze Zahlen $a, b \in \mathbb{Z}$, a, b nicht beide null, bezeichne

$$\operatorname{ggT}(a, b) := \max \{ c \in \mathbb{N} \colon c \,|\, a, c \,|\, b \} \in \mathbb{Z}$$

den größten gemeinsamen Teiler von a und b in $\mathbb{Z}$.

Für ganze Zahlen $a, a' \in \mathbb{Z}$ und eine natürliche Zahl $b \in \mathbb{N}$, bezeichnen wir mit $a \bmod b$ den ganzzahligen Rest der Ganzzahldivision a durch b mit gleichem Signum wie a,

$$\begin{aligned} a \bmod b &:= a - \operatorname{sgn}(a) \cdot b \cdot \left\lfloor \frac{|a|}{b} \right\rfloor \\ &= \begin{cases} a - b \cdot \left\lfloor \frac{a}{b} \right\rfloor & \in \{0, 1, \ldots, b-1\} \quad a \geq 0, \\ a + b \cdot \left\lfloor \frac{-a}{b} \right\rfloor & \in \{-b+1, \ldots, 1, 0\} \quad a < 0 \end{cases}. \end{aligned}$$

Wir schreiben $a' \equiv a \bmod b$ genau dann, wenn $b \,|\, (a - a')$.

Die Kardinalität einer Menge S werde bezeichnet mit $|S|$. Es bezeichne $\mathcal{P}(S)$ die Potenzmenge von S,

$$\mathcal{P}(S) := \left\{ S' \;\middle|\; S' \subseteq S \right\}.$$

2.2 Restklassenringe und Körper

Gegeben $N \in \mathbb{N}$, definieren wir

$$\mathbb{Z}_N \coloneqq \{0, 1, \ldots, N-1\} \subset \mathbb{Z}.$$

Zusammen mit der Addition modulo N und der Multiplikation modulo N wird die Menge $\mathbb{Z}_N$ zu einem (kommutativen) Ring (mit Eins), dem *Restklassenring modulo N*. Für jede Primzahlpotenz q bezeichne $\mathbb{F}_q$ den (bis auf Isomorphie eindeutig bestimmten) Körper mit q Elementen. Wir setzen $\mathbb{F}_q^* \coloneqq \mathbb{F}_q \setminus \{0\}$. Ist $q = p$ prim, so ist $\mathbb{F}_q = \mathbb{Z}_p$.

2.3 Vektorräume, affine Räume, Matrizen

Gegeben $m \in \mathbb{N}$ sowie $q \in \mathbb{N}$, eine Potenz der Primzahl p. Es bezeichne

$$\mathbb{F}_q^m \coloneqq \{(v_0, v_1, \ldots, v_{m-1}) \mid v_i \in \mathbb{F}_q\}$$

den Vektorraum der Zeilenvektoren der Dimension m über $\mathbb{F}_q$.

Ist eine Teilmenge U eines Vektorraums V selbst ein Vektorraum, kennzeichnen wir dieses durch die Schreibweise $U \leq V$ beziehungsweise bei einer echten Teilmenge $U \subsetneq V$ auch $U < V$. Es gibt $\dim U$ die Dimension des Untervektorraums U an. Für jede Teilmenge $U \subseteq \mathbb{F}_q^m$ ist

$$\langle U \rangle_{\mathbb{F}_q} \coloneqq \left\{ \sum_{\mathbf{w} \in U} \alpha_{\mathbf{w}} \mathbf{w} \;\middle|\; \forall \mathbf{w} \in U \colon \alpha_{\mathbf{w}} \in \mathbb{F}_q \right\}$$

der $\mathbb{F}_q$-Vektorraum, der durch die Vektoren aus U erzeugt wird.

Der kanonische i-te Einheitsvektor des $\mathbb{F}_q^m$, $i = 0, 1, \ldots, m-1$, werde bezeichnet mit $\mathbf{e}_i$,

$$\mathbf{e}_i \coloneqq (0, \ldots, 0, \underset{\substack{\uparrow \\ i}}{1}, 0, \ldots, 0) \in \{0,1\}^m .$$

(Die nullbasierte Indizierung ist zu beachten.) Sei $\mathbf{v} := (v_0, v_1, \ldots, v_{m-1}) \in \mathbb{F}_q^m$. Der korrespondierende Spaltenvektor ist

$$\mathbf{v}^\top := \begin{pmatrix} v_0 \\ v_1 \\ \ldots \\ v_{m-1} \end{pmatrix}.$$

Das Element v_i, $0 \leq i \leq m-1$, nennen wir den *i-ten Eintrag* respektive den *Eintrag an der Position i* von $\mathbf{v}$. Der Träger von $\mathbf{v}$, $\operatorname{supp} \mathbf{v}$, ist die Menge der Positionen, an denen der Eintrag ungleich null ist,

$$\operatorname{supp} \mathbf{v} := \{0 \leq i \leq m-1 \mid v_i \neq 0\}.$$

Das *Gewicht von* $\mathbf{v}$ ist die Anzahl der Einträge ungleich null,

$$\omega(\mathbf{v}) := |\operatorname{supp} \mathbf{v}| = |\{i = 0, 1, \ldots, m-1 \mid v_i \neq 0\}| \in \mathbb{N}_0.$$

Wir definieren eine symmetrische, nicht-ausgeartete $\mathbb{F}_q$-Bilinearform auf $\mathbb{F}_q^m$,

$$\begin{aligned} \circ : \mathbb{F}_q^m \times \mathbb{F}_q^m &\to \mathbb{F}_q \\ (\mathbf{v}, \mathbf{w}) &\mapsto \mathbf{v} \circ \mathbf{w} := \sum_{i=0}^{m-1} v_i \cdot w_i, \end{aligned}$$

und eine Metrik δ, die sogenannte *Hamming-Metrik*,

$$\begin{aligned} \delta : \mathbb{F}_q^m \times \mathbb{F}_q^m &\to \mathbb{N}_0, \\ (\mathbf{v}, \mathbf{w}) &\mapsto \delta(\mathbf{v}, \mathbf{w}) := |\{i = 0, 1, \ldots, m-1 \mid v_i \neq w_i\}|, \end{aligned}$$

wobei v_i bzw. w_i jeweils der i-te Eintrag von $\mathbf{v}$ bzw. $\mathbf{w}$ ist.

Wir halten fest, für beliebige Vektoren $\mathbf{v}, \mathbf{w} \in \mathbb{F}_q^m$ gleicht der *Hammingabstand* zwischen $\mathbf{v}$ und $\mathbf{w}$ dem Gewicht von $\mathbf{v} - \mathbf{w}$,

$$\delta(\mathbf{v}, \mathbf{w}) = \omega(\mathbf{v} - \mathbf{w}).$$

In Anlehnung an [22, § 8.4, Definition 8.2], [9, §1.3.] ist eine Menge von Vektoren

$$\{(v_{i,0}, v_{i,1}, \ldots, v_{i,m-1}) \mid 0 \leq i \leq N\} \subseteq \mathbb{F}_q^m,$$

$N \in \mathbb{N}_0$, *orthogonal bezüglich* $\mathbf{u} := (u_0, u_1, \ldots, u_{m-1}) \in \mathbb{F}_q^m$, falls

(i) an Positionen des Trägers von $\mathbf{u}$ alle Vektoren den gleichen Eintrag haben, $v_{i,s} = u_s$ für alle $s \in \operatorname{supp} \mathbf{u}$, für alle $0 \leq i \leq N$, und

(ii) an Positionen außerhalb des Trägers von $\mathbf{u}$ höchstens ein Vektor einen Eintrag ungleich null hat, so dass für alle $s \notin \operatorname{supp} \mathbf{u}$ folgende Implikation gilt:

$$v_{i,s} \neq 0 \text{ für ein } i \in \mathbb{N}_0, i \leq N$$
$$\Longrightarrow$$
$$v_{j,s} = 0 \text{ für alle } j \in \mathbb{N}_0, j \leq N, j \neq i.$$

Die Menge der $(m \times n)$-Matrizen, $n \in \mathbb{N}$, mit Elementen aus $\mathbb{F}_q$ werde bezeichnet mit $\operatorname{Mat}(m, n, \mathbb{F}_q)$. Die Multiplikation des Vektors $\mathbf{v}$ mit einer Matrix

$$\mathtt{M} := \left(\mathbf{u}_0^\top, \mathbf{u}_1^\top, \ldots, \mathbf{u}_{n-1}^\top\right) \in \operatorname{Mat}(m, n, \mathbb{F}_q)$$

ist gegeben durch

$$\mathbf{v} \cdot \mathtt{M} = (\mathbf{v} \circ \mathbf{u}_0, \mathbf{v} \circ \mathbf{u}_1, \ldots, \mathbf{v} \circ \mathbf{u}_{n-1},) \in \mathbb{F}_q^n.$$

Für beliebiges $D \in \mathbb{N}_0$ mit $D \leq m$ bezeichne $\mathcal{A}_{D,m,q} \subseteq \mathcal{P}\left(\mathbb{F}_q^m\right)$ die Menge der affinen D-dimensionalen Unterräume von $\mathbb{F}_q^m$,

$$\mathcal{A}_{D,m,q} := \left\{\mathbf{w} + U \subseteq \mathbb{F}_q^m \;\middle|\; U \leq \mathbb{F}_q^m, \dim(U) = D, \mathbf{w} \in \mathbb{F}_q^m\right\}.$$

Vorausgesetzt $D \in \mathbb{N}_0$, $D < m$, bezeichnen wir die Menge der affinen D-dimensionalen Unterräume von $\mathbb{F}_q^m$, die nicht den Nullvektor enthalten, mit $\mathcal{A}^*_{D,m,q} \subseteq \mathcal{P}\left(\mathbb{F}_q^m\right)$,

$$\mathcal{A}^*_{D,m,q} := \left\{\mathbf{w} + U \subseteq \mathbb{F}_q^m \;\middle|\; U \leq \mathbb{F}_q^m, \dim(U) = D, \mathbf{w} \in \mathbb{F}_q^m \backslash U\right\}.$$

2.4 Funktionen

Für eine Aussage T, die entweder wahr oder falsch ist, definieren wir

$$\mathbb{1}_T := \begin{cases} 1 & T \text{ ist wahr,} \\ 0 & T \text{ ist falsch} \end{cases}.$$

Sei $f\colon\ S \to \mathbb{R}$ eine reellwertige Funktion. Es ist $\arg\max_S f$ die Menge der Argumente aus S, die die Funktion f maximieren,

$$\arg\max_S f \coloneqq \left\{ s \in S \;\middle|\; \forall s^{'} \in S\colon f(s^{'}) \leq f(s) \right\} \subseteq S.$$

Gegeben eine natürliche Zahl $\eta \in \mathbb{N}$ und ein η-Tupel mit Elementen s_i aus S, $i \in \mathbb{Z}_\eta$. Die *Menge der Mehrheiten* bzgl. dieses Tupels $(s_i)_{i=0}^{\eta-1}$ sei definiert als

$$\mu_{(s_i)_{i=0}^{\eta-1}} \coloneqq \underset{\{s_i \,|\, 0 \leq i \leq \eta-1\}}{\arg\max} f,$$

wobei

$$\begin{array}{rccc} f\colon & \{s_i \mid 0 \leq i \leq \eta-1\} & \to & \mathbb{N}, \\ & s & \mapsto & \displaystyle\sum_{i=0}^{\eta-1} \mathbb{1}_{s_i=s}. \end{array}$$

Es ist also genau dann $s \in \mu_{(s_i)_{i=0}^{\eta-1}}$, wenn für alle $s^{'} \in \{s_i \mid 0 \leq i \leq \eta-1\}$ gilt, dass

$$|\{0 \leq i \leq \eta-1 \mid s_i = s\}| \geq \left| \left\{ 0 \leq i \leq \eta-1 \;\middle|\; s_i = s^{'} \right\} \right|.$$

Wir sagen, die Mehrheit in $(s_i)_{i=0}^{\eta-1}$ existiert, wenn $\left|\mu_{(s_i)_{i=0}^{\eta-1}}\right| = 1$. Von einer absoluten Mehrheit in $(s_i)_{i=0}^{\eta-1}$ sprechen wir, wenn es ein s gibt, so dass

$$\mu_{(s_i)_{i=0}^{\eta-1}} = \{s\} \text{ und } |\{i \in \mathbb{Z}_\eta \mid s_i = s\}| > \eta/2.$$

Darauf aufbauend definieren wir die *Majoritätsfunktion über S* für ein festes Γ,

$$\begin{array}{rccc} \mu^\Gamma\colon & \displaystyle\bigcup_{\eta\in\mathbb{N}} S^\eta & \to & S \cup \{\Gamma\} \\ & (s_0, s_1, \ldots, s_{\eta-1}) & \mapsto & \mu^\Gamma(s_0, s_1, \ldots, s_{\eta-1}), \end{array}$$

wobei

$$\mu^\Gamma(s_0, s_1, \ldots, s_{\eta-1}) \coloneqq \begin{cases} s_j & \mu_{(s_i)_{i=0}^{\eta-1}} = \{s_j\} \text{ für ein } j,\ 0 \leq j \leq \eta-1, \\ \Gamma & \left|\mu_{(s_i)_{i=0}^{\eta-1}}\right| > 1. \end{cases}$$

Werten wir die Majoritätsfunktion in η Argumenten aus, so sprechen wir von einer η-Mehrheitsentscheidung. Besondere Aufmerksamkeit verdient der Fall,

dass Γ just dem Wert entspricht, der mehrheitlich auftritt, $\mu_{(s_i)_{i=0}^{\eta-1}} = \{\Gamma\}$. Wir sehen, dass aus

$$\mu^{\Gamma}(s_0, s_1, \ldots, s_{\eta-1}) = \Gamma \in \{s_i \mid 0 \leq i \leq \eta - 1\}$$

kein Rückschluss möglich ist, ob eine Mehrheit existiert oder nicht.

2.5 Komplexität

Seien $f, g : \mathbb{N} \to \mathbb{R}^+$ Funktionen. Es ist $f = O(g)$ genau dann, wenn es eine Konstante $x \in \mathbb{R}$, $x > 0$ und ein $b \in \mathbb{N}$ gibt, so dass $f(a) \leq x \cdot g(a)$ für alle $a \in \mathbb{N}$ mit $a > b$. Weiterhin ist $f = \Theta(g)$ genau dann, wenn es Konstanten $x_1, x_2 \in \mathbb{R}$, $x_1, x_2 > 0$ und ein $b \in \mathbb{N}$ gibt, so dass $x_1 \cdot g(a) \leq f(a) \leq x_2 \cdot g(a)$ für alle $a \in \mathbb{N}$ mit $a > b$.

Neben der (sequenziellen) Laufzeit eines Algorithmus wollen wir auch seine parallele Laufzeit angeben. Unter *paralleler Laufzeit* versteht man die Laufzeit, die ein Algorithmus benötigt, wenn alle Rechenschritte weitestgehend parallelisiert von parallel nutzbaren Prozessoren ausgeführt werden. Je stärker ein Algorithmus sich parallelisieren lässt, desto höher das Verhältnis von sequenzieller gegenüber paralleler Laufzeit.

Auf Hardwareebene ist die Tiefe eines Schaltkreises „ein Maß für die Rechenzeit bei Parallelverarbeitung“ [38, S. 11]. Die Tiefe eines Schaltkreises ist die maximale Anzahl von Gattern auf einem gerichteten Weg von einem Eingang zu einem Ausgang des Schaltkreises(vgl. [38, 1.3.5 Definition], [37, 5.7 Definition]). Die Größe eines Schaltkreises ist die Anzahl seiner inneren Gatter, also der Gatter, die weder Eingang noch Ausgang darstellen [37, 5.7 Definition].

Die Hardwarekosten, insbesondere die Art und Anzahl der benutzen Gatter, sind ein wichtiges Kriterium, auf das wir eingehen werden. Die Anzahl der benutzten Verbindungsdrähte und die Chipfläche werden wir nicht explizit angeben, sie lassen sich jedoch bei den von uns betrachteten Algorithmen aufgrund der systematischen Struktur herleiten.

Kapitel 3

Die Majoritätsfunktion

Die Majority-Logic-Decodierung basiert, wie der Name besagt, auf Mehrheitsentscheidungen. Durch die Majoritätsfunktion ist eine mathematische Formulierung der Mehrheitsentscheidung gegeben. Wir werden zunächst ein paar wichtige Eigenschaften der Majoritätsfunktion aufzeigen, die wir sowohl in diesem als auch in den folgenden Kapiteln ausnutzen werden. Danach besprechen wir, wie die Majoritätsfunktion implementiert werden kann.

3.1 Eigenschaften der Majoritätsfunktion

Wir haben in Abschnitt 2.4 die Majoritätsfunktion eingeführt. Wenn es einen Wert in den Argumenten gibt, der mehrheitlich auftritt, so nimmt die Majoritätsfunktion diesen Wert an. Gibt es keine relative Mehrheit, so wird ein vordefinierter Wert angenommen. Dabei ist es mitunter nicht möglich, vom Funktionswert der Majoritätsfunktion darauf zurückzuschließen, dass (k)eine Mehrheit in den betrachteten Werten existiert. Im Folgenden präsentieren wir einige Eigenschaften der Majoritätsfunktion.

Proposition 3.1.1. *Sei $\eta \in \mathbb{N}$. Weiterhin seien α, β, Γ und $s_0, s_1, \ldots, s_{\eta-1}$ Elemente eines kommutativen Rings $(R, +, \cdot)$ mit Eins, wobei α in R invertierbar ist. Dann ist*

$$\alpha\mu^{\Gamma}(s_0, s_1, \ldots, s_{\eta-1}) + \beta = \mu^{\alpha\cdot\Gamma+\beta}(\alpha \cdot s_0 + \beta, \alpha \cdot s_1 + \beta, \ldots, \alpha \cdot s_{\eta-1} + \beta). \tag{3.1}$$

Beweis. Es ist $s \in \mu_{(s_i)_{i=0}^{\eta-1}}$ genau dann, wenn $(\alpha \cdot s + \beta) \in \mu_{(\alpha\cdot s_i+\beta)_{i=0}^{\eta-1}}$. Also ist

$$\begin{aligned}
\alpha \cdot \mu^{\Gamma}(s_0, s_1, \ldots, s_{\eta-1}) + \beta &\coloneqq \alpha \cdot \begin{cases} s \in \mu_{(s_i)_{i=0}^{\eta-1}} & \left|\mu_{(s_i)_{i=0}^{\eta-1}}\right| = 1, \\ \Gamma & \left|\mu_{(s_i)_{i=0}^{\eta-1}}\right| > 1 \end{cases} + \beta \\
&= \begin{cases} \alpha \cdot s + \beta \in \mu_{(\alpha\cdot s_i+\beta)_{i=0}^{\eta-1}} & \left|\mu_{(\alpha\cdot s_i+\beta)_{i=0}^{\eta-1}}\right| = 1, \\ \alpha \cdot \Gamma + \beta & \left|\mu_{(\alpha\cdot s_i+\beta)_{i=0}^{\eta-1}}\right| > 1 \end{cases} \\
&=: \mu^{\alpha\cdot\Gamma+\beta}(\alpha \cdot s_i + \beta \mid i \in \mathbb{Z}_\eta)
\end{aligned}$$

□

Die folgende Proposition stellt einen Zusammenhang zwischen der Majoritätsfunktion und dem Median her, sofern mindestens die Hälfte der Werte gleich ist.

Proposition 3.1.2. *Gegeben eine total geordnete Menge $(S, \leq)$, eine natürliche Zahl $\eta \in \mathbb{N}$ und ein η-Tupel $(s_i)_{i=0}^{\eta-1}$ mit $s_i \in S$, $i \in \mathbb{Z}_\eta$, derart indiziert, dass*

$$s_0 \leq s_1 \leq \ldots \leq s_{\eta-1}.$$

Der Median *des Tupels $(s_i)_{i=0}^{\eta-1}$ ist definiert als das Element, welches den Index $\left\lfloor \frac{\eta-1}{2} \right\rfloor$ besitzt.*

(a) *Existiert eine absolute Mehrheit in $s_0, s_1, \ldots, s_{\eta-1}$, so ist der Mehrheitswert gleich dem Median,*

$$\mu^{\Gamma}(s_0, s_1, \ldots, s_{\eta-1}) = s_{\left\lfloor \frac{\eta-1}{2} \right\rfloor}.$$

(b) *Ist mindestens die Hälfte der Werte $s_0, s_1, \ldots, s_{\eta-1}$ identisch, so ist*

$$\mu^{\Gamma}(s_0, s_1, \ldots, s_{\eta-1}) = \begin{cases} \Gamma & \eta \text{ gerade und} \\ & s_0 = s_{\eta/2-1} \neq s_{\eta/2} = s_{\eta-1}, \\ s_{\eta-1} & \eta \text{ gerade und} \\ & s_0 \neq s_{\eta/2-1} \neq s_{\eta/2} = s_{\eta-1}, \\ s_{\left\lfloor \frac{\eta-1}{2} \right\rfloor} & \text{sonst.} \end{cases}$$

(c) *Sind* $s_0, s_1, \ldots, s_{\eta-1} \in \{a, b\} \subseteq S$ *mit* $a \leq b$, *so ist*

$$\mu^a(s_0, s_1, \ldots, s_{\eta-1}) = s_{\lfloor \frac{\eta-1}{2} \rfloor}$$

Beweis. Wir setzen

$$M \coloneqq \mu^\Gamma(s_0, s_1, \ldots, s_{\eta-1}).$$

(a) Existiert eine absolute Mehrheit in $s_0, s_1, \ldots, s_{\eta-1}$, so sind mindestens $\lfloor \eta/2 \rfloor + 1$ der Werte gleich M. Angenommen, i ist der kleinste Index, für den $s_i = M$ gilt, dann ist $M = s_j$ für alle $j = i, i+1, \ldots, i + \lfloor \eta/2 \rfloor$. Da unabhängig vom konkreten Wert i stets gilt, dass

$$\lfloor (\eta - 1)/2 \rfloor \in \{i, i+1, \ldots, i + \lfloor \eta/2 \rfloor\} \subseteq \mathbb{Z}_\eta,$$

ist $s_{\lfloor (\eta-1)/2 \rfloor}$ gleich M. Mit anderen Worten, der Mehrheitswert ist gerade der Median.

(b) Ist η ungerade, so existiert eine absolute Mehrheit und die Behauptung folgt aus (a). Diskutieren wir den Fall, dass η gerade ist. Da mindestens die Hälfte der Werte identisch ist, gilt

$$M = \begin{cases} s_{\eta/2-1} & s_0 = s_{\eta/2-1} \neq s_{\eta/2} \neq s_{\eta-1} \text{ oder } s_{\eta/2-1} = s_{\eta/2}, \\ s_{\eta-1} & s_0 \neq s_{\eta/2-1} \neq s_{\eta/2} = s_{\eta-1}, \\ \Gamma & s_0 = s_{\eta/2-1} \neq s_{\eta/2} = s_{\eta-1}. \end{cases}$$

Die Behauptung folgt mit $\eta/2 - 1 = \lfloor \frac{\eta-1}{2} \rfloor$ für gerade Werte η.

(c) Angenommen, es existiert eine (absolute) Mehrheit, so folgt die Behauptung aus (a). Angenommen, es existiert keine Mehrheit. Dann ist $s_{\lfloor \frac{\eta-1}{2} \rfloor} = a$, η gerade und

$$M = \Gamma.$$

Indem wir $\Gamma \coloneqq a$ setzen, ist (c) bewiesen. □

Die Majoritätsfunktion über eine zweielementige Menge kann sehr einfach mit Hilfe von Summen in $\mathbb{Z}$ ausgewertet werden.

Proposition 3.1.3. *Gegeben* $\eta \in \mathbb{N}$, *Symbole* a, b, Γ, $a \neq b$, *und ein* η*-Tupel* $(s_i)_{i=0}^{\eta-1}$ *mit* $s_i \in \{a, b\}$, $0 \leq i \leq \eta - 1$.

(a) *Es ist*

$$\mu^{\Gamma}(s_0, s_1, \ldots, s_{\eta-1}) = \begin{cases} a & \sum_{i=0}^{\eta-1} \mathbb{1}_{s_i=b} < \eta/2, \\ b & \sum_{i=0}^{\eta-1} \mathbb{1}_{s_i=b} > \eta/2, \\ \Gamma & \sum_{i=0}^{\eta-1} \mathbb{1}_{s_i=a} = \eta/2. \end{cases}$$

(b) *Sind $a, b \in \mathbb{Z}$ und $a < b$, dann gilt*

$$\mu^{\Gamma}(s_0, s_1, \ldots, s_{\eta-1}) = \begin{cases} a & \sum_{i=0}^{\eta-1} s_i < \eta/2 \cdot (a+b)\,, \\ b & \sum_{i=0}^{\eta-1} s_i > \eta/2 \cdot (a+b)\,, \\ \Gamma & \sum_{i=0}^{\eta-1} s_i = \eta/2 \cdot (a+b)\,. \end{cases}$$

Beweis. (a) ergibt sich direkt aus der Definition der Majoritätsfunktion.

(b) Wegen

$$\mathbb{1}_{s_i=b} = (s_i - a) \,/\, (b - a)$$

ist

$$\sum_{i=0}^{\eta-1} \mathbb{1}_{s_i=b} < \eta/2$$

genau dann, wenn

$$\sum_{i=0}^{\eta-1} s_i < \eta/2 \cdot (a+b)\,.$$

Diese Äquivalenz gilt ebenso für „>“ und „=“. Die zweite Aussage folgt. □

An dieser Stelle weisen wir noch einmal daraufhin, dass Γ gleich a oder b gewählt werden kann, um nur zwei Fälle unterscheiden zu müssen. Gibt die Majoritätsfunktion in diesem Fall den Wert $\Gamma \in \{a, b\}$ wieder, so darf daraus keine Aussage abgeleitet werden, ob eine Mehrheit existiert oder nicht existiert.

3.2 Implementierung der Majoritätsfunktion

Wir sind sowohl an Software- als auch an Hardwarelösungen interessiert, wie sich die Majoritätsfunktion realisieren lässt.

3.2.1 Implementierung der Majoritätsfunktion in Software

Um verschiedene Decodierverfahren, die auf Mehrheitsentscheidungen basieren, untereinander vergleichen zu können, wollen wir die Komplexität einer Mehrheitsentscheidung untersuchen. Sicherlich hängt diese von der Implementierung ab. Wir stellen drei Varianten vor, wie die Majoritätsfunktion realisiert werden kann.

Sofern nicht zusätzliche Informationen über die η Argumente, die an die Majoritätsfunktion übergeben werden, zur Verfügung stehen, muss mehr als die Hälfte der übergebenen η Argumente betrachtet werden, um die Majoritätsfunktion auszuwerten. In diesem Fall ist sowohl die Laufzeit als auch der Speicherbedarf bestenfalls linear in η.

Algorithmus 3.2.1: berechnet $\mu^{\Gamma}(s_0, s_1, \ldots, s_{\eta-1})$.

```
1  for i = 0 to η − 1 do
2      count[s_i] = 0
3  for i = 0 to η − 1 do
4      count[s_i] = count[s_i] + 1
5  s = s_0
6  no_majority = true
7  for i = 1 to η − 1 do
8      if (count[s_i] > count[s]) then
9          s = s_i
10         no_majority = false
11     else if ((s_i ≠ s) & (count[s_i] == count[s])) then
12         no_majority = true
13 if no_majority then
14     Ausgabe: Γ
15 else
16     Ausgabe: s
```

Algorithmus 3.2.1 Die Majoritätsfunktion μ^Γ über eine Menge S lässt sich naiv implementieren (siehe Algorithmus 3.2.1). Laufzeit- und Speicherkomplexität sind dabei linear in η.

Ist bereits bekannt, dass es eine relative Mehrheit in $s_0, s_1, \ldots, s_{\eta-1}$ gibt,

$$\left|\mu_{(s_i)_{i=1}^{\eta-1}}\right| = 1,$$

so können Zeile 10 bis Zeile 16 in Algorithmus 3.2.1 weggelassen werden.

Algorithmus 3.2.2 Algorithmus 3.2.2 basiert auf Proposition 3.1.2(b) und berechnet den Mehrheitswert unter anderem anhand des Medians. In diesem Algorithmus bezeichne $s_{\pi(i)}$, $0 \leq i \leq \eta - 1$, jenes Element, das beim Ordnen aller Elemente des Tupels $(s_i)_{i=0}^{\eta-1}$ in aufsteigender Reihenfolge an i-ter Stelle auftritt, so dass

$$s_{\pi(0)} \leq s_{\pi(1)} \leq \ldots \leq s_{\pi(\eta-1)}$$

gewährleistet ist.

Zu beachten ist, dass es genügt, die Werte $s_{\pi(i)}$ für

$$i = 0, \lfloor(\eta-1)/2\rfloor, \lfloor\eta/2\rfloor, \eta - 1$$

zu berechnen. Die gesamte Ordnung π muss nicht zwingend ermittelt werden.

Existiert sogar eine absolute Mehrheit in den η Werten $s_0, s_1, \ldots, s_{\eta-1}$, so können Zeile 2 bis Zeile 11 in Algorithmus 3.2.2 weggelassen werden, vgl. Proposition 3.1.2(a). Diese Zeilen können ebenso ausgelassen werden, wenn es $a, b \in S$ gibt, so dass jeder an die Majoritätsfunktion übergebene Wert entweder gleich a oder gleich b ist und außerdem $\Gamma = \min\{a, b\}$ gewählt wird, vgl. Proposition 3.1.2(c).

Die Laufzeit und der Speicherbedarf von Algorithmus 3.2.2 hängen unmittelbar vom Aufwand, die Werte $s_{\pi(i)}$ für $i = 0, \lfloor(\eta-1)/2\rfloor, \lfloor\eta/2\rfloor, \eta - 1$ zu bestimmen, ab. Bestenfalls sind die Werte bereits vor dem Ausführen des Algorithmus bekannt, so dass nur diese statt der gesamten η Werte übergeben werden müssen. In diesem Fall wären sowohl Laufzeit als auch Speicherbedarf von Algorithmus 3.2.2 konstant.

Allgemeine Lösungsansätze bieten Selektionsalgorithmen, deren Aufgabe es ist, das i-tgrößte Element, $i = 0, 1, \ldots, \eta - 1$, aus einer Reihe von η (unsortierten) Elementen zu berechnen. Blum, Floyd, Pratt, Rivest, and Tarjan haben einen auf dem Prinzip „teile und herrsche“ basierenden Selektionsalgorithmus namens „median-of-medians algorithm“ [5] entwickelt. Dieser Algorithmus selektiert das i-t größte Element im besten sowie im schlechtesten Fall in linearer Laufzeit bei linearem Speicherbedarf in η. Damit ließe sich die Majoritätsfunktion wie in Algorithmus 3.2.2 vorgestellt ebenfalls in linearer Laufzeit- und Speicherkomplexität auswerten.

Algorithmus 3.2.2: berechnet $\mu^{\Gamma}(s_0, s_1, \ldots, s_{\eta-1})$.

Vorbedingung: Die Elemente in $s_0, s_1, \ldots, s_{\eta-1}$ lassen sich total ordnen, mindestens die Hälfte dieser Werte ist identisch.

```
1  berechne s_{π(⌊(η-1)/2⌋)}
2  if η gerade then
3      berechne s_{π(η/2)}
4      if s_{π(⌊(η-1)/2⌋)} ≠ s_{π(η/2)} then
5          berechne s_{π(η-1)}
6          if s_{π(η/2)} == s_{π(η-1)} then
7              berechne s_{π(0)}
8              if s_{π(0)} == s_{π(⌊(η-1)/2⌋)} then
9                  Ausgabe: Γ
10             else
11                 Ausgabe: s_{π(η-1)}
   Ausgabe: s_{π(⌊(η-1)/2⌋)}
```

Algorithmus 3.2.3 Nehmen die Argumente der Majoritätsfunktion nur zwei verschiedene Werte aus $\mathbb{Z}$ an, so kann man, wie in Proposition 3.1.3(b) dargelegt, die Mehrheit sehr einfach durch Aufsummieren der Argumente bestimmen (siehe Algorithmus 3.2.3). Dieses leicht zu implementierende Verfahren hat eine lineare Laufzeit in η. Ist bereits vor dem Auswerten die Summe aller Werte

Algorithmus 3.2.3: berechnet $\mu^{\Gamma}(s_0, s_1, \ldots, s_{\eta-1})$.

Vorbedingung: $s_0, s_1, \ldots, s_{\eta-1} \in \{a, b\} \subset \mathbb{Z}$, $a < b$

```
1  s=0
2  for i = 0 to η − 1 do
3      s = s_i + s
4  r = η/2 · (a + b)
5  if s < r then
6      Ausgabe: a
7  else if s > r then
8      Ausgabe: b
9  else
10     Ausgabe: Γ
```

bekannt, so können Zeile 2 und Zeile 3 ausgelassen werden, so dass die Laufzeit sogar konstant ist. Werden die zu betrachtenden Werte nacheinander einzeln eingelesen, so ist der Speicherbedarf ebenfalls konstant, ansonsten linear in η.

3.2.2 Das Majoritätsgatter – Implementierung der Majoritätsfunktion in Hardware

Ein η-*Majoritätsgatter* ist ein elektronisches Bauteil mit η Eingängen und einem Ausgang, das die Majoritätsfunktion μ^{Γ} mit η Argumenten über $\mathbb{Z}_2$ in Hardware realisiert Es kann auf verschiedene Arten implementiert werden. Je nach Aufbau unterscheiden sich die einzelnen Lösungen beispielsweise hinsichtlich der Zeitverzögerung, des Stromverbrauchs und der maximalen Anzahl der Eingänge des Gatters (*Fan-In*).

Die in Abschnitt 3.2.1 vorgestellten Konzepte lassen sich mitunter in Hardware unter Verwendung von grundlegenden Logikgattern wie NICHT-Gattern (*NOT gates*), NAND-Gattern (*NAND gates*), UND-Gattern (*AND gates*), ODER-Gattern (*or gates*), XOR-Gattern (*XOR gates*), sowie komplexeren Schaltungen wie Komparatoren und Addierern umsetzen. Beispielhaft ist ei-

ne auf Logikgattern basierende Implementierung eines 3-Majoritätsgatters in Abbildung 3.1 und eines 4-Majoritätsgatters in Abbildung 3.2 dargestellt.

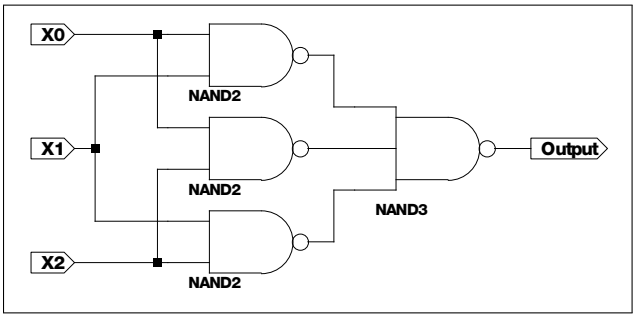


Abbildung 3.1: 3-Majoritätsgatter realisiert durch NAND-Gatter mit zwei und drei Eingängen. Unter einem NAND-Gatter mit drei Eingängen, in der Abbildung mit NAND3 bezeichnet, verstehen wir ein Gatter, das die Boolesche Funktion $f(x_0, x_1, x_2) = \neg(x_0 \wedge x_1 \wedge x_2)$ realisiert.

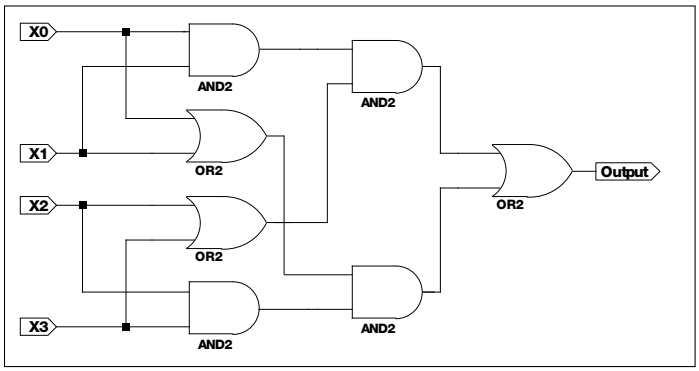


Abbildung 3.2: 4-Majoritätsgatter realisiert durch AND- und OR-Gatter mit jeweils zwei Eingängen.

Das Majoritätsgatter ist ein spezielles *Schwellen(wert)gatter* (*threshold gate*). Ein Schwellengatter ist ein elektronisches Bauteil, welches die Schwellenwertfunktion implementiert. Die Schwellenwertfunktion über $\mathbb{Z}_2$ ist für gegebene binäre Gewichte $w_1, \ldots, w_s \in \mathbb{Z}_2$ und für einen fixen Schwellenwert $s \in \mathbb{Z}_2$ die Funktion

$$
\begin{array}{lccl}
T^s: & \mathbb{Z}_2^\eta & \to & \mathbb{Z}_2 \\
 & (a_0, a_1, \ldots, a_{\eta-1}) & \mapsto & \begin{cases} 1 & \sum_{i=0}^{\eta-1} w_i \cdot a_i \geq s, \\ 0 & \sum_{i=0}^{\eta-1} w_i \cdot a_i < s. \end{cases}
\end{array}
$$

Werden die Gewichte alle auf eins gesetzt, so ist für alle $(a_0, a_1, \ldots, a_{\eta-1}) \in \mathbb{Z}_2^\eta$

$$\mu^1 (a_0, a_1, \ldots, a_{\eta-1}) = T^{\eta/2} (a_0, a_1, \ldots, a_{\eta-1})$$

und

$$\mu^0 (a_0, a_1, \ldots, a_{\eta-1}) = T^{(\eta+1)/2} (a_0, a_1, \ldots, a_{\eta-1}) .$$

Die Schwellenwertfunktion ist damit eine Verallgemeinerung der Majoritätsfunktion μ^Γ über $\mathbb{Z}_2$.

Einen detaillierten Überblick über verschiedene Ansätze, wie Schwellengatter in Hardware gebaut wurden und werden, liefern Beiu et al. [1]. In ihrer Arbeit unterscheiden die Autoren die verschiedenen Lösungen unter anderem danach, ob die (gewichtete) Summe der Eingangssignale durch eine analoge Größe wie beispielsweise die elektrische Stromstärke, die elektrische Spannung oder die elektrische Ladung repräsentiert wird. Im Besonderen werden Entwürfe, die Kondensatoren verwenden [1, §III] oder auf dem elektrischen Leitwert (Konduktanz) respektive der elektrischen Stromstärke basieren [1, Abschnitt IV], vorgestellt. Beiu et al. betrachten außerdem Architekturen mit geringem Stromverbrauch, die auf komplementären Metall-Oxid-Halbleitern (*CMOS*) basieren [1, Abschnitt II]. Darüber hinaus gehen die Autoren auf nanoelektronische Ansätze, die auf Einzelelektronentransistoren oder Resonanztunneldioden beruhen [1, §V], ein.

Nicht nur bei Resonanztunneldioden und Einzelelektronentransistoren nutzt man quantenmechanische Effekte aus. In der Theorie um zellulare Automaten aus gekoppelten Quantenpunkten (*Quantum (Dot) Cellular Automata*, QCA), begründet 1993 durch Lent et al. [19], spielt das Majoritätsgatter mit drei Eingängen eine grundlegende Rolle. Es lässt sich sehr effizient aus nur fünf QCA-Zellen bauen, wobei jede QCA-Zelle aus vier quadratisch angeordneten und miteinander gekoppelten Quantenpunkten besteht (siehe Abbildung 3.3, entnommen aus [34, Fig. 2. (c)]).

Darauf aufbauend wurden QCA-Majoritätsgatter mit fünf [28] und sieben Eingängen [27] vorgestellt.

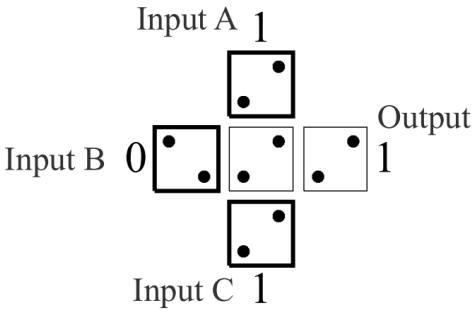


Abbildung 3.3: QCA-Majoritätsgatter mit den Eingangssignalen $1, 0, 1$, entnommen aus [34, Fig. 2. (c)]. Die drei QCA-Zellen links, oben und unten bilden die Eingänge, die QCA-Zelle rechts dient als Ausgang. Für die eigentliche Mehrheitsentscheidung wird nur die im Zentrum positionierte QCA-Zelle benötigt.

Kapitel 4

Die Grundlagen der Codierungstheorie und das Prinzip der Majority-Logic-Decodierung

Das Kapitel beginnen wir mit einer Einführung in die Codierungstheorie. Anschließend stellen wir das Prinzip vor, auf welchem die Majority-Logic-Decodierung beruht. Im Anschluss zeigen wir Möglichkeiten auf, wie diese Decodierung effizienter gestaltet werden kann.

4.1 Lineare Codes

Sei $q_{\mathcal{C}}$ eine Potenz der Primzahl p. Ein $[n,k,d]_{q_{\mathcal{C}}}$-Code $\mathcal{C}$ ist ein linearer Code über $\mathbb{F}_{q_{\mathcal{C}}}$ der Länge n, der Dimension k und mit Minimaldistanz d[1],

$$\begin{aligned} \{0\} \leq \mathcal{C} &\leq \mathbb{F}_{q_{\mathcal{C}}}^{n}, \\ k &= \dim(\mathcal{C}), \\ d &= \begin{cases} \min\left\{\delta\left(\mathtt{c},\mathtt{c}'\right) \mid \mathtt{c} \neq \mathtt{c}' \in \mathcal{C}\right\} & \mathcal{C} \neq \{0\}, \\ 0 & \mathcal{C} = \{0\} \end{cases}. \end{aligned}$$

[1] An späterer Stelle werden wir q als Potenz von $q_{\mathcal{C}}$ einführen und Codes über $\mathbb{F}_{q_{\mathcal{C}}}$ betrachten, deren Längen sich über q definieren.

Eine *Erzeugermatrix* von $\mathcal{C} \neq \{0\}$ ist eine Matrix $\mathtt{G} \in \mathrm{Mat}(k, n, \mathbb{F}_{q_\mathcal{C}})$, deren Zeilen eine Basis von $\mathcal{C}$ bilden,

$$\mathcal{C} = \left\{ \mathbf{v} \cdot \mathtt{G} \;\middle|\; \mathbf{v} \in \mathbb{F}_{q_\mathcal{C}}^k \right\}.$$

Im Fall $\mathcal{C} = \{0\}$ sei $\mathtt{G} = (0, \ldots, 0) \in \mathrm{Mat}(1, n, \mathbb{F}_{q_\mathcal{C}})$ die Erzeugermatrix.

Ein Vektor $\mathbf{v} \in \mathbb{F}_{q_\mathcal{C}}^k$ lässt sich zu einem Codewort mittels $\mathbf{v} \cdot \mathtt{G} \in \mathcal{C}$ codieren.

Der *Dualcode von* $\mathcal{C}$ werde mit $\mathcal{C}^\perp$ bezeichnet,

$$\mathcal{C}^\perp \coloneqq \left\{ \mathbf{v} \in \mathbb{F}_{q_\mathcal{C}}^n \;\middle|\; \forall \mathtt{c} \in \mathcal{C} \colon \mathbf{v} \circ \mathtt{c} = 0 \right\}.$$

Eine *Kontrollmatrix* von $\mathcal{C}$ ist gegeben durch eine Erzeugermatrix von $\mathcal{C}^\perp$.

Die Matrix $\mathtt{G} = (g_{i,j})$ ist in *systematischer Form*, wenn es k Positionen

$$j_0, j_1, \ldots, j_{k-1} \in \mathbb{Z}_n$$

gibt, so dass für alle $i = 0, 1, \ldots, k-1$ die j_i-te Spalte in $\mathtt{G}$ (als Zeilenvektor aufgefasst) gerade der kanonische i-te Einheitsvektor des $\mathbb{F}_{q_\mathcal{C}}^k$ ist. In diesem Fall lässt sich der Vektor $\mathbf{v}$ explizit an den Positionen $j_0, j_1, \ldots, j_{k-1}$ des Codeworts $\mathbf{v} \cdot \mathtt{G}$ ablesen. Die k Positionen $j_0, j_1, \ldots, j_{k-1}$ nennt man *Informationspositionen*. Die übrigen $n - k$ Positionen heißen *Redundanzpositionen*. Für jeden linearen Code gibt es eine Erzeugermatrix in systematischer Form [4, S. 53].

Ein $\left[n', k', d'\right]_{q_\mathcal{C}}$-Code $\mathcal{C}'$ wird als *äquivalent* zum $[n, k, d]_{q_\mathcal{C}}$-Code $\mathcal{C}$ bezeichnet, wenn $n' = n$, $k' = k$ und es eine Matrix $\mathtt{M} \in \mathrm{Mat}(n, n, \mathbb{F}_{q_\mathcal{C}})$ gibt, so dass $\mathcal{C}\mathtt{M} = \mathcal{C}'$ sowie

$$\delta(\mathbf{v}, \mathbf{w}) = \delta(\mathbf{v} \cdot \mathtt{M}, \mathbf{w} \cdot \mathtt{M})$$

für alle $\mathbf{v}, \mathbf{w} \in \mathcal{C}$, [39, Definition 1.2.18]. Wir veranschaulichen dies durch die Schreibweise $\mathcal{C}' \cong \mathcal{C}$. Ist $\mathcal{C}' \leq \mathcal{C}$, so bezeichnen wir $\mathcal{C}'$ als *Subcode* von $\mathcal{C}$.

Der lineare Code $\mathcal{C}$ heißt *zyklisch*, falls aus $(c_0, c_1, \ldots, c_{n-1}) \in \mathcal{C}$ stets auch $(c_{n-1}, c_0, \ldots, c_{n-2}) \in \mathcal{C}$ folgt , [39, Definition 1.2.11]. Wir können jedem Polynom in $\mathbb{F}_{q_\mathcal{C}}[X]$ vom Grad kleiner n

$$\sum_{i=0}^{n-1} w_i X^i \in \mathbb{F}_{q_\mathcal{C}}[X]$$

eindeutig den *Koeffizientenvektor* zuordnen.

$$(w_0, w_1, \ldots, w_{n-1}) \in \mathbb{F}_{q_\mathcal{C}}^n.$$

Ein Polynom vom Grad $n-k$

$$g(X) \coloneqq \sum_{i=0}^{n-k} a_i X^i \in \mathbb{F}_{q_\mathcal{C}}[X]$$

heißt *Erzeugerpolynom* des zyklischen Codes $\mathcal{C}$, falls die zu

$$g(X), X \cdot g(X), \ldots, X^{k-1} \cdot g(X)$$

korrespondierenden Koeffizientenvektoren den Code $\mathcal{C}$ erzeugen. Für jeden zyklischen Code $\mathcal{C}$ gibt es ein eindeutig bestimmtes normiertes Erzeugerpolynom [39, Satz 6.1.2].

4.2 Decodierung

Die Notationen, die in diesem Abschnitt eingeführt werden, sind in allen folgenden Kapiteln gültig. Sei $q_\mathcal{C}$ eine Potenz der Primzahl p. Es bezeichne $\mathcal{C} \leq \mathbb{F}_{q_\mathcal{C}}^n$ einen $[n,k,d]_{q_\mathcal{C}}$-Code mit $(k \times n)$-Erzeugermatix $\mathtt{G}$.

Aus einer Information $\mathtt{I} \in \mathbb{F}_{q_\mathcal{C}}^k$ kann durch Multiplikation mit $\mathtt{G}$ ein Codewort $\mathtt{C} \in \mathcal{C}$ generiert werden,

$$\mathtt{C} = (c_0, c_1, \ldots, c_{n-1}) \coloneqq \mathtt{I} \cdot \mathtt{G}.$$

Wir nennen c_i, $i = 0, 1, \ldots, n-1$ das *i-te Codewortsymbol*.

Über einen gestörten Kanal wird das Codewort $\mathtt{C}$, aus n Symbolen bestehend, übertragen. Wir gehen davon aus, dass bei der Übertragung keine Daten ausgelöscht werden, so dass der Empfänger ebenfalls n Symbole erhält. Sei $\mathtt{Z} \coloneqq (z_0, z_1, \ldots, z_{n-1}) \in \mathbb{F}_{q_\mathcal{C}}^n$ das erhaltene Wort, wobei wir z_i, $i \in \mathbb{Z}_n$, als das *i-te Wortsymbol* bezeichnen. Das empfangene Wort $\mathtt{Z}$ unterscheidet sich in τ der n Positionen vom Codewort $\mathtt{C}$

$$\tau \coloneqq \left|\{i = 0, 1, \ldots, n-1 \mid z_i \neq c_i\}\right|.$$

Die Differenz des empfangenen Worts $\mathtt{z}$ und des Codeworts $\mathtt{c}$ bezeichnen wir als Fehlerwort $\mathtt{e} \coloneqq (\mathtt{e}_0, \mathtt{e}_1, \ldots, \mathtt{e}_{n-1}) \in \mathbb{F}_{q_\mathcal{C}}^n$,

$$\mathtt{e} \coloneqq \mathtt{z} - \mathtt{c}.$$

Wir nennen $\mathtt{e}_i$, $i \in \mathbb{Z}_n$, das *i-te Fehlersymbol.*

Treten nicht mehr als

$$\tau_{\max} \coloneqq \lfloor (d-1)/2 \rfloor$$

Fehler beim Übertragen auf, so ist $\mathtt{c}$ das (einzige) zum empfangenen Wort$\mathtt{z}$ nächstgelegene Codewort,

$$\omega(\mathtt{z} - \mathtt{c}) = \min_{\dot{\mathtt{c}} \in \mathcal{C}} \omega\left(\mathtt{z} - \dot{\mathtt{c}}\right).$$

Indem gemäß dem Prinzip der *Hamming-Decodierung* das zu $\mathtt{z}$ nächstgelegene Codewort bestimmt wird, ermittelt man das ursprünglich übertragene. Allerdings ist eine korrekte Decodierung nicht mehr gewährleistet, sobald mehr als $\tau_{\max}$ Fehler auftreten. In diesem Fall kann es ein oder mehrere nächstgelegene Codewörter geben, das ursprünglich übertragene Codewort ist nicht zwingend eines davon.

Die Frage, die sich stellt, ist, wie man das nächstgelegene oder die nächstgelegenen Codewörter bestimmt. Die Brute-Force-Methode, bei der man für alle Codewörter im Code den Abstand zum erhaltenen Wort ermittelt, ist sicherlich eine, wenn auch nicht besonders elegante, Möglichkeit. Abhängig vom Code gibt es Decodierverfahren wie Syndromdecodierung und Majority-Logic-Decodierung, die effizienter als die Brute-Force-Methode sein können.

Syndromdecodierung

Sei $\mathcal{C}$ ein beliebiger $[n, k, d]_{q_\mathcal{C}}$-Code mit Minimaldistanz $d \geq 3$und Kontrollmatrix $\mathtt{H}$

Das Syndrom eines Vektors $\mathbf{v} \in \mathbb{F}_{q_\mathcal{C}}^n$ bezüglich $\mathtt{H}$ ist gegeben durch $\mathbf{v} \cdot \mathtt{H}^\top \in \mathbb{F}_{q_\mathcal{C}}^{n-k}$. Zwei Vektoren derselben Nebenklasse von $\mathcal{C}$ haben das gleiche Syndrom [15, Theorem 1.11.5],

$$\mathbf{v} + \mathcal{C} = \mathbf{w} + \mathcal{C} \quad \Longleftrightarrow \quad \mathbf{v} \cdot \mathtt{H}^\top = \mathbf{w} \cdot \mathtt{H}^\top.$$

Wir wählen in jeder der ${q_\mathcal{C}}^{n-k}$ verschiedenen Nebenklassen von $\mathcal{C}$ einen Vektor mit minimalem Gewicht als Nebenklassenführer. Ist das Gewicht maximal $\lfloor (d-1)/2 \rfloor$, so ist der Nebenklassenführer eindeutig bestimmt. Vor dem eigentlichen Decodieren wird einmalig von jedem der ${q_\mathcal{C}}^{n-k}$ Nebenklassenführer das Syndrom bestimmt und in einer Lookup-Tabelle abgelegt [15, S. 41 ff.].

Bei der Syndromdecodierung berechnet man nach Empfangen des Worts $\mathtt{z}$ im ersten Schritt sein Syndrom $\mathtt{z} \cdot \mathtt{H}^\top \in \mathbb{F}_{q_\mathcal{C}}^{n-k}$. Ist dieses Syndrom der Nullvektor, so ist das Wort $\mathtt{z}$ bereits ein Codewort und wir beenden die Decodierung mit der Annahme, dass kein Fehler aufgetreten ist.

Die sequenzielle bzw. parallele Zeitkomplexität, das Syndrom zu berechnen, liegt bei $O\left(n^2 - nk\right)$ bzw. $O(\log n)$. Wir geben mit der O-Notation eine obere Abschätzung an, da der tatsächliche Aufwand geringer sein kann, wenn die Kontrollmatrix $\mathtt{H}$ dünn besetzt ist.

Im zweiten Schritt suchen wir in der Lookup-Tabelle nach dem Nebenklassenführer $\dot{\mathtt{E}}$ mit demselben Syndrom. Als übertragenes Codewort wird $\mathtt{z} - \dot{\mathtt{E}} \in \mathcal{C}$ angenommen und ausgegeben.

Die Syndromdecodierung erfolgt damit in einer sequenziellen bzw. parallelen Gesamtlaufzeit von $O\left(n^2 - nk\right)$ bzw. $O(\log n)$.

Majority-Logic-Decodierung

Wir werden uns auf fehlerkorrigierende Majority-Logic-Decodierverfahren für Euklidische-Geometrie-Codes konzentrieren. Wie bereits erwähnt, wird bei diesen mit Hilfe von Mehrheitsentscheidungen (oder Majoritätsentscheidungen) decodiert. Die Grenze, bis zu welcher diese Verfahren das übertragene Codewort korrekt rekonstruieren, werde mit τ_{MLG} bezeichnet. Abhängig vom Code kann τ_{MLG} unterhalb von $\tau_{\max}$ liegen, wie wir in Kapitel 7 sehen werden. Treten maximal τ_{MLG} Fehler auf, so entspricht die ML-Decodierung dem Prinzip der Hamming-Decodierung. Treten mehr als $\tau_{\max}$ Fehler auf, ist nicht gesichert, dass das Codewort $\mathtt{c}$ wieder hergestellt werden kann. Ist $\tau_{\mathrm{MLG}} < \tau \leq \tau_{\max}$, so kann leicht mit Hilfe der Kontrollmatrix $\mathtt{H} \in \mathrm{Mat}(n-k, n, \mathbb{F}_{q_\mathcal{C}})$ von $\mathcal{C}$ überprüft

werden, ob richtig decodiert wurde. Dafür muss das vom Decoder zurückgegebene Wort $\dot{\mathtt{z}}$ ein Codewort in $\mathcal{C}$ sein, dessen Abstand zu $\mathtt{z}$ höchstens $\tau_{\max}$ beträgt,

$$\dot{\mathtt{z}} \in \mathcal{C}, \quad \omega(\mathtt{z} - \dot{\mathtt{z}}) \leq \tau_{\max}.$$

Dabei gilt $\dot{\mathtt{z}} \in \mathcal{C}$ genau dann, wenn $\dot{\mathtt{z}} \cdot \mathtt{H}^\top \in \mathbb{F}_{qc}^{n-k}$ der Nullvektor ist.

Gegeben $\mathbf{v} \in \mathbb{F}_{qc}^n$. Die *Anzahl der Fehler in* $\mathbf{v}$ werde bezeichnet mit

$$\tau_{\mathbf{v}} \coloneqq |\operatorname{supp} \mathbf{v} \cap \operatorname{supp} \mathtt{E}| \,.$$

Es bezeichne $\mathtt{E} \circ \mathbf{v} \in \mathbb{F}_{qc}$ die *Fehlersumme von* $\mathbf{v}$ und $\mathtt{Z} \circ \mathbf{v} \in \mathbb{F}_{qc}$ die *Checksumme von* $\mathbf{v}$. Wir halten ein paar grundlegende Aussagen fest. Für alle $\mathbf{v} \in C^\perp$ ist wegen der Bilinearität der Multiplikation

$$\mathtt{E} \circ \mathbf{v} = \mathtt{Z} \circ \mathbf{v} - \underbrace{\mathtt{C} \circ \mathbf{v}}_{=0} = \mathtt{Z} \circ \mathbf{v}. \tag{4.1}$$

Für alle $i = 0, 1, \ldots, n-1$ ist das i-te Fehlersymbol gleich der Fehlersumme von $\mathbf{e}_i$,

$$\mathtt{E}_i = \mathtt{E} \circ \mathbf{e}_i. \tag{4.2}$$

Außerdem gelten die Implikationen

$$\begin{aligned} \tau_{\mathbf{v}} = 0 &\Rightarrow \mathtt{E} \circ \mathbf{v} = 0, \\ \tau_{\mathbf{v}} = 1 &\Rightarrow \mathtt{E} \circ \mathbf{v} \neq 0. \end{aligned} \tag{4.3}$$

4.3 Prinzip der Majority-Logic-Decodierung

Bei der Majority-Logic-Decodierung werden die einzelnen Fehlersymbole durch aufeinander aufbauende Mehrheitsentscheidungen geschätzt.

Genauer gesagt werden die Fehlersummen von Vektoren, die (paarweise) orthogonal bezüglich eines gegebenen Vektors $\mathbf{v}$ sind, bestimmt. Anschließend wird überprüft, welchen Wert die Fehlersummen am häufigsten annehmen. Dieser mehrheitlich auftretende Wert wird dann als Schätzung für die Fehlersumme des Vektors $\mathbf{v}$ gewertet. Die Grundlage für die Majority-Logic-Decodierung bildet daher die folgende Proposition, die sich an [25, Theorem 1] orientiert.

Proposition 4.3.1 (vgl. [25, Theorem 1]). *Sei $\eta \in \mathbb{N}$, $\eta \geq 2\tau$ beliebig. Weiterhin sei $\mathbf{v} \in \mathbb{F}_{q^c}^n$ beliebig.*

Angenommen, die Vektormenge

$$\{\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_{\eta-1}\} \subseteq \mathbb{F}_{q^c}^n$$

ist orthogonal bezüglich des Vektors $\mathbf{v}$.

Dann ist

$$\mathtt{E} \circ \mathbf{v} = \mu^0 \left(\mathtt{E} \circ \mathbf{v}_0, \mathtt{E} \circ \mathbf{v}_1, \ldots, \mathtt{E} \circ \mathbf{v}_{\eta-1}\right) \qquad [4.4]$$

Beweis. Beim Übertragen des Codeworts sind τ Fehler aufgetreten, $\tau_{\mathbf{v}}$ davon an Positionen im Träger von $\mathbf{v}$ und bis zu $\tau - \tau_{\mathbf{v}}$ an Positionen in den Trägern von $\mathbf{v}_i - \mathbf{v}$, $0 \leq i \leq \eta - 1$. Da die Träger von $\mathbf{v}_i - \mathbf{v}$, $0 \leq i \leq \eta - 1$ nach Voraussetzung paarweise disjunkt sind, gilt also

$$|\{0 \leq i \leq \eta - 1 \mid \tau_{\mathbf{v}_i - \mathbf{v}} = 0\}| \geq \eta - (\tau - \tau_{\mathbf{v}}).$$

Aus $\tau_{\mathbf{v}_i - \mathbf{v}} = 0$ folgt

$$\mathtt{E} \circ \mathbf{v}_i = \mathtt{E} \circ \mathbf{v}$$

für jedes $0 \leq i \leq \eta - 1$, und daher

$$|\{0 \leq i \leq \eta - 1 \mid \mathtt{E} \circ \mathbf{v}_i = \mathtt{E} \circ \mathbf{v}\}| \geq \eta - (\tau - \tau_{\mathbf{v}}).$$

Ist $\tau - \tau_{\mathbf{v}} < \eta/2$, so gilt Gleichung [4.4].

Ist $\tau - \tau_{\mathbf{v}} \geq \eta/2$, dann muss $\tau = \eta/2$ mit $\tau_{\mathbf{v}} = 0$ gelten. Unmittelbar folgt $\mathtt{E} \circ \mathbf{v} = 0$ und $\mathtt{E} \circ \mathbf{v} \in \mu_{(\mathtt{E} \circ \mathbf{v}_i)_{i=0}^{\eta-1}}$. Es gilt Gleichung [4.4]. □

In Worten: Sofern nicht mehr als $\eta/2$ Fehler auftreten, lässt sich die Fehlersumme des Vektors $\mathbf{v}$ ausgehend von den Fehlersummen der Vektoren $\mathbf{v}_i$, $i \in \mathbb{Z}_\eta$, bestimmen. Sind letztere unbekannt, so können diese im Vorfeld ebenso mit Hilfe von Proposition 4.3.1 bestimmt werden, vorausgesetzt man kennt geeignete Vektormengen und die dazugehörigen Fehlersummen.

Es besteht also die Aufgabe zu gegebener Position s, an der decodiert werden soll, eine Vektormenge $\left\{\mathbf{v}_0^1, \mathbf{v}_1^1, \ldots, \mathbf{v}_{\eta-1}^1\right\}$ zu definieren, die orthogonal bezüglich $\mathbf{v}_0^0 \coloneqq \mathbf{e}_s$ ist. Anschließend wird für jeden Vektor $\mathbf{v}_i^1$, $0 \leq i \leq \eta - 1$, dessen

Fehlersumme unbekannt ist, wiederum eine zu diesem Vektor orthogonale Vektormenge $\left\{\mathbf{v}^2_{i\cdot\eta+0}, \mathbf{v}^2_{i\cdot\eta+1}, \ldots, \mathbf{v}^2_{(i+1)\cdot\eta-1}\right\}$ definiert. Der Prozess wird fortgesetzt, bis man die Fehlersummen zu den Vektoren kennt. Dies tritt ein, sobald ein Vektor im Dualcode liegt, da dann die Fehlersumme der leicht zu berechnenden Checksumme gleicht, siehe Gleichung [4.1] auf Seite 38.

Aus graphentheoretischer Sicht ergibt sich ein sogenannter *Decodierbaum* (*decoding tree* [15, S. 354]), den wir in Abbildung 4.1 veranschaulicht haben. Die Wurzel ist der Einheitsvektor $\mathbf{e}_s$, die Blätter sind Vektoren des Dualcodes. Die Höhe des Baumes ς entspricht der Anzahl der Majority-Logic-Stufen. Knoten derselben Tiefe gehören derselben Majority-Logic-Stufe an. Jeder Elternknoten hat η Kindknoten bzw. jeder Knoten außer der Wurzel hat $\eta - 1$ Geschwisterknoten.

Ausgehend von diesem Decodierbaum ergibt sich ein iterativer Decodieralgorithmus, formuliert in Algorithmus 4.3.1. Zunächst werden die Checksummen und damit die Fehlersummen zu den Vektoren korrespondierend zu den Blättern berechnet. Darauf aufbauend werden die Fehlersummen der Vektoren von Knoten der nächstliegenden Majority-Logic-Stufe mit Hilfe von Mehrheitsentscheidungen wie in Proposition 4.3.1 berechnet. Durch diese aufeinander aufbauenden Mehrheitsentscheidungen können letztlich die einzelnen Fehlersymbole bestimmt werden.

Algorithmus 4.3.1: berechnet das Fehlersymbol $\mathtt{E}_s$.

Eingabe: $\mathtt{Z}, \eta, \varsigma, s \in \mathbb{Z}_n, \mathbf{v}_i^\varsigma \in \mathbb{F}_q^m$, $0 \leq i \leq \eta^\varsigma - 1$ wie in Abbildung 4.1

Vorbedingung: $2\tau \leq \eta$

Zusicherung: $e_i^j = \mathtt{E} \circ \mathbf{v}_i^j, 0 \leq j \leq \varsigma$, $0 \leq i \leq \eta^j - 1$;
im Besonderen $e_0^0 = \mathtt{E} \circ \mathbf{e}_s = \mathtt{E}_s$.

1 **for** $i = 0$ **to** $\eta^\varsigma - 1$ **do**
2 $e_i^\varsigma \coloneqq \mathtt{Z} \circ \mathbf{v}_i^\varsigma$
3 **for** $j = \varsigma - 1$ **to** 0 **do**
4 **for** $i = 0$ **to** $\eta^j - 1$ **do**
5 $e_i^j \coloneqq \mu^0\left(e_{i\cdot\eta}^{j+1}, e_{i\cdot\eta+1}^{j+1}, \ldots, e_{(i+1)\cdot\eta-1}^{j+1}\right)$

Ausgabe: e_0^0

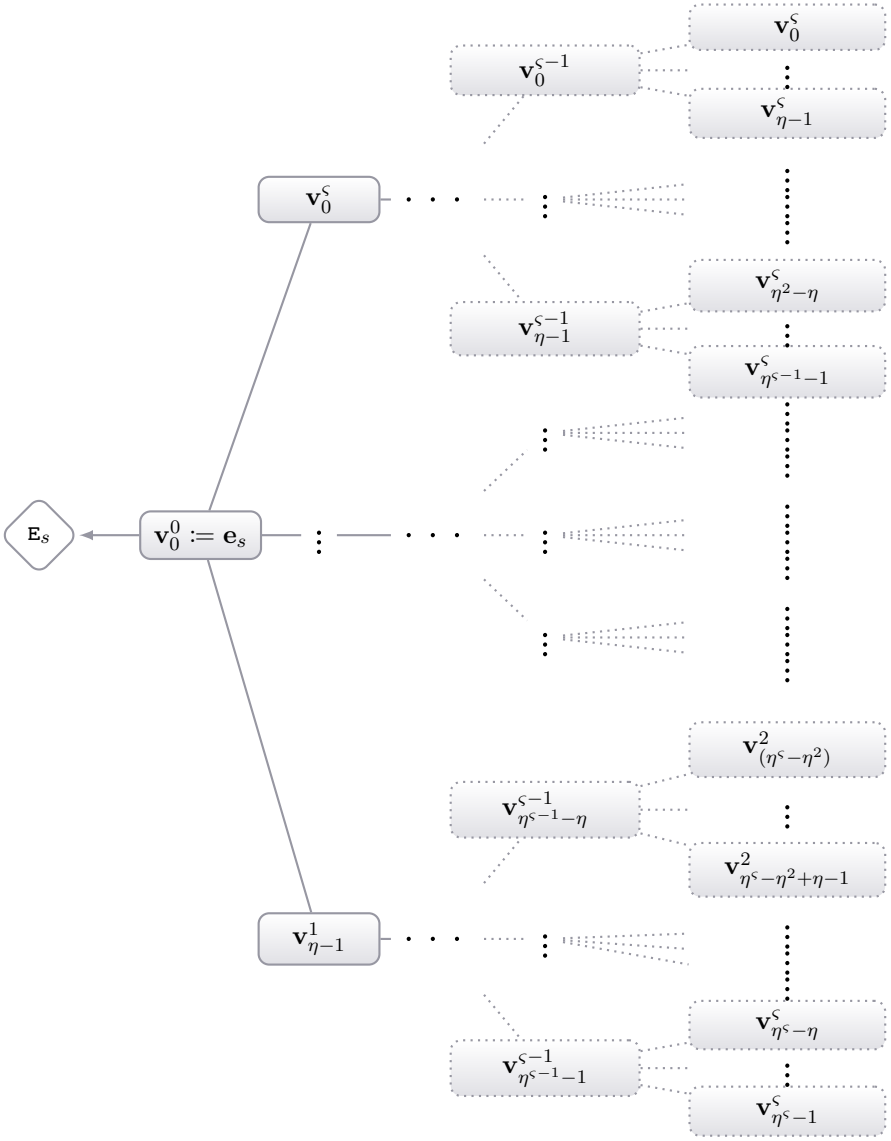


Abbildung 4.1: Decodierbaum, um den Fehler an der Position s zu bestimmen. Die Vektoren von Geschwisterknoten bilden eine Menge, die orthogonal bezüglich des Vektors vom Elternknoten ist.

4.4 Strategien zur Optimierung der Majority-Logic-Decodierung

Die Majority-Logic-Decodierung kann im Wesentlichen in drei Schritte unterteilt werden,

1. Checksummen berechnen,
2. Mehrheitsentscheidungen im Rahmen von einer oder mehreren Majority-Logic-Stufen mit Hilfe eines Decodierbaums treffen,
3. die ursprüngliche Information `I` bzw. das Codewort `C` anhand des Fehlerworts `E` rekonstruieren.

Um die Checksumme eines Vektors $\mathbf{v}$ zu berechnen, ist die Formel

$$\sum_{i \in \operatorname{supp} \mathbf{v} \cap \operatorname{supp} \mathbf{z}} \mathbf{v}_i \cdot z_i$$

der Formel

$$\sum_{i=0}^{n-1} \mathbf{v}_i \cdot z_i$$

vorzuziehen, da sich die Anzahl der notwendigen Multiplikationen und Additionen mitunter signifikant reduziert.

Bei einer Mehrheitsentscheidung wird ermittelt, ob ein – und wenn ja, welcher – Wert am häufigsten von den übergebenen Parametern angenommen wird. Dazu müssen im schlimmsten Fall alle übergebenen Parameter berücksichtigt werden. Die Komplexität einer Mehrheitsentscheidung steigt in der Regel mit der Anzahl der zu berücksichtigenden Parameter. Daher ist es generell empfehlenswert, möglichst wenige Parameter bei einer Mehrheitsentscheidung heranzuziehen.

Ein Ansatz, sowohl Checksummen als auch Mehrheitsentscheidungen einzusparen, ist, Zwischenergebnisse wiederholt zu nutzen. So kann beispielsweise ein Vektor, dessen Träger die Positionen i und j enthält, sowohl für die Korrektur an der i-ten als auch an der j-ten Position eingesetzt werden. Die Check- respektive Fehlersumme dieses Vektors wäre nur einmal zu berechnen.

In erster Linie ist es wünschenswert, die Anzahl der zu treffenden Mehrheitsentscheidungen zu reduzieren. Dies kann zum Beispiel durch eine geringere Anzahl von Majority-Logic-Stufen, also durch weniger Iterationen, erreicht werden. Optimal wäre eine einzige Majority-Logic-Stufe, so dass mit nur einer Mehrheitsentscheidung ein Fehlersymbol bestimmt werden kann. Ferner lässt sich die Anzahl der Mehrheitsentscheidungen reduzieren, indem ein und dieselbe Mehrheitsentscheidung zur Berechnung mehrerer Fehlersummen herangezogen wird, wir wir in Proposition 4.4.1 sehen werden. So kann die Vektormenge $\{\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_{\eta-1}\}$, die orthogonal bezüglich $\mathbf{v}$ ist, nicht nur eingesetzt werden, um die Fehlersumme von $\mathbf{v}$ sondern auch um die Fehlersummen von $\mathbf{v}_0 - \mathbf{v}, \mathbf{v}_1 - \mathbf{v}, \ldots, \mathbf{v}_{\eta-1} - \mathbf{v}$ zu berechnen – mit Hilfe einer einzigen Mehrheitsentscheidung.

Proposition 4.4.1. *Sei $\eta \in \mathbb{N}$ mit $\eta \geq 2\tau$ beliebig. Weiterhin sei $\mathbf{v} \in \mathbb{F}_{qc}^n$ beliebig und $\Gamma \coloneqq -\mathtt{Z} \circ \mathbf{v}$. Angenommen, die Vektormenge*

$$\{\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_{\eta-1}\} \subseteq \mathcal{C}^{\perp},$$

ist orthogonal bezüglich des Vektors $\mathbf{v}$.

Dann ist

$$\mathtt{E} \circ \mathbf{v} = \mathtt{Z} \circ \mathbf{v} + \mu^{\Gamma}\left(\mathtt{Z} \circ (\mathbf{v}_i - \mathbf{v}) \mid 0 \leq i \leq \eta - 1\right)$$

und

$$\mathtt{E} \circ (\mathbf{v}_j - \mathbf{v}) = \mathtt{Z} \circ (\mathbf{v}_j - \mathbf{v}) - \mu^{\Gamma}\left(\mathtt{Z} \circ (\mathbf{v}_i - \mathbf{v}) \mid 0 \leq i \leq \eta - 1\right) \qquad [4.5]$$

für alle $j \in \mathbb{N}_0$, $0 \leq j \leq \eta - 1$.

Beweis. Die beiden Gleichungen folgen aus Proposition 4.3.1, Gleichung [4.1] auf Seite 38 und Proposition 3.1.1. □

In Besonderen wird anhand von Gleichung [4.5] im Vergleich zu Gleichung [4.4] auf Seite 39 deutlich, dass wir die Checksummen der im Dualcode liegenden Vektoren als solche gar nicht explizit berechnen müssen. Indem wir nun Checksummen von dünner besetzten Vektoren berechnen, können wir den Aufwand gegenüber der in Proposition 4.3.1 vorgestellten Methode weiterhin reduzieren. In welcher Größenordnung sehen wir am besten anhand von Beispielen in Kapitel 7.

Wir werden in Kapitel 5 zwei neue Decodierverfahren, die verbesserte und die hybride Decodierung, präsentieren, bei denen wir die genannten Vorschläge aufgreifen.

Eine weitere Möglichkeit, die Decodierung zu optimieren, bietet eine Codierung mit systematischer Erzeugermatrix. Diese erlaubt, die n Positionen eines Codeworts in Informationspositionen und in Redundanzpositionen zu unterteilen. Bei der Decodierung beschränkt man sich zunächst darauf, die Einträge an den Informationspositionen zu korrigieren. Häufig muss man auf diese Art und Weise weniger Checksummen berechnen und weniger Mehrheitsentscheidungen treffen, als würde man an allen Positionen, auch an den Redundanzpositionen, decodieren. Mitunter ist es bereits akzeptabel, nur die ursprüngliche Information `I` statt des gesamten Codeworts `C` wiederherzustellen. Manchmal ist es jedoch erforderlich, alle n Codewortsymbole zu rekonstruieren. In diesem Fall kann man einfach die rekonstruierte Information `I` unter Verwendung der Erzeugermatrix erneut codieren statt die Einträge an Redundanzpositionen zu decodieren. Diese Vorgehensweise ist dann von Vorteil, wenn das Codieren einer Information effizienter als das Decodieren an den Redundanzpositionen ist. Diese Idee, sich auf die Informationspositionen zu konzentrieren, wurde in [13] für Reed-Muller-Codes eingehend studiert. In dieser Arbeit werden wir diese nicht weiter verfolgen und uns stattdessen auf die vorgenannten Ansätze konzentrieren. Gleichwohl stellen wir für Reed-Muller-Codes die Ergebnisse aus dieser Arbeit und aus [13] in Abschnitt 8.2.1 nebeneinander.

Kapitel 5

Majority-Logic-Decodierverfahren für über affine Räume definierte Codes

In diesem Kapitel stellen wir drei verschiedene Majority-Logic-Decodierverfahren vor. Alle basieren auf dem in Abschnitt 4.3 vorgestellten Prinzip der Majority-Logic-Decodierung. Das erste Verfahren, das wir als „klassisch“ betiteln, stellt eine Verallgemeinerung des von Reed für die Muller-Codes entwickelten Decodieralgorithmus [30] dar, siehe auch [29, §10.2, S. 325 f.], [22, §8.6, S. 312 f.]. Es wendet die aus der Literatur bekannte Proposition 4.3.1 an.

Das klassische Verfahren ist sehr allgemein beschrieben und wirft daher unmittelbar die Frage auf, wie es sich effizienter gestalten lässt. Indem wir uns an den in Abschnitt 4.4 präsentierten Optimierungsmöglichkeiten orientierten, konnten wir wir zwei Majority-Logic-Decodierverfahren entwickeln, die der klassischen Decodierung hinsichtlich der Effizienz überlegen sind. Den beiden Verfahren haben wir als „verbesserte“ und „hybride“ Decodierung benannt.

Im verbesserten Verfahren konkretisieren wir die zugrundeliegende mathematische Struktur. Diese zusätzlichen Forderungen an die Struktur eröffnen uns die Möglichkeit, die Anzahl der zu berechnenden Fehlersummen und damit auch der Mehrheitsentscheidungen zu senken. Darüber hinaus reduziert sich

die Schaltkreisgröße bei gleichbleibender Schaltkreistiefe. Weitere Verbesserungen sind je nach Decodierparametern (Anzahl der Majority-Logic-Stufen, Abstufung, Größe der Mehrheitsentscheidungen) denkbar. Beispielhaft werden wir zwei Spezialfälle von Decodierparametern betrachten. Für diese zeigen wir auf, wie die jeweilige Struktur optimiert und somit der Decodieraufwand reduziert werden kann.

Das hybride Verfahren baut wiederum auf der verbesserten Decodierung auf. Ziel ist es, die Anzahl der Mehrheitsentscheidungen weiter zu mindern – unter der Annahme, dass eine Mehrheitsentscheidung einen höheren Rechenaufwand als eine festgelegte Zahl von Additionen respektive Subtraktionen in einem endlichen Körper verursacht. Im Resultat werden beim hybriden Verfahren sowohl Mehrheitsentscheidungen als auch Additionen und Subtraktionen in einem endlichen Körper miteinander verknüpft, um zu decodieren. Die geringere Anzahl an Mehrheitsentscheidungen geht jedoch mit einer geringfügig größeren Schaltkreistiefe einher. Die Hybriddecodierung bietet das Potential, weitere Berechnungen, insbesondere Mehrheitsentscheidungen, einzusparen: Wenn eine Mehrheitsentscheidung durch Additionen/Subtraktionen ersetzt wird und somit entfällt, ist denkbar, dass auch die Werte, die bei dieser Mehrheitsentscheidung herangezogen würden, nicht berechnet werden müssten. Diesem Gedanken geben wir Raum in Abschnitt 5.4.3.

Zu erwähnen ist, dass in der Literatur die Korrektheit des klassischen Verfahrens gezeigt wird, ohne auf den damit verbundenen Aufwand einzugehen. Ohne diesen Aufwand zu kennen, ist ein Vergleich mit unseren Algorithmen wenig effektiv bis unmöglich. Aus diesem Grund analysieren wir eingehend für alle drei Algorithmen den Aufwand hinsichtlich der Art und Gesamtzahl der Operationen sowie der Gesamtzahl der Berechnungsschritte unter Parallelisierung. Dies gibt Aufschluss über Schaltkreisgröße und -tiefe, sequenzieller und paralleler Laufzeit. Alle drei Verfahren veranschaulichen wir anhand von mehreren Beispielen.

Die im folgenden Abschnitt verwendete Notation wird für das gesamte Kapitel gültig sein.

5.1 Affine Räume als Strukturen der Majority-Logic-Decodierung

Sei $\mathcal{C}$ ein $[n, k, d]_{q_\mathcal{C}}$-Code, wobei q eine Potenz von $q_\mathcal{C}$ und $q_\mathcal{C}$ eine Potenz der Primzahl p mit $n \coloneqq q^m - 1$ für ein $m \in \mathbb{N}$, $m \geq 2$ ist[1]. Wir setzen

$$\mathbb{F}_q^m \coloneqq \{\mathbf{w}_0, \mathbf{w}_1, \ldots, \mathbf{w}_{n-1}, 0\}.$$

Jede Position $s \in \mathbb{Z}_n$ eines Worts wird mit einem Vektor $\mathbf{w}_s \in \mathbb{F}_q^m$ ungleich null identifiziert. Wir erhalten die Bijektion,

$$\begin{aligned} \mathcal{P}(\mathbb{Z}_n) &\longleftrightarrow \mathcal{P}\left(\mathbb{F}_q^m \setminus \{0\}\right) \\ S &\mapsto \left\{\mathbf{w}_i \in \mathbb{F}_q^m \;\middle|\; i \in S\right\} \\ \{i \in \mathbb{Z}_n \mid \mathbf{w}_i \in V\} &\mapsfrom V. \end{aligned}$$

In diesem Kapitel werden wir nur Teilmengen des $\mathbb{F}_q^m$ betrachten, die den Nullvektor nicht enthalten. In Kapitel 7 dagegen werden wir die bis dort erhaltenen Ergebnisse auf alle Teilmengen einschließlich der echten Unterräume des $\mathbb{F}_q^m$ ausweiten. Daher definieren wir für beliebige Teilmengen V des Vektorraums $\mathbb{F}_q^m$ den *Inzidenzvektor zu V*, bezeichnet mit χ_V,

$$\begin{aligned} \mathcal{P}\left(\mathbb{F}_q^m\right) &\longrightarrow \{0,1\}^n \\ V \mapsto \chi_V &\coloneqq \sum_{i \in \mathbb{Z}_n : \mathbf{w}_i \in V} \mathbf{e}_i. \end{aligned}$$

Insbesondere ist $\chi_{\{0\}}$ der Nullvektor in $\{0,1\}^n$. Es bezeichne $\mathtt{E} \circ \chi_V \in \mathbb{F}_{q_\mathcal{C}}$ die *Fehlersumme von V* und $\mathtt{Z} \circ \chi_V \in \mathbb{F}_{q_\mathcal{C}}$ die *Checksumme von V*.

Zu beachten ist, dass diese Abbildung nicht injektiv ist, da wir nicht zwischen χ_V und $\chi_{V \setminus \{0\}}$ unterscheiden, insbesondere ist

$$\chi_{\{0\}} = \chi_\emptyset = 0 \in \{0,1\}^n.$$

Für unsere Zwecke ist diese Unterscheidung nicht notwendig, da wir mit einer Ausnahme nur zyklische Codes betrachten werden, bei denen dem Nullvektor

[1] $\mathcal{C}$ ist ein Code, dessen Länge sich über q definiert. Die Codewortsymbole von $\mathcal{C}$ sind Elemente aus dem Körper $\mathbb{F}_{q_\mathcal{C}}$, einem Teilkörper von $\mathbb{F}_q$. In Kapitel 7 werden wir gewisse Codes betrachten, denen diese Struktur zugrunde liegt.

in $\mathbb{F}_q^m$ keine Position in $\mathbb{Z}_n$ zugeordnet ist. Diese eine Ausnahme ist der in Abschnitt 7.5 eingeführte Reed-Muller-Code, den wir als Erweiterung eines zyklischen Codes beschreiben werden.

Wir werden Codes betrachten, deren Strukturen sich über affine Räume beschreiben lassen. Daher führen wir folgende Definitionen ein.

Definition 5.1.1. Für $D_0 \in \mathbb{N}_0, D_1 \in \mathbb{N}$ mit $D_0 < D_1 \leq m$ und für einen beliebigen D_0-dimensionalen affinen Unterraum $A \in \mathcal{A}_{D_0,m,q}$ setzen wir

$$\ell_{D_0,D_1,m,q} \coloneqq \max \left(l \in \mathbb{N} \;\middle|\; \begin{array}{c} \exists\ A_0, A_1, \ldots, A_{l-1} \in \mathcal{A}_{D_1,m,q}, \\ A \subseteq A_i \text{ für alle } 0 \leq i \leq l-1, \\ A_i \cap A_j = A \text{ für alle } 0 \leq i < j \leq l-1 \end{array} \right).$$

Definition 5.1.2. Für $D_0 \in \mathbb{N}_0, D_1 \in \mathbb{N}$ mit $D_0 < D_1 < m$ und für einen beliebigen D_0-dimensionalen affinen Unterraum $A \in \mathcal{A}^*_{D_0,m,q}$ setzen wir

$$\ell^*_{D_0,D_1,m,q} \coloneqq \max \left(l \in \mathbb{N} \;\middle|\; \begin{array}{c} \exists\ A_0, A_1, \ldots, A_{l-1} \in \mathcal{A}^*_{D_1,m,q}, \\ A \subseteq A_i \text{ für alle } 0 \leq i \leq l-1, \\ A_i \cap A_j = A \text{ für alle } 0 \leq i < j \leq l-1 \end{array} \right).$$

Bemerkung 5.1.3. Die Werte $\ell_{D_0,D_1,m,q}$ und $\ell^*_{D_0,D_1,m,q}$ sind unabhängig von A. Gegeben $B \in \mathcal{A}_{D_0,m,q}$. Angenommen, es existieren affine Räume $A_i \in \mathcal{A}_{D_1,m,q}$, $i \in \mathbb{Z}_l$, die sich paarweise in $A \in \mathcal{A}_{D_0,m,q}$ schneiden. Es gibt eine affine Abbildung $f\colon \mathbb{F}_q^m \longrightarrow \mathbb{F}_q^m$, die die beiden affinen Räume ineinander überführt, $f(A) = B$. Dann sind $f(A_0), f(A_1), \ldots, f(A_{l-1})$ affine Räume in $\mathcal{A}_{D_1,m,q}$, die sich paarweise in B schneiden.

Diese Definitionen lassen offen, wie eine bestimmte Anzahl affiner Unterräume (gleicher Dimension) des $\mathbb{F}_q^m$, die sich paarweise in einem gegebenen affinen Unterraum schneiden, konstruiert werden können. Eine Konstruktionsidee zusammen mit einer oberen sowie unteren Grenze für $\ell_{D_0,D_1,m,q}$ lieferten bereits Eisfeld und Storme in ihren Ausführungen zu projektiven Räumen [10].

Proposition 5.1.4. *Seien $D_0, D_1 \in \mathbb{N}_0$ mit $D_0 < D_1 \leq m$. Wir definieren*

$$R \coloneqq (m - D_0)\, mod\, (D_1 - D_0) \in \mathbb{N}_0$$

als den ganzzahligen Rest der Division $(m - D_0)$ *durch* $(D_1 - D_0)$ *und setzen*

$$\ell := \frac{q^{m-D_0} - q^{D_1-D_0+R}}{q^{D_1-D_0} - 1}.$$

(Insbesondere ist genau dann $\ell = 0$*, wenn* $m - D_0 < 2(D_1 - D_0)$*.)*

Dann ist

(a) *im Fall* $R = 0$*,*

$$\ell_{D_0,D_1,m,q} = \ell + 1,$$

(b) *im Fall* $R > 0$*,*

$$\ell + 1 \leq \ell_{D_0,D_1,m,q} \leq \ell + q^R - q + 1.$$

Gilt darüber hinaus $D_1 < m$*, so ist*

(c) stets

$$\ell_{D_0,D_1,m,q} - 1 \leq \ell^*_{D_0,D_1,m,q} \leq \ell_{D_0,D_1,m,q}.$$

(d) im Fall $R = 0$*,*

$$\ell^*_{D_0,D_1,m,q} = \ell.$$

(e) im Fall $R > 0$*,*

$$\ell + 1 \leq \ell^*_{D_0,D_1,m,q} \leq \ell + q^R - q + 1$$

Für den Beweis von Proposition 5.1.4 benötigen wir mehrere aufeinander aufbauende Aussagen. Diese liefern gleichzeitig eine Anleitung, wie affine Unterräume des $\mathbb{F}_q^m$, die sich paarweise in einem gegebenen affinen Raum schneiden, konstruiert werden können. Um vom Grundmotiv der Majority-Logic-Decodierung an dieser Stelle nicht zu weit abzurücken, ist der Beweis von Proposition 5.1.4 in Anhang A zu finden.

5.2 Klassische Decodierung

Wir zeigen, wie sich das Fehlerwort ermitteln lässt, sofern die Fehlersummen von affinen Räumen bestimmter Dimension bekannt sind. Die folgende Proposition liefert eine Verallgemeinerung des von Reed für die Muller-Codes entwickelten Decodieralgorithmus [30], siehe auch [29, §10.2, S. 325 f.], [22, §8.6, S. 312 f.]. Der zugrundeliegende Beweis wendet Proposition 4.3.1 an.

Proposition 5.2.1. *Angenommen, es existieren* $D_{\mathcal{C}} \in \mathbb{N}$, $D_{\mathcal{C}} < m$, $\eta \in \mathbb{N}$, $\eta \geq 2$, $\varsigma \in \mathbb{N}$ *und* $D_1, \ldots, D_{\varsigma-1} \in \mathbb{N}$ *mit*

$$m > D_\varsigma \coloneqq D_{\mathcal{C}} > D_{\varsigma-1} > \ldots > D_1 > D_0 \coloneqq 0,$$

so dass

- *zu jedem* $A \in \mathcal{A}^*_{D_{\mathcal{C}},m,q}$ *die Fehlersumme* $\mathtt{E} \circ \chi_A$ *berechenbar ist und*
- $\ell^*_{D_t,D_{t+1},m,q} \geq \eta \geq 2\tau$ *für alle* $t = \varsigma - 1, \varsigma - 2, \ldots, 0$.

Es gibt ein Majority-Logic-Verfahren, das anhand von maximal $\mathcal{K}_{\mathtt{E}}$ *Fehlersummen zu affinen Räumen aus* $\mathcal{A}^*_{D_{\mathcal{C}},m,q}$ *in* ς *Majority-Logic-Stufen,*

$$D_\varsigma \to D_{\varsigma-1} \ldots \to D_1 \to D_0 \coloneqq 0$$

mit Hilfe von maximal $\mathcal{K}_\mu$ η*-Mehrheitsentscheidungen alle* n *Fehlersymbole* $\mathtt{E}_i$, $0 \leq i \leq n-1$, *korrekt berechnet, wobei*

$$\mathcal{K}_{\mathtt{E}} \coloneqq n \cdot \eta^\varsigma, \tag{5.1}$$

$$\mathcal{K}_\mu \coloneqq n \cdot \frac{\eta^\varsigma - 1}{\eta - 1}. \tag{5.2}$$

Beweis. Sei die Position $i \in \mathbb{N}_0$, $0 \leq i \leq n-1$ beliebig. Konstruiere einen vollständigen Decodierbaum $\mathcal{G}_i$ mit folgenden Eigenschaften:

- Die Höhe des Baumes ist ς (die Anzahl der Majority-Logic-Stufen).
- Jeder Knoten außer den Blättern hat η Kindknoten. (Der Kindknoten eines Knotens der Tiefe t hat selbst Tiefe $t+1$. Die Blätter haben Tiefe ς.)

- Jeder Knoten K der Tiefe t, $0 \leq t \leq \varsigma$, repräsentiert einen affinen Raum aus $\mathcal{A}^*_{D_t,m,q}$.
- Der Wurzelknoten W repräsentiert $\{\mathbf{w}_i\}$, er hat Tiefe 0.
- Für beliebige zwei (verschiedene) Geschwisterknoten, K_j und K_l, eines Elternknotens K gilt, die von K_j und K_l repräsentierten affinen Räume schneiden sich in jenem von K repräsentierten.

Solch ein Graph existiert, da laut Voraussetzung $\eta \leq \ell^*_{D_t,D_{t+1},m,q}$ für alle $t = \varsigma - 1, \varsigma - 2, \ldots, 0$.

Per Induktion über $t \in \mathbb{N}_0$, $t \leq \varsigma$, zeigen wir, dass wir die Fehlersummen zu allen von Knoten der Tiefe t repräsentierten affinen Räumen bestimmen können. Bei $t = \varsigma$ beginnend und t in jedem Schritt um eins dekrementierend, werden bei $t = 0$ schließlich die Fehlersummen von $\mathbf{w} + \{0\}$, $\mathbf{w} \in \mathbb{F}_q^m \setminus \{0\}$ berechnet. Diese entsprechen gerade den Fehlersymbolen $\mathtt{E}_i$, $0 \leq i \leq n - 1$.

Induktionsanfang. Laut Voraussetzung sind die Fehlersummen der affinen Räume, die von den η^ς Blattknoten repräsentiert werden, berechenbar.

Sei nun t, $0 \leq t \leq \varsigma - 1$, beliebig und sei K ein beliebiger der η^t Knoten der Tiefe t. Die η Kindknoten von K werden mit K_i, $i \in \mathbb{Z}_\eta$, bezeichnet. Die Knoten $K, K_0, K_1, \ldots, K_{\eta-1}$ repräsentieren jeweils die affinen Räume $A, A_0, A_1, \ldots, A_{\eta-1}$. Nach Induktionsannahme sind die Fehlersummen der affinen Räume A_i, $0 \leq i \leq \eta$, bekannt. Die Vektormenge $\left\{\chi_{A_0}, \chi_{A_1}, \ldots, \chi_{A_{\eta-1}}\right\}$ ist orthogonal bezüglich χ_A. Wir wenden Proposition 4.3.1 an und erhalten die Fehlersumme von A (Induktionsschritt),

$$\mathtt{E} \circ \chi_A = \mu^0 \left(\mathtt{E} \circ \chi_{A_0}, \mathtt{E} \circ \chi_{A_1}, \ldots, \mathtt{E} \circ \chi_{A_{\eta-1}}\right).$$

Da es η^t Knoten der Tiefe t, $0 \leq t \leq \varsigma - 1$, gibt, treffen wir im Schritt $D_{t+1} \to D_t$ also η^t und insgesamt

$$\sum_{t=0}^{\varsigma-1} \eta^t = \frac{\eta^\varsigma - 1}{\eta - 1}$$

η-Mehrheitsentscheidungen. □

Bemerkung 5.2.2. Abhängig vom Code müssen die n Decodierbäume nur ein einziges Mal konstruiert werden und können dann bei jedem zu decodierenden Wort des Codes verwendet werden.

Die (sequenzielle) Laufzeit bestimmt sich durch $\mathcal{K}_\mu$ und $\mathcal{K}_{\mathtt{E}}$. Die Anzahl der Majority-Logic-Stufen ς geben Aufschluss über die parallele Laufzeit.

Der Decodieralgorithmus ist in Algorithmus 5.2.1 dargestellt.

Algorithmus 5.2.1: Klassische Decodierung .

Vorbedingung: Voraussetzungen aus Proposition 5.2.1,

Graphen $\mathcal{G}_i$ aus dem Beweis von Proposition 5.2.1.

Zusicherung: $\mathtt{E}$ wird zurückgegeben.

for $i = 0$ **to** $n - 1$ **do**
  foreach Knoten in $\mathcal{G}_i$ der Tiefe ς korrespondierend zu A **do**
    $e_A = \mathtt{E} \circ \chi_A$
  for $t = \varsigma - 1$ **to** 0 **do**
    foreach Knoten in $\mathcal{G}_i$ der Tiefe t korrespondierend zu A mit η Kindknoten korrespondierend zu $A_0, A_1, \ldots, A_{\eta-1}$ **do**
      $e_A = \mu^0\left(e_{A_0}, e_{A_1}, \ldots, e_{A_{\eta-1}}\right)$

Ausgabe: $\left(e_{\{\mathbf{w}_0\}}, e_{\{\mathbf{w}_1\}}, \ldots, e_{\{\mathbf{w}_{n-1}\}}\right)$.

Beispiel 5.2.3. Sei

$$\mathbb{F}_2^3 \coloneqq \{\mathbf{w}_0, \mathbf{w}_1, \ldots, \mathbf{w}_7 = 0\}\,.$$

Um die Notation abzukürzen, fassen wir jeden Vektor $(a_2, a_1, a_0) \in \mathbb{F}_2^3$ als Binärzahl auf und identifizieren ihn eindeutig durch die Zahl $\sum_{i=0}^{2} a_i 2^i$,

$$(a_2, a_1, a_0) \in \mathbb{F}_2^3 \quad \longleftrightarrow \quad \sum_{i=0}^{2} a_i 2^i \in \mathbb{Z}_8.$$

Sei $\mathcal{C} \leq \mathbb{F}_2^7$ der Code, dessen Dualcode durch die Inzidenzvektoren zu allen zweidimensionalen echten affinen Räumen des $\mathbb{F}_2^3$ generiert wird,

$$\mathcal{C}^\perp \coloneqq \left\langle \chi_{\mathbf{v}+U} \in \mathbb{F}_2^7 \mid U \leq \mathbb{F}_2^3, \dim U = 2, \mathbf{v} \in \mathbb{F}_2^3 \backslash U \right\rangle_{\mathbb{F}_2}.$$

Man kann leicht überprüfen, dass

$$\mathcal{C} = \mathcal{C}^\perp \oplus \langle 1, 1, 1, 1, 1, 1, 1 \rangle_{\mathbb{F}_2}\,.$$

(Wir werden in Abschnitt 7.6 sehen, dass $\mathcal{C}$ äquivalent zum *binären Hamming-Code* der Länge sieben ist.)

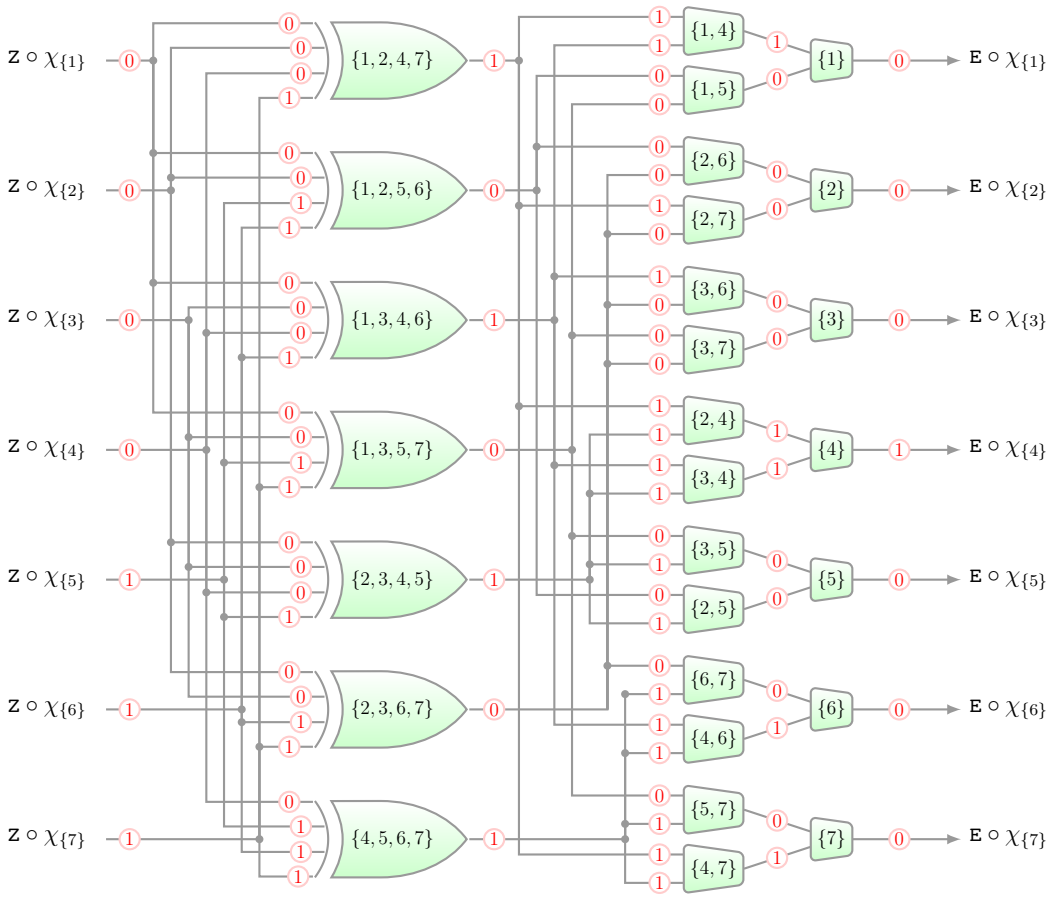
Für alle affinen Räume $A \in \mathcal{A}_{2,3,2}{}^*$ ist die Fehlersumme von A identisch mit der leicht zu berechnenden Checksumme von A,

$$\mathtt{E} \circ \chi_A = \mathtt{Z} \circ \chi_A.$$

Mit Blick auf Proposition 5.2.1 seien $\eta := 2$, $\varsigma := 2$ und

$$3 > D_2 := 2 > D_1 := 1 > D_0 := 0.$$

Konstruieren wir nun einen Decoder, der die im Beweis von Proposition 5.2.1 eingeführten Decodierbäume $\mathcal{G}_i$ für alle Positionen $i \in \mathbb{Z}_7$ in sich vereint.



Die Decodierung verläuft in obigem Schema von links nach rechts. Es gibt sieben durch $\mathtt{z}$ definierte Eingangssignale und sieben Ausgangssignale, die die

geschätzten Fehlersymbole liefern. Der Decoder ist aus zwei verschiedenen Gattertypen aufgebaut, dem XOR-Gatter mit vier Eingängen, symbolisiert durch  , und dem Majoritätsgatter mit zwei Eingängen, symbolisiert durch  . Jedes Gatter repräsentiert einen affinen Raum. Der Ausgang eines Gatters liefert die geschätzte Fehlersumme dieses affinen Raums. Mit Hilfe der XOR-Gatter werden zu Beginn die Checksummen beziehungsweise die Fehlersummen der zweidimensionalen affinen Räume berechnet. Insgesamt besteht der Decoder aus sieben XOR-Gattern und entsprechend Term 5.2 aus

$$21 = 7 \cdot \frac{2^2 - 1}{2 - 1} = n \cdot \frac{\eta^\varsigma - 1}{\eta - 1}$$

Majoritätsgattern.

Beispielhaft decodieren wir das Wort $\mathtt{z} \coloneqq \chi_{\{5,6,7\}}$ und beschriften die Eingänge und Ausgänge der Gatter entsprechend. Der Decoder liefert $\chi_{\{4\}}$ als potentielles Fehlerwort mit einem Fehler an Position $t \in \mathbb{Z}_7$, wobei $\mathbf{w}_t = (1, 0, 0) \longleftrightarrow 4$. Daraus leiten wir das potentielle Codewort $\mathtt{z} - \chi_{\{4\}} = \chi_{\{4,5,6,7\}}$ her. Das potentielle Codewort $\chi_{\{4,5,6,7\}}$ ist tatsächlich das übertragene Codewort $\mathtt{c}$, wenn gewährleistet werden kann, dass nicht mehr als ein Fehler aufgetreten ist. Dies liegt darin begründet, dass im Fall $\tau \leq 1$ mit obig gewählter Abstufung die Voraussetzungen aus Proposition 5.2.1 erfüllt sind,

$$\ell^*_{D_1=1,D_2=2,m=3,2} = 2 = 2\tau$$

und

$$\ell^*_{D_0=0,D_1=1,m=3,2} = 6 > 2 = 2\tau.$$

Eine Bemerkung zur Anzahl der XOR-Gatter. Es fällt auf, dass entgegen der gemäß Term 5.1 zu erwartenden $28 = 7 \cdot 2^2$ XOR-Gatter nur ein Viertel dessen benötigt wird. Gerade bei „kleinen“ Codes wie in diesem Beispiel, bei denen die Menge $\mathcal{A}^*_{D_\varsigma,m,q}$ nur wenige affine Räume enthält (hier $\left|\mathcal{A}^*_{D_\varsigma,m,q}\right| = 7$), werden diese wenigen affinen Räume zur Decodierung an mehreren Positionen eingesetzt. Es wäre unsinnig, die jeweiligen Checksummen mehrfach zu berechnen. Bei entsprechend großen Codes schließen wir nicht aus, dass im schlechtesten Fall tatsächlich $n \cdot \eta^\varsigma$ XOR-Gatter zum Einsatz kommen. ◁

5.3 Verbesserte Decodierung

In Abschnitt 4.4 haben wir bereits besprochen, wie sich die Majority-Logic-Decodierung grundsätzlich optimieren lässt. Diese Ideen wollen wir nun hier anwenden. Zunächst präsentieren wir die von uns entwickelte verbesserte Decodierung, um dann anschließend zwei Spezialfälle vorzustellen, die eine weitere Optimierung bieten. Bei diesen Spezialfällen verkleinern wir den zugrundeliegenden Decodierbaum, indem wir Vektorraumpaare mehrfach nutzen. Abschließend geben wir für kurze binäre Codes explizit die maximale Größe der Decodierbäume unter Berücksichtigung der Spezialfälle an.

5.3.1 Verbesserte Decodierung im Allgemeinen

Bei der klassischen Decodierung wird für jede Position, an der decodiert werden soll, ein eigener Decodierbaum konstruiert. Jeder der Bäume besteht aus Knoten, die zu affinen Räumen korrespondieren. Diese affinen Räume können in einem gewissen Rahmen frei gewählt werden. Wir wissen nicht, ob und wenn ja, in welchem Maß, verschiedene Knoten (des gleichen Decodierbaums oder von verschiedenen Decodierbäumen) denselben affinen Raum repräsentieren. Das führt zu den oberen Abschätzungen die Anzahl der benötigten Fehlersummen in Term 5.1 auf Seite 50 und der Mehrheitsentscheidungen in Term 5.2 auf Seite 50 betreffend.

Diese Wahlmöglichkeiten schränken wir ein und beschreiben eine Struktur, welche fordert affine Unterräume mehrfach zu verwenden. Indem wir verlangen, die benötigten Fehlersummen nur jeweils einmal zu berechnen, reduzieren wir die Anzahl der Mehrheitsentscheidungen. Außerdem werden wir statt n Decodierbäumen nur einen einzigen konstruieren.

Theorem 5.3.1. *Angenommen, es existieren* $D_{\mathcal{C}} \in \mathbb{N}$, $D_{\mathcal{C}} < m$, $\eta \in \mathbb{N}$, $\eta \geq 2$, $\varsigma \in \mathbb{N}$ *und* $D_1, \ldots, D_{\varsigma-1} \in \mathbb{N}$ *mit*

$$m > D_\varsigma \coloneqq D_{\mathcal{C}} > D_{\varsigma-1} > \ldots > D_1 > D_0 \coloneqq 0,$$

so dass

- *zu jedem* $A \in \mathcal{A}^*_{D_\mathcal{C},m,q}$ *die Fehlersumme* $\mathtt{E} \circ \chi_A$ *berechenbar ist und*
- $\ell_{D_t,D_{t+1},m,q} - 1 \geq \eta \geq 2\tau$ *für alle* $t = \varsigma - 1, \varsigma - 2, \ldots, 0$.

Es gibt ein Majority-Logic-Verfahren, das anhand von maximal $\mathcal{V}_\mathtt{E}$ *Fehlersummen zu affinen Räumen aus* $\mathcal{A}^*_{D_\mathcal{C},m,q}$ *in* ς *Majority-Logic-Stufen,*

$$D_\mathcal{C} \coloneqq D_\varsigma \to D_{\varsigma-1} \ldots \to D_1 \to D_0 \coloneqq 0$$

mit Hilfe von maximal $\mathcal{V}_\mu$ η*-Mehrheitsentscheidungen alle* n *Fehlersymbole* $\mathtt{E}_i$*,* $0 \leq i \leq n-1$*, korrekt berechnet, wobei*

$$\mathcal{V}_\mathtt{E} \coloneqq (\eta+1)^\varsigma \left(q^{m-D_\mathcal{C}} - 1\right) \tag{5.3}$$

$$\mathcal{V}_\mu \coloneqq \sum_{t=0}^{\varsigma-1} (\eta+1)^t \left(q^{m-D_t} - 1\right) \tag{5.4}$$

Beweis. Konstruiere einen vollständigen Decodierbaum $\mathcal{G}$ mit folgenden Eigenschaften:

- Die Höhe des Baumes ist ς (die Anzahl der Majority-Logic-Stufen).
- Der Wurzelknoten W repräsentiert $(\{0\}, \mathbb{F}_q^m)$, er hat Tiefe 0.
- Jeder Knoten außer den Blättern hat $\eta+1$ Kindknoten. (Der Kindknoten eines Knotens der Tiefe t hat selbst Tiefe $t+1$. Die Blätter haben Tiefe ς.)
- Jeder Knoten K der Tiefe t repräsentiert ein geordnetes Paar von Vektorräumen aus der Menge

$$\left\{ (U, U') \;\middle|\; U, U' \leq \mathbb{F}_q^m, U \oplus U' = \mathbb{F}_q^m, \dim(U) = D_t \right\}$$

- Für zwei beliebige Geschwisterknoten, korrespondierend zu (U_i, U_i') und (U_j, U_j'), eines Elternknotens, der (U, U') repräsentiert, gilt, die Unterräume U_i und U_j schneiden sich in U.

Solch ein Graph existiert, da laut Voraussetzung $\eta + 1 \leq \ell_{D_t,D_{t+1},m,q}$ für alle $t = \varsigma - 1, \varsigma - 2, \ldots, 0$.

Gehen wir nun zur Decodierung über. Per Induktion über $t \in \mathbb{N}_0$, $t \leq \varsigma$, zeigen wir, dass wir die Fehlersummen zu affinen Räume $\mathbf{u}^{'}+U$ für alle $\mathbf{u}^{'} \in \mathbb{F}_q^m \backslash U$, für alle von Knoten der Tiefe t repräsentierten Unterräume U bestimmen können. Bei $t = \varsigma$ beginnend und t in jedem Schritt um eins dekrementierend, werden bei $t = 0$ schließlich die Fehlersummen von $\mathbf{w} + \{0\}$, $\mathbf{w} \in \mathbb{F}_q^m \backslash \{0\}$ berechnet. Diese entsprechen gerade den Fehlersymbolen $\mathtt{E}_i$, $0 \leq i \leq n-1$.

Für $t = \varsigma$ (Induktionsanfang) sei K ein beliebiger der maximal $(\eta+1)^{\varsigma}$ Blattknoten. K repräsentiere $(U, U^{'})$. Laut Voraussetzung sind die Fehlersummen der affinen Räume $\mathbf{u}^{'} + U$ für alle $\mathbf{u}^{'} \in U^{'} \backslash \{0\}$ berechenbar.

Sei nun t, $0 \leq t \leq \varsigma - 1$, beliebig und sei K ein beliebiger der $(\eta+1)^t$ Knoten der Tiefe t. K repräsentiere $(U, U^{'})$. Die $\eta+1$ Kindknoten von K repräsentieren die Paare $(U_i, U_i^{'})$, $0 \leq i \leq \eta$. Sei $\mathbf{u}^{'} \in U^{'} \backslash \{0\}$ beliebig. Der Vektor $\mathbf{u}^{'}$ kann in keinen zwei verschiedenen Unterräumen U_i, U_j der $U_0, U_1, \ldots, U_\eta$ enthalten sein, denn

$$U_i \cap U_j \cap U^{'} \backslash \{0\} = U \cap U^{'} \backslash \{0\} = \emptyset$$

für alle $0 \leq i < j \leq \eta$. Ohne Beschränkung der Allgemeinheit nehmen wir an, $\mathbf{u}^{'}$ ist nicht in $U_0, U_1, \ldots, U_{\eta-1}$ enthalten. Für jedes i, $0 \leq i \leq \eta-1$ existiert ein $\mathbf{u}_i^{'} \in U_i^{'} \backslash \{0\}$, so dass $\mathbf{u}^{'} + U_i = \mathbf{u}_i^{'} + U_i$. Diese Zuordnung zwischen $(\mathbf{u}^{'}, i)$ und $\mathbf{u}_i^{'}$ ist bereits nach Konstruktion des Decodierbaums und vor der eigentlichen Decodierung bekannt, sie kann fest im Algorithmus verankert (*hard-coded*) werden und beeinflusst dessen Laufzeit nicht. Nach Induktionsannahme kennen wir die Fehlersummen zu den affinen Räumen $\mathbf{u}^{'} + U_i = \mathbf{u}_i^{'} + U_i$.

Die Vektormenge $\left\{\chi_{\mathbf{u}'+U_0}, \chi_{\mathbf{u}'+U_1}, \ldots, \chi_{\mathbf{u}'+U_{\eta-1}}\right\}$ ist orthogonal bezüglich des Vektors $\chi_{\mathbf{u}'+U}$. Wir wenden Proposition 4.3.1 an und erhalten die Fehlersumme von $\mathbf{u}^{'} + U$,

$$\mathtt{E} \circ \chi_{\mathbf{u}'+U} = \mu^0 \left(\mathtt{E} \circ \chi_{\mathbf{u}'+U_0}, \mathtt{E} \circ \chi_{\mathbf{u}'+U_1}, \ldots, \mathtt{E} \circ \chi_{\mathbf{u}'+U_{\eta-1}}\right). \qquad [5.5]$$

Zusammenfassend halten wir fest, dass wir für jeden der (maximal) $(\eta+1)^{\varsigma}$ Blattknoten $q^{m-D_\varsigma}-1$ Fehlersummen berechnen müssen. Da es $(\eta+1)^t$ Knoten

der Tiefe t, $0 \leq t \leq \varsigma - 1$, gibt, treffen wir im Schritt $D_{t+1} \to D_t$ also $(\eta+1)^t \cdot \left(q^{m-D_t} - 1\right)$ und insgesamt

$$\sum_{t=0}^{\varsigma-1} (\eta+1)^t \left(q^{m-D_t} - 1\right)$$

η-Mehrheitsentscheidungen. □

Bemerkung 5.3.2. Wir erinnern uns an Bemerkung 5.2.2 und halten fest, dass abhängig vom Code jeder Decodierbaum nur ein einziges Mal konstruiert werden muss. Die sequenzielle bzw. parallele Laufzeit bestimmt sich durch $\mathcal{V}_\mu$ und $\mathcal{V}_{\mathtt{E}}$ bzw. durch ς.

Bemerkung 5.3.3. Zu beachten ist, dass im Vergleich zu Proposition 5.2.1 die in Theorem 5.3.1 gestellten Voraussetzungen etwas stärker sind. Genauer gesagt fordern wir für das klassische Verfahren

$$\ell^*_{D_t,D_{t+1},m,q} \geq \eta \geq 2\tau$$

und für das verbesserte Verfahren

$$\ell_{D_t,D_{t+1},m,q} - 1 \geq \eta \geq 2\tau,$$

jeweils für alle $t = \varsigma - 1, \varsigma - 2, \ldots, 0$. Aus Proposition 5.1.4 ergibt sich unmittelbar, dass die letzte Forderung stärker ist. Bei den von uns im Folgenden betrachteten Abstufungen wird jedoch stets der Fall eintraten, dass für das größte $t = \varsigma - 1$ zum einen die Werte $\ell^*_{D_t,D_{t+1},m,q}$ und $\ell_{D_t,D_{t+1},m,q} - 1$ minimal werden und zum anderen für dieses maximale t gilt

$$\ell_{D_t,D_{t+1},m,q} - 1 = \ell^*_{D_t,D_{t+1},m,q}.$$

Folglich wird es für die von uns betrachteten Abstufungen keinen Unterschied hinsichtlich der gestellten Voraussetzungen geben.

Bemerkung 5.3.4. Beim klassischen Verfahren in Proposition 5.2.1 wird separat für jede Position, an der decodiert werden soll, eine Auswahl von affinen Räumen nach bestimmten Kriterien getroffen. Die im verbesserten Verfahren umgesetzte Idee ist, dieses Vorgehen zu kanonisieren und eine Beziehung zwischen den verschiedenen Positionen herzustellen. Praktisch gesehen vereinigen

wir die Decodierbäume aller Positionen in einem: Statt n Decodierbäume separat zu betrachten, beschreiben wir die gesamte Decodierlogik mittels eines einzigen Decodierbaums. Mit Hilfe eines ausgezeichneten Nebenklassenvertretersystems geben wir genau vor, mit welchen affinen Räumen an den jeweiligen Positionen decodiert werden soll. Ein affiner Raum kann zur Decodierung an jeder Position, die zu einem in diesem Raum enthaltenen Vektor korrespondiert, eingesetzt werden. Das verbesserte Verfahren zielt darauf ab, einen affinen Raum der Dimension D tatsächlich zur Decodierung an bis zu q^D Positionen zu verwenden. Dadurch gelingt es uns, die Anzahl der zu betrachtenden affinen Räume zu reduzieren.

Die verbesserte Decodierung stellt hinsichtlich der Maximalzahl der benötigten Mehrheitsentscheidungen und Fehlersummen tatsächlich eine Verbesserung gegenüber der klassischen Decodierung dar. Es gilt $\mathcal{V}_\mu \leq \mathcal{K}_\mu$, wobei Gleichheit nur bei $\varsigma = 1$ besteht, sowie $\mathcal{V}_{\mathtt{E}} < \mathcal{K}_{\mathtt{E}}$. Anschaulich wird die Verbesserung gegenüber der klassischen Decodierung durch Beispiel 5.3.5, in welchem wir das gleiche Wort wie in Beispiel 5.2.3 decodieren. Außerdem setzen wir in Abschnitt 6.3.4 die konkreten Werte für $\mathcal{K}_\mu$ und $\mathcal{V}_\mu$ bei festem $q, m, D_\mathcal{C}, \eta$ und gegebener Abstufung zueinander in Beziehung.

Beweis. Ist $t = 0$, so ist

$$(\eta + 1)^t (q^{m-D_t} - 1) = q^m - 1 = n = \eta^t \cdot n,$$

so dass bei $\varsigma = 1$ Gleichheit besteht. Sei also t, $0 < t \leq \varsigma$ beliebig. Es ist $q \geq 2$ und $D_t \geq t$ nach Definition, so dass

$$q^m - 1 > (q^{m-D_t+t} - q^t) \geq 2^t (q^{m-D_t} - 1).$$

Also ist wegen $2\eta \geq \eta + 1$

$$\begin{aligned} \eta^t \cdot n &= \eta^t (q^m - 1) \\ &> \eta^t \cdot 2^t (q^{m-D_t} - 1) \\ &\geq (\eta + 1)^t (q^{m-D_t} - 1). \end{aligned}$$

Es folgen

$$\mathcal{V}_\mu \coloneqq \sum_{t=0}^{\varsigma-1} (\eta + 1)^t \left(q^{m-D_t} - 1\right) \leq \sum_{t=0}^{\varsigma-1} \eta^t \cdot n = n \cdot \frac{\eta^\varsigma - 1}{\eta - 1} =: \mathcal{K}_\mu,$$

$$\mathcal{V}_{\mathtt{E}} \coloneqq (\eta+1)^{\varsigma}\left(q^{m-D_{\varsigma}}-1\right) < n\cdot\eta^{\varsigma} =: \mathcal{K}_{\mathtt{E}}.$$

□

Das im Beweis von Theorem 5.3.1 beschriebene Verfahren ist in Algorithmus 5.3.1 veranschaulicht.

Algorithmus 5.3.1: Verbesserte Decodierung .

Vorbedingung: Voraussetzungen aus Theorem 5.3.1,

Graph $\mathcal{G}$ aus dem Beweis von Theorem 5.3.1.

Zusicherung: $\mathtt{E}$ wird zurückgegeben.

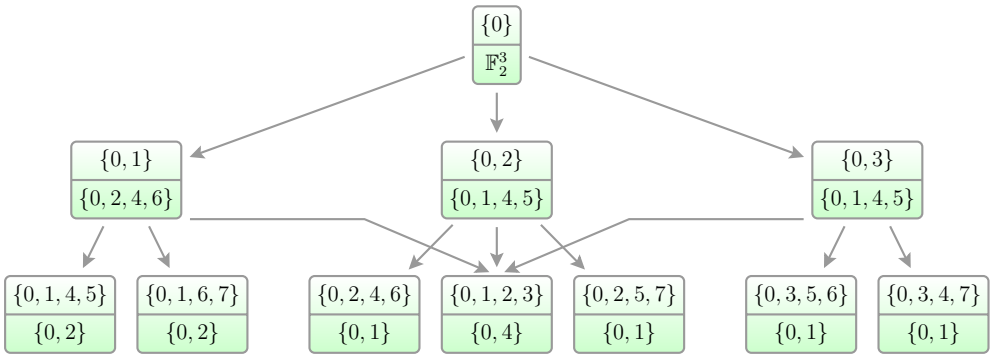
foreach Knoten der Tiefe ς korrespondierend zu $(U, U^{'})$ **do**
  foreach $\mathbf{u}^{'} \in U^{'}$, $\mathbf{u}^{'} \neq 0$ **do**
    $e_{\mathbf{u}'+U} = \mathtt{E} \circ \chi_{\mathbf{u}'+U}$
for $t = \varsigma - 1$ **to** 0 **do**
  foreach Knoten der Tiefe t korrespondierend zu $(U, U^{'})$ mit $\eta+1$ Kindknoten korrespondierend zu $\left(\left(U_0, U_0^{'}\right), \ldots, \left(U_{\eta}, U_{\eta}^{'}\right)\right)$ **do**
    foreach $\mathbf{u}^{'} \in U^{'}$, $\mathbf{u}^{'} \neq 0$ **do**
      wähle $I \subseteq \left\{0 \leq i \leq \eta \;\middle|\; \mathbf{u}^{'} \notin U_i\right\}$ mit $|I| = \eta$
      $e_{\mathbf{u}'+U} = \mu^0\left(e_{\mathbf{u}'+U_i} \mid i \in I\right)$

Ausgabe: $\left(e_{\{\mathbf{w}_0\}}, e_{\{\mathbf{w}_1\}}, \ldots, e_{\{\mathbf{w}_{n-1}\}}\right)$.

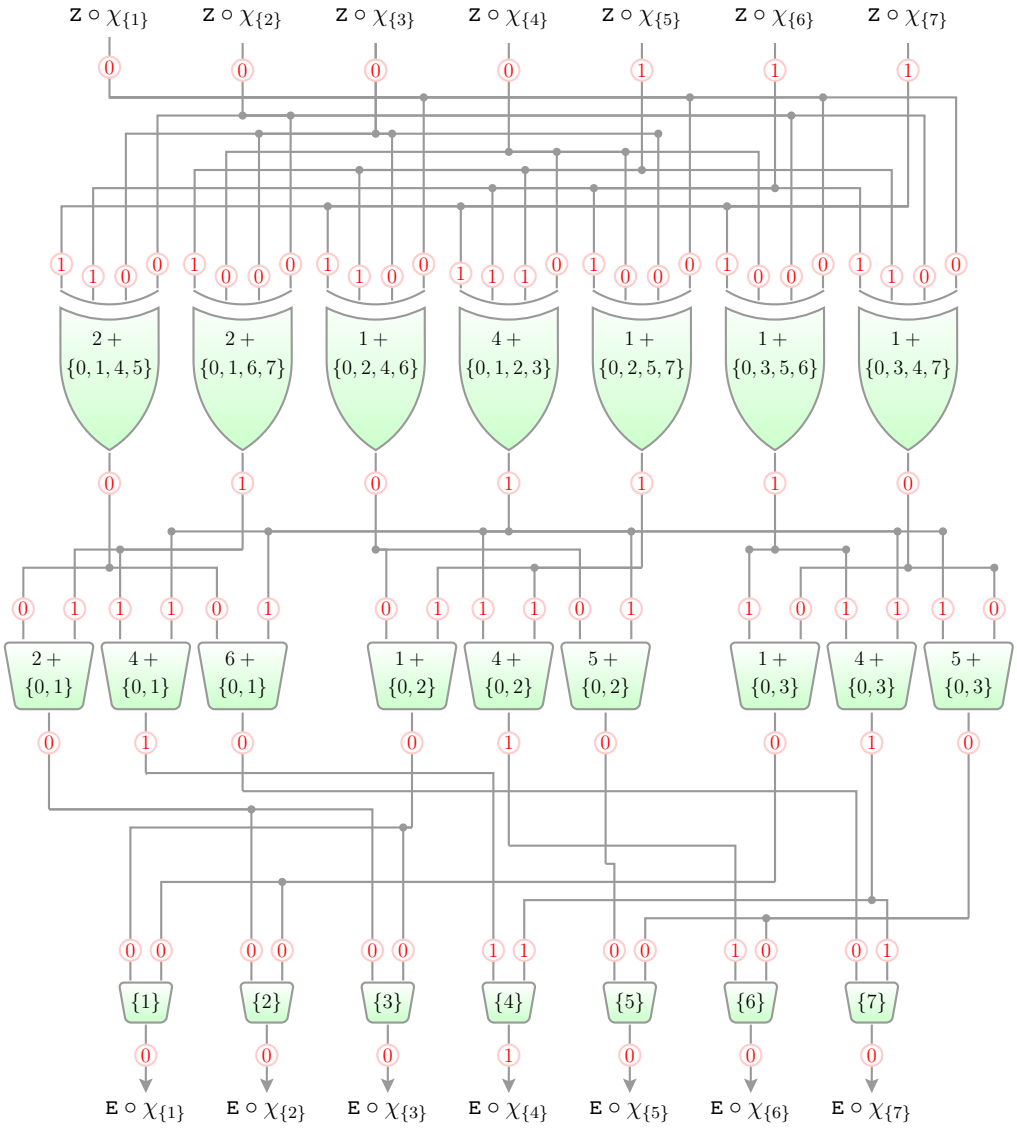
Beispiel 5.3.5. Wir verwenden dieselben Notationen, betrachten denselben Code und wählen dieselbe Abstufung wie in Beispiel 5.2.3,

$$\mathcal{C}^{\perp} \coloneqq \left\langle \chi_{\mathbf{v}+U} \in \mathbb{F}_2^7 \mid U \leq \mathbb{F}_2^3, \dim U = 2, \mathbf{v} \in \mathbb{F}_2^3 \backslash U \right\rangle_{\mathbb{F}_2} \leq F_2^7,$$
$$\mathcal{C} = \mathcal{C}^{\perp} \oplus \langle 1,1,1,1,1,1,1 \rangle_{\mathbb{F}_2} \leq F_2^7,$$

sowie $\eta \coloneqq 2$, $\varsigma \coloneqq 2$ und $3 > D_2 \coloneqq 2 > D_1 \coloneqq 1 > D_0 \coloneqq 0$. Konstruieren wir einen Decodierbaum wie in Theorem 5.3.1 skizziert. Jeder Knoten repräsentiert ein geordnetes Paar von Unterräumen, deren innere direkte Summe gerade $\mathbb{F}_2^3$ ist.



Dieser Decodierbaum definiert das Decodierschema für alle $a \in \mathbb{Z}_7$.



Die Decodierung verläuft in obigem Schema von oben nach unten. Die Symbolik des Decodierschemas ist die gleiche, wie wir sie in Beispiel 5.2.3 gesehen haben: XOR-Gatter und Majoritätsgatter werden durch  beziehungsweise  dargestellt und repräsentieren jeweils einen affinen Raum. Die XOR-Gatter dienen wie zuvor dazu, zu Beginn der Decodierung die Checksummen beziehungsweise die Fehlersummen der zweidimensionalen affinen Räume zu berechnen. Der Decoder ist aus sieben XOR-Gattern und

$$16 = 7 + (2+1)\cdot(2^2-1) = \sum_{j=0}^{\varsigma-1} (\eta+1)^j \left(q^{m-D_j}-1\right)$$

Majoritätsgattern aufgebaut.

Erneut weisen wir daraufhin, dass bei „kleinen“ Codes wie in diesem Beispiel, die wenigen affinen Räume der Menge $\mathcal{A}^*_{D_\varsigma,m,q}$ an mehreren Positionen zur Decodierung eingesetzt werden. Entgegen der gemäß Term 5.3 zu erwartenden $9 = (2+1)^2$ XOR-Gatter werden sieben verwendet. Bei größeren Codes können im schlechtesten Fall tatsächlich $(\eta+1)^\varsigma \left(q^{m-D_\varsigma}-1\right)$ XOR-Gatter zum Einsatz kommen (siehe Beispiel 5.3.6).

Beispielhaft decodieren wir erneut das Wort $\mathtt{z} \coloneqq \chi_{\{5,6,7\}}$ und beschriften die Eingänge und Ausgänge der Gatter entsprechend.

Der Decoder liefert wie zuvor $\chi_{\{4\}}$ als potentielles Fehlerwort und $\mathtt{z} - \chi_{\{4\}} = \chi_{\{4,5,6,7\}}$ als potentielles Codewort. Dieses ist tatsächlich das übertragene Codewort, wenn sichergestellt ist, dass nicht mehr als ein Fehler aufgetreten ist. In diesem Fall, $\tau \leq 1$, sind mit obig gewählter Abstufung die Voraussetzungen aus Theorem 5.3.1 erfüllt,

$$\ell_{D_1=1,D_2=2,m=3,2} - 1 = 2 = 2\tau$$

und

$$\ell_{D_0=0,D_1=1,m=3,2} - 1 = 6 > 2 = 2\tau. \qquad \triangleleft$$

Das folgende Beispiel belegt, dass es tatsächlich Codes gibt, bei denen bei schlechter Wahl der Vektorraumpaare keines von mehr als einem Knoten repräsentiert wird. Währenddessen eine geschickte Wahl der Unterräume die Effizienz der Decodierung weiter steigert.

Beispiel 5.3.6. Sei

$$\mathbb{F}_3^3 \coloneqq \{\mathbf{w}_0, \mathbf{w}_1, \ldots, \mathbf{w}_{26} = 0\}.$$

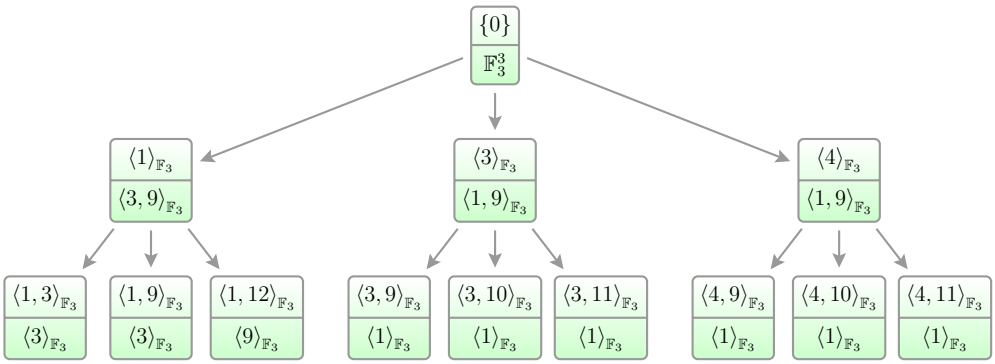
Um die Notation abzukürzen, fassen wir jeden Vektor $(a_2, a_1, a_0) \in \mathbb{F}_3^3$ als Ternärzahl auf und identifizieren ihn eindeutig durch die Zahl $\sum_{i=0}^{2} a_i 3^i$,

$$(a_2, a_1, a_0) \in \mathbb{F}_3^3 \quad \longleftrightarrow \quad \sum_{i=0}^{2} a_i 3^i \in \mathbb{Z}_{27}.$$

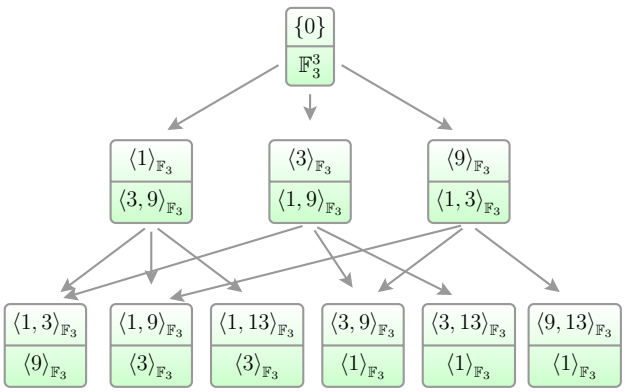
Weiterhin seien $\eta \coloneqq 2$, $\varsigma \geq 2$ und

$$D_2 \coloneqq 2 > D_1 \coloneqq 1 > D_0 \coloneqq 0.$$

Die Knoten der Tiefe $0, 1, 2$ eines Decodierbaums, wie in Theorem 5.3.1 skizziert, könnten wie folgt aussehen.



Keine zwei Knoten repräsentieren das gleiche Paar von Vektorräumen. Darüber hinaus wäre auch dieser Graph denkbar.



Statt neun sind es nur noch sechs Vektorraumpaare, die von den Knoten der Tiefe zwei repräsentiert werden. Wir haben jedes Vektorraumpaar nur einmal aufgeführt und stattdessen durch gerichtete Kanten die Knoten der Tiefe zwei symbolisiert. Streng genommen ist es nicht mehr korrekt, von einem Decodierbaum zu sprechen. ◁

Im Folgenden betrachten wir zwei Spezialfälle von Decodierparametern, die uns erlauben, noch effizienter zu decodieren.

5.3.2 Der Spezialfall I: $\varsigma \geq s$ für ein s, $2 \leq s \leq m - \eta + 1$, mit $D_s = s$

Ist $\varsigma \geq s$ für ein s, $2 \leq s \leq m-\eta+1$, mit $D_s = s$, so ist es möglich, dass mehrere Knoten das gleiche Vektorraumpaar repräsentieren und so Vektorraumpaare zur Decodierung an mehreren Positionen eingesetzt werden können.

Korollar 5.3.7. *Die Voraussetzungen aus Theorem 5.3.1 seien erfüllt. Darüber hinaus gelte $\eta \geq 2$, $m - \eta > 0$ und $\varsigma \geq s$ für ein s, $2 \leq s \leq m - \eta + 1$, mit $D_s = s$.*

Es gibt ein Majority-Logic-Verfahren, das anhand von maximal

$$\left(q^{m-D_\varsigma} - 1\right) \cdot \begin{cases} \binom{\eta+s}{s} \cdot (\eta+1)^{\varsigma-s} & s \leq \varsigma - 1, \\ \binom{\eta+\varsigma}{\varsigma} & s = \varsigma \end{cases} \qquad [5.6]$$

Fehlersummen zu affinen Räumen aus $\mathcal{A}^*_{D_\mathcal{C},m,q}$ in ς Majority-Logic-Stufen,

$$D_\varsigma \to D_{\varsigma-1} \ldots \to D_1 \to D_0 \coloneqq 0.$$

mit Hilfe von maximal

$$\begin{cases} \sum\limits_{t=0}^{s} \binom{\eta+t}{t} \left(q^{m-t} - 1\right) + \sum\limits_{t=s+1}^{\varsigma-1} \binom{\eta+s}{s} \cdot (\eta+1)^{t-s} \left(q^{m-D_t} - 1\right) & s < \varsigma - 1, \\ \sum\limits_{t=0}^{\varsigma-1} \binom{\eta+t}{t} \left(q^{m-t} - 1\right) & s \geq \varsigma - 1 \end{cases} \qquad [5.7]$$

η-Mehrheitsentscheidungen alle n Fehlersymbole $\mathtt{E}_i$, $0 \leq i \leq n-1$, korrekt berechnet.

Beweis. Zunächst weisen wir darauf hin, dass das Decodierverfahren identisch ist zu jenem aus Theorem 5.3.1. Einzig der zugrundeliegende Decodierbaum ist genauer spezifiziert. Die Voraussetzung $D_s = s$ impliziert $D_t = t$ für alle t, $0 \leq t \leq s$. Wir konstruieren den Decodierbaum wie folgt.

Für Knoten der Tiefe t, $1 \leq t \leq \min(s, m-\eta)$, ziehen wir nur die kanonischen Einheitsvektoren des $\mathbb{F}_q^m$ heran. Sei also t, $1 \leq t \leq \min(s, m-\eta)$, beliebig. Die Knoten der Tiefe t repräsentieren die $\binom{\eta+t}{t}$ verschiedenen Paare

$$\left(\langle \mathbf{e}_i \mid i \in I \rangle_{\mathbb{F}_q}, \langle \mathbf{e}_i \mid i \in \mathbb{Z}_m \backslash I \rangle_{\mathbb{F}_q}\right) \quad \text{für alle } I \subseteq \mathbb{Z}_{\eta+t} \text{ mit } |I| = t.$$

Ist $s = m - \eta + 1$, repräsentieren die Knoten der Tiefe s die verschiedenen Paare

$$\left(\langle \mathbf{e}_i \mid i \in I \rangle_{\mathbb{F}_q}, \langle \mathbf{e}_i \mid i \in \mathbb{Z}_m \backslash I \rangle_{\mathbb{F}_q}\right)$$

für alle $I \subseteq \mathbb{Z}_m$ mit $|I| = m - \eta + 1$ sowie

$$\left(U_I \coloneqq \langle \mathbf{e}_i \mid i \in I \rangle_{\mathbb{F}_q} \oplus \langle \mathbf{v}_I \rangle_{\mathbb{F}_q}, U_I^{'}\right),$$

für alle $I \subseteq \mathbb{Z}_m$ mit $|I| = m-\eta$, $\mathbf{v}_I \notin \bigcup\limits_{j \in \mathbb{Z}_m \backslash I} \langle \mathbf{e}_i, \mathbf{e}_j \mid i \in I \rangle_{\mathbb{F}_q}$ beliebig, aber fest. (Solch ein $\mathbf{v}_I$ existiert, da laut Voraussetzung $\eta \geq 2$. Beispielsweise kann $\mathbf{v}_I \coloneqq \sum_{j \in \mathbb{Z}_m \backslash I} \mathbf{e}_j$ gewählt werden.) Die Anzahl der Vektorraumpaare ist gegeben durch

$$\binom{m}{m-\eta+1} + \binom{m}{m-\eta} = \binom{m+1}{\eta} = \binom{\eta+s}{s}.$$

Zusammenfassend halten wir fest, dass wir keine

$$(\eta+1)^t \cdot (q^{m-t} - 1)$$

sondern nur

$$\binom{\eta+t}{t} \cdot (q^{m-t} - 1)$$

Mehrheitsentscheidungen in der Majority-Logic-Stufe $D_{t+1} \to D_t$ für alle t, $0 \leq t \leq s$ benötigen.

Indem wir die zu betrachtenden Vektorraumpaare für Knoten der Tiefe s minimieren, senken wir auch die Anzahl der zu betrachtenden Vektorraumpaare für Knoten der Tiefe größer s. □

Als Beispiel dient der untere Graph in Beispiel 5.3.6.

5.3.3 Der Spezialfall II: $\varsigma \geq 2$, $D_2 = 2$, $2^{m-1} \geq \eta + 2$

Vorausgesetzt $\varsigma \geq 2$, $D_2 = 2$ ($D_1 = 1$ implizierend) und $2^{m-1} \geq \eta + 2$ ist es grundsätzlich möglich, $\binom{\eta+2}{2}$ (verschiedene) Vektorraumpaare so zu wählen, dass jeder der $(\eta+1)^2$ Knoten der Tiefe zwei eines dieser Vektorraumpaare repräsentiert. Mit anderen Worten, wir benötigen keine

$$(\eta+1)^2 \cdot (q^{m-2} - 1)$$

sondern nur

$$\binom{\eta+2}{2} \cdot (q^{m-2} - 1)$$

Mehrheitsentscheidungen in der Majority-Logic-Stufe $D_3 \to 2$.

Korollar 5.3.8. *Die Voraussetzungen aus Theorem 5.3.1 seien erfüllt. Darüber hinaus gelte $\varsigma \geq 2$, $D_2 = 2$ und $2^{m-1} \geq \eta + 2$.*

Es gibt ein Majority-Logic-Verfahren, das anhand von maximal

$$\left(q^{m-D_\varsigma} - 1\right) \cdot \begin{cases} \binom{\eta+2}{2} \cdot (\eta+1)^{\varsigma-2} & 2 \leq \varsigma - 1, \\ \binom{\eta+\varsigma}{\varsigma} & 2 = \varsigma \end{cases} \qquad [5.8]$$

Fehlersummen zu affinen Räumen aus $\mathcal{A}^*_{D_\mathcal{C},m,q}$ in ς Majority-Logic-Stufen,

$$D_\varsigma \to D_{\varsigma-1} \ldots \to D_1 \to D_0 \coloneqq 0.$$

mit Hilfe von maximal

$$\begin{cases} \sum_{t=0}^{2} \binom{\eta+t}{t} (q^{m-t} - 1) + \sum_{t=3}^{\varsigma-1} \binom{\eta+2}{2} \cdot (\eta+1)^{t-2} \left(q^{m-D_t} - 1\right) & 2 < \varsigma - 1, \\ \sum_{t=0}^{\varsigma-1} \binom{\eta+t}{t} (q^{m-t} - 1) & 2 \geq \varsigma - 1 \end{cases} \qquad [5.9]$$

η-Mehrheitsentscheidungen alle n Fehlersymbole $\mathtt{E}_i$, $0 \leq i \leq n-1$, korrekt berechnet.

Beweis. Wie auch beim vorherigen Spezialfall ist das Decodierverfahren identisch zu jenem aus Theorem 5.3.1 und einzig den zugrundeliegende Decodierbaum spezifizieren wir.

Dazu ordnen wir die Vektoren aus $\{0,1\}^m \subseteq \mathbb{F}_q^m$ mit ungeradem Gewicht,

$$S \coloneqq \left\{ \mathbf{v} \in \{0,1\}^m \subseteq \mathbb{F}_q^m \;\middle|\; \omega(\mathbf{v}) \text{ ungerade} \right\}.$$

Für zwei (verschiedene) Vektoren $\mathbf{v}, \mathbf{v}^{'} \in S$ sei $\mathbf{v} <_S \mathbf{v}^{'}$ genau dann, wenn eine der beiden Bedingungen zutrifft:

- Das Gewicht von $\mathbf{v}$ ist kleiner als jenes von $\mathbf{v}^{'}$, $\omega(\mathbf{v}) < \omega(\mathbf{v}^{'})$;
- das Gewicht von $\mathbf{v}$ gleicht jenem von $\mathbf{v}^{'}$, $\omega(\mathbf{v}) = \omega(\mathbf{v}^{'})$, und es gibt ein $s \in \mathbb{Z}_m$, so dass $v_i = v_i^{'}$ für alle $0 \leq i \leq s-1$ und $v_s = 0, v_s^{'} = 1$;

Damit ist die Relation $<_S$ auf S eine strenge Totalordnung. Die Kardinalität der Menge S ist 2^{m-1}, was man per Induktion über m nachweisen kann.

$$\mathbf{e}_{m-1} <_S \mathbf{e}_{m-2} <_S \ldots <_S \mathbf{e}_0 <_S \mathbf{e}_{m-1} + \mathbf{e}_{m-2} + \mathbf{e}_{m-3} <_S \ldots. \quad [5.10]$$

Die $\eta + 1$ Knoten der Tiefe eins repräsentieren die Paare $\left(\langle \mathbf{v}_i \rangle_{\mathbb{F}_q}, \langle \mathbf{v}_i \rangle_{\mathbb{F}_q}^{'}\right)$, $0 \leq i \leq \eta$, wobei

$$\mathbf{v}_0 <_S \mathbf{v}_1 <_S \ldots <_S \mathbf{v}_{\eta+1}$$

hinsichtlich der obigen Ordnung die kleinsten $\eta + 2$ Vektoren aus der Menge S sind und für jedes $0 \leq i \leq \eta$ der Vektorraum $\langle \mathbf{v}_i \rangle_{\mathbb{F}_q}^{'}$ ein beliebiger Komplementärraum zu $\langle \mathbf{v}_i \rangle_{\mathbb{F}_q}$ ist. Darauf aufbauend repräsentieren die Knoten der Tiefe zwei die $\binom{\eta+2}{2}$ Paare $\left(\langle \mathbf{v}_i, \mathbf{v}_j \rangle_{\mathbb{F}_q}, \langle \mathbf{v}_i, \mathbf{v}_j \rangle_{\mathbb{F}_q}^{'}\right)$, $0 \leq i < j \leq \eta + 1$.

An dieser Stelle bleibt zu begründen, warum die von Geschwisterknoten repräsentierten zweidimensionalen Vektorraumpaare tatsächlich verschieden sind,

$$\langle \mathbf{v}_i, \mathbf{v}_j \rangle_{\mathbb{F}_q} \neq \langle \mathbf{v}_i, \mathbf{v}_l \rangle_{\mathbb{F}_q}$$

für beliebiges i, $0 \leq i \leq \eta$, für alle $j, l \neq i$, $0 \leq j < l \leq \eta + 1$. Wären beide Vektorräume gleich, so wäre

$$a_i \cdot \mathbf{v}_i + a_j \cdot \mathbf{v}_j = \mathbf{v}_l \in S$$

für gewisse $a_i, a_j \in \mathbb{F}_q^*$. Wir erinnern uns, $\mathbf{v}_l$ hat ungerades Gewicht mit Einträgen aus $\{0, 1\}$. Ein Widerspruch: Im Fall $p = 2$ ist das Gewicht von $a_i \cdot \mathbf{v}_i + a_j \cdot \mathbf{v}_j$ gerade; im Fall $p > 2$ kann der Vektor $a_i \cdot \mathbf{v}_i + a_j \cdot \mathbf{v}_j$ nicht sowohl in $\{0, 1\}^m$ liegen als auch ein ungerades Gewicht haben. □

Bemerkung 5.3.9. Wir weisen daraufhin, dass im Fall $q = 2$, $\varsigma \geq 2$ die Ungleichung

$$2^{m-1} \geq \eta + 1$$

unmittelbar gilt, sofern die Voraussetzungen aus Theorem 5.3.1 erfüllt sind. Denn, um Theorem 5.3.1 anwenden zu können, fordern wir, dass

$$\eta \leq \ell_{D_t, D_{t+1}, m, 2} - 1$$

für alle $t = \varsigma - 1, \varsigma - 2, \ldots, 0$ ist. Also insbesondere mit Blick auf Proposition 5.1.4

$$\eta \leq \ell_{D_1, D_2, m, 2} - 1 = \ell_{1,2,m,2} - 1 = 2^{m-1} - 2.$$

Neben dem unteren Decodierbaum in Beispiel 5.3.6 geben wir ein weiteres Beispiel.

Beispiel 5.3.10. Sei

$$\mathbb{F}_2^5 \coloneqq \{\mathbf{w}_0, \mathbf{w}_1, \ldots, \mathbf{w}_{31} = 0\}.$$

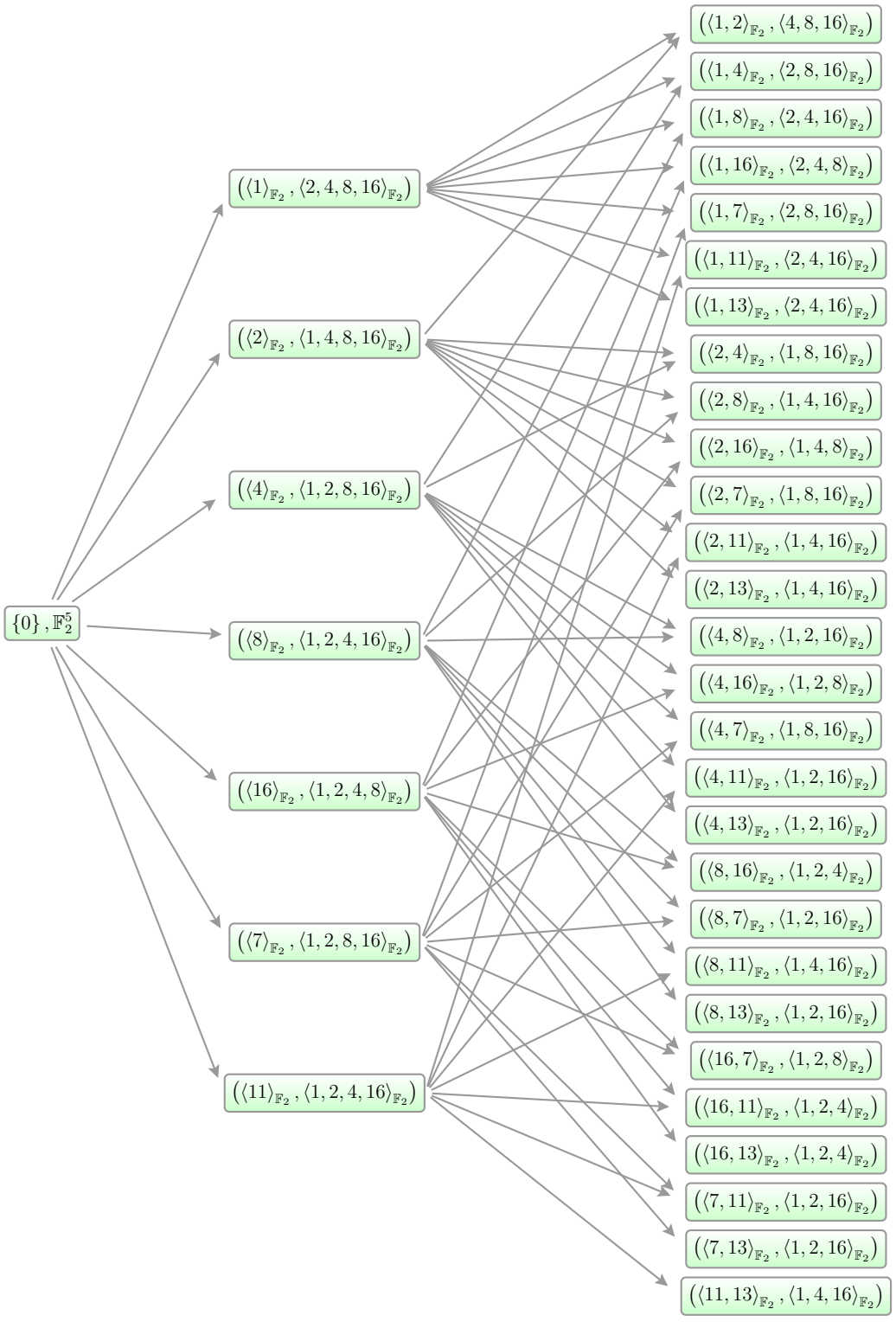
Um die Notation abzukürzen, fassen wir jeden Vektor $(a_4, \ldots, a_1, a_0) \in \mathbb{F}_2^5$ als Binärzahl auf und identifizieren ihn eindeutig durch die Zahl $\sum_{i=0}^{4} a_i 2^i$,

$$(a_4, \ldots, a_1, a_0) \in \mathbb{F}_2^5 \quad \longleftrightarrow \quad \sum_{i=0}^{4} a_i 2^i \in \mathbb{Z}_{32}.$$

Weiterhin seien $\eta \coloneqq 6$, $\varsigma \geq 2$ und

$$D_2 \coloneqq 2 > D_1 \coloneqq 1 > D_0 \coloneqq 0.$$

Dann repräsentieren die Knoten der Tiefe zwei $28 = \binom{\eta+2}{2}$ verschiedene Vektorraumpaare.


$(\{0\}, \mathbb{F}_2^5)$
$(\langle 1\rangle_{\mathbb{F}_2}, \langle 2,4,8,16\rangle_{\mathbb{F}_2})$
$(\langle 2\rangle_{\mathbb{F}_2}, \langle 1,4,8,16\rangle_{\mathbb{F}_2})$
$(\langle 4\rangle_{\mathbb{F}_2}, \langle 1,2,8,16\rangle_{\mathbb{F}_2})$
$(\langle 8\rangle_{\mathbb{F}_2}, \langle 1,2,4,16\rangle_{\mathbb{F}_2})$
$(\langle 16\rangle_{\mathbb{F}_2}, \langle 1,2,4,8\rangle_{\mathbb{F}_2})$
$(\langle 7\rangle_{\mathbb{F}_2}, \langle 1,2,8,16\rangle_{\mathbb{F}_2})$
$(\langle 11\rangle_{\mathbb{F}_2}, \langle 1,2,4,16\rangle_{\mathbb{F}_2})$
$(\langle 1,2\rangle_{\mathbb{F}_2}, \langle 4,8,16\rangle_{\mathbb{F}_2})$
$(\langle 1,4\rangle_{\mathbb{F}_2}, \langle 2,8,16\rangle_{\mathbb{F}_2})$
$(\langle 1,8\rangle_{\mathbb{F}_2}, \langle 2,4,16\rangle_{\mathbb{F}_2})$
$(\langle 1,16\rangle_{\mathbb{F}_2}, \langle 2,4,8\rangle_{\mathbb{F}_2})$
$(\langle 1,7\rangle_{\mathbb{F}_2}, \langle 2,8,16\rangle_{\mathbb{F}_2})$
$(\langle 1,11\rangle_{\mathbb{F}_2}, \langle 2,4,16\rangle_{\mathbb{F}_2})$
$(\langle 1,13\rangle_{\mathbb{F}_2}, \langle 2,4,16\rangle_{\mathbb{F}_2})$
$(\langle 2,4\rangle_{\mathbb{F}_2}, \langle 1,8,16\rangle_{\mathbb{F}_2})$
$(\langle 2,8\rangle_{\mathbb{F}_2}, \langle 1,4,16\rangle_{\mathbb{F}_2})$
$(\langle 2,16\rangle_{\mathbb{F}_2}, \langle 1,4,8\rangle_{\mathbb{F}_2})$
$(\langle 2,7\rangle_{\mathbb{F}_2}, \langle 1,8,16\rangle_{\mathbb{F}_2})$
$(\langle 2,11\rangle_{\mathbb{F}_2}, \langle 1,4,16\rangle_{\mathbb{F}_2})$
$(\langle 2,13\rangle_{\mathbb{F}_2}, \langle 1,4,16\rangle_{\mathbb{F}_2})$
$(\langle 4,8\rangle_{\mathbb{F}_2}, \langle 1,2,16\rangle_{\mathbb{F}_2})$
$(\langle 4,16\rangle_{\mathbb{F}_2}, \langle 1,2,8\rangle_{\mathbb{F}_2})$
$(\langle 4,7\rangle_{\mathbb{F}_2}, \langle 1,8,16\rangle_{\mathbb{F}_2})$
$(\langle 4,11\rangle_{\mathbb{F}_2}, \langle 1,2,16\rangle_{\mathbb{F}_2})$
$(\langle 4,13\rangle_{\mathbb{F}_2}, \langle 1,2,16\rangle_{\mathbb{F}_2})$
$(\langle 8,16\rangle_{\mathbb{F}_2}, \langle 1,2,4\rangle_{\mathbb{F}_2})$
$(\langle 8,7\rangle_{\mathbb{F}_2}, \langle 1,2,16\rangle_{\mathbb{F}_2})$
$(\langle 8,11\rangle_{\mathbb{F}_2}, \langle 1,4,16\rangle_{\mathbb{F}_2})$
$(\langle 8,13\rangle_{\mathbb{F}_2}, \langle 1,2,16\rangle_{\mathbb{F}_2})$
$(\langle 16,7\rangle_{\mathbb{F}_2}, \langle 1,2,8\rangle_{\mathbb{F}_2})$
$(\langle 16,11\rangle_{\mathbb{F}_2}, \langle 1,2,4\rangle_{\mathbb{F}_2})$
$(\langle 16,13\rangle_{\mathbb{F}_2}, \langle 1,2,4\rangle_{\mathbb{F}_2})$
$(\langle 7,11\rangle_{\mathbb{F}_2}, \langle 1,2,16\rangle_{\mathbb{F}_2})$
$(\langle 7,13\rangle_{\mathbb{F}_2}, \langle 1,2,16\rangle_{\mathbb{F}_2})$
$(\langle 11,13\rangle_{\mathbb{F}_2}, \langle 1,4,16\rangle_{\mathbb{F}_2})$

◁

5.3.4 Maximale Größe der Graphen

Für verschiedene Tripel (q, m, η) haben wir Decodierbäume konstruiert und dabei versucht, möglichst wenige Vektorraumpaare heranziehen zu müssen. Dabei setzen wir die in Korollar 5.3.7 (Spezialfall I) und Korollar 5.3.8 (Spezialfall II) vorgeschlagenen Konstruktionsanweisungen um. Im Spezialfall I geben wir explizit vor, welche Vektorraumpaare für die Knoten bis zur Tiefe s des Decodierbaums verwendet werden können. Im Spezialfall II definieren wir die Vektorraumpaare für Knoten der Tiefe eins und zwei.

Indem wir die zu betrachtenden Vektorraumpaare für Knoten der Tiefe t minimieren, senken wir auch die Anzahl der zu betrachtenden Vektorraumpaare für Knoten der Tiefe größer t. Wir wollten wissen, wie groß dieser Einfluss ist. Dazu haben wir weitere Vektorraumpaare für Knoten größerer Tiefe mit Hilfe eines eigens geschriebenen Computerprogramms konstruiert. Mit dem Ergebnis, dass auch bei größerer Tiefe die Anzahl der benötigten Vektorraumpaare wesentlich(!) kleiner als die Anzahl der Knoten ist. Genau genommen ist – entgegen unserer Erwartungen – die Anzahl der Vektorraumpaare der Tiefe t stets signifikant kleiner als das Produkt aus $(\eta + 1)$ und der Anzahl der Vektorraumpaare der Tiefe $t - 1$. Der positive Effekt pflanzt sich also fort: Die durch die Spezialfälle vorgegebene Struktur bis zu einer gewissen Tiefe bewirkt auch bei Knoten größerer Tiefe, dass häufig mehrere Knoten dasselbe Vektorraumpaar repräsentieren.

Tabelle 5.1 gibt für einige Tripel (q, m, η) konkret an, wie viele verschiedene Vektorraumpaare für Knoten der Tiefe t ausreichend sind, damit eine korrekte Decodierung gewährleistet ist. Wir können jedoch nicht ausschließen, dass mitunter noch weniger Vektorraumpaare als angegeben ausreichen. Die jeweils erste Zeile enthält die Potenzen von $\eta + 1$ und damit die gemäß Theorem 5.3.1 maximale Anzahl der zu betrachtenden Vektorraumpaare. Einträge mit einem hochgestellten Stern wurden mit Hilfe von Korollar 5.3.7 (Spezialfall I) ermittelt; für Einträge mit zwei hochgestellten Sternen haben wir Korollar 5.3.8 (Spezialfall II) angewendet. Einträge mit drei hochgestellten Sternen können aus beiden Korollaren hergeleitet werden.

Tabelle 5.1: Mögliche Anzahl der verschiedenen Vektorraumpaare repräsentiert durch Knoten der Tiefe t, wobei $D_t = t$

q	m	η	$t < m$						
			0	1	2	3	4	5	6
2		2	1	3	9	27	81	243	729
	3		1	3	6***				
	4		1	3	6***	10*			
	5		1	3	6***	10*	15*		
	6		1	3	6***	10*	15*	21*	
	7		1	3	6***	10*	15*	21*	28*
	8		1	3	6***	10*	15*	21*	28*
2		6	1	7	49	343	2.401	16.807	117.649
	4		1	7	28**				
	5		1	7	28**	105			
	6		1	7	28**	93	329		
	7		1	7	28***	87	262	894	
	8		1	7	28***	84*	228	648	2.164
	9		1	7	28***	84*	210*	526	
2		14	1	15	225	3.375			
	5		1	15	120**				
	6		1	15	120**	995			
	7		1	15	120**	911			

q	m	η	$D_t = t$					
			0	1	2	3	4	5
3	>t	2	1	3	9	27	81	243
	3		1	3	6***			
	4		1	3	6***	10*		
	5		1	3	6***	10*	15*	
	6		1	3	6***	10*	15*	21*
3		12	1	13	169	2.197	28.561	
	4		1	13	104			
	5		1	13	91**	608		
	6		1	13	91**	566	3.574	
3		38	1	39	1.521	59.319		
	5		1	39	953			
	6		1	39	824	16.314		
4		4	1	5	25	125	625	3.125
	3		1	5	17			
	4		1	5	15**	44		
	5		1	5	15***	38	104	
	6		1	5	15***	35*	82	217
4		20	1	21	441	9.261	194.481	
	4		1	21	292			
	5		1	21	246	2.810		
	6		1	21	231**	2.548	27.075	

5.4 Hybriddecodierung

Wir möchten ein Hybriddecodierverfahren vorstellen, bei welchem Mehrheitsentscheidungen mit Additionen und Subtraktionen in $\mathbb{F}_{q_C}$ verknüpft werden, um das Fehlerwort zu berechnen. Genauer gesagt können wir unter gewissen Voraussetzungen eine Mehrheitsentscheidung durch $q-1$ Additionen bzw. Subtraktionen ersetzen. Dies ist dann von Vorteil, wenn sich die $q-1$ Additionen/Subtraktionen effizienter als eine Mehrheitsentscheidung implementieren lässt.

Die Decodierung ist wie auch zuvor in mehreren Stufen aufgebaut, wobei nun neben Majority-Logic-Stufen auch Addition-Subtraktion-Stufen zum Tragen kommen. Die Grundidee der Hybriddecodierung, gewissermaßen der Induktionsdecodierschritt, ist in Lemma 5.4.1 verankert. Zur Veranschaulichung der Idee geben wir direkt im Anschluss an Lemma 5.4.1 zwei Beispiele. Das vollständige Hybriddecodierverfahren ist dann in Theorem 5.4.4 erklärt. Anschließend zeigen wir, dass es häufig möglich ist, die Hybriddecodierung hinsichtlich der Anzahl der zu bestimmenden Fehlersummen zu optimieren. Abschließend demonstrieren wir das Hybriddecodierverfahren anhand eines Beispiels, in welchem wir diese Optimierungsmöglichkeit aufgreifen.

5.4.1 Induktionsschritt der Hybriddecodierung

Lemma 5.4.1. *Seien $D \in \mathbb{N}_0$, $D < m-1$ und $\eta \in \mathbb{N}$ mit*

$$\ell_{D,D+1,m,q} - 1 \geq \eta$$

beliebig.

Gegeben seien $(D+1)$-dimensionale Vektorräume $U_i \leq \mathbb{F}_q^m$, $0 \leq i \leq \eta$, die sich paarweise in einem D-dimensionalen Vektorraum $U \leq \mathbb{F}_q^m$ schneiden. Sei $U' \leq \mathbb{F}_q^m$ ein Komplementärraum zu U. Weiterhin sei V ein nichttrivialer Unterraum von U', der vollständig in der Vereinigung aller U_i, $0 \leq i \leq \eta$, liegt,

$$\{0\} \subset V \subseteq \bigcup_{0 \leq i \leq \eta} U_i \cap U'.$$

Wir gehen davon aus, es treten bei der Übertragung des Codeworts maximal $\eta/2$ *Fehler auf,*

$$2\tau \leq \eta.$$

Angenommen, wir kennen die Fehlersummen zu den affinen Räumen $\mathbf{u}' + U_i$, $\mathbf{u}' \in \mathbb{F}_q^m \backslash U_i$, $0 \leq i \leq \eta$. *Dann können wir mit* $\mathcal{H}_\mu(D, \dim V)$ η*-Mehrheitsentscheidungen sowie* $\mathcal{H}_\pm(D, \dim V)$ *Additionen/Subtraktionen in* $\mathbb{F}_{q_\mathcal{C}}$ *die Fehlersummen zu allen affinen Räumen* $\mathbf{u}' + U$, $\mathbf{u}' \in \mathbb{F}_q^m \backslash U$ *in zwei Schritten, einem Majority-Logic-Schritt und einem Addition-Subtraktion-Schritt, berechnen, wobei*

$$\mathcal{H}_\mu(D, \dim V) \coloneqq q^{m-D-\dim V} \cdot \left(1 + (q-2) \cdot \frac{q^{\dim V} - 1}{q-1}\right), \qquad [5.11]$$

$$\mathcal{H}_\pm(D, \dim V) \coloneqq q^{m-D-\dim V} \cdot \left(q^{\dim V} - 1\right) - q + 1. \qquad [5.12]$$

Beweis. Jeder Vektorraum U_i, $0 \leq i \leq \eta$, ist von der Form

$$U_i = U \oplus \langle \mathbf{v}_i \rangle_{\mathbb{F}_q}$$

für ein $\mathbf{v}_i \in U'$, $\mathbf{v}_i \neq 0$. Insbesondere ist

$$\langle \mathbf{v}_i \rangle_{\mathbb{F}_q} \neq \langle \mathbf{v}_j \rangle_{\mathbb{F}_q}$$

für alle $0 \leq i < j \leq \eta$ und

$$V \subseteq \bigcup_{i=0}^{\eta} \langle \mathbf{v}_i \rangle_{\mathbb{F}_q} \subseteq U'.$$

Die Indexmenge J gibt an, welche der Vektoren $\mathbf{v}_i$, $0 \leq i \leq \eta$, in V liegen,

$$J \coloneqq \{0 \leq i \leq \eta \mid \mathbf{v}_i \in V\}.$$

Die Kardinalität der Indexmenge J ist $\frac{q^{\dim V}-1}{q-1}$. Ein Komplementärraum von V in U' werde mit V' bezeichnet, so dass

$$U \oplus V \oplus V' = F_q^m.$$

Wir wählen einen Vektor aus $\{\mathbf{v}_j \mid j \in J\} \subset V$ und bezeichnen diesen mit $\mathbf{v}^*$. Die Decodierung verläuft in zwei Stufen.

Der Majority-Logic-Schritt. Sei $\mathbf{v} \in V' \setminus \{0\} \dot{\cup} \{\mathbf{v}^*\}$ beliebig. Der Vektor $\mathbf{v}$ liegt in U' und kann daher in höchstens einem der Unterräume $U_0, U_1, \ldots, U_\eta$ liegen, denn

$$U_i \cap U_j \cap U' \setminus \{0\} = U \cap U' \setminus \{0\} = \emptyset$$

für alle $0 \leq i < j \leq \eta$.

Wir bestimmen eine η-elementige Indexmenge I,

$$I \subseteq \{0 \leq i \leq \eta \mid \mathbf{v} \notin U_i\},$$

und treffen die η-Mehrheitsentscheidung

$$e_0 \coloneqq \mu^0 \left(\mathtt{E} \circ \chi_{\mathbf{v}+U_i} \mid i \in I\right) \in F_{q_C}. \qquad [5.13]$$

Gemäß Proposition 4.3.1 ist die Fehlersumme von $\mathbf{v} + U$ gerade e_0,

$$\mathtt{E} \circ \chi_{\mathbf{v}+U} = e_0.$$

Sofern $q > 2$, seien außerdem $j \in J$, und $\alpha \in \mathbb{F}_q \setminus \{0, p-1\}$ beliebig. Wir halten fest,

$$0 \neq \mathbf{v} + \underbrace{\alpha \cdot \mathbf{v}_j}_{\in V}.$$

Mit dem gleichen Argument wie zuvor ist jeder Vektor $\mathbf{v} + \alpha \cdot \mathbf{v}_j$ in höchstens einem der Unterräume $U_0, U_1, \ldots, U_\eta$ enthalten. Wir treffen die η-Mehrheitsentscheidung,

$$e_{j,\alpha} \coloneqq \mu^0 \left(\mathtt{E} \circ \chi_{\mathbf{v}+\alpha\cdot\mathbf{v}_j+U_i} \;\middle|\; i \in I_{j,\alpha}\right) \in F_q, \qquad [5.14]$$

basierend auf einer Indexmenge $I_{j,\alpha}$ mit

$$I_{j,\alpha} \subseteq \{0 \leq i \leq \eta \mid \mathbf{v} + \alpha \cdot \mathbf{v}_j \notin U_i\}, \quad |I_{j,\alpha}| = \eta.$$

Erneut Proposition 4.3.1 anwendend, ist die Fehlersumme von $\mathbf{v} + \alpha \cdot \mathbf{v}_j + U$ gerade $e_{j,\alpha}$,

$$\mathtt{E} \circ \chi_{\mathbf{v}+\alpha\cdot\mathbf{v}_j+U} = e_{j,\alpha}.$$

So verfahren wir für alle $\mathbf{v} \in V' \setminus \{0\} \,\dot\cup\, \{\mathbf{v}^*\}$ und sofern $q > 2$, für alle $j \in J$ sowie für alle $\alpha \in \mathbb{F}_q \setminus \{0, p-1\}$, so dass wir zusammengenommen

$$\begin{aligned}\left|V'\right| \cdot (1 + (q-2) \cdot |J|) &= q^{m-D-\dim V} \cdot \left(1 + (q-2) \cdot \frac{q^{\dim V} - 1}{q-1}\right) \\ &= \mathcal{H}_\mu\left(D, \dim V\right)\end{aligned}$$

η-Mehrheitsentscheidungen treffen.

Der Addition-Subtraktion-Schritt. Für alle $\mathbf{v} \in V' \setminus \{0\} \,\dot\cup\, \{\mathbf{v}^*\}$, für alle $j \in J$ mit $\mathbf{v} - \mathbf{v}_j \notin U$, $\mathbf{v} \notin U_j$ leiten wir mit Hilfe der Gleichung

$$\mathbf{v} + U_j = (\mathbf{v} - \mathbf{v}_j + U) \;\dot\cup \bigdot_{\alpha \in \mathbb{F}_q \setminus \{p-1\}} \mathbf{v} + \alpha \cdot \mathbf{v}_j + U$$

die Fehlersumme zu $\mathbf{v} - \mathbf{v}_j + U$ her,

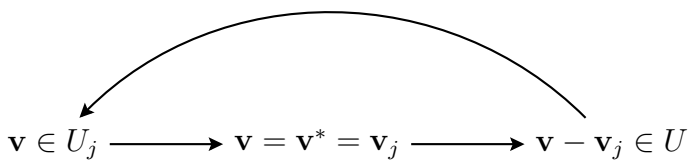
$$\begin{aligned}\mathtt{E} \circ \chi_{\mathbf{v}-\mathbf{v}_j+U} &= \mathtt{E} \circ \left(\chi_{\mathbf{v}+U_j} - \sum_{\alpha \in \mathbb{F}_q \setminus \{p-1\}} \chi_{\mathbf{v}+\alpha\cdot\mathbf{v}_j+U}\right) \\ &= \mathtt{E} \circ \chi_{\mathbf{v}+U_j} - e_0 - \sum_{\alpha \in \mathbb{F}_q \setminus \{0,p-1\}} e_{j,\alpha}. \qquad [5.15]\end{aligned}$$

(Sicherlich ließe sich die Fehlersumme von $\mathbf{v} - \mathbf{v}_j + U$ ebenfalls mittels einer Mehrheitsentscheidung berechnen (siehe verbesserte Decodierung in Abschnitt 5.3). Ist jedoch, wie eingangs bereits erwähnt, eine Mehrheitsentscheidung aufwändiger als $q-1$ Additionen zu realisieren, dann ist es vorteilhafter mit Hilfe von Gleichung [5.15] die Fehlersumme zu bestimmen.)

Wir wollen einen Blick auf die beiden Forderungen $\mathbf{v} - \mathbf{v}_j \notin U$ und $\mathbf{v} \notin U_j$ werfen. Eine von beiden kann ohne Einschränkung weggelassen werden, da beide äquivalent sind. Denn wegen

$$U_j \cap \left(V' \setminus \{0\} \,\dot\cup\, \{\mathbf{v}^*\}\right) = \langle \mathbf{v}_j \rangle_{\mathbb{F}_q} \cap \{\mathbf{v}^*\} = \begin{cases} \{\mathbf{v}^*\} & \mathbf{v}^* = \mathbf{v}_j, \\ \emptyset & \text{sonst} \end{cases}$$

gelten folgende Implikationen für alle $\mathbf{v} \in V' \setminus \{0\} \,\dot\cup\, \{\mathbf{v}^*\}$.



Für die Berechnung dieser Fehlersummen bedarf es insgesamt

$$\begin{aligned}\left(\left|V^{'}\setminus\{0\}\,\dot{\cup}\,\{\mathbf{v}^{*}\}\right|\cdot|J|-1\right)\cdot(q-1) &= q^{m-D-\dim V}\cdot\left(q^{\dim V}-1\right)-q+1\\ &= \mathcal{H}_{\pm}\left(D,\dim V\right)\end{aligned}$$

Additionen/Subtraktionen in $\mathbb{F}_{qc}$.

An dieser Stelle bleibt noch zu zeigen, dass wir auf diese Art und Weise die Fehlersummen zu allen affinen Räumen $\mathbf{u}^{'}+U$ mit $\mathbf{u}^{'}\in\mathbb{F}_q^m\setminus U$ bestimmt haben:

$$\begin{aligned}&\left\{\mathbf{v}+\alpha\cdot\mathbf{v}_j+U\;\middle|\;\alpha\in\mathbb{F}_q, j\in J, \mathbf{v}\in V^{'}, \mathbf{v}\neq 0\right\}\\ \cup&\left\{\mathbf{v}^{*}+\alpha\cdot\mathbf{v}_j+U\mid\alpha\in\mathbb{F}_q, j\in J\right\}\setminus\{U\}\\ &=\left\{\mathbf{u}^{'}+U\;\middle|\;\mathbf{u}^{'}\in\left(V^{'}\setminus\{0\}+V\right)\cup\left(V\setminus\{0\}\right)\right\}\\ &=\left\{\mathbf{u}^{'}+U\;\middle|\;\mathbf{u}^{'}\in\left(V+V^{'}\right)\setminus\{0\}\right\}\\ &=\left\{\mathbf{u}^{'}+U\;\middle|\;\mathbf{u}^{'}\in\mathbb{F}_q^m\setminus U\right\}.\end{aligned}$$

□

Bevor wir das vollständige Decodierverfahren vorstellen, möchten wir den in Lemma 5.4.1 präsentierten Decodierschritt anhand von zwei Beispielen erläutern. Beginnen werden wir mit einem Beispiel über $q=2$, das an die verwendeten Strukturen in vereinfachter Weise heranführt. Anschließend betrachten wir noch ein komplexeres Beispiel über $q=3$, bei dem alle Strukturen des Decodierschritts veranschaulicht werden.

Beispiel 5.4.2. Seien $q=2$, $m\coloneqq 3>1=:D$ und $\eta\coloneqq 2$. Wir halten fest, $\eta=2=\ell_{1,2,3,2}-1$. Sei

$$\mathbb{Z}_2^3\coloneqq\{\mathbf{w}_0,\mathbf{w}_1,\ldots,\mathbf{w}_7=0\}.$$

Um die Notation abzukürzen, fassen wir wie in Beispiel 5.2.3 jeden Vektor $(a_2,a_1,a_0)\in\mathbb{Z}_2^3$ als Binärzahl auf und identifizieren ihn eindeutig durch die Zahl $\sum_{i=0}^{2}a_i2^i$,

$$(a_2,a_1,a_0)\in\mathbb{Z}_2^3\quad\longleftrightarrow\quad\sum_{i=0}^{2}a_i2^i\in\mathbb{Z}_8.$$

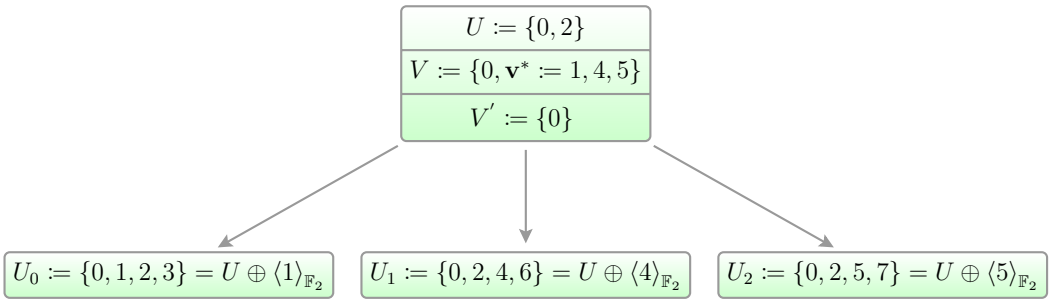
Die drei zweidimensionalen Untervektorräume

$$U_0\coloneqq\{0,1,2,3\},U_1\coloneqq\{0,2,4,6\},U_2\coloneqq\{0,2,5,7\}\leq\mathbb{F}_2^3$$

schneiden sich paarweise im Untervektorraum $U \coloneqq \{0, 2\}$. Der Untervektorraum $U^{'} \coloneqq \{0, 1, 4, 5\}$ ist ein Komplementärraum zu U und definiert die Vektoren $\mathbf{v}_0 \coloneqq 1$, $\mathbf{v}_1 \coloneqq 4$, $\mathbf{v}_2 \coloneqq 5$. Wir wählen $V \coloneqq U^{'}$, so dass

$$\{0\} \subset V \coloneqq \{0, 1, 4, 5\} = \bigcup_{0 \leq i \leq \eta} U^{'} \cap U_i$$

gewährleistet ist. Dann ist der Nullvektorraum der Komplementärraum von V in $U^{'}$. Wir setzen $\mathbf{v}^* \coloneqq 1 \in \{\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2\}$.



Angenommen, $\tau \leq 1$. Beispielhaft seien die Fehlersummen von $1 + U_1$ und $1 + U_2$ gerade 0 beziehungsweise 1. Die Menge $V^{'} \setminus \{0\} \dot{\cup} \{\mathbf{v}^*\}$ enthält nur den Vektor $\mathbf{v}^* = 1 \in U_0$.

Der Majority-Logic-Schritt. Wir treffen die 2-Mehrheitsentscheidung

$$e_0 \coloneqq \mu^0 \left(\mathtt{E} \circ \chi_{1+U_1}, \mathtt{E} \circ \chi_{1+U_2} \right) = \mu^0 \left(0, 1 \right) = 0, \qquad [5.16]$$

vgl. Gleichung [5.13] auf Seite 74. Gemäß Proposition 4.3.1 ist die Fehlersumme von $1 + U$ gerade e_0,

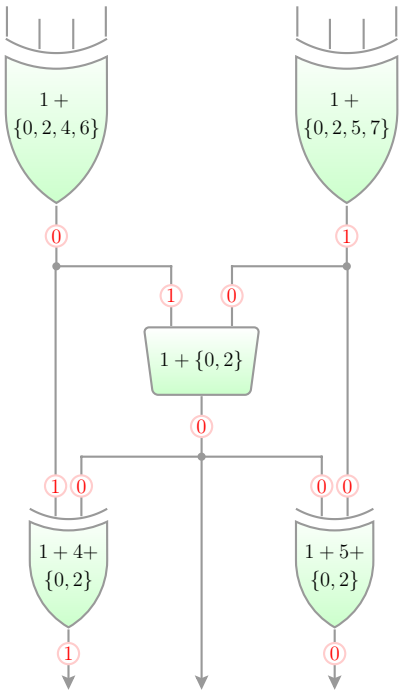
$$\mathtt{E} \circ \chi_{1+U} = 0.$$

Der Addition-Subtraktion-Schritt. Es ist

$$\mathtt{E} \circ \chi_{1+4+U} = \mathtt{E} \circ \chi_{\mathbf{v}^*+\mathbf{v}_1+U} = \mathtt{E} \circ \chi_{\mathbf{v}^*+U_1} + e_0 = 0 + 0 = 0,$$
$$\mathtt{E} \circ \chi_{1+5+U} = \mathtt{E} \circ \chi_{\mathbf{v}^*+\mathbf{v}_2+U} = \mathtt{E} \circ \chi_{\mathbf{v}^*+U_2} + e_0 = 1 + 0 = 1.$$

Damit haben wir die Fehlersummen zu $1 + U$, $5 + U$, $4 + U$ bestimmt.

Der Decodierschritt unter Kenntnis der Fehlersummen zu $1+U_1$ und $1+U_2$ ist in der Grafik rechts veranschaulicht. Decodiert wird erneut von oben nach unten. Wir übernehmen die Symbolik aus Beispiel 5.2.3 beziehungsweise Beispiel 5.3.5. Es werden entsprechend Lemma 5.4.1 ein Majoritätsgatter,  , und zwei XOR-Gatter,  , mit jeweils zwei Eingängen benutzt.



◁

Beispiel 5.4.3. Seien $q = 3$, $m \coloneqq 4 > 2 =: D$ und $\eta \coloneqq 2$. Wir halten fest, $\eta \coloneqq 2 < 3 = \ell_{2,3,4,3} - 1$. Sei

$$\mathbb{Z}_3^4 \coloneqq \{\mathbf{w}_0, \mathbf{w}_1, \ldots, \mathbf{w}_{80} = 0\}\,.$$

Um die Notation abzukürzen, fassen wir jeden Vektor $(a_3, a_2, a_1, a_0) \in \mathbb{Z}_3^4$ als Ternärzahl auf und identifizieren ihn eindeutig durch die Zahl $\sum_{i=0}^{3} a_i 3^i$,

$$(a_3, a_2, a_1, a_0) \in \mathbb{Z}_3^4 \quad \longleftrightarrow \quad \sum_{i=0}^{3} a_i 3^i \in \mathbb{Z}_{81}.$$

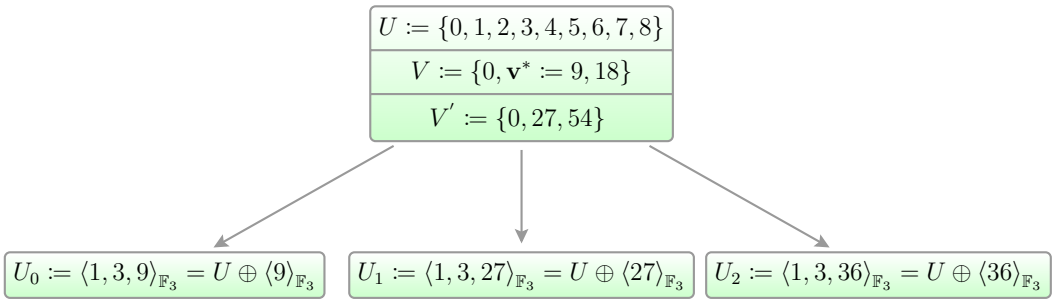
Die drei dreidimensionalen Untervektorräume

$$U_0 \coloneqq \langle 1, 3, 9\rangle_{\mathbb{F}_3}\,, U_1 \coloneqq \langle 1, 3, 27\rangle_{\mathbb{F}_3}\,, U_2 \coloneqq \langle 1, 3, 36\rangle_{\mathbb{F}_3} \leq \mathbb{F}_3^4$$

schneiden sich paarweise im Untervektorraum $U \coloneqq \langle 1, 3\rangle_{\mathbb{F}_3}$. Der Untervektorraum $U' \coloneqq \langle 9, 27\rangle_{\mathbb{F}_3}$ ist ein Komplementärraum zu U und definiert die Vektoren $\mathbf{v}_0 \coloneqq 9$, $\mathbf{v}_1 \coloneqq 27$, $\mathbf{v}_2 \coloneqq 36$. Wir wählen $V \coloneqq \langle 9\rangle_{\mathbb{F}_3} < U'$, so dass

$$\{0\} \subset V \coloneqq \langle 9\rangle_{\mathbb{F}_3} \leq \langle 9\rangle_{\mathbb{F}_3} \cup \langle 27\rangle_{\mathbb{F}_3} \cup \langle 36\rangle_{\mathbb{F}_3} = \bigcup_{0 \leq i \leq \eta} U' \cap U_i$$

gewährleistet ist. Dann ist $V' \coloneqq \langle 27\rangle_{\mathbb{F}_3}$ ein Komplementärraum von V in U' und der Vektor $\mathbf{v}^*$ eindeutig definiert, $\mathbf{v}^* \coloneqq 9$.



Angenommen, $\tau \leq 1$. Die Menge $V' \setminus \{0\} \dot{\cup} \{\mathbf{v}^*\}$ ist gegeben durch $\{9, 27, 54\}$.

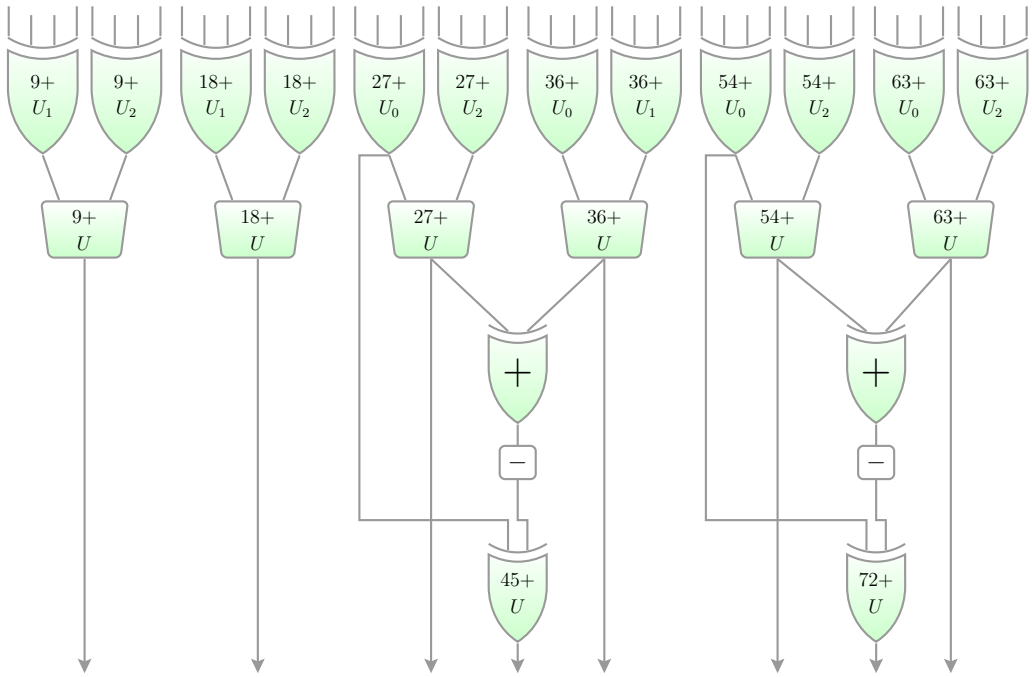
Der Majority-Logic-Schritt. Wir treffen die sechs folgenden 2-Mehrheitsentscheidungen und fassen die zurückgegebenen Mehrheitswerte als die Fehlersummen zu $9+U$, $18+U$, $27+U$, $36+U$, $54+U$ und $63+U$ auf,

$$\begin{aligned}
\mu^0\left(\mathtt{E} \circ \chi_{9+U_1}, \mathtt{E} \circ \chi_{9+U_2}\right) &=: \mathtt{E} \circ \chi_{9+U}, \\
\mu^0\left(\mathtt{E} \circ \chi_{9+9+U_1}, \mathtt{E} \circ \chi_{9+9+U_2}\right) &=: \mathtt{E} \circ \chi_{9+9+U}, \\
\mu^0\left(\mathtt{E} \circ \chi_{27+U_0}, \mathtt{E} \circ \chi_{27+U_2}\right) &=: \mathtt{E} \circ \chi_{27+U}, \\
\mu^0\left(\mathtt{E} \circ \chi_{27+9+U_0}, \mathtt{E} \circ \chi_{27+9+U_1}\right) &=: \mathtt{E} \circ \chi_{27+9+U}, \\
\mu^0\left(\mathtt{E} \circ \chi_{54+U_0}, \mathtt{E} \circ \chi_{54+U_2}\right) &=: \mathtt{E} \circ \chi_{54+U}, \\
\mu^0\left(\mathtt{E} \circ \chi_{54+9+U_0}, \mathtt{E} \circ \chi_{54+9+U_2}\right) &=: \mathtt{E} \circ \chi_{54+9+U}.
\end{aligned}$$

Der Addition-Subtraktion-Schritt. Es ist

$$\begin{aligned}
\mathtt{E} \circ \chi_{45+U} = \mathtt{E} \circ \chi_{27-\mathbf{v}_0+U} &= \mathtt{E} \circ \chi_{27+U_0} - \left(\mathtt{E} \circ \chi_{27+U} + \mathtt{E} \circ \chi_{27+9+U}\right), \\
\mathtt{E} \circ \chi_{72+U} = \mathtt{E} \circ \chi_{54-\mathbf{v}_0+U} &= \mathtt{E} \circ \chi_{54+U_0} - \left(\mathtt{E} \circ \chi_{54+U} + \mathtt{E} \circ \chi_{54+9+U}\right).
\end{aligned}$$

Der Decodierschritt ist in der Grafik unten veranschaulicht. Decodiert wird erneut von oben nach unten. Unter Kenntnis der Fehlersummen zu den dreidimensionalen affinen Räumen werden entsprechend Lemma 5.4.1 ein Majoritätsgatter,  , und vier Addierer über $\mathbb{F}_3$,  , mit jeweils zwei Eingängen benutzt.



◁

5.4.2 Die Hybriddecodierung im Ganzen

Nachdem wir nun einen einzelnen Decodierschritt der Hybriddecodierung vorgestellt haben, wollen wir diese nun vollumfänglich vorstellen.

Theorem 5.4.4. *Angenommen, es existieren* $D_{\mathcal{C}} \in \mathbb{N}$, $D_{\mathcal{C}} < m$, $\eta \in \mathbb{N}$, $\eta \geq 2$, $\varsigma \in \mathbb{N}$ *und* $D_1, \ldots, D_{\varsigma-1} \in \mathbb{N}$ *mit*

$$m > D_\varsigma \coloneqq D_{\mathcal{C}} > D_{\varsigma-1} > \ldots > D_1 > D_0 \coloneqq 0,$$

so dass

- *zu jedem* $A \in \mathcal{A}^*_{D_{\mathcal{C}},m,q}$ *die Fehlersumme* $\mathtt{E} \circ \chi_A$ *berechenbar ist und*
- $\ell_{D_t,D_{t+1},m,q} - 1 \geq \eta \geq 2\tau$ *für alle* $t = \varsigma - 1, \varsigma - 2, \ldots, 0$.

Wir definieren

$$T \coloneqq \{t \in \mathbb{N}_0, t \leq \varsigma - 1 \mid D_{t+1} = D_t + 1\} \qquad [5.17]$$

und

$$\dot{D} \coloneqq \left\lfloor \log_q((q-1)(\eta+1)+1) \right\rfloor \in \mathbb{N}.$$

Es gibt ein Majority-Logic-Verfahren, das anhand von maximal $\mathcal{H}_{\mathtt{E}}$ Fehlersummen zu affinen Räumen aus $\mathcal{A}^*_{D_C,m,q}$ in ς Majority-Logic-Schritten

$$D_\varsigma \to D_{\varsigma-1} \ldots \to D_1 \to D_0 \coloneqq 0$$

und $|T|$ Addition-Subtraktion-Schritten mit Hilfe von maximal $\mathcal{H}_\mu$ η-Mehrheitsentscheidungen sowie $\mathcal{H}_\pm$ Additionen/Subtraktionen über $\mathbb{F}_{q_C}$ alle n Fehlersymbole $\mathtt{E}_i$, $0 \leq i \leq n-1$, korrekt berechnet, wobei

$$\mathcal{H}_\mu \coloneqq \sum_{\substack{0 \leq t \leq \varsigma-1 \\ t \notin T}} (\eta+1)^t \cdot \left(q^{m-D_t}-1\right) + \sum_{\substack{0 \leq t \leq \varsigma-1 \\ t \in T}} (\eta+1)^t \cdot \mathcal{H}_\mu\left(D_t, \dot{D}\right), \quad [5.18]$$

$$\mathcal{H}_\pm \coloneqq \sum_{\substack{0 \leq t \leq \varsigma-1 \\ t \in T}} (\eta+1)^t \cdot \mathcal{H}_\pm\left(D_t, \dot{D}\right), \quad [5.19]$$

$$\mathcal{H}_{\mathtt{E}} \coloneqq (\eta+1)^\varsigma \cdot \left(q^{m-D_C}-1\right) = \mathcal{V}_{\mathtt{E}}. \quad [5.20]$$

Beweis. Der Beweis ist in drei Abschnitte gegliedert. Zunächst zeigen wir in Beweisteil A für alle $t \in T$, dass $1 \leq \dot{D} \leq m - D_t$ gilt. Anschließend konstruieren wir in Beweisteil B einen vollständigen Decodierbaum $\mathcal{G}$, der jenen aus dem Beweis von Theorem 5.3.1 erweitert, indem Knoten der Tiefe $t+1$ für alle $t \in T$ genauer spezifiziert werden. Abschließend widmen wir uns in Beweisteil C der Decodierung, die sich in den Stufen $D_{t+1} \to D_t$ für alle $t \in T \cup \{\varsigma-1\}$ vom Prozedere in Theorem 5.3.1 unterscheidet.

Beweisteil A. Sei $t \in T$ beliebig. Aus Proposition 5.1.4 und der Voraussetzung leiten wir ab, dass

$$q < (q-1)2+1 \leq (q-1)(\eta+1)+1 \leq (q-1) \cdot \ell_{D_t,D_{t+1},m,q} + 1 = q^{m-D_t}.$$

Also ist

$$1 \leq \dot{D} = \left\lfloor \log_q((q-1)(\eta+1)+1) \right\rfloor \leq m - D_t.$$

Beweisteil B. Konstruiere einen vollständigen Decodierbaum $\mathcal{G}$, der ähnlich ist zu jenem aus dem Beweis von Theorem 5.3.1, diesen aber weiter spezifiziert,

- Die Höhe des Baumes ist ς (die Anzahl der Majority-Logic-Stufen); der Wurzelknoten hat Tiefe 0.
- Jeder Knoten K außer den Blättern hat $\eta + 1$ Kindknoten.
- Jeder Knoten K der Tiefe $t \in \mathbb{N}_0$, $0 \leq t \leq \varsigma$, $t \notin T$, repräsentiert ein geordnetes Tripel aus

$$\left\{(U, V, \{0\}) \;\middle|\; U, V \leq \mathbb{F}_q^m, U \oplus V = \mathbb{F}_q^m, \dim(U) = D_t\right\}.$$

  Jeder Knoten K der Tiefe $t \in T$ repräsentiert ein geordnetes Tripel aus

$$\left\{\left(U, V, V^{'}\right) \;\middle|\; \begin{array}{c} U, V, V^{'} \leq \mathbb{F}_q^m, U \oplus V \oplus V^{'} = \mathbb{F}_q^m, \\ \dim U = D_t, \dim V = \dot{D} \end{array}\right\}.$$

- Für zwei beliebige Geschwisterknoten, korrespondierend zu $(U_i, V_i, V_i^{'})$ und $(U_j, V_j, V_j^{'})$, eines Elternknotens, der $(U, V, V^{'})$ repräsentiert, gilt, die Unterräume U_i und U_j schneiden sich in U.
- Für jeden Knoten K korrespondierend zu $(U, V, V^{'})$ der Tiefe $t \in T$ repräsentieren die $\eta + 1$ Kindknoten von K die Tripel

$$\left(U \oplus \langle \mathbf{v}_i \rangle_{\mathbb{F}_q}, V_i, V_i^{'}\right),$$

  $0 \leq i \leq \eta$, wobei erstens $\mathbf{v}_i \in V \oplus V^{'}$, $\mathbf{v}_i \neq 0$ für alle $0 \leq i \leq \eta$, zweitens

$$\langle \mathbf{v}_i \rangle_{\mathbb{F}_q} \neq \langle \mathbf{v}_j \rangle_{\mathbb{F}_q}$$

  für alle $0 \leq i < j \leq \eta$ und drittens

$$V \subseteq \bigcup_{i=0}^{\eta} \langle \mathbf{v}_i \rangle_{\mathbb{F}_q} \subseteq V \oplus V^{'} \qquad [5.21]$$

  gewährleistet sein muss.

Solch ein Graph existiert, da laut Voraussetzung für alle $t = \varsigma - 1, \varsigma - 2, \ldots, 0$

$$\eta + 1 \leq \ell_{D_t, D_{t+1}, m, q}$$

sowie für alle $t \in T$

$$\dot{D} \leq m - D_t,$$

wie wir in Beweisteil A gesehen haben. Außerdem lässt sich ein Vektorraum V der Dimension $\dot{D}$ als Vereinigung von $(q^{\dot{D}}-1)/(q-1)$ eindimensionalen Räumen darstellen, wobei nach Definition von $\dot{D}$

$$(q^{\dot{D}}-1)/(q-1) \leq \eta + 1$$

gilt. Indem wir den Decodierbaum nach und nach aufbauen, also bei der Wurzel beginnend die Kindknoten sukzessive definieren, lässt sich Anforderung [5.21] tatsächlich stets erfüllen.

Beweisteil C Widmen wir uns nun der Decodierung. Per Induktion über $t \in \mathbb{N}_0$, $t \leq \varsigma$, zeigen wir, dass wir die Fehlersummen zu affinen Räume $\mathbf{u}^{'} + U$ für alle $\mathbf{u}^{'} \in \mathbb{F}_q^m \backslash U$, für alle von Knoten der Tiefe t repräsentierten Unterräume U bestimmen können. Bei $t = \varsigma$ beginnend und t in jedem Schritt um eins dekrementierend, werden bei $t = 0$ schließlich die Fehlersummen von $\mathbf{w} + \{0\}$, $\mathbf{w} \in \mathbb{F}_q^m \backslash \{0\}$ berechnet. Diese entsprechen gerade den Fehlersymbolen $\mathtt{E}_i$, $0 \leq i \leq n-1$.

Für $t = \varsigma$ (Induktionsanfang) sei K ein beliebiger der $(\eta+1)^{\varsigma}$ Blattknoten. K repräsentiere $(U, V, V^{'})$. Laut Voraussetzung sind die Fehlersummen der affinen Räume $\mathbf{u}^{'} + U$ für alle $\mathbf{u}^{'} \in V \oplus V^{'}$, $\mathbf{u}^{'} \neq 0$ berechenbar.

Sei nun t, $0 \leq t \leq \varsigma - 1$, beliebig und sei K ein beliebiger der $(\eta+1)^t$ Knoten der Tiefe t. K repräsentiere $(U, V, V^{'})$. Nach Induktionsannahme sind die Fehlersummen zu $\mathbf{u}^{'} + U_i$ für alle $\mathbf{u}^{'} \in \mathbb{F}_q^m$, für alle von Knoten der Tiefe $t+1$ repräsentierten Unterräume U_i bekannt.

Ist $D_t + 1 < D_{t+1}$, verfahren wir wie im Beweis von Theorem 5.3.1 gesehen: Wir bestimmen die Fehlersummen der $q^{m-D_t} - 1$ affinen Räume $\mathbf{u}^{'} + U$, $\mathbf{u}^{'} \in U^{'} \backslash \{0\}$, mit Hilfe von insgesamt $q^{m-D_t} - 1$ η-Mehrheitsentscheidungen (vgl. Beweis von Theorem 5.3.1) in einem Majority-Logic-Schritt.

Ist $D_t + 1 = D_{t+1}$, decodieren wir, wie wir es im Beweis von Lemma 5.4.1 gesehen haben. Die Fehlersummen aller affinen Räume $\mathbf{u}^{'} + U$, $\mathbf{u}^{'} \in U^{'} \backslash \{0\}$ werden in einem Majority-Logic-Schritt und einem Addition-Subtraktion-Schritt mittels $\mathcal{H}_{\mu}\left(D_t, \dot{D}\right)$ η-Mehrheitsentscheidungen, $\mathcal{H}_{\pm}\left(D_t, \dot{D}\right)$ Additionen über $\mathbb{F}_{qc}$ berechnet. $\square$

Bemerkung 5.4.5. Die (sequenzielle) Laufzeit bestimmt sich durch $\mathcal{H}_\mu$, $\mathcal{H}_\pm$ und $\mathcal{H}_{\mathtt{E}}$. Die Anzahl der Majority-Logic-Stufen ς sowie der Addition-Subtraktion-Schritte, gegeben durch $|T_I|$, geben Aufschluss über die parallele Laufzeit.

Bemerkung 5.4.6. Ist T nicht leer, so müssen im Vergleich zur klassischen Decodierung (Proposition 5.2.1) und zur verbesserten Decodierung (Theorem 5.3.1) bei der Hybriddecodierung (Theorem 5.4.4) weniger Mehrheitsentscheidungen getroffen werden, denn

$$(q-2)\frac{q^{\dot{D}}-1}{q-1} < q^{\dot{D}}-1$$

und damit

$$\mathcal{H}_\mu(D_t,\dot{D}) \coloneqq q^{m-D_t-\dot{D}} \cdot \underbrace{\left(1+(q-2)\frac{q^{\dot{D}}-1}{q-1}\right)}_{<q^{\dot{D}}} \leq q^{m-D_t}-1$$

für alle $t \in T$. Genauer gesagt, können wir im Schritt $D_t+1 \to D_t$ gegenüber Theorem 5.3.1

$$(\eta+1)^t \cdot \left(-1+q^{m-D_t-\dot{D}} \cdot \left(q^{\dot{D}}-1-(q-2)\frac{q^{\dot{D}}-1}{q-1}\right)\right)$$

Mehrheitsentscheidungen einsparen. Im Gegenzug sind wir in diesem Schritt auf zusätzliche Additionen/Subtraktionen über $\mathbb{F}_{q_\mathcal{C}}$ angewiesen. Die Anzahl der zusätzlich benötigten Additionen/Subtraktionen ist bestenfalls gleich der Anzahl der eingesparten Mehrheitsentscheidungen (bei $q=2$) und ansonsten größer, denn

$$\begin{aligned}
&\mathcal{H}_\pm(D_t,\dot{D}) + \mathcal{H}_\mu(D_t,\dot{D}) - \left(q^{m-D_t}-1\right) \\
&= (q-2)\left(\underbrace{q^{m-D_t-\dot{D}} \cdot \frac{q^{\dot{D}}-1}{q-1}}_{>1} - 1\right) \\
&\geq 0.
\end{aligned}$$

Das im Beweis von Theorem 5.4.4 beschriebene Verfahren ist in Algorithmus 5.4.1 dargestellt.

Algorithmus 5.4.1: Hybriddecodierung.

Vorbedingung: Voraussetzungen aus Theorem 5.4.4,

Graph $\mathcal{G}$ aus dem Beweis von Theorem 5.4.4.

Zusicherung: $\mathtt{E}$ wird zurückgegeben.

$\dot{D} = \left\lfloor \log_q \left((q-1)(\eta+1)+1 \right) \right\rfloor$

foreach Knoten der Tiefe ς korrespondierend zu (U, V, V') **do**

  foreach $\mathbf{v} \in V \oplus V'$, $\mathbf{v} \neq 0$ **do**

    $e_{\mathbf{v}+U} = \mathtt{E} \circ \chi_{\mathbf{v}+U}$

for $t = \varsigma - 1$ **to** 0 **do**

  foreach Knoten der Tiefe t korrespondierend zu $\left(U, V, V'\right)$ mit $\eta+1$ Kindknoten korrespondierend zu $\left(\left(U_0, V_0, V_0'\right), \ldots, \left(U_\eta, V_\eta, V_\eta'\right)\right)$ **do**

    if $D_{t+1} \neq D_t + 1$ **then**

      foreach $\mathbf{v} \in V$, $\mathbf{v} \neq 0$ **do**

        wähle $I \subseteq \{0 \leq i \leq \eta \mid \mathbf{v} \notin U_i\}$ mit $|I| = \eta$

        $e_{\mathbf{v}+U} = \mu^0 \left(e_{\mathbf{v}+U_i} \mid i \in I \right)$

    else

      $J = \{0 \leq j \leq \eta \mid U_j \cap V \neq \{0\}\}$

      foreach $j \in J$ **do**

        wähle $\mathbf{v}_j \in V \cap U_j \backslash \{0\}$

      wähle $\mathbf{v}^* \in \{\mathbf{v}_j \mid j \in J\}$

      foreach $\mathbf{v} \in V' \backslash \{0\} \dot{\cup} \{\mathbf{v}^*\}$ **do**

        wähle $I \subseteq \{0 \leq i \leq \eta \mid \mathbf{v} \notin U_i\}$ mit $|I| = \eta$

        $e_{\mathbf{v}+U} = \mu^0 \left(e_{\mathbf{v}+U_i} \mid i \in I \right)$

        foreach $j \in J$ **do**

          foreach $\alpha \in \mathbb{F}_q \backslash \{0, p-1\}$ **do**

            wähle $I_{j,\alpha} = \{0 \leq i \leq \eta \mid \mathbf{v} + \alpha \cdot \mathbf{v}_j \notin U_i\}$ mit $|I_{j,\alpha}| = \eta$

            $e_{\mathbf{v}+\alpha\cdot\mathbf{v}_j+U} = \mu^0 \left(e_{\mathbf{v}+\alpha\cdot\mathbf{v}_j+U_i} \;\middle|\; i \in I_{j,\alpha} \right)$

          if $(\mathbf{v} \neq \mathbf{v}^*) \,||\, (\mathbf{v}^* \neq \mathbf{v}_j)$ **then**

            $e = e_{\mathbf{v}+U}$

            foreach $\alpha \in \mathbb{F}_q \backslash \{0, p-1\}$ **do**

              $e = e + e_{\mathbf{v}+\alpha\cdot\mathbf{v}_j+U}$

            $e_{\mathbf{v}-\mathbf{v}_j+U} = e_{\mathbf{v}+U_j} - e$

Ausgabe: $\left(e_{\{\mathbf{w}_0\}}, e_{\{\mathbf{w}_1\}}, \ldots, e_{\{\mathbf{w}_{n-1}\}}\right)$.

5.4.3 Optimierung der Hybriddecodierung

Die Hybriddecodierung kann häufig noch effizienter gestaltet werden. Wir machen uns noch einmal bewusst, dass die Fehlersummen zu einigen affinen Räumen der Form $\mathbf{v}-\mathbf{v}_j+U$ nicht über Mehrheitsentscheidungen ermittelt werden (siehe Zeile 27 in Algorithmus 5.4.1). Dies wirft die Frage auf, ob wir die Fehlersummen zu $\mathbf{v}-\mathbf{v}_j+U_i$, $0 \leq i \leq \eta$ überhaupt bestimmen müssen? Die Antwort darauf geben wir in Proposition 5.4.7. Erwähnenswert ist, dass wir bereits vor Beginn des eigentlichen Decodierens, also nicht während der Laufzeit, feststellen können, welche Fehlersummen wir nicht benötigen. Dies hängt einzig vom Decodierbaum ab.

Proposition 5.4.7. *Wir übernehmen die Notationen, Voraussetzungen und den Decodierbaum aus Theorem 5.4.4.*

Sei $t \in T$ beliebig und sei K ein beliebiger der $(\eta+1)^t$ Knoten der Tiefe t. K repräsentiere $(U, V, V^{'})$. Die $\eta+1$ Kindknoten von K repräsentieren die Tripel $(U_i, V_i, V_i^{'})$, $0 \leq i \leq \eta$. Es existieren $\mathbf{v}_j \in \left(V \oplus V^{'}\right)$, so dass $U \oplus \langle \mathbf{v}_j \rangle_{\mathbb{F}_q} = U_j$ für alle $0 \leq j \leq \eta$.

Wir definieren

$$J \coloneqq \{0 \leq j \leq \eta \mid \mathbf{v}_j \in V\}$$

und

$$I \coloneqq \left\{0 \leq i \leq \eta \;\middle|\; \mathbf{v}_i \in V \cup V^{'}\right\} \supseteq J.$$

Seien $j^{'} \in J$ und $\mathbf{v}^{'} \in V^{'} \setminus \{0\} \dot{\cup} \{\mathbf{v}^*\}$ beliebig. Angenommen, $i \in I \backslash J$ oder

$$\langle \mathbf{v}_i \rangle_{\mathbb{F}_q} \subseteq \left\{\mathbf{v}_{j'} - \mathbf{v}_j \;\middle|\; j \in J\right\}. \qquad [5.22]$$

Dann benötigen wir die Fehlersumme zum affinen Raum $\mathbf{v}^{'} - \mathbf{v}_{j'} + U_i$ nicht, um Fehlersummen zu affinen Räumen $\mathbf{u}^{'} + U$, $\mathbf{u}^{'} \in \left(V \oplus V^{'}\right)$ zu bestimmen.

Beweis. Im Beweis von Theorem 5.4.4 haben wir die Fehlersummen zu den affinen Räumen $\mathbf{v} - \mathbf{v}_j + U$ mit $\mathbf{v} \neq \mathbf{v}_j$, $\mathbf{v} \in V^{'} \dot{\cup} \{\mathbf{v}^*\}$, $j \in J$ nicht mittels Mehrheitsentscheidung bestimmt.

Sei $i \in I$ beliebig. Die Fehlersumme zum affinen Raum $\mathbf{v}^{'} - \mathbf{v}_{j^{'}} + U_i$ wird nicht benötigt, falls $\mathbf{v}^{'} - \mathbf{v}_{j^{'}} + U_i$ nicht in der Menge S liegt,

$$S \coloneqq \bigcup_{\substack{\mathbf{v} \in V^{'} \dot{\cup} \{\mathbf{v}^*\} \\ \mathbf{v} \neq 0}} \bigcup_{j \in J} \{\mathbf{v} + \alpha \cdot \mathbf{v}_j + U_i \mid \alpha \in \mathbb{F}_q, \alpha \neq -1\}.$$

Seien $\mathbf{v} \in V^{'} \setminus \{0\} \dot{\cup} \{\mathbf{v}^*\}, \mathbf{v} \neq 0$, $j \in J$ und $\alpha \in \mathbb{F}_q$, $\alpha \neq -1$ beliebig. Nun ist

$$\mathbf{v}^{'} - \mathbf{v}_{j^{'}} + U_i \neq \mathbf{v} + \alpha \cdot \mathbf{v}_j + U_i \qquad [5.23]$$

genau dann, wenn

$$\mathbf{v}^{'} - \mathbf{v} - \mathbf{v}_{j^{'}} - \alpha \cdot \mathbf{v}_j \notin U_i \coloneqq \langle \mathbf{v}_i \rangle_{\mathbb{F}_q} \oplus U$$

genau dann, wenn

$$\left(\mathbf{v}^{'} - \mathbf{v}\right) - \left(\mathbf{v}_{j^{'}} + \alpha \cdot \mathbf{v}_j\right) \notin \langle \mathbf{v}_i \rangle_{\mathbb{F}_q},$$

da $\mathbf{v}^{'} - \mathbf{v} \in V^{'} \setminus \{0\} \dot{\cup} \{\mathbf{v}^*\}$ und $\mathbf{v}_{j^{'}} + \alpha \cdot \mathbf{v}_j \in V$.

Wir unterscheiden vier sich gegenseitig ausschließende Fälle,

1. $i \in J$, $\mathbf{v}^{'} \neq \mathbf{v}$,
2. $i \in J$, $\mathbf{v}^{'} = \mathbf{v}$, $\left(\mathbf{v}_{j^{'}} + \alpha \cdot \mathbf{v}_j\right) \notin \langle \mathbf{v}_i \rangle_{\mathbb{F}_q}$,
3. $i \in I \setminus J$, $\mathbf{v}_{j^{'}} + \alpha \cdot \mathbf{v}_j \neq 0$,
4. $i \in I \setminus J$, $\mathbf{v}_{j^{'}} + \alpha \cdot \mathbf{v}_j = 0$, $\mathbf{v}^{'} - \mathbf{v} \notin \langle \mathbf{v}_i \rangle_{\mathbb{F}_q}$.

In allen vier Fällen gilt Ungleichung [5.23]. Folglich liegt $\mathbf{v}^{'} - \mathbf{v}_{j^{'}} + U_i$ nicht in der Menge S , falls für alle $\mathbf{v} \in V^{'} \setminus \{0\} \dot{\cup} \{\mathbf{v}^*\}, \mathbf{v} \neq 0$, für alle $j \in J$ und für alle $\alpha \in \mathbb{F}_q$, $\alpha \neq -1$ eine dieser vier Bedingungen gilt.

Fall 2 näher betrachtend ist $\left(\mathbf{v}_{j^{'}} + \alpha \cdot \mathbf{v}_j\right) \notin \langle \mathbf{v}_i \rangle_{\mathbb{F}_q}$ für alle $j \in J$ und für alle $\alpha \in \mathbb{F}_q$, $\alpha \neq -1$ genau dann, wenn

$$\left(\mathbf{v}_{j^{'}} + V \setminus \{-\mathbf{v}_j \mid j \in J\}\right) \cap U_i = \emptyset,$$

was äquivalent ist zu Aussage 5.22. Für alle $\alpha \in \mathbb{F}_q$, $\alpha \neq -1$ tritt der Fall 2 nicht ein.

Zusammenfassend halten wir fest, die Fehlersumme zu $\mathbf{v}^{'} - \mathbf{v}_{j^{'}} + U_i$ wird nicht benötigt, falls $i \in I \setminus J$ oder Aussage 5.22 gilt. □

5.4.4 Vollständiges Beispiel zur Hybriddecodierung

Wie bereits angekündigt, demonstrieren wir die Hybriddecodierung an einem vollständigen Beispiel. Wir greifen zudem die Ergebnisse aus Proposition 5.4.7 auf.

Beispiel 5.4.8. Erneut benutzen wir dieselben Notationen, betrachten denselben Code und wählen dieselbe Abstufung wie in Beispiel 5.2.3,

$$\begin{aligned}\mathcal{C}^{\perp} &\coloneqq \left\langle \chi_{\mathbf{v}+U} \in \mathbb{F}_2^7 \mid U \leq \mathbb{F}_2^3, \dim U = 2, \mathbf{v} \in \mathbb{F}_2^3 \backslash U \right\rangle_{\mathbb{F}_2} \leq F_2^7, \\ \mathcal{C} &= \mathcal{C}^{\perp} \oplus \langle 1,1,1,1,1,1,1 \rangle_{\mathbb{F}_2} \leq F_2^7,\end{aligned}$$

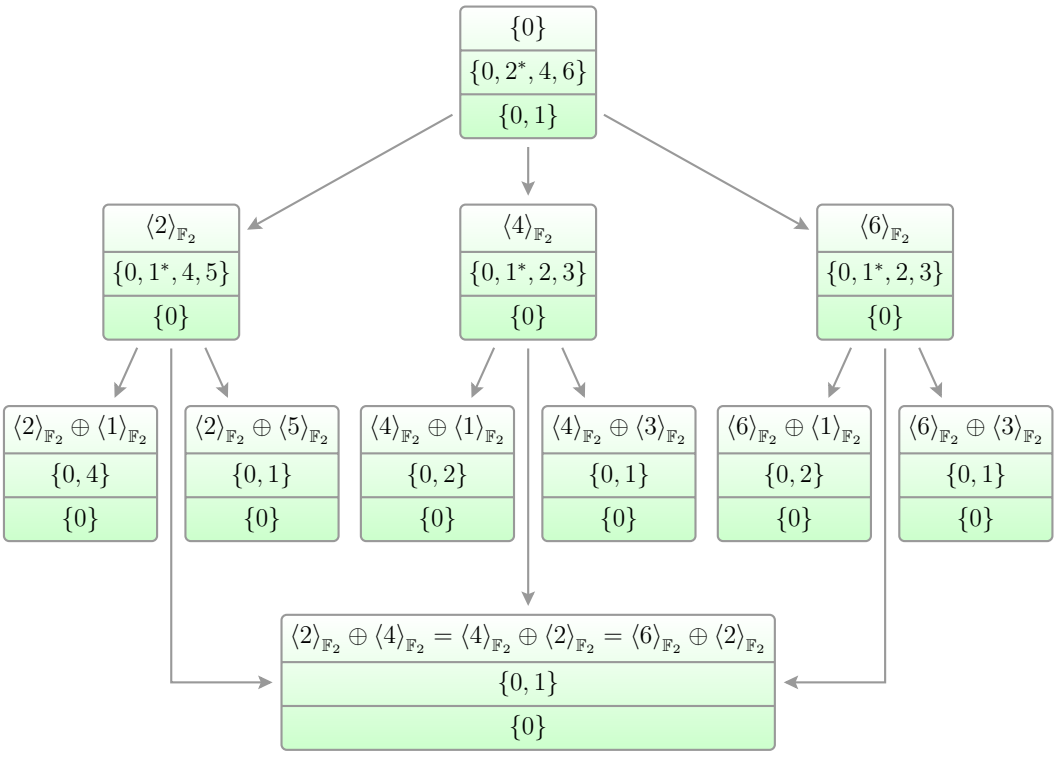
sowie $\eta \coloneqq 2$, $\varsigma \coloneqq 2$ und $3 > D_2 \coloneqq 2 > D_1 \coloneqq 1 > D_0 \coloneqq 0$.

Dann ist

$$T \coloneqq \{t \in \mathbb{N}_0, t \leq \varsigma - 1 \mid D_{t+1} = D_t + 1\} = \{0,1\}$$

und

$$\dot{D} \coloneqq \left\lfloor \log_q((q-1)(\eta+1)+1) \right\rfloor = 2.$$



Konstruieren wir einen Decodierbaum wie in Theorem 5.4.4 skizziert. Jeder Knoten repräsentiert ein geordnetes Tripel von Unterräumen, deren innere direkte Summe gerade $\mathbb{F}_2^3$ ist.

Die mit Sternchen markierten Zahlen repräsentieren jene Vektoren, die wir beim Beschreiben des Decodierverfahrens in Lemma 5.4.1 mit $\mathbf{v}^*$ bezeichnet haben. Sie bedingen nicht den Decodierbaum, jedoch das Decodierschema.

Von diesem Decodierbaum ausgehend, wollen wir Proposition 5.4.7 anwenden. Dazu prüfen wir für beide Schritte, sowohl für $D_1 = 1 \to D_0 = 0$ als auch für $D_2 = 2 \to D_1 = 1$, wann die Aussage 5.22 gilt. Für den Schritt $D_1 = 1 \to D_0 = 0$ halten wir fest,

$$\langle 2 \rangle_{\mathbb{F}_2} \subseteq \left\{ \mathbf{v}_{j'} + 2, \mathbf{v}_{j'} + 4, \mathbf{v}_{j'} + 6 \right\}$$

für $\mathbf{v}_{j'} \in \{4, 6\}$,

$$\langle 4 \rangle_{\mathbb{F}_2} \subseteq \left\{ \mathbf{v}_{j'} + 2, \mathbf{v}_{j'} + 4, \mathbf{v}_{j'} + 6 \right\}$$

für $\mathbf{v}_{j'} \in \{2, 6\}$ und

$$\langle 6 \rangle_{\mathbb{F}_2} \subseteq \left\{ \mathbf{v}_{j'} + 2, \mathbf{v}_{j'} + 4, \mathbf{v}_{j'} + 6 \right\}$$

für $\mathbf{v}_{j'} \in \{2, 4\}$. Folglich benötigen wir sowohl für $\mathbf{v}^{'} = 1$ als auch für $\mathbf{v}^{'} = \mathbf{v}^*$ die Fehlersummen zu den affinen Räumen

$$\begin{aligned} &\mathbf{v}^{'} + 4 + \langle 2 \rangle_{\mathbb{F}_2} \left(= \mathbf{v}^{'} + 6 + \langle 2 \rangle_{\mathbb{F}_2} \right), \\ &\mathbf{v}^{'} + 2 + \langle 4 \rangle_{\mathbb{F}_2} \left(= \mathbf{v}^{'} + 6 + \langle 4 \rangle_{\mathbb{F}_2} \right), \\ &\mathbf{v}^{'} + 2 + \langle 6 \rangle_{\mathbb{F}_2} \left(= \mathbf{v}^{'} + 4 + \langle 6 \rangle_{\mathbb{F}_2} \right), \end{aligned}$$

nicht. Durch die Wahl des Vektors $\mathbf{v}^* \in \{2, 4, 6\}$ beeinflussen wir, für welchen einen affinen Raum, $4 + \langle 2 \rangle_{\mathbb{F}_2}$, $2 + \langle 4 \rangle_{\mathbb{F}_2}$ oder $2 + \langle 6 \rangle_{\mathbb{F}_2}$, wir keine Fehlersumme bestimmen müssen. Indem wir $\mathbf{v}^* = 2$ wählen, entscheiden wir uns gegen $4 + \langle 2 \rangle_{\mathbb{F}_2}$. Alles zusammengenommen müssen im Schritt $D_1 = 1 \to D_0 = 0$ die vier affinen Räume

$$5 + \langle 2 \rangle_{\mathbb{F}_2}, \quad 3 + \langle 4 \rangle_{\mathbb{F}_2}, \quad 3 + \langle 6 \rangle_{\mathbb{F}_2}, \quad 4 + \langle 2 \rangle_{\mathbb{F}_2}$$

nicht berücksichtigt werden.

Wir gehen zum Schritt $D_2 = 2 \to D_1 = 1$ über und stellen fest, indem wir jeweils $\mathbf{v}^* \neq 2, 4$ wählen, wird der affine Raum $1 + \langle 2, 4 \rangle_{F_2}$ dreimal zur Decodierung herangezogen. Beispielweise werden die affinen Räume

$$4 + \langle 1, 2 \rangle_{\mathbb{F}_2}, \quad 2 + \langle 1, 4 \rangle_{\mathbb{F}_2}, \quad 2 + \langle 1, 6 \rangle_{\mathbb{F}_2}$$

nicht weiter benötigt, wenn wir in diesem Schritt stets $\mathbf{v}^* = 1$ wählen. An dieser Stelle wird deutlich, dass die Wahl der Vektoren $\mathbf{v}^*$ die Effizienz der Decodierung beeinflusst. Diese kann wie folgt grafisch veranschaulicht werden.

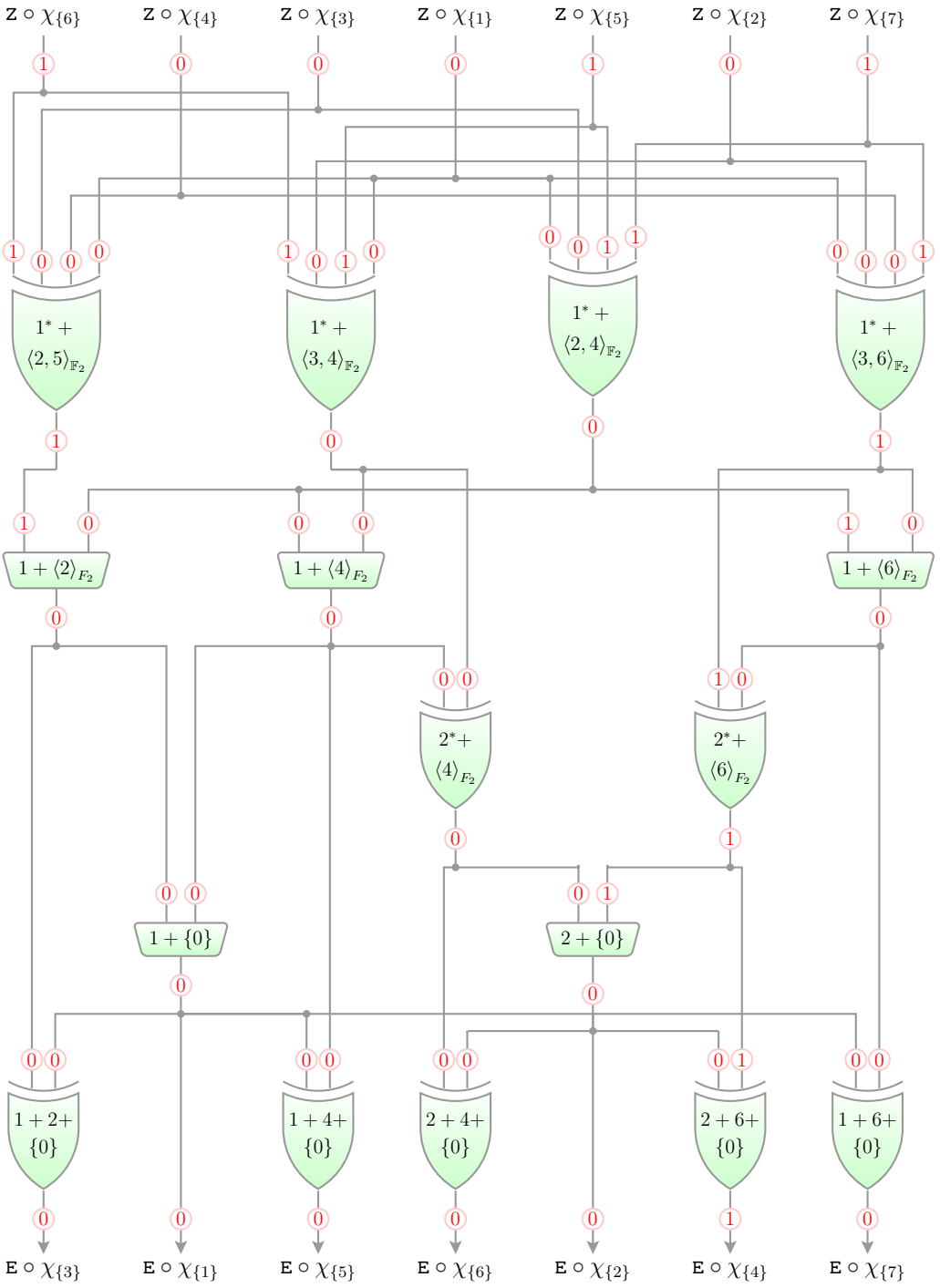
Die Decodierung verläuft erneut von oben nach unten. Neben vier XOR-Gattern mit vier Eingängen, mit Hilfe derer zu Beginn der Decodierung die Checksummen beziehungsweise die Fehlersummen der zweidimensionalen affinen Räume berechnet werden, besteht der Decoder aus sieben XOR-Gattern mit zwei Eingängen,  , und entsprechend Term 5.18 fünf Majoritätsgattern mit zwei Eingängen,  .

Vier XOR-Gatter mit je vier Eingängen und sieben XOR-Gatter mit je zwei Eingängen entspricht insgesamt $19 = 7 \cdot 1 + 4 \cdot 3$ Additionen. Dies ist deutlich weniger als die in Term 5.19 angegebenen

$$38 = 5 + 6 + 27$$

Additionen. Die geringere Zahl an Additionen lässt sich vollständig, wie oben gesehen, mit Proposition 5.4.7 begründen.

Der Decoder liefert wie zuvor $\chi_{\{4\}}$ als potentielles Fehlerwort und $\mathtt{z} - \mathtt{e} = \chi_{\{4,5,6,7\}}$ als potentielles Codewort. In gleicher Weise argumentiert wie zuvor in Theorem 5.3.1 ist das potentielle Codewort tatsächlich das übertragene Codewort, sofern sichergestellt ist, dass nicht mehr als ein Fehler aufgetreten ist.


Z ∘ χ{6}
Z ∘ χ{4}
Z ∘ χ{3}
Z ∘ χ{1}
Z ∘ χ{5}
Z ∘ χ{2}
Z ∘ χ{7}
1* + ⟨2, 5⟩F2
1* + ⟨3, 4⟩F2
1* + ⟨2, 4⟩F2
1* + ⟨3, 6⟩F2
1 + ⟨2⟩F2
1 + ⟨4⟩F2
1 + ⟨6⟩F2
2*+ ⟨4⟩F2
2*+ ⟨6⟩F2
1 + {0}
2 + {0}
1 + 2+ {0}
1 + 4+ {0}
2 + 4+ {0}
2 + 6+ {0}
1 + 6+ {0}
E ∘ χ{3}
E ∘ χ{1}
E ∘ χ{5}
E ∘ χ{6}
E ∘ χ{2}
E ∘ χ{4}
E ∘ χ{7}

◁

Kapitel 6

Drei verschiedene Abstufungen und ihr Einfluss auf die Decodierverfahren

Zu jedem der drei in Kapitel 5 präsentierten Decodierverfahren haben wir angegeben, wie viele Operationen allgemein benötigt werden. Zudem haben wir die drei Verfahren untereinander hinsichtlich ihres Aufwands verglichen (siehe Bemerkung 5.3.4 und Bemerkung 5.4.6) und gesehen, dass die wenigstens Mehrheitsentscheidungen bei der Hybriddecodierung getroffen werden. Bislang haben wir konkrete Abstufungen bei der Aufwandsabschätzung noch außen vor gelassen.

Drei Abstufungen, jene nach Reed [30] und Chen [7,8] sowie eine von uns entworfene, stellen wir im ersten Abschnitt vor. Bei letzterer haben wir uns durch Chens Abstufung inspirieren lassen und diese, vereinfacht gesagt, invertiert. Im zweiten Abschnitt zeigen wir, dass alle drei hinsichtlich der Fehlerkorrekturfähigkeit optimal sind: Sie garantieren, bei höchstens τ_{MLG} Fehlern das ursprüngliche Codewort mit obigen Verfahren korrekt rekonstruieren zu können, sofern

$$\eta = 2\tau_{\mathrm{MLG}}$$

gewählt wird. Das Kapitel abschließend widmen wir uns dem konkreten Decodieraufwand in Abhängigkeit der einzelnen Abstufungen.

6.1 Definition der drei Abstufungen

In diesem Abschnitt stellen wir die Abstufungen nach Reed und Chen und die von uns entwickelte invertierte Abstufung vor.

6.1.1 Reeds Abstufung

Die sicherlich naheliegende Abstufung

$$D_{\mathcal{C}} \to D_{\mathcal{C}} - 1 \to D_{\mathcal{C}} - 2 \to \ldots \to 1 \to 0,$$

bei der

$$D_{\varsigma - j} = D_{\mathcal{C}} - j$$

für alle $0 \leq j \leq D_{\mathcal{C}}$ ist, geht auf Reed zurück [30]. Reed setzt diese voraus, um Reed-Muller-Codes $\mathrm{RM}(D_{\mathcal{C}} - 1, m)$ wie in Proposition 5.2.1 gesehen klassisch zu decodieren. Reed-Muller-Codes werden wir in Abschnitt 7.5 näher beleuchten.

6.1.2 Chens Abstufung

Chen befasst sich in [7,8] mit Klassen von Codes, die auf euklidischen oder projektiven Geometrien basieren, den sogenannten Euklidische-Geometrie-Codes und Projektive-Geometrie-Codes. Zu diesen zählen auch die oben erwähnten Reed-Muller-Codes. Die von Chen vorgeschlagene Abstufung sieht wie folgt aus: Im Trivialfall $D_{\mathcal{C}} = 1$ gilt genau wie bei Reed

$$D_{\mathcal{C}} = 1 \to 0.$$

Falls hingegen $D_{\mathcal{C}} > 1$, so ist Chens Abstufung gegeben durch

$$\begin{aligned} D_{\mathcal{C}} &\to m - 2^0(m+1-D_{\mathcal{C}}) \\ &\to m - 2^1(m+1-D_{\mathcal{C}}) \\ &\to \dots \\ &\to m - 2^{\varsigma-2}(m+1-D_{\mathcal{C}}) \\ &\to 0, \end{aligned}$$

wobei

$$\varsigma := 1 + \lceil \log_2 (m/(m+1-D_{\mathcal{C}})) \rceil .$$

Es ist also für alle $j \in \mathbb{N}$, $1 \leq j \leq \varsigma - 1$,

$$D_j := m - 2^{\varsigma-1-j} (m+1-D_{\mathcal{C}}) . \qquad [6.1]$$

6.1.3 Invertierte Abstufung

Sei weiterhin genau wie bei Chens Abstufung

$$\varsigma := 1 + \lceil \log_2 (m/(m+1-D_{\mathcal{C}})) \rceil .$$

Ist $\varsigma = 1$ oder $\varsigma = 2$, bleibt die Abstufung unverändert,

$$D_1 \to 0$$

beziehungsweise

$$D_2 \to D_2 - 1 \to 0.$$

Betrachten wir den Fall $\varsigma \geq 3$. Wir übernehmen die Stufe

$$D_\varsigma := D_{\mathcal{C}} \to D_{\mathcal{C}} - 1 =: D_{\varsigma-1}$$

von Reed und Chen, um die Fehlerkorrektureigenschaften zu bewahren, wie wir in Abschnitt 6.2 sehen werden. Für alle $j \in \mathbb{N}$, $1 \leq j \leq \varsigma - 2$, wählen wir bei gegebenem D_{j-1} den Wert D_j maximal unter der Nebenbedingung

$$m \geq 2D_j - D_{j-1}.$$

Bei $D_0 \coloneqq 0$ beginnend definieren wir also

$$D_1 \coloneqq \lfloor (m + D_0)/2 \rfloor ,$$

$$D_2 \coloneqq \lfloor (m + D_1)/2 \rfloor$$

und setzen dies fort bis hin zu

$$D_{\varsigma-2} \coloneqq \lfloor (m + D_{\varsigma-3})/2 \rfloor .$$

Neben dieser rekursiven Notation lassen sich die Werte auch in expliziter Form darstellen.

Lemma 6.1.1. *Für alle $j \in \mathbb{N}_0$, $j \leq \varsigma - 2$, ist*

$$D_j = \left\lfloor m \cdot \left(2^j - 1\right) / 2^j \right\rfloor$$

Beweis. Per Induktion über j. Klar, für $D_0 = 0$ (Induktionsanfang).

Nach Induktionsannahme ist

$$D_j = \left\lfloor m \cdot \left(2^j - 1\right) / 2^j \right\rfloor$$

für ein $j \in \mathbb{N}_0$, $j \leq \varsigma - 3$. Dann ist wegen

$$m + \lfloor x \rfloor = \lfloor m + x \rfloor$$

und

$$\lfloor \lfloor x \rfloor / 2 \rfloor = \lfloor x/2 \rfloor$$

für alle $x \in \mathbb{R}$

$$\begin{aligned} D_{j+1} &\coloneqq \lfloor (m + D_j) / 2 \rfloor \\ &= \left\lfloor \left(m + \left\lfloor m \cdot \left(2^j - 1\right) / 2^j \right\rfloor\right) / 2 \right\rfloor \\ &= \left\lfloor \left\lfloor m \cdot \left(2^{j+1} - 1\right) / 2^j \right\rfloor / 2 \right\rfloor \\ &= \left\lfloor m \cdot \left(2^{j+1} - 1\right) / 2^{j+1} \right\rfloor \end{aligned}$$

□

An dieser Stelle ist noch offen, ob die eingeführte Abstufung streng monoton ist. Diese Lücke wollen wir nun schließen.

Proposition 6.1.2. *Für die invertierte Abstufung gilt*

$$D_\varsigma > D_{\varsigma-1} > D_{\varsigma-2} > \ldots > D_0.$$

Für alle $j \in \mathbb{N}_0$, $j \leq \varsigma - 3$, ist die Differenz $D_{j+1} - D_j$ mindestens zwei.

Beweis. Nach Definition von ς ist

$$2^{\varsigma-2} < \frac{m}{m+1-D_\varsigma}.$$

Diese Ungleichung umgestellt nach $D_{\varsigma-1} \coloneqq D_\varsigma - 1$ und Lemma 6.1.1 anwendend erhalten wir

$$D_{\varsigma-2} \leq \frac{2^{\varsigma-2}-1}{2^{\varsigma-2}} \cdot m < D_{\varsigma-1}.$$

Wir stellen die erste Ungleichung nach m um,

$$m > 2^{\varsigma-2} \cdot \underbrace{(m+1-D_\varsigma)}_{\geq 2} \geq 2^{\varsigma-1}.$$

Wiederum auf Lemma 6.1.1 verweisend ist für alle $j \in \mathbb{N}_0$, $j \leq \varsigma - 3$, die Differenz $D_{j+1} - D_j$ mindestens zwei, denn

$$\begin{aligned} D_{j+1} - D_j &= \left\lfloor \left(2^{j+1}-1\right)/2^{j+1} \cdot m \right\rfloor - \left\lfloor \left(2^j-1\right)/2^j \cdot m \right\rfloor \\ &= \left\lceil m/2^j \right\rceil - \left\lceil m/2^{j+1} \right\rceil \\ &= 2^{\varsigma-j-1} + \left\lceil \frac{m-2^{\varsigma-1}}{2^j} \right\rceil - 2^{\varsigma-j-2} - \left\lceil \frac{m-2^{\varsigma-1}}{2^{j+1}} \right\rceil \\ &\geq 2^{\varsigma-j-2} \geq 2. \end{aligned}$$

□

6.1.4 Resümee: Allgemeiner Vergleich der Abstufungen

Reeds Abstufung hat genauso viele oder mehr Stufen als die anderen beiden Abstufungen, deren Stufenanzahl identisch ist.

In Abbildung 6.1 ist die Anzahl der Stufen ς beim Decodieren unter Chens bzw. unter der invertierten Abstufung dargestellt. Zum einen als 2D-Plot in Abhängigkeit des Verhältnisses $(D_\mathcal{C} - 1)/m$, zum anderen als 3D-Plot in Abhängigkeit der Parameter $D_\mathcal{C}$ und m. Abbildung 6.1 veranschaulicht, dass für

eine Vielzahl von Codes sehr wenige Majority-Logic-Stufen genügen, um zu decodieren.

Eine reduzierte Anzahl von Majority-Logic-Stufen bedeutet auf den ersten Blick vor allem einen geringeren Rechenaufwand. Auf dem zweiten Blick wird ein weiterer Vorteil ersichtlich: Parallelisiert man alle Berechnungen weitestgehend, so ist die Decodierzeit in der Regel wesentlich verkürzt.

Ein Vorteil der invertierten Abstufung ist zudem, dass die Stufen zunächst unabhängig von $D_{\mathcal{C}}$ nur in Abhängigkeit von m definiert werden. Dies ermöglicht bei verschiedenen $D_{\mathcal{C}}$ aber gleichem m bis zu einer gewissen Stufe dieselbe Abstufung, so dass Teilgraphen der Graphen, die für die Decodierung zu konstruieren sind, wiederverwendet werden können.

Wir werden in den nächsten beiden Abschnitten sehen, dass die Fehlerkorrekturfähigkeit bei allen drei Abstufungen gleich ist und unter der invertierten Abstufung der Decodieraufwand hinsichtlich der Anzahl der heranzuziehenden Fehlersummen und der zu treffenden Mehrheitsentscheidungen kleiner oder genauso hoch ist wie unter den anderen beiden Abstufungen. Bei der Anzahl der Additionen und Subtraktionen muss hingegen differenziert werden.

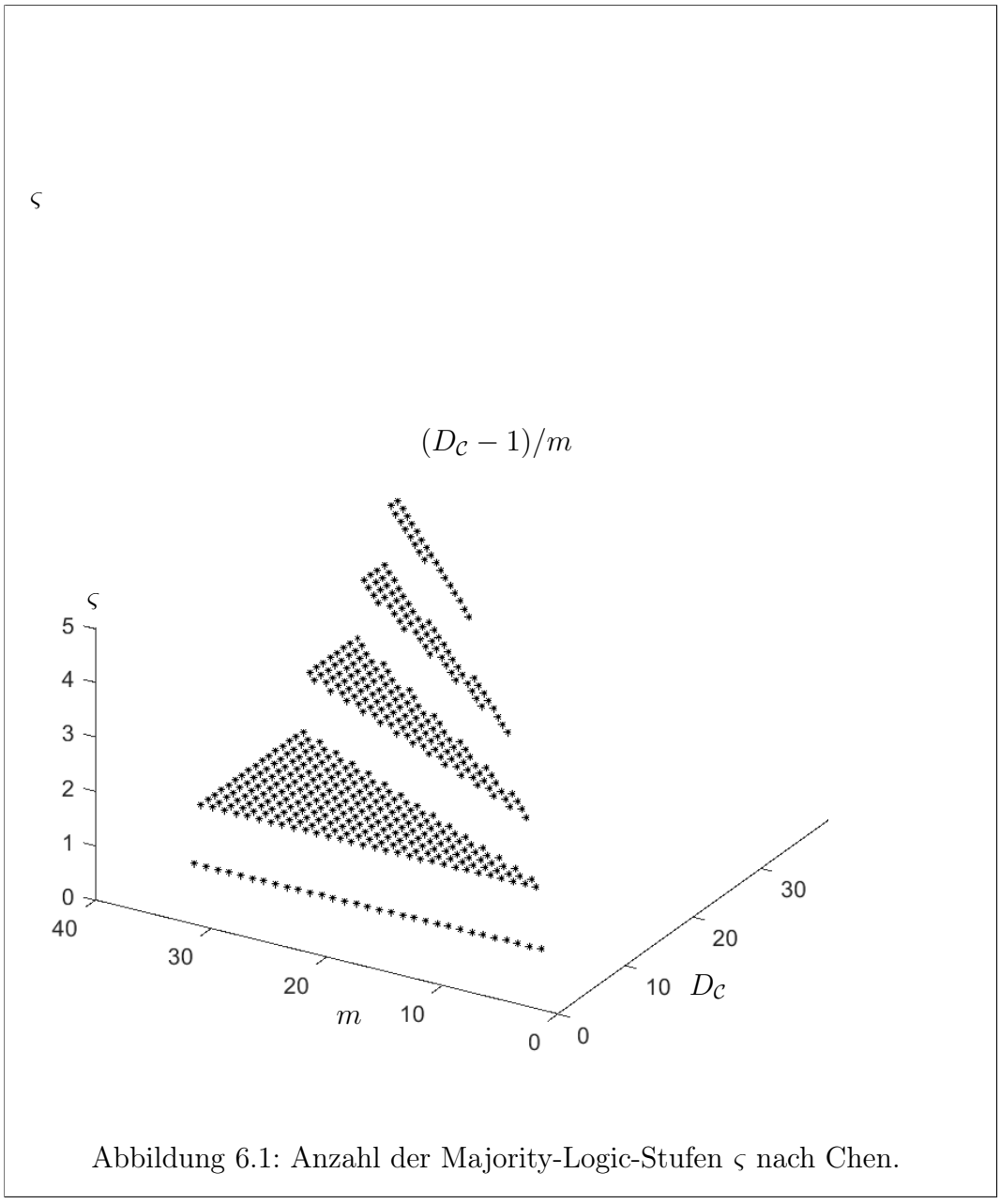


Abbildung 6.1: Anzahl der Majority-Logic-Stufen ς nach Chen.

6.2 Anzahl korrigierbarer Fehler unter verschiedenen Abstufungen

Für eine bessere Vergleichbarkeit der Abstufungen nehmen wir im Folgenden an, dass $D_{\mathcal{C}}$ bei den Verfahren der klassischen, verbesserten und hybriden Decodierung aus Proposition 5.2.1, Theorem 5.3.1 und Theorem 5.4.4 identisch ist.

Mit Blick auf die Decodiervoraussetzungen stellen wir fest, dass $\ell^*_{D_j,D_{j+1},m,q}$ und $\ell_{D_j,D_{j+1},m,q}$ für alle $j \in \mathbb{Z}_\varsigma$ neben η, der Anzahl der bei jeder Mehrheitsentscheidung herangezogenen Werte, vorgeben, wie viele Fehler mit obigen Verfahren auf jeden Fall – unabhängig vom gesendeten Codewort und an welcher Stelle die Fehler auftreten – korrigiert werden können.

Wir fragen uns daher, unter welcher Abstufung

$$\min_{j\in\mathbb{Z}_\varsigma}\left(\ell_{D_j,D_{j+1},m,q}-1\right) \quad \text{respektive} \quad \min_{j\in\mathbb{Z}_\varsigma}\left(\ell^*_{D_j,D_{j+1},m,q}\right)$$

maximal ist und wie dieses Maximum aussieht. Wir zeigen zunächst, dass, unabhängig von der konkret gewählten Abstufung, eine obere Schranke bei

$$\frac{q^{m+1-D_{\mathcal{C}}}-q}{q-1}$$

liegt. Diese wird jedoch nicht angenommen, wenn $D_{\varsigma-1} \neq D_{\mathcal{C}}-1$ gewählt wird. Mit anderen Worten, obige Verfahren erlauben

$$\tau_{\mathrm{MLG}} \coloneqq \left\lfloor \frac{q^{m+1-D_{\mathcal{C}}}-q}{2(q-1)} \right\rfloor$$

Fehler zu korrigieren.

Proposition 6.2.1. *Unabhängig von der konkret gewählten Abstufung in Proposition 5.2.1, Theorem 5.3.1 oder Theorem 5.4.4 ist sowohl*

$$\min_{j\in\mathbb{Z}_\varsigma}\left(\ell_{D_j,D_{j+1},m,q}-1\right) \leq \frac{q^{m+1-D_{\mathcal{C}}}-q}{q-1}.$$

als auch

$$\min_{j\in\mathbb{Z}_\varsigma}\left(\ell^*_{D_j,D_{j+1},m,q}\right) \leq \frac{q^{m+1-D_{\mathcal{C}}}-q}{q-1}.$$

Die Ungleichungen sind strikt, falls $D_{\varsigma-1} \neq D_{\mathcal{C}}-1$ gewählt wird.

Beweis. Wir wissen von Proposition 5.1.4 und Korollar A.0.6, dass

$$\frac{q^{m+1-D_{\mathcal{C}}}-q}{q-1} = \ell_{D_{\mathcal{C}}-1,D_{\mathcal{C}},m,q} - 1 = \ell^*_{D_{\mathcal{C}}-1,D_{\mathcal{C}},m,q} \\ > \ell^*_{D_{\varsigma-1},D_{\mathcal{C}},m,q} \geq \ell_{D_{\varsigma-1},D_{\mathcal{C}},m,q} - 1.$$

für alle $D_{\varsigma-1} < D_{\mathcal{C}} - 1$. □

6.2.1 Fehlerkorrekturfähigkeit unter Reeds Abstufung

Tatsächlich sind mit Reeds Abstufung

$$D_{\mathcal{C}} \to D_{\mathcal{C}} - 1 \to D_{\mathcal{C}} - 2 \to \ldots \to 1 \to 0,$$

bei den oben genannten Verfahren stets τ_{MLG} Fehler korrigierbar: Es ist zum einen wegen Proposition 5.1.4

$$\ell_{j,j+1,m,q} - 1 = \ell^*_{j,j+1,m,q}$$

für alle $j \in \mathbb{Z}_{D_\varsigma}$ und zum anderen wegen Proposition A.0.5

$$\ell^*_{0,1,m,q} \geq \ell^*_{1,2,m,q} \geq \ldots \geq \ell^*_{D_\varsigma-1,D_\varsigma,m,q} \geq 2\tau_{\mathrm{MLG}}.$$

6.2.2 Fehlerkorrekturfähigkeit unter Chens Abstufung

Chen konnte zeigen, dass es statt $D_{\mathcal{C}}$ Stufen wie bei Reed nur

$$\varsigma \coloneqq 1 + \lceil \log_2 (m/(m+1-D_{\mathcal{C}})) \rceil$$

Majority-Logic-Stufen bedarf bei gleicher Fehlerkorrekturgrenze. Insbesondere reichen für Reed-Muller-Codes $\mathrm{RM}(r,m)$ mit $r \leq m/2$ zwei Majority-Logic-Stufen aus. Reed hatte für diese Codes $r+1$ Majority-Logic-Stufen vorgesehen.

Dieselbe Fehlerkorrekturgrenze wie bei Reed ergibt sich dadurch, dass Chen die erste Stufe $D_{\mathcal{C}} \to D_{\mathcal{C}} - 1 =: D_{\varsigma-1}$ von Reed übernimmt und die folgenden Stufen $D_j \to D_{j-1}$ so wählt, dass für alle $j \in \mathbb{N}$, $2 \leq j \leq \varsigma - 1$, gilt

$$m = 2D_j - D_{j-1}.$$

Proposition 6.2.2. *Unter Chens Abstufung können mit den Decodierverfahren aus Proposition 5.2.1, Theorem 5.3.1 und Theorem 5.4.4 stets τ_{MLG} Fehler korrigiert werden.*

Beweis. Nach Definition von ς ist

$$2^{\varsigma-1} \geq \frac{m}{m+1-D_\varsigma},$$

umgestellt zu

$$0 \geq m - 2^{\varsigma-1}\left(m+1-D_\varsigma\right).$$

Für die letzte Stufe

$$D_1 \coloneqq m - 2^{\varsigma-2}\left(m+1-D_\varsigma\right) \to D_0 \coloneqq 0$$

gilt dann ebenfalls

$$m \geq m + m - 2^{\varsigma-1}\left(m+1-D_\varsigma\right) = 2D_1 - D_0.$$

Aus Korollar A.0.7 folgt

$$\ell^*_{D_{j-1},D_j,m,q} \geq \ell_{D_{j-1},D_j,m,q} - 1 \geq q^{m-D_j} \geq q^{m-D_{\varsigma-1}} = q^{m+1-D_\varsigma}$$

für alle $j \in \mathbb{N}$, $1 \leq j \leq \varsigma - 1$. Andererseits ist

$$q^{m+1-D_\varsigma} > \frac{q^{m+1-D_\varsigma} - q}{q-1} = \ell_{D_\varsigma-1,D_\varsigma,m,q} - 1 = \ell^*_{D_\varsigma-1,D_\varsigma,m,q}.$$

Unter Chens Abstufung können mit obigen Verfahren also ebenso τ_{MLG} Fehler korrigiert werden. □

6.2.3 Fehlerkorrekturfähigkeit unter der invertierten Abstufung

Chens Abstufung analysierend haben wir festgestellt, dass beginnend bei $D_{\mathcal{C}}$ die Stufen in absteigender Folge derart gewählt werden, dass bei gegebenem D_j der Wert D_{j-1} maximal unter der Nebenbedingung

$$m \geq 2D_j - D_{j-1}$$

ist. Genauso gut ist denkbar, die Stufen beginnend bei null in aufsteigender Folge unter dieser Nebenbedingung zu wählen. Diese Idee umsetzend und sie im Namen verankernd haben wir die *invertierte* Abstufung hergeleitet.

Zu Beginn des Kapitels behaupteten wir, dass auch mit dieser Abstufung τ_{MLG} Fehler korrigiert werden können. Dies werden wir nun zeigen.

Proposition 6.2.3. *Mit der invertierten Abstufung können stets bis zu τ_{MLG} Fehler korrigiert werden.*

Beweis. Die Argumentation ist dieselbe wie bei Chens Abstufung.

Nach Konstruktion ist

$$m \geq 2D_j - D_{j-1}$$

für alle $1 \leq j \leq \varsigma - 2$. Diese Ungleichung wollen für auch für $j = \varsigma - 1$ zeigen.

Nach Definition von ς ist

$$\frac{m}{m + 1 - D_\varsigma} \leq 2^{\varsigma - 1},$$

umgestellt nach $D_{\varsigma-1} \in \mathbb{N}_0$

$$D_{\varsigma-1} \leq m \cdot \left(2^{\varsigma-1} - 1\right) / 2^{\varsigma-1}.$$

Da $2 \cdot D_{\varsigma-1} \in \mathbb{N}_0$, ist

$$\begin{aligned} 2D_{\varsigma-1} &\leq \left\lfloor m \cdot \left(2^{\varsigma-1} - 1\right) / 2^{\varsigma-2} \right\rfloor \\ &= m + \left\lfloor m \cdot \left(2^{\varsigma-2} - 1\right) / 2^{\varsigma-2} \right\rfloor \\ &= m + D_{\varsigma-2} \end{aligned}$$

Mit Korollar A.0.7 schlussfolgern wir,

$$\ell^*_{D_{j-1},D_j,m,q} \geq \ell_{D_{j-1},D_j,m,q} - 1 \geq q^{m-D_j} \geq q^{m-D_{\varsigma-1}} = q^{m+1-D_\varsigma}$$

für alle $1 \leq j \leq \varsigma - 1$. Indem wir die Stufe $D_\varsigma \to D_\varsigma - 1$ beibehalten, stellen wir sicher, dass tatsächlich bis zu τ_{MLG} Fehler korrigiert werden können, denn

$$q^{m+1-D_\varsigma} > \frac{q^{m+1-D_\varsigma} - q}{q - 1} = \ell^*_{D_\varsigma - 1, D_\varsigma, m, q} = \ell_{D_\varsigma - 1, D_\varsigma, m, q} - 1. \qquad \square$$

6.2.4 Resümee: Vergleich der Abstufungen hinsichtlich der Anzahl korrigierbarer Fehler

Hinsichtlich der Fehlerkorrekturfähigkeit haben Reed und Chen die Majority-Logic-Decodierung bereits optimiert. Wie wir in Abschnitt 6.2.3 gesehen haben, weist auch die invertierte Abstufung die gleiche Fehlerkorrekturgrenze auf. Diese liegt bei τ_{MLG}, vorausgesetzt $\eta = 2 \cdot \tau_{\mathrm{MLG}}$ der zur Verfügung stehenden Fehlersummen werden in jeder Mehrheitsentscheidung einer jeden Majority-Logic-Stufe berücksichtigt. Treten mehr als τ_{MLG} Fehler auf, können wir keine allgemeingültige Aussage treffen. Das vom Decoder ausgegebene Wort ist mitunter ein Codewort, jedoch nicht zwangsläufig das übertragene.

Weitere Abstufungen mit Fehlerkorrekturgrenze τ_{MLG} sind denkbar. So wird in [29, §10.4, S. 337 f.] folgende rekursiv definierte Abstufung vorgestellt,

$$D_{\mathcal{C}} \to D_{\varsigma-1} \coloneqq D_{\mathcal{C}} - 1 \to D_{\varsigma-2} \coloneqq D_{\mathcal{C}} - 1 - \mathrm{ggT}\left(D_{\mathcal{C}} - 1, m\right) \to \ldots \to 0,$$

wobei

$$D_j \coloneqq D_{j+1} - \mathrm{ggT}\left(D_{j+1}, m\right)$$

für alle $0 \leq j \leq \varsigma - 2$. Wenn m prim ist, ist sie also identisch zu Reeds Abstufung. Bestenfalls besteht sie aus genauso vielen Majority-Logic-Stufen wie Chens und die invertierte Abstufung. In diesem Fall ist sie identisch zu einer der beiden Abstufungen. Daher werden wir uns im Folgenden auf die Abstufungen von Reed und Chen sowie die invertierte fokussieren.

6.3 Decodieraufwand unter den verschiedenen Abstufungen

Zu jedem der Decodierverfahren, das wir in Kapitel 5 vorgestellt haben, haben wir angegeben, wie viele Operationen allgemein benötigt werden. Zudem haben wir die drei Verfahren untereinander hinsichtlich ihres Aufwands verglichen (siehe Bemerkung 5.3.4 und Bemerkung 5.4.6) und gesehen, dass die wenigstens Mehrheitsentscheidungen bei der Hybriddecodierung getroffen werden. Bislang

haben wir konkrete Abstufungen bei der Aufwandsabschätzung noch außen vor gelassen. Diese wollen wir jetzt mit einbeziehen.

Um die drei Abstufungen des vorherigen Abschnitts untereinander besser vergleichen zu können, setzen wir in Abhängigkeit der Parameter q, m, $D_{\mathcal{C}}$

$$
\begin{aligned}
n &\coloneqq q^m - 1, \\
\eta &\coloneqq 2 \cdot \left\lfloor \frac{q^{m+1-D_{\mathcal{C}}} - q}{2(q-1)} \right\rfloor = \frac{q^{m+1-D_{\mathcal{C}}} - q}{q-1} - \begin{cases} 1 & p \neq 2, m - D_{\mathcal{C}} \text{ ungerade}, \\ 0 & \text{sonst} \end{cases}, \\
\varsigma &\coloneqq 1 + \lceil \log_2 \left(m/(m+1-D_{\mathcal{C}}) \right) \rceil, \\
\dot{D} &\coloneqq \left\lfloor \log_q((q-1)(\eta+1)+1) \right\rfloor = m - D_{\mathcal{C}} + \begin{cases} 0 & p \neq 2, m - D_{\mathcal{C}} \text{ ungerade}, \\ 1 & \text{sonst} \end{cases}.
\end{aligned}
$$

Beachte, dass die Anzahl der Majority-Logic-Stufen unter Reeds Abstufung $D_{\mathcal{C}}$ und nicht ς ist.

Die Operationen μ und $\pm$ symbolisieren jeweils eine Mehrheitsentscheidung beziehungsweise eine Addition/Subtraktion über $\mathbb{F}_{q_{\mathcal{C}}}$. Die Buchstaben R, C und I repräsentieren jeweils Reeds, Chens und die invertierte Abstufung. Es bezeichne $\mathcal{K}_{\sim,A}$ bzw. $\mathcal{V}_{\sim,A}$ bzw. $\mathcal{H}_{\sim,A}$ die Anzahl der zur klassischen bzw. verbesserten bzw. hybriden Decodierung benötigten Operationen $\sim$ unter der Abstufung A, wobei $\sim \in \{\mu, \pm\}$ und $A \in \{R, C, I\}$. Weiterhin werde für alle $A \in \{R, C, I\}$ der Parameter D_t unter der Abstufung A bezeichnet durch $D_{t,A}$.

Wir werden sehen, dass zum einen das klassische Verfahren und zum anderen Reeds Abstufung mit den beiden anderen Verfahren beziehungsweise Abstufungen hinsichtlich des Decodieraufwands nicht konkurrieren kann. Legt man Wert auf eine möglichst geringe Anzahl von Mehrheitsentscheidungen ist die Hybriddecodierung unter der invertierten Abstufung allen anderen Varianten stets vorzuziehen.

6.3.1 Vergleich der Parameter der Abstufung nach Chen mit jenen der invertierten Abstufung

Wir wollen der Frage nachgehen, unter welchen Bedingungen und inwiefern sich die beiden Abstufungen, jene nach Chen und die invertierte, voneinander unterscheiden. Kurz gesagt gilt, dass der Parameter D_t unter der invertierten Abstufung stets am größten ist – was wir in diesem Abschnitt formal beweisen werden. Wir werden in den folgenden Abschnitten sehen, dass dies und die Tatsache $\varsigma \leq D_{\mathcal{C}}$ die Effizienz der invertierten Abstufung primär erklären.

Proposition 6.3.1. *Sei $j \in \mathbb{N}$, $0 \leq j \leq \varsigma$, beliebig.*

Dann ist

$$D_{j,R} \coloneqq j \leq D_{j,C} \leq D_{j,I}.$$

Beweis. Klar, $D_{j,C} \geq j$, da die Folge $(D_{i,C})_{i=0}^{\varsigma}$ beginnend bei null streng monoton wächst. Also ist

$$D_{j,C} \geq j = D_{j,R}.$$

Bleibt zu zeigen, $D_{j,I} \geq D_{j,C}$. Nach Definition gilt $D_{i,C} = D_{i,I}$ für alle $i = 0, \varsigma - 1, \varsigma$. Betrachten wir daher den Fall $0 < j \leq \varsigma - 2$. Die Definitionen von $D_{j,C}$ und $D_{j,I}$ entnehmen wir Gleichung [6.1] auf Seite 95 und Lemma 6.1.1,

$$D_{j,C} \coloneqq m - 2^{\varsigma-1-j}\left(m + 1 - D_{\mathcal{C}}\right), \qquad D_{j,I} \coloneqq \left\lfloor \left(2^j - 1\right)/2^j \cdot m \right\rfloor.$$

Nach Definition von ς ist

$$m/(m + 1 - D_{\mathcal{C}}) \leq 2^{\varsigma-1},$$

woraus folgt

$$\left\lceil m/2^{\varsigma-1} \right\rceil \leq m + 1 - D_{\mathcal{C}}. \tag{6.2}$$

Des weiteren ist

$$2^j \left\lceil m/2^j \right\rceil \leq 2^{\varsigma-1} \left\lceil m/2^{\varsigma-1} \right\rceil. \tag{6.3}$$

Dies lässt sich leicht nachweisen, indem wir für beliebiges $i \in \mathbb{N}$ den Parameter m in Form von $m = a \cdot 2^i + b$ für gewisse $a, b \in \mathbb{N}_0$ mit $0 \leq b < 2^i$ schreiben, so dass

$$2^{i+1} \left\lceil m/2^{i+1} \right\rceil = 2^{i+1} \cdot \begin{cases} a/2 + \mathbb{1}_{b>0} & a \text{ gerade}, \\ (a-1)/2 + 1 & a \text{ ungerade} \end{cases}$$

$$\begin{aligned} &\geq 2^i(a + \mathbb{1}_{b>0}) \\ &= 2^i \left\lceil m/2^i \right\rceil . \end{aligned}$$

Alles in allem ist

$$\begin{aligned} D_{j,I} &:= \left\lfloor \left(2^j - 1\right) / 2^j \cdot m \right\rfloor \\ &= m - \left\lceil m/2^j \right\rceil \\ &\underset{[6.3]}{\geq} m - 2^{\varsigma-1-j} \left\lceil m/2^{\varsigma-1} \right\rceil \\ &\underset{[6.2]}{\geq} m - 2^{\varsigma-1-j} \left(m + 1 - D_{\mathcal{C}}\right) =: D_{j,C}. \end{aligned}$$ □

An dieser Stelle wollen wir die Frage beantworten, unter welchen Voraussetzungen, Chens und die invertierte Abstufung identisch sind.

Proposition 6.3.2. *Genau dann, wenn $\varsigma \leq 2$ oder $m/(m+1-D_{\mathcal{C}})$ oder $(m+1)/(m+1-D_{\mathcal{C}})$ eine Zweierpotenz ist, fallen Chens und die invertierte Abstufung zusammen.*

Beweis. Wir wissen, $D_{\varsigma,C} = D_{\mathcal{C}} = D_{\varsigma,I}$ und $D_{\varsigma-1,C} = D_{\mathcal{C}} - 1 = D_{\varsigma-1,I}$. Offensichtlich sind beide Abstufungen identisch, wenn $\varsigma \leq 2$. Betrachten wir also den Fall $\varsigma > 2$. Folgende Äquivalenzen sind in absteigender Reihenfolge sofort ersichtlich

$$\begin{aligned} &m/(m+1-D_{\mathcal{C}}) \text{ oder } (m+1)/(m+1-D_{\mathcal{C}}) \text{ ist eine Zweierpotenz;} \\ \iff\ &m \in \{2^{\varsigma-1}\left(m+1-D_{\mathcal{C}}\right), 2^{\varsigma-1}\left(m+1-D_{\mathcal{C}}\right) - 1\}; \\ \iff\ &2^{\varsigma-2}\left(m+1-D_{\mathcal{C}}\right) = \lceil m/2 \rceil; \\ \iff\ &D_{1,C} = D_{1,I}. \end{aligned}$$

Damit ist die Rückrichtung bereits bewiesen. Bleibt die Hinrichtung zu zeigen. Unter der Annahme

$$m \in \left\{2^{\varsigma-1}\left(m+1-D_{\mathcal{C}}\right), 2^{\varsigma-1}\left(m+1-D_{\mathcal{C}}\right) - 1\right\}$$

gilt für alle j, $1 < j < \varsigma - 2$,

$$D_{j,C} - D_{j,I} = \left\lceil m/2^j \right\rceil - 2^{\varsigma-1-j}(m+1-D_{\mathcal{C}}) = 0.$$ □

6.3.2 Vergleich des Aufwands der klassischen und verbesserten Decodierung unter drei Abstufungen

Zunächst vergleichen wir in Tabelle 6.1 die drei Abstufungen nur für das klassische und das verbesserte Verfahren. Die dort getroffenen Aussagen lassen sich formal mit Hilfe von Proposition 6.3.1 beweisen.

Tabelle 6.1: Aufwand der klassischen und verbesserten Decodierung unter drei Abstufungen

Klassische Decodierung nach Proposition 5.2.1					Verbesserte Decodierung nach Theorem 5.3.1				
Operation	Reed		Chen	Inv.	Operation	Reed		Chen	Inv.
$\mathcal{K}_\mu$	$\mathcal{K}_{\mu,R}$	$\geq$	$\mathcal{K}_{\mu,C}$ $=$	$\mathcal{K}_{\mu,I}$	$\mathcal{V}_\mu$	$\mathcal{V}_{\mu,R}$	$\geq$	$\mathcal{V}_{\mu,C}$ $\geq$	$\mathcal{V}_{\mu,I}$
$\mathcal{K}_{\mathtt{E}}$	$\mathcal{K}_{\mathtt{E},R}$	$\geq$	$\mathcal{K}_{\mathtt{E},C}$ $=$	$\mathcal{K}_{\mathtt{E},I}$	$\mathcal{V}_{\mathtt{E}}$	$\mathcal{V}_{\mathtt{E},R}$	$\geq$	$\mathcal{V}_{\mathtt{E},C}$ $=$	$\mathcal{V}_{\mathtt{E},I}$

Es liegt auf der Hand, dass die beiden Verfahren unter Reeds Abstufung die größte Komplexität haben. Dies hat zwei Ursachen. Zum einen gibt es bei Reeds Abstufung gleich viele oder mehr Majority-Logic-Stufen gegenüber den anderen beiden Abstufungen, $D_{\mathcal{C}} \geq \varsigma$; zum anderen ist für alle $j \in \mathbb{N}$, $1 \leq j \leq \varsigma - 1$ unter Reeds Abstufung der Parameter D_j mit $D_j \coloneqq j$ stets minimal (vgl. Proposition 6.3.1) und damit $m - D_j$ stets maximal gewählt.

Da die Anzahl der Stufen bei Chens und der invertierten Abstufung identisch ist, ist der Aufwand bei der klassischen Decodierung unter diesen beiden Abstufungen gleich. Sofern sich beide Abstufungen voneinander unterscheiden, liegt der Vorteil der invertierten gegenüber Chens Abstufung darin begründet, dass für alle $j \in \mathbb{N}$, $1 \leq j \leq \varsigma - 1$, der Parameter D_j unter der invertierten Abstufung stets mindestens so groß wie der Parameter D_j unter Chens Abstufung ist, wie wir in Proposition 6.3.1 gesehen haben.

Wir halten zusammenfassend fest, bei reiner Majority-Logic-Decodierung bietet die invertierte Abstufung die größte Effizienz.

6.3.3 Vergleich des Aufwands der Hybriddecodierung unter drei Abstufungen

Jede der drei vorgestellten Abstufungen definiert eine Menge T, wie wir sie in Gleichung [5.17] auf Seite 80 eingeführt haben. Wie diese konkret aussehen, geben wir in Lemma 6.3.3 an.

Lemma 6.3.3. *Gegeben $D_{\mathcal{C}}$ und $\varsigma \coloneqq 1+\lceil \log_2 (m/(m+1-D_{\mathcal{C}}))\rceil$. Wir führen drei Bedingungen ein,*

(i) $\varsigma = 2 = D_{\mathcal{C}}$;

(ii) $\varsigma \geq 3$, $m = 1 + 2^{\varsigma-2}(m+1-D_{\mathcal{C}})$;

(iii) $\varsigma \geq 3$, $m+1-D_{\mathcal{C}} = \lceil m/2^{\varsigma-2}\rceil - 1$.

Bei Reeds Abstufung ist

$$T = T_R \coloneqq \mathbb{Z}_{D_{\mathcal{C}}}.$$

Bei Chens Abstufung ist

$$T = T_C \coloneqq \begin{cases} \{0, \varsigma-1\} & \textit{Bedingung (i) oder (ii) ist erfüllt,} \\ \{\varsigma-1\} & \textit{sonst} \end{cases}.$$

Bei der invertierten Abstufung ist

$$T = T_I \coloneqq \begin{cases} \{\varsigma-2, \varsigma-1\} & \textit{Bedingung (i) oder (iii) ist erfüllt,} \\ \{\varsigma-1\} & \textit{sonst} \end{cases}.$$

Beweis. Bei Reeds Abstufung ist nichts zu beweisen. Die Abstufungen sind so definiert, dass $\varsigma - 1$ stets in T enthalten ist. Im Fall $\varsigma = D_{\mathcal{C}} = 2$ ist

$$D_1 \coloneqq D_2 - 1 = 1,$$

also $0 \in T$ beziehungsweise $\varsigma - 2 \in T$. Ist jedoch $\varsigma = 2 < D_{\mathcal{C}}$, dann ist $D_1 > 1$ und $0 \notin T$.

Betrachten wir Chens Abstufung für $\varsigma \geq 3$. Nach Definition ist

$$D_j \coloneqq m - 2^{\varsigma-1-j}\left(m + 1 - D_{\mathcal{C}}\right)$$

für alle $1 \leq j \leq \varsigma - 1$. Dann ist

$$\begin{aligned} D_{j+1} - D_j &= -2^{\varsigma-2-j}\left(m + 1 - D_{\mathcal{C}}\right) + 2^{\varsigma-1-j}\left(m + 1 - D_{\mathcal{C}}\right) \\ &= \underbrace{2^{\varsigma-2-j}}_{\geq 1}\underbrace{\left(m + 1 - D_{\mathcal{C}}\right)}_{\geq 2} > 1 \end{aligned}$$

für alle $1 \leq j \leq \varsigma - 2$. Es ist $D_1 - D_0 = 1$ genau dann, wenn

$$m = 1 + 2^{\varsigma-2}\left(m + 1 - D_{\mathcal{C}}\right).$$

Abschließend überprüfen wir die Behauptungen für die invertierte Abstufung im Fall $\varsigma \geq 3$. Aus Proposition 6.1.2 wissen wir bereits, dass für alle $j \in \mathbb{N}_0$, $0 \leq j \leq \varsigma - 3$ die Differenz von D_{j+1} und D_j mindestens zwei ist.

Bleibt also $D_{\varsigma-1} - D_{\varsigma-2}$ zu überprüfen:

$$D_{\varsigma-1} - D_{\varsigma-2} = D_{\mathcal{C}} - 1 - \left\lfloor \left(2^{\varsigma-2} - 1\right) / 2^{\varsigma-2} \cdot m \right\rfloor,$$

gemäß der expliziten Notation von $D_{\varsigma-2}$ aus Lemma 6.1.1. Es ist also genau dann $D_{\varsigma-1} - D_{\varsigma-2} = 1$, wenn

$$D_{\mathcal{C}} = \left\lfloor \left(2^{\varsigma-2} - 1\right) / 2^{\varsigma-2} \cdot m \right\rfloor + 2$$

respektive

$$m + 1 - D_{\mathcal{C}} = \left\lceil m / 2^{\varsigma-2} \right\rceil - 1. \qquad \square$$

In Tabelle 6.2 vergleichen wir die drei vorgestellten Abstufungen hinsichtlich des Decodieraufwands für die Hybriddecodierung. In einer Vielzahl der Fälle bietet die invertierte Abstufung die effizienteste Decodierung. Die Ungleichungen formal nachzuweisen, ist langwierig, weshalb der Beweis in Anhang B zu finden ist.

Tabelle 6.2: Aufwand der Hybriddecodierung nach Theorem 5.4.4 unter drei Abstufungen

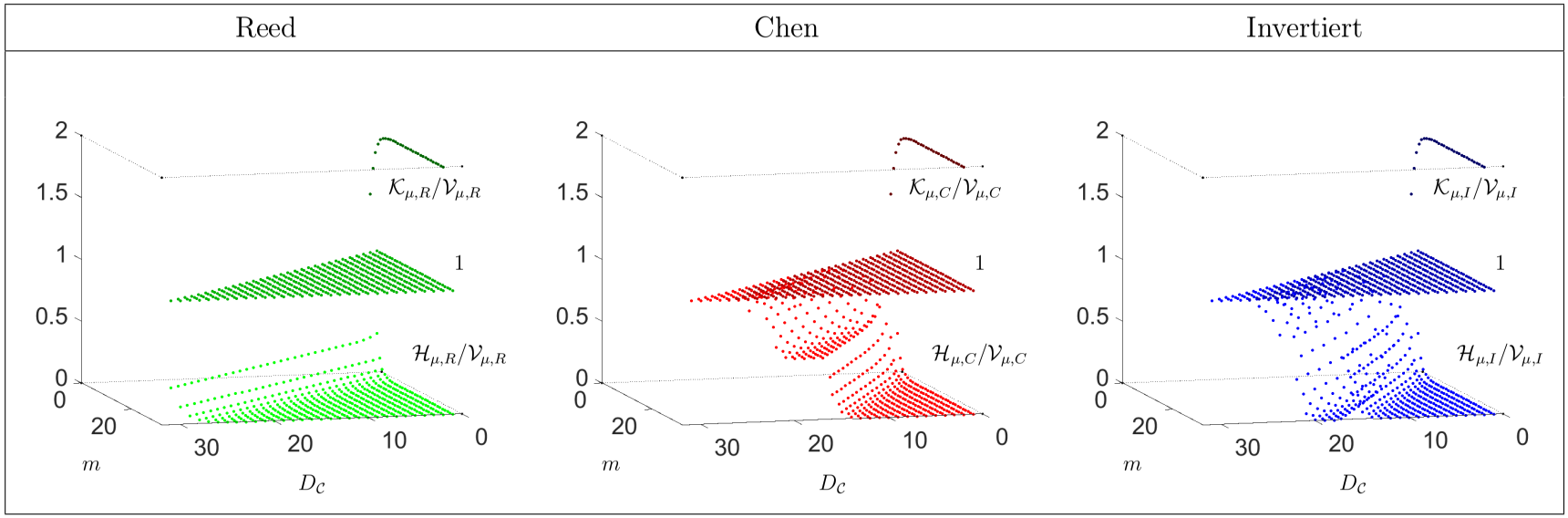
Operation		Reed		Chen		Invertiert		Reed
$\mathcal{H}_\mu$		$\mathcal{H}_{\mu,R}$	$\geq$	$\mathcal{H}_{\mu,C}$	$\geq$	$\mathcal{H}_{\mu,I}$		
$\mathcal{H}_{\mathtt{E}}$		$\mathcal{H}_{\mathtt{E},R}$	$\geq$	$\mathcal{H}_{\mathtt{E},C}$	$=$	$\mathcal{H}_{\mathtt{E},I}$		
$\mathcal{H}_\pm$	$T_C = T_I$	$\mathcal{H}_{\pm,R}$	$\geq$	$\mathcal{H}_{\pm,C}$	$=$	$\mathcal{H}_{\pm,I}$		
	$T_C \subset T_I$			$\mathcal{H}_{\pm,C}$	$<$	$\mathcal{H}_{\pm,I}$	$\leq$	$\mathcal{H}_{\pm,R}$
	$T_C \not\subseteq T_I$	$\mathcal{H}_{\pm,R}$	$\geq$	$\mathcal{H}_{\pm,C}$	$>$	$\mathcal{H}_{\pm,I}$		

6.3.4 Grafische Darstellung des Decodieraufwands

Zur Veranschaulichung haben wir in Abbildung 6.4 den Zweierlogarithmus der Gesamtzahl der Mehrheitsentscheidungen in Abhängigkeit von $D_{\mathcal{C}}, m$ unter der Annahme $q = 2$ geplottet. Datenpunkte zum klassischen Verfahren, verbesserten und hybriden Verfahren sind jeweils dunkel-, mittel- und hellfarben gehalten. Reeds, Chens und der invertierten Abstufung werden jeweils die Farben grün, rot und blau zugewiesen. Magentafarbene Datenpunkte bedeuten, dass unter Chens und der invertierten Abstufung dieselbe Anzahl von Mehrheitsentscheidungen benötigt wird. Schwarze Datenpunkte symbolisieren, dass unter allen drei Abstufungen mit derselben Anzahl von Mehrheitsentscheidungen decodiert wird.

Abbildung 6.4 stellt die Absolutzahlen der benötigten Mehrheitsentscheidungen dar. Diese setzen wir in Abbildung 6.2 und Abbildung 6.3 zueinander ins Verhältnis. Anhand der Abbildungen in Abbildung 6.2 sieht man unmittelbar, dass die klassische Decodierung nicht mit den anderen Verfahren konkurrieren kann und die Hybriddecodierung ihren Vorteil gegenüber der verbesserten Decodierung gerade bei einem großen Verhältnis $m\colon D_{\mathcal{C}}$ besitzt. In den Abbildungen aus Abbildung 6.3 sind nur wenige Datenpunkte zu Reeds Abstufung aufgeführt, da unter dieser in den meisten Fällen mehr als das Doppelte an Mehrheitsentscheidungen getroffen werden müssen.

Abbildung 6.2: Verhältnisse der Decodierverfahren zum verbesserten Decodierverfahren unter fester Abstufung jeweils hinsichtlich der Gesamtzahl der benötigten Mehrheitsentscheidungen für $q = 2$



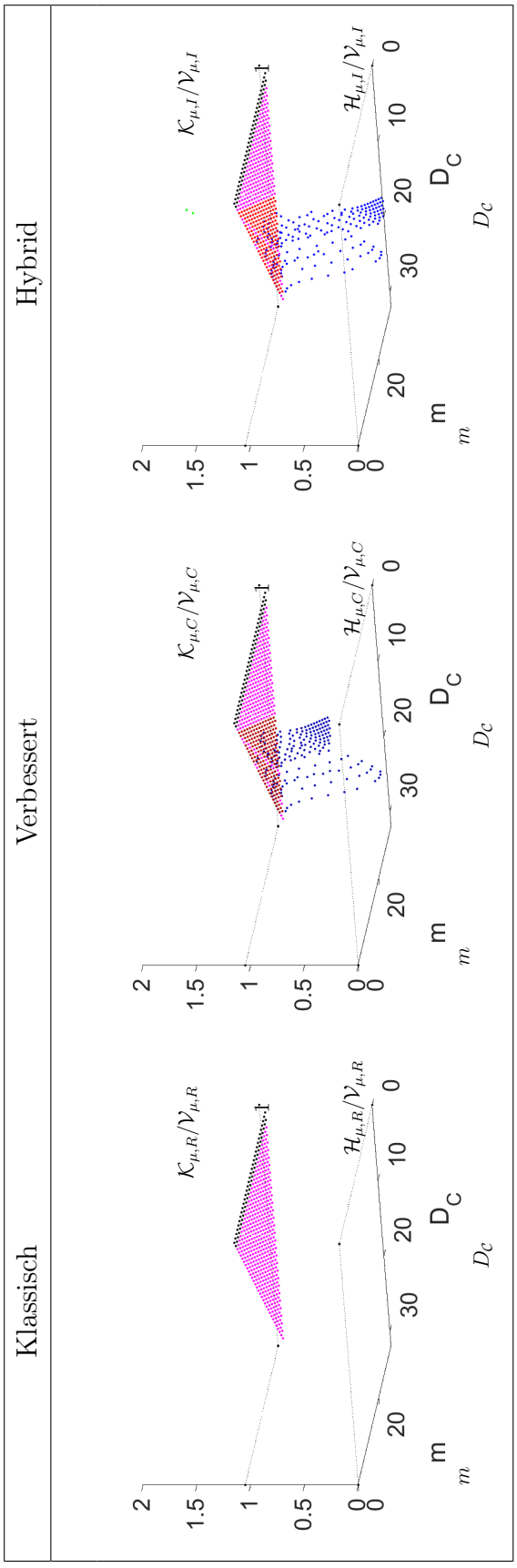
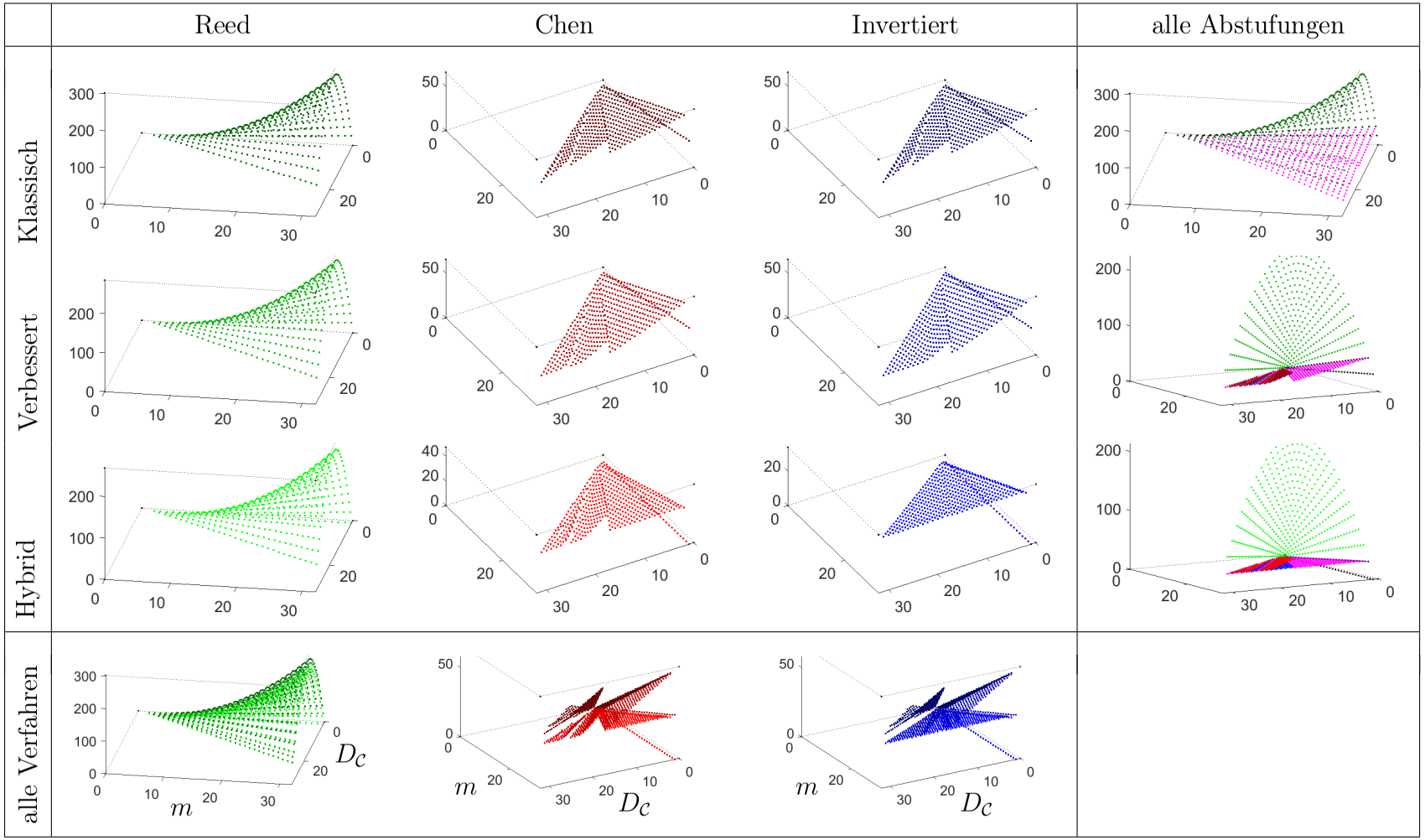


Abbildung 6.3: Verhältnisse der Abstufungen zu Chens Abstufung bei festem Decodierverfahren jeweils hinsichtlich der Gesamtzahl der benötigten Mehrheitsentscheidungen für $q = 2$

Abbildung 6.4: Zweierlogarithmus der Gesamtzahl der benötigten Mehrheitsentscheidungen für $q = 2$



Kapitel 7

Euklidische-Geometrie-Codes

In diesem Kapitel widmen wir uns bekannten Klassen der Euklidische-Geometrie-Codes, die gemeinsam haben, dass sich ihre Strukturen über affine Räume definieren, so dass die im vorherigen Kapitel präsentierten Decodierverfahren angewandt werden können. Diese Codes gehören zu den (zyklischen) Polynomcodes über $\mathbb{F}_{q_\mathcal{C}}$ [18].

Die im folgenden Abschnitt verwendete Notation wird für das gesamte Kapitel gültig sein.

7.1 Euklidische-Geometrie-Codes und die allgemeine Beschreibung ihrer Dualcodes

Sei wie in Kapitel 5 q eine Potenz von $q_\mathcal{C}$, $m \in \mathbb{N}$, $m \geq 2$, $n := q^m - 1$ sowie

$$\mathbb{F}_q^m := \{\mathbf{w}_0, \mathbf{w}_1, \ldots, \mathbf{w}_{n-1}, 0\}$$

und

$$\begin{aligned} \mathcal{P}\left(\mathbb{F}_q^m\right) &\longrightarrow \{0,1\}^n \\ V &\mapsto \chi_V := \sum_{i \in \mathbb{Z}_n : \mathbf{w}_i \in V} \mathbf{e}_i. \end{aligned}$$

Wir erinnern daran, dass die Abbildung nicht injektiv ist. Die leere Menge und der Nullvektorraum werden beide auf den Nullvektor der Länge n abgebildet. Gegeben $r \in \mathbb{N}_0$ mit $r < m$ und $\alpha, \beta \in \mathbb{F}_{q_\mathcal{C}}$. Wir definieren

$$\mathcal{C}_{r,m,q,q_\mathcal{C},\alpha,\beta} \coloneqq \left\langle \alpha\chi_{\mathbf{v}+U} + \beta\chi_{\mathbf{w}+U} \;\middle|\; \begin{array}{l} U \leq \mathbb{F}_q^m, \dim U = r, \\ \mathbf{v}, \mathbf{w}, \mathbf{v}-\mathbf{w} \in \mathbb{F}_q^m \backslash U \end{array} \right\rangle_{\mathbb{F}_{q_\mathcal{C}}}.$$

Wir zeigen in diesem Abschnitt, dass $\mathcal{C}_{r,m,q,q_\mathcal{C},\alpha,\beta}$ für viele Paare (α, β) mit

$$\mathcal{C}_{r,m,q,q_\mathcal{C}} \coloneqq \left\langle \chi_{\mathbf{v}+U} \mid U \leq \mathbb{F}_q^m, \dim U = r, \mathbf{v} \in \mathbb{F}_q^m \backslash U \right\rangle_{\mathbb{F}_{q_\mathcal{C}}}$$

oder

$$\bar{\mathcal{C}}_{r,m,q,q_\mathcal{C}} \coloneqq \left\langle \chi_{\mathbf{v}+U} - \chi_{\mathbf{w}+U} \mid U \leq \mathbb{F}_q^m, \dim U = r, \mathbf{v}, \mathbf{w}, \mathbf{v}-\mathbf{w} \in \mathbb{F}_q^m \backslash U \right\rangle_{\mathbb{F}_{q_\mathcal{C}}}.$$

zusammenfällt. Alle anderen Fälle sind für unsere Zwecke uninteressant.

In den folgenden Abschnitten des Kapitels erklären wir, wie die Decodierverfahren aus Kapitel 5 für die Dualcodes von $\mathcal{C}_{r,m,q,q_\mathcal{C}}$ oder $\bar{\mathcal{C}}_{r,m,q,q_\mathcal{C}}$ eingesetzt werden können. Zu diesen Dualcodes und ihren Erweiterungen gehören der (reguläre) Euklidische-Geometrie-Code [18], [16], [7], [29, §10.2], [4, §13.8], der zweifache Euklidische-Geometrie-Code [21], [22, §8.7], [23], der binäre Reed-Muller-Code [29, §5.5], [22, §4.3], [24, Ch. 13.], [40, S. 21 ff.] und der binäre Hamming-Code [22, §4.1], [24, Ch. 1. §7.], [29, §5.1], [39, Beispiele 1.2.9], [4, §1.5].

Am Ende des Kapitels werden wir demonstriert haben, dass für alle nichttrivialen Dualcodes von $\mathcal{C}_{r,m,q,q_\mathcal{C},\alpha,\beta}$ mit Minimaldistanz von mindestens drei die Decodierverfahren aus Kapitel 5 eingesetzt werden können.

Proposition 7.1.1. *Es ist*

$$\mathcal{C}_{r,m,q,q_\mathcal{C},\alpha,\beta} = \begin{cases} \{0\} & \alpha = -\beta = 0 \text{ oder } q^{m-r} = 2, \\ \bar{\mathcal{C}}_{r,m,q,q_\mathcal{C}} & \alpha = -\beta \neq 0, q^{m-r} > 2, \\ \mathcal{C}_{m-1,m,3,3,1,1} & \alpha = \beta \neq -\beta, q^{m-r} = 3, \\ \mathcal{C}_{r,m,q,q_\mathcal{C}} & \text{sonst} \end{cases}.$$

Beweis. Sei $U \leq \mathbb{F}_q^m$ mit $\dim U = r$ beliebig. Es bezeichne $U' \leq \mathbb{F}_q^m$ einen Komplementärraum zu U, so dass $U \oplus U' = \mathbb{F}_q^m$. Dieser enthält q^{m-r} Elemente. Weiterhin sei $\mathbf{v}_0 \in \mathbb{F}_q^m \backslash U$ beliebig. Es existiert $0 \neq \mathbf{v}_0' \in U'$, so dass

$$\mathbf{v}_0 + U = \mathbf{v}_0' + U.$$

1. Ganz augenscheinlich ist

$$\mathcal{C}_{r,m,q,q_{\mathcal{C}},0,0} = \{0\}\,.$$

2. Falls $q^{m-r} = 2$ (d.h. $q = 2$, $m - r = 1$), enthält $U^{'}$ neben dem Nullvektor nur den Vektor $\mathbf{v}_0^{'}$. Für zwei beliebige Vektoren $\mathbf{v}, \mathbf{w} \in \mathbb{F}_q^m \backslash U$ gilt $\mathbf{v}, \mathbf{w} \in \mathbf{v}_0^{'} + U$, so dass der Differenzvektor $\mathbf{v} - \mathbf{w}$ in U enthalten ist. Damit ist $\mathcal{C}_{r,m,q,q_{\mathcal{C}},\alpha,\beta}$ der Nullraum.

3. Gehen wir also davon aus, dass $U^{'}$ mindestens drei Vektoren enthält. Dann ist $\mathcal{C}_{r,m,q,q_{\mathcal{C}},\alpha,\beta} > \{0\}$. Offensichtlich ist

$$\mathcal{C}_{r,m,q,q_{\mathcal{C}},\alpha,\beta} \leq \mathcal{C}_{r,m,q,q_{\mathcal{C}}}.$$

   (a) Falls $0 \neq \alpha = -\beta$, ist α in $\mathbb{F}_{q_{\mathcal{C}}}^*$ invertierbar und damit,

$$\mathcal{C}_{r,m,q,q_{\mathcal{C}},\alpha,\beta} = \mathcal{C}_{r,m,q,q_{\mathcal{C}},\alpha,-\alpha} = \bar{\mathcal{C}}_{r,m,q,q_{\mathcal{C}}}.$$

   (b) Angenommen, $0 \neq \alpha = \beta \neq -\beta$ und $q^{m-r} > 3$. Dann ist $p \neq 2$ ungerade. Neben $\mathbf{v}_0^{'}$ enthält $U^{'}$ mindestens zwei weitere (voneinander verschiedene) Vektoren ungleich null, sagen wir $\mathbf{v}_1^{'}$, $\mathbf{v}_2^{'}$. Die Differenzvektoren $\mathbf{v}_i^{'} - \mathbf{v}_j^{'}$, $0 \leq i < j \leq 2$ sind nicht in U enthalten. Dann ist

$$\begin{aligned}\chi_{\mathbf{v}_0^{'}+U} &= (p+1)/2 \cdot \alpha^{-1} \cdot \left(\alpha\chi_{\mathbf{v}_0^{'}+U} + \alpha\chi_{\mathbf{v}_1^{'}+U}\right) \\ &\quad + (p-1)(p+1)/2 \cdot \alpha^{-1} \cdot \left(\alpha\chi_{\mathbf{v}_1^{'}+U} + \alpha\chi_{\mathbf{v}_2^{'}+U}\right) \\ &\quad + (p+1)/2 \cdot \alpha^{-1} \cdot \left(\alpha\chi_{\mathbf{v}_0^{'}+U} + \alpha\chi_{\mathbf{v}_2^{'}+U}\right) \\ &\in \mathcal{C}_{r,m,q,q_{\mathcal{C}},\alpha,\alpha} = \mathcal{C}_{r,m,q,q_{\mathcal{C}},\alpha,\beta}.\end{aligned}$$

   (c) Falls $0 \neq \alpha = \beta \neq -\beta$ und $q^{m-r} = 3$, ist $q = 3$, $r = m - 1$, α in $\mathbb{F}_{q_{\mathcal{C}}}^*$ invertierbar und damit

$$\mathcal{C}_{r,m,q,q_{\mathcal{C}},\alpha,\beta} = \mathcal{C}_{m-1,m,3,3,1,1}.$$

   (d) Angenommen, $\alpha \neq \pm\beta$.

      i. Falls $\alpha = 0$, ist β in $\mathbb{F}_{q_{\mathcal{C}}}^*$ invertierbar und damit

$$\mathcal{C}_{r,m,q,q_{\mathcal{C}},\alpha,\beta} = \mathcal{C}_{r,m,q,q_{\mathcal{C}},0,\beta} = \mathcal{C}_{r,m,q,q_{\mathcal{C}}}.$$

ii. Äquivalent: Falls $\beta = 0$, ist α in $\mathbb{F}_{q_\mathcal{C}}^*$ invertierbar und

$$\mathcal{C}_{r,m,q,q_\mathcal{C},\alpha,\beta} = \mathcal{C}_{r,m,q,q_\mathcal{C},q_\mathcal{C},\alpha,0} = \mathcal{C}_{r,m,q,q_\mathcal{C}}.$$

iii. Angenommen, $\alpha, \beta \neq 0$. Da

$$\alpha \neq \pm\beta$$

ist

$$\alpha^2 \neq \beta^2$$

und

$$\left(\alpha^{-1}\beta - \alpha\beta^{-1}\right)$$

invertierbar in $\mathbb{F}_{q_\mathcal{C}}^*$. Neben $\mathbf{v}_0^{'}$ enthält $U^{'}$ mindestens einen weiteren Vektor ungleich null, sagen wir $\mathbf{v}_1^{'}$. Der Differenzvektor $\mathbf{v}_0^{'} - \mathbf{v}_1^{'}$ ist nicht in U enthalten. Dann ist

$$\begin{aligned}
\chi_{\mathbf{v}_0^{'}+U} &= \left(\alpha^{-1}\beta - \alpha\beta^{-1}\right)^{-1}(\alpha^{-1}\beta - \alpha\beta^{-1})\chi_{\mathbf{v}_0^{'}+U} \\
&= \left(\alpha^{-1}\beta - \alpha\beta^{-1}\right)^{-1}\alpha^{-1}\left(\alpha\chi_{\mathbf{v}_1^{'}+U} + \beta\chi_{\mathbf{v}_0^{'}+U}\right) \\
&- \left(\alpha^{-1}\beta - \alpha\beta^{-1}\right)^{-1}\beta^{-1}\left(\alpha\chi_{\mathbf{v}_0^{'}+U} + \beta\chi_{\mathbf{v}_1^{'}+U}\right) \\
&\in \mathcal{C}_{r,m,q,q_\mathcal{C},\alpha,\beta}.
\end{aligned}$$

□

Der Code $\mathcal{C}_{m-1,m,3,3,1,1}$ ist für unsere Zwecke uninteressant, da die Minimaldistanz seines Dualcodes zwei beträgt und somit keine Fehler korrigiert werden können.

Proposition 7.1.2. *Die Minimaldistanz des Dualcodes von $\mathcal{C}_{m-1,m,3,3,1,1}$ beträgt zwei.*

Beweis. Nach Definition ist

$$\begin{aligned}
\mathcal{C}_{m-1,m,3,3,1,1} &\coloneqq \left\langle \chi_{\mathbf{v}+U} + \chi_{\mathbf{w}+U} \;\middle|\; \begin{array}{c} U \leq \mathbb{F}_3^m, \dim U = m-1 \\ \mathbf{v}, \mathbf{w}, \mathbf{v}-\mathbf{w} \in \mathbb{F}_3^m \backslash U \end{array} \right\rangle_{\mathbb{F}_3} \\
&= \left\langle \chi_{\mathbf{v}+U} + \chi_{2\cdot\mathbf{v}+U} \;\middle|\; \begin{array}{c} \mathbf{v} \in \mathbb{F}_3^m, U \leq \mathbb{F}_3^m, \dim U = m-1, \\ \langle \mathbf{v} \rangle_{\mathbb{F}_3} \oplus U = \mathbb{F}_3^m \end{array} \right\rangle_{\mathbb{F}_3} \\
&= \left\langle (1,\ldots,1) - \chi_{U\backslash\{0\}} \;\middle|\; U \leq \mathbb{F}_3^m, \dim U = m-1 \right\rangle_{\mathbb{F}_3}.
\end{aligned}$$

Sei $\mathbf{u} \in \mathbb{F}_3^m$ ein beliebiger Vektor ungleich null. Dann ist $\chi_{\{\mathbf{u}\}} - \chi_{\{2\cdot\mathbf{u}\}}$ ein Codewort aus ${\mathcal{C}_{m-1,m,3,3,1,1}}^{\perp}$ mit Gewicht zwei, denn

$$\big(\chi_{\{\mathbf{u}\}} - \chi_{\{2\cdot\mathbf{u}\}}\big) \circ \big((1,\ldots,1) - \chi_{U\setminus\{0\}}\big) = 0$$

für alle $U \leq \mathbb{F}_3^m$ mit $\dim U = m-1$. Dies liegt darin begründet, dass jeder $(m-1)$-dimensionale Unterraum des $\mathbb{F}_3^m$, der $\mathbf{u}$ enthält, auch $2\cdot\mathbf{u}$ enthält, und umgekehrt. □

Proposition 7.1.1 und Proposition 7.1.2 zeigen, dass es ausreicht, sich mit $\mathcal{C}_{r,m,q,q_\mathcal{C}}$ und $\bar{\mathcal{C}}_{r,m,q,q_\mathcal{C}}$ zu befassen, um $\mathcal{C}_{r,m,q,q_\mathcal{C},\alpha,\beta}$ umfassend abzuhandeln. Bemerkenswert an dieser Stelle ist, dass für $r \leq m-2$

$$\mathcal{C}_{r+1,m,2,2} = \bar{\mathcal{C}}_{r,m,2,2}$$

ist.

Proposition 7.1.3. *Falls $q = 2$, $r+1 < m$, ist*

$$\mathcal{C}_{r+1,m,2,2} = \bar{\mathcal{C}}_{r,m,2,2}.$$

Beweis. Sei $U \leq \mathbb{F}_2^m$ mit $\dim U = r$ beliebig. Weiterhin seien $\mathbf{v}, \mathbf{w} \in \mathbb{F}_2^m \setminus U$ mit $\mathbf{v} - \mathbf{w} \notin U$ beliebig. (Die Existenz solcher Vektoren ist gesichert durch $r \leq m-2$.) Dann ist

$$(\mathbf{v}+U)\,\dot\cup\,(\mathbf{w}+U) = \mathbf{v} + \langle \mathbf{w}+\mathbf{v}\rangle_{\mathbb{F}_2} \oplus U$$

ein $(r+1)$-dimensionaler affiner Unterraum von $\mathbb{F}_2^m$, wobei

$$\mathbf{v} \notin \langle \mathbf{w}+\mathbf{v}\rangle_{\mathbb{F}_2} \oplus U.$$

Also

$$\begin{aligned}\chi_{\mathbf{v}+U} - \chi_{\mathbf{w}+U} &= \chi_{\mathbf{v}+U} + \chi_{\mathbf{w}+U} \\ &= \chi_{(\mathbf{v}+U)\dot\cup(\mathbf{w}+U)} \in \mathcal{C}_{r+1,m,2,2}.\end{aligned}$$

Andererseits gibt es für beliebige $\mathbf{v} \in \mathbb{F}_2^m \setminus U$, $U \leq \mathbb{F}_2^m$ mit $\dim U = r+1$ einen Unterraum W von U und einen Vektor $\mathbf{w} \in U$, so dass

$$U = \langle \mathbf{w}\rangle_{\mathbb{F}_2} \oplus W = (\mathbf{w}+W)\,\dot\cup W.$$

Es ist weder $\mathbf{v}$ noch $\mathbf{v}+\mathbf{w}$ ein Element von W. Somit

$$\begin{aligned}\chi_{\mathbf{v}+U} &= \chi_{(\mathbf{v}+\mathbf{w}+W)\dot\cup(\mathbf{v}+W)} \\ &= \chi_{\mathbf{v}+\mathbf{w}+W} - \chi_{\mathbf{v}+W} \in \bar{\mathcal{C}}_{r,m,2,2}\end{aligned}$$

□

7.2 (Reguläre) Euklidische-Geometrie-Codes

In diesem Abschnitt widmen wir uns den (zyklischen) Euklidische-Geometrie-Codes, [18], [16], [7], [29, §10.2]„ [4, §13.8]. Diese werden in zwei Typen unterschieden, je nachdem, ob 1 eine Nullstelle des Erzeugerpolynoms ist (Typ-0) oder nicht (Typ-1), vgl. [16], [21]. Dies hat Auswirkungen auf die Dimension des Codes, die beim Typ-0-Code geringer ist, auf den Decodieraufwand und mitunter auf die Anzahl der korrigierbaren Fehler, die beim Typ-0-Code höher liegen kann, stets im Vergleich zum korrespondierenden Typ-1-Code.

Sei $q_{\mathcal{C}} \coloneqq p$ prim. Weiterhin sei γ ein multiplikatives Erzeugendes von $\mathbb{F}_{q^m}^*$ und $r+1 < m$.

7.2.1 Zyklische Typ-1-EG(m,q)-Codes

Wir erinnern uns an die Definition von $\mathcal{C}_{r+1,m,q,p}$,

$$\mathcal{C}_{r+1,m,q,p} \coloneqq \left\langle \chi_{\mathbf{v}+U} \mid U \leq \mathbb{F}_q^m, \dim U = r+1, \mathbf{v} \in \mathbb{F}_q^m \backslash U \right\rangle_{\mathbb{F}_p}.$$

Der Code $\mathcal{C}_{r+1,m,q,p}$ ist äquivalent zum sogenannten (zyklischen) *primitiven Polynomcode* über F_p der Ordnung $(m-r-1)(q-1)$ [18], dessen Erzeugerpolynom um den Faktor $(x-1)$ erweitert wird, [16, Theorem 6]. Genauer gesagt besitzt das Erzeugerpolynom (ausschließlich) die Nullstellen γ^i, $i \in I$ mit

$$I \coloneqq \left\{ i \in \mathbb{Z}_n \;\middle|\; 0 \leq \min_{0 \leq j < \log_p(q)} \omega_q\left(i \cdot p^j \bmod n\right) < (r+1)(q-1) \right\},$$

[18, Theorem 6], [16, Theorem 1][1], [29, Theorem 10.14][2].

[1] Ungleichung (10) in [16, Theorem 1] weist einen Fehler auf. Der letzte Teil der Ungleichung muss strikt sein, vgl. [18, Theorem 6].

[2] Beachte den Fehler in [29, Theorem 10.14]. Es muss heißen

$$0 < j < \left\lceil \frac{m(q^s-1)}{b} \right\rceil.$$

statt

$$0 < j < \left\lceil \frac{m(q-1)}{b} \right\rceil.$$

Es sei $\mathcal{C}$ der Dualcode von $\mathcal{C}_{r+1,m,q,p}$,

$$\mathcal{C} \coloneqq \mathcal{C}_{r+1,m,q,p}^{\perp}.$$

Der Code $\mathcal{C}$ ist äquivalent zum sogenannten (zyklischen) *Typ-1-EG*(m,q)-*Code*[3] [4] der Ordnung r, [16, Theorem 6][5]. Dessen Erzeugerpolynom besitzt (ausschließlich) die Nullstellen γ^i, $i \in \bar{I}$ mit

$$\bar{I} \coloneqq \left\{ i \in \mathbb{Z}_n \;\middle|\; 0 < \max_{0 \le j < \log_p(q)} \omega_q\left(i \cdot p^j \bmod n\right) \le (m-r-1)(q-1) \right\},$$

[16, Theorem 6]. Eine untere Schranke für die Minimaldistanz d von $\mathcal{C}$ liefert die BCH-Schranke [39, Satz 6.2.1],

$$d \ge q^{m-r-1} + p \cdot q^{m-r-2} - 1 \qquad \text{[29, (10.16)].}$$

Also ist $\mathcal{C}$ ein $\tau_{\max}$-fehlerkorrigierender Code mit

$$\tau_{\max} \ge \left\lfloor \frac{q^{m-r-1} + p \cdot q^{m-r-2}}{2} \right\rfloor - 1 =: \tau_{\mathrm{BCH}}. \qquad [7.1]$$

Anzahl der korrigierbaren Fehler und Decodieraufwand

Alle in Kapitel 5 vorgestellten Decodierverfahren unter jeder der drei präsentierten Abstufungen mit $D_{\mathcal{C}} \coloneqq r+1$ und

$$\eta \coloneqq 2 \cdot \left\lfloor \frac{q^{m+1-D_{\mathcal{C}}} - q}{2(q-1)} \right\rfloor = \frac{q^{m-r} - q}{q-1} - \begin{cases} 1 & p \neq 2, m-r \text{ gerade,} \\ 0 & \text{sonst} \end{cases}$$

können für den Code $\mathcal{C}$ angewandt werden. Unter Reeds Abstufung bedarf es $r+1$, unter den beiden anderen Abstufungen jeweils $1 + \lceil \log_2(m/m-r) \rceil$ Majority-Logic-Stufen. Wie wir in Abschnitt 6.2 gesehen haben, verläuft die Decodierung erfolgreich, wenn nicht mehr als

$$\tau_{\mathrm{MLG}} \coloneqq \left\lfloor \frac{q^{m-r} - q}{2(q-1)} \right\rfloor \qquad [7.2]$$

[3] EG ist die Abkürzung für Euklidische Geometrie.

[4] S. Lin nennt diese Codeklasse *regular EG codes* [21], vermutlich, um zu verdeutlichen, dass sie sich von seinen *multifold EG codes*, die auch auf Euklidischen Geometrien basieren, abheben.

[5] Kasami und Lin bezeichnen den Dualcode des primitiven Polynomcode über $\mathbb{F}_p$ der Ordnung $(m-r-1)(q-1)$ als zyklischen Typ-1-EG(m,q)-Code der Ordnung $r+1$ (und nicht der Ordnung r), verweisen jedoch selbst auf [29].

Fehler auftreten. Der konkrete Aufwand entspricht in Abhängigkeit der gewählten Abstufung bei der klassischen Decodierung den Werten $\mathcal{K}_{\text{E}}$ und $\mathcal{K}_{\mu}$ aus Term 5.1 und Term 5.2, bei der verbesserten Decodierung den Werten $\mathcal{V}_{\text{E}}$ und $\mathcal{V}_{\mu}$ aus Term 5.3 und Term 5.4 und bei der hybriden Decodierung mit

$$\dot{D} \coloneqq \begin{cases} m-r-1 & p \neq 2, m-r \text{ gerade,} \\ m-r & \text{sonst} \end{cases}$$

den Werten $\mathcal{H}_{\mu}$, $\mathcal{H}_{\pm}$ und $\mathcal{H}_{\text{E}}$ aus Term 5.18, Term 5.19 und Term 5.20.

In Tabelle 7.1 listen wir für einige (zyklische) Typ-1-EG(m, q)-Codes die Parameter Länge n, k, τ_{BCH} und τ_{MLG} sowie den Aufwand sowohl der verbesserten als auch der hybriden Decodierung unter der invertierten Abstufung. Der zweite bzw. der dritte Block der Tabelle 7.1 geben Aufschluss über die sequenzielle bzw. parallele Laufzeit.

Tabelle 7.1: Parameter und Decodieraufwand unter der invertierten Abstufung beim (zyklischen) Typ-1-EG(m, q)-Code der Ordnung r.

n	k	$\frac{k}{n}$	τ_{BCH}	τ_{MLG}	$\mathcal{V}_{\mu,I}$	$\mathcal{H}_{\mu,I}$	$\mathcal{H}_{\pm,I}$	$\mathcal{H}_{\text{E},I} = \mathcal{V}_{\text{E},I}$	ς	$\lvert T_I \rvert$	η	q	r
3	1	0,33	1	1	3	1	2	3	1	1	2	2	0
7	1	0,14	3	3	7	1	6	21	1	1	6	2	0
7	4	0,57	1	1	16	5	11	9	2	2	2	2	1
15	5	0,33	3	3	64	9	55	147	2	2	6	2	1
15	11	0,73	1	1	24	18	6	9	2	1	2	2	2
15	7	0,47	2	2	15	11	12	15	1	1	4	4	0
31	6	0,19	7	7	256	17	239	1.575	2	2	14	2	1
31	16	0,52	3	3	80	38	42	147	2	1	6	2	2
31	26	0,84	1	1	79	46	33	27	3	2	2	2	3
63	7	0,11	15	15	1.024	33	991	14.415	2	2	30	2	1
63	22	0,35	7	7	288	78	210	1.575	2	1	14	2	2
63	42	0,67	3	3	112	70	42	147	2	1	6	2	3
63	57	0,9	1	1	111	78	33	27	3	2	2	2	4
63	13	0,21	11	10	63	43	60	315	1	1	20	4	0
63	48	0,76	2	2	138	99	117	75	2	2	4	4	1
63	37	0,59	4	4	63	55	56	63	1	1	8	8	0

Fortsetzung der Tabelle auf der nächsten Seite

Tabelle 7.1 – Fortsetzung der Tabelle

n	k	$\frac{k}{n}$	τ_{BCH}	τ_{MLG}	$\mathcal{V}_{\mu,I}$	$\mathcal{H}_{\mu,I}$	$\mathcal{H}_{\pm,I}$	$\mathcal{H}_{\mathbb{E},I}=\mathcal{V}_{\mathbb{E},I}$	ς	$\lvert T_I\rvert$	η	q	r
127	29	0,23	15	15	1.088	158	930	14.415	2	1	30	2	2
127	64	0,5	7	7	352	142	210	1.575	2	1	14	2	3
127	99	0,78	3	3	575	190	385	1.029	3	2	6	2	4
127	120	0,94	1	1	199	181	18	27	3	1	2	2	5
255	37	0,15	31	31	4.224	318	3.906	123.039	2	1	62	2	2
255	93	0,36	15	15	1.216	286	930	14.415	2	1	30	2	3
255	163	0,64	7	7	480	270	210	1.575	2	1	14	2	4
255	219	0,86	3	3	703	318	385	1.029	3	2	6	2	5
255	247	0,97	1	1	327	309	18	27	3	1	2	2	6
255	127	0,5	11	10	1.578	1.075	1.509	6.615	2	2	20	4	1
255	231	0,91	2	2	330	310	60	75	2	1	4	4	2
255	175	0,69	8	8	255	239	240	255	1	1	16	16	0
511	130	0,25	31	31	4.480	574	3.906	123.039	2	1	62	2	3
511	256	0,5	15	15	1.472	542	930	14.415	2	1	30	2	4
511	382	0,75	7	7	4.351	766	3.585	23.625	3	2	14	2	5
511	466	0,91	3	3	1.071	777	294	1.029	3	1	6	2	6
511	502	0,98	1	1	748	649	99	81	4	2	2	2	7
511	139	0,27	39	36	511	439	504	4.599	1	1	72	8	0
511	448	0,88	4	4	1.078	935	1.001	567	2	2	8	8	1

7.2.2 Zyklische Typ-0-EG(m,q)-Codes

Wir betrachten den Code, der erzeugt wird durch die Inzidenzvektoren zu allen affinen Räumen der Dimension $r+1$, einschließlich aller $(r+1)$-dimensionalen Unterräume des $\mathbb{F}_q^m$,

$$\left\langle \chi_{\mathbf{v}+U} \mid U \leq \mathbb{F}_q^m, \dim U = r+1, \mathbf{v} \in \mathbb{F}_q^m \right\rangle_{\mathbb{F}_p},$$

wobei $\chi_{\{0\}}$ der Nullvektor ist. Man sieht leicht, dass dieser Code aufgespannt wird durch $\mathcal{C}_{r+1,m,q,p}$ und dem nur aus Einsen bestehenden Vektor. Dazu muss

man sich nur vor Augen führen, dass für jeden $(r+1)$-dimensionalen Unterraum $U \leq \mathbb{F}_q^m$ mit einem Komplementärraum $U' \leq \mathbb{F}_q^m$ gilt,

$$\mathbb{F}_q^m = U \quad \dot\cup \quad \dot{\bigcup_{\substack{\mathbf{u}' \in U' \\ \mathbf{u}' \neq 0}}} \mathbf{u}' + U,$$

so dass

$$\chi_U = \chi_{\mathbb{F}_q^m} - \dot{\bigcup_{\substack{\mathbf{u}' \in U' \\ \mathbf{u}' \neq 0}}} \chi_{\mathbf{u}'+U}.$$

Dieser Code ist äquivalent zum (zyklischen) primitiven Polynomcode über $\mathbb{F}_p$ der Ordnung $(m-r-1)(q-1)$, [18], [16, Theorem 5]. Das Erzeugerpolynom besitzt (ausschließlich) die Nullstellen γ^i, $i \in I$ mit

$$I \coloneqq \left\{ i \in \mathbb{Z}_n \;\middle|\; 0 < \min_{0 \leq j < \log_p(q)} \omega_q\left(i \cdot p^j \bmod n\right) < (r+1)(q-1) \right\},$$

[18, Theorem 6], [16, Theorem 1][1], [29, Theorem 10.14][2].

Es sei $\mathcal{C}$ der Dualcode von $\langle (1, \ldots, 1) \rangle_{\mathbb{F}_p} \oplus \mathcal{C}_{r+1,m,q,p}$. Der Code $\mathcal{C}$ ist äquivalent zum sogenannten (zyklischen) *Typ-0-EG*(m,q)*-Code*der Ordnung r, [16, Theorem 5][6]. Dessen Erzeugerpolynom besitzt (ausschließlich) die Nullstellen γ^i, $i \in \bar{I}$ mit

$$\bar{I} \coloneqq \left\{ i \in \mathbb{Z}_n \;\middle|\; 0 \leq \max_{0 \leq j < \log_p(q)} \omega_q\left(i \cdot p^j \bmod n\right) \leq (m-r-1)(q-1) \right\},$$

[16, Theorem 2]. Eine untere Schranke für die Minimaldistanz d von $\mathcal{C}$ liefert die BCH-Schranke [39, Satz 6.2.1],

$$d \geq q^{m-r-1} + p \cdot q^{m-r-2} \qquad \text{[18, Theorem 17].}$$

Also ist $\mathcal{C}$ ein $\tau_{\max}$-fehlerkorrigierender Code mit

$$\tau_{\max} \geq \left\lfloor \frac{q^{m-r-1} + p \cdot q^{m-r-2} - 1}{2} \right\rfloor \eqqcolon \tau_{\mathrm{BCH}}. \qquad [7.3]$$

[6] Kasami und Lin bezeichnen den Dualcode des primitiven Polynomcode über $\mathbb{F}_p$ der Ordnung $(m-r-1)(q-1)$ in der früheren Publikation [17] noch als verallgemeinerten EG-Code der Ordnung $(m-r-1, \log_p q)$.

Da $\mathcal{C}$ ein Subcode des Typ-1-EG(m, q)-Codes ist, können Codewörter aus $\mathcal{C}$ selbstverständlich mit allen in Kapitel 5 vorgestellten Decodierverfahren unter jeder der drei präsentierten Abstufungen mit $D_{\mathcal{C}} \coloneqq r + 1$ und

$$\eta \coloneqq \frac{q^{m-r} - q}{q - 1} - \begin{cases} 1 & p \neq 2, m - r \text{ gerade}, \\ 0 & \text{sonst} \end{cases}$$

rekonstruiert werden, sofern nicht mehr als

$$\left\lfloor \frac{q^{m-r} - q}{2(q-1)} \right\rfloor$$

Fehler auftreten. Darüber hinaus gibt es mitunter die Möglichkeit, den Aufwand zu reduzieren oder einen zusätzlichen Fehler zu korrigieren.

Anpassung der Decodierung auf den Typ-0-EG(m, q)-Code

Da nun im Gegensatz zum Typ-1-EG(m, q)-Code auch Inzidenzvektoren zu Unterräumen des $\mathbb{F}_q^m$ im Dualcode liegen und für die Decodierung herangezogen werden können, können wir die jeweils gestellte Forderung

$$\ell^*_{D_t, D_{t+1}, m, q} \geq \eta \geq 2\tau \qquad \text{für alle } t = \varsigma - 1, \varsigma - 2, \ldots, 0$$

respektive

$$\ell_{D_t, D_{t+1}, m, q} - 1 \geq \eta \geq 2\tau \qquad \text{für alle } t = \varsigma - 1, \varsigma - 2, \ldots, 0$$

dahingehend lockern, dass wir sie durch

$$\ell_{D_t, D_{t+1}, m, q} \geq \eta \geq 2\tau \qquad \text{für alle } t = \varsigma - 1, \varsigma - 2, \ldots, 0$$

ersetzen.

Die für die verbesserte und hybride Decodierung benötigten Decodierbäume können kleiner als beim Typ-1-EG(m, q)-Code konstruiert werden, indem jeder Elternknoten statt $\eta + 1$ nur noch η Kindknoten besitzt. Beim Decodieren müssen allerdings nicht nur die Fehlersummen von echten affinen Räumen sondern auch von Unterräumen (bis auf den Nullraum) bestimmt werden. Dies führt zu einem veränderten Decodieraufwand gegenüber dem Typ-1-EG(m, q)-Code. Insbesondere führen wir für jeden zu betrachtenden Knoten, auf den wir

die Hybriddecodierung anwenden (vgl. Lemma 5.4.1), statt einer zusätzlichen Mehrheitsentscheidung $q-1$ weitere Additionen bzw. Subtraktionen aus.

Falls $p \neq 2$, $m-r$ gerade ist, ist auch $\ell_{r,r+1,m,q}$ gerade. Dann ist es möglich im Vergleich zum Typ-1-EG(m,q)-Code einen zusätzlichen Fehler zu korrigieren,

$$\tau_{\mathrm{MLG}} \coloneqq \frac{q^{m-r}-1}{2\,(q-1)}, \tag{7.4}$$

indem man

$$\eta \coloneqq \ell_{r,r+1,m,q} = \frac{q^{m-r}-q}{q-1} + 1 = 2 \cdot \tau_{\mathrm{MLG}}$$

wählt. Der Decodieraufwand erhöht sich. Zu beachten ist, dass sich in diesem Fall auch der für die Hybriddecodierung relevante Parameter $\dot{D}$ angepasst werden muss,

$$\dot{D} \coloneqq \left\lfloor \log_q((q-1)\cdot\eta+1) \right\rfloor = m-r.$$

Ist hingegen $p=2$ oder $m-r$ ungerade, so behalten wir die Anzahl der bei einer Mehrheitsentscheidung herangezogenen Werte η bei,

$$\begin{aligned} \eta &\coloneqq \ell_{r,r+1,m,q} - 1 = \frac{q^{m-r}-q}{q-1} = 2 \cdot \tau_{\mathrm{MLG}}, \\ \tau_{\mathrm{MLG}} &\coloneqq \frac{q^{m-r}-q}{2\,(q-1)}. \end{aligned} \tag{7.5}$$

Der Parameter $\dot{D}$ wird auf $m-r-1$ gesetzt,

$$\dot{D} \coloneqq \left\lfloor \log_q((q-1)\cdot\eta+1) \right\rfloor = m-r-1.$$

Der Decodieraufwand kann explizit angegeben werden,

$$\begin{aligned} \mathcal{V}_{\mathbf{e}}^{\mathrm{EG}-0} &\coloneqq \mathcal{H}_{\mathbf{e}}^{\mathrm{EG}-0} \coloneqq \eta^{\varsigma} \cdot q^{m-D_{\mathcal{C}}} \\ \mathcal{V}_{\mu}^{\mathrm{EG}-0} &\coloneqq -1 + \sum_{t=0}^{\varsigma-1} \eta^t \cdot q^{m-D_t} \\ \mathcal{H}_{\mu}^{\mathrm{EG}-0} &\coloneqq -\mathbb{1}_{0\notin T} + \sum_{0\leq t\leq \varsigma-1} \eta^t \cdot \begin{cases} q^{m-D_t} & t \notin T, \\ \mathcal{H}_{\mu}\left(D_t, \dot{D}\right) & t \in T \end{cases}, \\ \mathcal{H}_{\pm}^{\mathrm{EG}-0} &\coloneqq \mathbb{1}_{0\in T} \cdot \mathcal{H}_{\pm}\left(0, \dot{D}\right) + \sum_{\substack{1\leq t\leq \varsigma-1 \\ t\in T}} \eta^t \cdot \left(\mathcal{H}_{\pm}\left(D_t, \dot{D}\right) + q - 1\right). \end{aligned}$$

In Tabelle 7.2 listen wir für einige (zyklische) Typ-0-EG(m, q)-Codes die Parameter Länge n, k, τ_{BCH} und τ_{MLG} sowie den Aufwand sowohl der verbesserten als auch der hybriden Decodierung unter der invertierten Abstufung. Wie zuvor geben der zweite bzw. der dritte Block der Tabelle 7.2 Aufschluss über die sequenzielle bzw. parallele Laufzeit.

Tabelle 7.2: Parameter und Decodieraufwand unter der invertierten Abstufung beim (zyklischen) Typ-0-EG(m, q)-Code der Ordnung r.

n	k	$\frac{k}{n}$	τ_{BCH}	τ_{MLG}	$\mathcal{V}_{\mu,I}$	$\mathcal{H}_{\mu,I}$	$\mathcal{H}_{\pm,I}$	$\mathcal{H}_{\mathbf{E},I} = \mathcal{V}_{\mathbf{E},I}$	ς	$\lvert T_I \rvert$	η	q	r
7	3	0,43	1	1	15	8	7	8	2	2	2	2	1
15	4	0,27	3	3	63	16	47	144	2	2	6	2	1
15	10	0,67	1	1	23	19	4	8	2	1	2	2	2
31	5	0,16	7	7	255	32	223	1.568	2	2	14	2	1
31	15	0,48	3	3	79	43	36	144	2	1	6	2	2
31	25	0,81	1	1	63	47	16	16	3	2	2	2	3
63	6	0,1	15	15	1.023	64	959	14.400	2	2	30	2	1
63	21	0,33	7	7	287	91	196	1.568	2	1	14	2	2
63	41	0,65	3	3	111	75	36	144	2	1	6	2	3
63	56	0,89	1	1	95	79	16	16	3	2	2	2	4
63	47	0,75	2	2	127	96	93	64	2	2	4	4	1
127	28	0,22	15	15	1.087	187	900	14.400	2	1	30	2	2
127	63	0,5	7	7	351	155	196	1.568	2	1	14	2	3
127	98	0,77	3	3	511	223	288	864	3	2	6	2	4
127	119	0,94	1	1	175	167	8	16	3	1	2	2	5
255	36	0,14	31	31	4.223	379	3.844	123.008	2	1	62	2	2
255	92	0,36	15	15	1.215	315	900	14.400	2	1	30	2	3
255	162	0,64	7	7	479	283	196	1.568	2	1	14	2	4
255	218	0,85	3	3	639	351	288	864	3	2	6	2	5
255	246	0,96	1	1	303	295	8	16	3	1	2	2	6
255	126	0,49	11	10	1.535	1.056	1.437	6.400	2	2	20	4	1
255	230	0,9	2	2	319	303	48	64	2	1	4	4	2
511	129	0,25	31	31	4.479	635	3.844	123.008	2	1	62	2	3
511	255	0,5	15	15	1.471	571	900	14.400	2	1	30	2	4

Fortsetzung der Tabelle auf der nächsten Seite

Tabelle 7.2 – Fortsetzung der Tabelle

n	k	$\frac{k}{n}$	τ_{BCH}	τ_{MLG}	$\mathcal{V}_{\mu,I}$	$\mathcal{H}_{\mu,I}$	$\mathcal{H}_{\pm,I}$	$\mathcal{H}_{\mathbf{E},I} = \mathcal{V}_{\mathbf{E},I}$	ς	$\lvert T_I \rvert$	η	q	r
511	381	0,75	7	7	4.095	959	3.136	21.952	3	2	14	2	5
511	465	0,91	3	3	991	775	216	864	3	1	6	2	6
511	501	0,98	1	1	639	607	32	32	4	2	2	2	7
511	447	0,87	4	4	1.023	896	889	512	2	2	8	8	1

7.3 Zweifache EG(m, q)-Codes

Lin veröffentlichte im Jahr 1973 mit den *Multifold Euclidean Geometry Codes* eine neue Klasse von Codes [21], [22, §8.7], [23]. Diese Klasse von Codes enthält die zuvor betrachteten (zyklischen) Euklidischen-Geometrie-Codes als Subklasse: Der sogenannte *q-fache EG(m, q)-Code* und der EG(m, q)-Code jeweils der gleichen Ordnung fallen zusammen [21, S. 540]. Wir wollen uns an dieser Stelle den zweifachen EG(m, q)-Codes zuwenden. Diese bilden Lin zufolge die interessanteste und effizienteste Subklasse [21, S. 541]. Lin geht sogar so weit zu behaupten, die zweifachen EG(m, q)-Codes seien die effizientesten mit Majority-Logic decodierbaren Codes, die bis dahin konstruiert worden sind. Dies begründet er unter anderem mit den ausgezeichneten Fehlerkorrektureigenschaften: Bis zur vom Code vorgegebenen Fehlerkorrekturgrenze, die mindestens so groß wie beim EG-Code ist, kann mit Majority-Logic decodiert werden.

Wie bei den EG-Codes lassen sich die zweifachen EG-Codes zwei Typen zuordnen, je nachdem, ob 1 eine Nullstelle des Erzeugerpolynoms ist (Typ-0) oder nicht (Typ-1).

Sei $q_{\mathcal{C}} \coloneqq p$. Weiterhin sei γ ein multiplikatives Erzeugendes von $\mathbb{F}_{q^m}^*$ und $q^{m-r} > 2$.

7.3.1 Zweifache Typ-1-EG(m, q)-Codes

Wir betrachten den Code $\bar{\mathcal{C}}_{r,m,q,q_\mathcal{C}}$, den wir wie folgt definiert hatten,

$$\bar{\mathcal{C}}_{r,m,q,q_\mathcal{C}} \coloneqq \left\langle \chi_{\mathbf{v}+U} - \chi_{\mathbf{w}+U} \mid U \leq \mathbb{F}_q^m, \dim U = r, \mathbf{v}, \mathbf{w}, \mathbf{v}-\mathbf{w} \in \mathbb{F}_q^m \backslash U \right\rangle_{\mathbb{F}_{q_\mathcal{C}}}.$$

Der Code $\bar{\mathcal{C}}_{r,m,q,q_\mathcal{C}}$ ist äquivalent zum (zyklischen) primitiven Polynomcode über $\mathbb{F}_p$ der Ordnung $(m-r)(q-1)-1$ [18], dessen Erzeugerpolynom um den Faktor $(x-1)$ erweitert wird, [21, Theorem 3, Theorem 12, §V. S.543]. Das Erzeugerpolynom dieses Codes besitzt also (ausschließlich) die Nullstellen γ^i, $i \in I$ mit

$$I \coloneqq \left\{ i \in \mathbb{Z}_n \;\middle|\; 0 \leq \min_{0 \leq j < \log_p(q)} \omega_q\left(i \cdot p^j \bmod n\right) < r(q-1)+1 \right\},$$

[18, Theorem 6, Corollary 7].

Es sei $\mathcal{C}$ der Dualcode von $\bar{\mathcal{C}}_{r,m,q,q_\mathcal{C}}$,

$$\mathcal{C} \coloneqq \bar{\mathcal{C}}^{\perp}_{r,m,q,q_\mathcal{C}}.$$

Der Code $\mathcal{C}$ ist äquivalent zum sogenannten *zweifachen Typ-1-EG(m, q)-Code* der Ordnung r (engl. Type-1 twofold $(r, \log_p(q))$th-order EG code), [21, Theorem 12]. Dessen Erzeugerpolynom besitzt (ausschließlich) die Nullstellen γ^i, $i \in \bar{I}$ mit

$$\bar{I} \coloneqq \left\{ i \in \mathbb{Z}_n \;\middle|\; 0 < \max_{0 \leq j < \log_p(q)} \omega_q\left(i \cdot p^j \bmod n\right) \leq (m-r)(q-1)-1 \right\},$$

[21, §IV. Theorem 3, §V. S.543]. Die Minimaldistanz d von $\mathcal{C}$ beträgt

$$d \coloneqq q^{m-r} - 1, \qquad [7.6]$$

siehe [21, §V.]. Also ist $\mathcal{C}$ ein $\tau_{\max}$-fehlerkorrigierender Code mit

$$\tau_{\max} \coloneqq \left\lfloor \frac{q^{m-r}-2}{2} \right\rfloor.$$

Decodierung und Anzahl der korrigierbaren Fehler

Lin schlägt eine Decodierung in $r+1$ Majority-Logic-Stufen vor [21, § III.], Zunächst treffe für jeden affinen Raum

$$A \coloneqq \mathbf{v} + U \in \mathcal{A}^*_{r,m,q},$$

die η-Mehrheitsentscheidung

$$\mu^0\left(\mathtt{z}\circ(\chi_A-\chi_{\mathbf{w}+A})\;\middle|\;\mathbf{w}\in U',\mathbf{w}\neq-\mathbf{v},\mathbf{w}\neq 0\right), \tag{7.7}$$

wobei $U' \leq \mathbb{F}_q^m$, $U \oplus U' = \mathbb{F}_q^m$ und $\eta \coloneqq q^{m-r}-2$. Werte das Ergebnis dieser Mehrheitsentscheidung als Fehlersumme von A, vgl. Proposition 4.3.1. Anschließend wird unter Reeds Abstufung beginnend bei $D_{\mathcal{C}} \coloneqq r$ in weiteren r Majority-Logic-Stufen klassisch decodiert, wie in Proposition 5.2.1 gesehen. Dieses Decodierverfahren ermöglicht die Korrektur von bis zu

$$\tau_{\mathrm{MLG}} \coloneqq \left\lfloor \frac{q^{m-r}-2}{2} \right\rfloor = \tau_{\max} \tag{7.8}$$

Fehlern, vgl. [21, § V., S. 543]. Lin legt in seiner Publikation [21] den Fokus auf die Eigenschaften der von ihm neu entwickelten *Multifold Euclidean Geometry Codes* und die Korrektheit des Decodierverfahrens. Er verweist nur kurz auf Chens Publikation [7], ohne die darin präsentierten Ergebnisse aufzugreifen:

> „Therefore, at most $(\mu+1)$ steps are required for decoding a $(d+1)$-fold (μ,s)th-order EG code (some decoding steps between the second step and the last may be saved [...])."

Wie können wir nun die von Lin vorgeschlagene Decodierung optimieren? Beginnen wir damit, dass wir η so setzen, dass es stets gerade ist,

$$\eta \coloneqq 2\cdot\tau_{\mathrm{MLG}} = \begin{cases} q^{m-r}-3 & p\neq 2, \\ q^{m-r}-2 & p=2 \end{cases}.$$

Der Effekt auf den Decodieraufwand ist zugegeben gering.

Wir sehen uns die erste Majority-Logic-Stufe an. Sei $U \leq \mathbb{F}_q^m$ ein r-dimensionaler Raum mit einem Komplementärraum $U' \leq \mathbb{F}_q^m$, so dass $U \oplus U' = \mathbb{F}_q^m$. Die Fehlersummen zu allen echten affinen Räumen $\mathbf{u}' + U$, $\mathbf{u}' \in U' \setminus \{0\}$ sind zu bestimmen.

Entweder wir treffen für jeden dieser echten affinen Räume eine Mehrheitsentscheidung wie von Lin vorgeschlagen, allerdings stets, auch im Fall $p \neq 2$, nur

auf einer geraden Anzahl von Werten basierend. Oder wir wenden Proposition 4.4.1 an und reduzieren damit die Gesamtzahl der Mehrheitsentscheidungen sowie der benötigten Checksummen mit Hilfe eines zusätzlichen Addition-Subtraktion-Schritts ähnlich wie bei der Hybriddecodierung.

Seien dafür $\mathbf{v}', \mathbf{w}' \in U' \setminus \{0\}$, $\mathbf{v}' \neq \mathbf{w}'$ beliebig, jedoch fest gewählt. Wir regen an, eine einzige η-Mehrheitsentscheidung zu treffen,

$$e \coloneqq \begin{cases} \mu^0\left(\mathtt{Z} \circ \chi_{\mathbf{u}'+U} \;\middle|\; \mathbf{u}' \in U' \setminus \{\mathbf{v}', \mathbf{w}', 0\}\right) & p \neq 2, \\ \mu^0\left(\mathtt{Z} \circ \chi_{\mathbf{u}'+U} \;\middle|\; \mathbf{u}' \in U' \setminus \{\mathbf{v}', 0\}\right) & p = 2 \end{cases}. \tag{7.9}$$

Aus Proposition 4.4.1 wissen wir: Sofern nicht mehr als τ_{MLG} Übertragungsfehler aufgetreten sind, entspricht die Fehlersumme von $\mathbf{u}' + U$, $\mathbf{u}' \in U' \setminus \{0\}$, der Checksumme von $\mathbf{u}' + U$ abzüglich e,

$$\begin{aligned} \mathtt{E} \circ \chi_{\mathbf{u}'+U} &= \mathtt{E} \circ \left(\chi_{\mathbf{u}'+U} - \chi_{\mathbf{v}'+U}\right) + \mathtt{E} \circ \chi_{\mathbf{v}'+U} \\ &= \mathtt{Z} \circ \left(\chi_{\mathbf{u}'+U} - \chi_{\mathbf{v}'+U}\right) + \mathtt{Z} \circ \chi_{\mathbf{v}'+U} - e \\ &= \mathtt{Z} \circ \chi_{\mathbf{u}'+U} - e. \end{aligned} \tag{7.10}$$

Die Fehlersummen dieser echten affinen Räume erhalten wir also mittels einer Mehrheitsentscheidung und anschließenden $q^{m-r} - 1$ Subtraktionen.

Natürlich ist es möglich, nach dieser ersten Majority-Logic-Stufe für die verbleibenden zwischen der klassischen, verbesserten und hybriden Decodierung unter Reeds, Chens und der invertierten Abstufung zu wählen. Dazu setzen wir $D_{\mathcal{C}} \coloneqq r+1$, wählen eine der drei Abstufungen aus und decodieren fast genauso wie in den entsprechenden Verfahen gesehen. Nur die im jeweiligen Verfahren verankerte erste Majority-Logic-Stufe wird durch die oben beschriebene auf den zweifachen Typ-1-EG(m,q)-Code zugeschnittene ersetzt.

Die Fehlerkorrekturgrenze von $\tau_{\mathrm{MLG}} = \tau_{\max}$ bleibt erhalten. Dies lässt sich in aller Kürze mit Korollar A.0.7 oder ausführlich mit der in den Beweisen von Proposition 6.2.2 und Proposition 6.2.3 angeführten Argumentation begründen.

Zu beachten ist: Fällt die Wahl auf die Hybriddecodierung, so ist im Gegensatz zum EG-Code $\dot{D}$ stets auf $m - r$ zu setzen, denn

$$\dot{D} \coloneqq \left\lfloor \log_q((q-1)(\eta+1)+1) \right\rfloor = m - r.$$

Beispiel zur Hybriddecodierung (einschließlich modifizierter erster Majority-Logic-Stufe)

Beispiel 7.3.1. Wir definieren $q \coloneqq 2$, $m \coloneqq 5$ und $r \coloneqq 2$. Sei γ ein multiplikatives Erzeugendes von

$$\mathbb{F}_{32}^{*} \cong \mathbb{F}_2[\gamma]/(\gamma^5 + \gamma^2 + 1)\mathbb{F}_2[\gamma].$$

Weiterhin sei

$$\mathbb{F}_2^5 \coloneqq \{\mathbf{w}_0, \mathbf{w}_1, \ldots, \mathbf{w}_{31} = 0\},$$

wobei

$$\mathbf{w}_i \coloneqq (\mathbf{w}_{i,0}, \mathbf{w}_{i,1}, \mathbf{w}_{i,2}, \mathbf{w}_{i,3}, \mathbf{w}_{i,4}) \quad \longleftrightarrow \quad \sum_{s=0}^{4} \mathbf{w}_{i,s} \cdot \gamma^s \cong \gamma^i,$$

$0 \leq i \leq 30$. Mit anderen Worten, die i-te Position eines Worts aus $\mathbb{F}_2^{31}$ korrespondiert zu γ^i, $0 \leq i \leq 30$.

Das Produkt zweier Elemente γ^i und γ^j, $0 \leq i < j \leq 30$, ist leicht zu berechnen, indem man i und j addiert und mod 31 reduziert. Die Summe dieser beiden Elemente ist gegeben durch

$$\gamma^i + \gamma^j \cong \sum_{s=0}^{4} ((\mathbf{w}_{i,s} + \mathbf{w}_{j,s}) \bmod 2) \cdot \gamma^s.$$

Um die Notation abzukürzen identifizieren wir jedes Element

$$\gamma^i \cong \sum_{j=0}^{4} \mathbf{w}_{i,j} \cdot \gamma^j,$$

eindeutig durch die korrespondierende Binärzahl

$$\sum_{j=0}^{4} \mathbf{w}_{i,j} \cdot 2^j \in \mathbb{Z}_{32} \setminus \{0\},$$

$0 \leq i \leq 30$. Es ergibt sich Tabelle 7.3.

Diese Notation hat den Vorteil, dass die Summe zweier Elemente leichter abgelesen werden kann. Dadurch wird die Decodierung im Gesamten anschaulicher.

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Binärzahl zu γ^i	2	4	8	16	5	10	20	13	26	17	7	14	28	29	31
i	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Binärzahl zu γ^i	27	19	3	6	12	24	21	15	30	25	23	11	22	9	18

Tabelle 7.3: Konversion zwischen Position im Codewort und repräsentierender Binärzahl

Wir betrachten den zweifachen Typ-1-EG$(5, 2)$-Code der Ordnung zwei. Dessen Erzeugerpolynom lautet

$$\begin{aligned} g(X) &\coloneqq \prod_{i\in\{1,2,3,4,5,6,8,9,10,12,16,17,18,20,24\}} (X-\gamma^i) \\ &= X^{15}+X^{11}+X^{10}+X^9+X^8+X^7+X^5+X^3+X^2+X^1+1. \end{aligned}$$

Angenommen, das Codewort

$$\mathsf{c} \coloneqq (111101011111000100000000000000000)$$

wurde übertragen und

$$\mathsf{z} \coloneqq (011101011111000000000000000000001)$$

mit Fehlern an den Positionen $\{0, 15, 30\}$ wurde empfangen. An den Positionen $\{1, 2, 3, 5, 7, 8, 9, 10, 11, 30\}$ korrespondierend zu den Elementen mit den Binärzahlen $\{2, 4, 8, 5, 20, 13, 26, 17, 7, 18\}$ ist das jeweilige Wortsymbol eins.

Chens Abstufung und die invertierte Abstufung bestehen beide aus $\varsigma = 2$ Stufen,

$$m = 5 > D_\varsigma = r+1 = 3 > D_{\varsigma-1} = 2 > D_0 = 0.$$

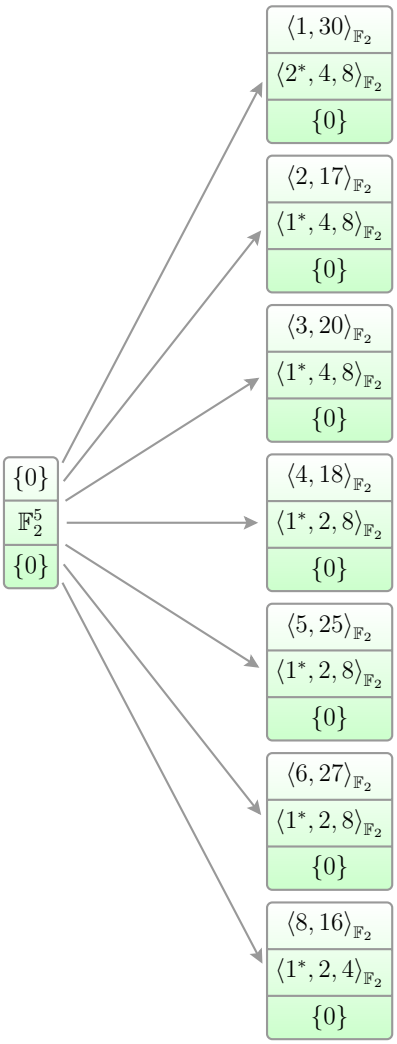
Es ist $\eta = 6$ und gemäß Theorem 5.4.4

$$\begin{aligned} T &= \{t \in \mathbb{N}_0, t \leq \varsigma - 1 \mid D_{t+1} = D_t + 1\} = \{2\}, \\ \dot{D} &= 3. \end{aligned}$$

Ähnlich wie in Theorem 5.4.4 skizziert, konstruieren wir einen Decodierbaum der Höhe $\varsigma - 1 = 1$. (Der Decodierbaum in Theorem 5.4.4 hat die Höhe ς, wir benötigen allerdings beim zweifachen EG-Code nur eine Höhe von $\varsigma - 1$.) Jeder Knoten repräsentiert ein geordnetes Tripel von Unterräumen, deren innere direkte Summe gerade $\mathbb{F}_2^5$ ist.

Die mit Sternchen markierten Zahlen repräsentieren jene Vektoren, die wir in Gleichung [7.9] auf Seite 131 mit $\mathbf{v}'$ bezeichnet haben. Sie bedingen nicht den Decodierbaum, jedoch das Decodierschema.

Von diesem Decodierbaum auf der rechten Seite ausgehend berechnen wir zunächst die benötigten $49 = 7 \cdot 7$ Checksummen. Mit Blick auf Gleichung [7.9] auf Seite 131 treffen wir sieben 6-Mehrheitsentscheidungen. Darauf aufbauend addieren wir wie in Gleichung [7.10] auf Seite 131 gesehen $49 = 7 \cdot 7$ Werte in $\mathbb{F}_2$. Auf diese Art erhalten wir Schätzwerte (SW) für die Fehlersummen zu den zweidimensionalen affinen Räumen.



$U_0 = \langle 1, 30\rangle_{\mathbb{F}_2}$	$\mathbf{u}' \in U' \setminus \{0\}$:	2	4	6	8	10	12	14	
	$\mathtt{Z} \circ \chi_{\mathbf{u}'+U_0}$:	1	1	1	1	1	0	1	
	μ^0 (		1	1	1	1	0	1	) = 1
	SW $\mathtt{E} \circ \chi_{\mathbf{u}'+U_0}$:	0	0	0	0	0	1	0	
$U_1 = \langle 2, 17\rangle_{\mathbb{F}_2}$	$\mathbf{u}' \in U' \setminus \{0\}$:	1	4	5	8	9	12	13	
	$\mathtt{Z} \circ \chi_{\mathbf{u}'+U_1}$:	1	1	1	1	1	0	0	
	μ^0 (		1	1	1	1	0	0	) = 1
	SW $\mathtt{E} \circ \chi_{\mathbf{u}'+U_1}$:	0	0	0	0	0	1	1	

$U_2 = \langle 3, 20\rangle_{\mathbb{F}_2}$	$\mathbf{u}' \in U' \setminus \{0\}$:	1	4	5	8	9	12	13	
	$\mathtt{Z} \circ \chi_{\mathbf{u}'+U_2}$:	0	0	1	1	0	0	0	
	μ^0 (		0	1	1	0	0	0	) = 0
	SW $\mathtt{E} \circ \chi_{\mathbf{u}'+U_2}$:	0	0	1	1	0	0	0	
$U_3 = \langle 4, 18\rangle_{\mathbb{F}_2}$	$\mathbf{u}' \in U' \setminus \{0\}$:	1	2	3	8	9	10	11	
	$\mathtt{Z} \circ \chi_{\mathbf{u}'+U_3}$:	1	0	0	0	1	0	0	
	μ^0 (		0	0	0	1	0	0	) = 0
	SW $\mathtt{E} \circ \chi_{\mathbf{u}'+U_3}$:	1	0	0	0	1	0	0	
$U_4 = \langle 5, 25\rangle_{\mathbb{F}_2}$	$\mathbf{u}' \in U' \setminus \{0\}$:	1	2	3	8	9	10	11	
	$\mathtt{Z} \circ \chi_{\mathbf{u}'+U_4}$:	1	0	1	0	0	0	1	
	μ^0 (		0	1	0	0	0	1	) = 0
	SW $\mathtt{E} \circ \chi_{\mathbf{u}'+U_4}$:	1	0	1	0	0	0	1	
$U_5 = \langle 6, 27\rangle_{\mathbb{F}_2}$	$\mathbf{u}' \in U' \setminus \{0\}$:	1	2	3	8	9	10	11	
	$\mathtt{Z} \circ \chi_{\mathbf{u}'+U_5}$:	0	0	1	1	1	1	1	
	μ^0 (		0	1	1	1	1	1	) = 1
	SW $\mathtt{E} \circ \chi_{\mathbf{u}'+U_5}$:	1	1	0	0	0	0	0	
$U_6 = \langle 8, 16\rangle_{\mathbb{F}_2}$	$\mathbf{u}' \in U' \setminus \{0\}$:	1	2	3	4	5	6	7	
	$\mathtt{Z} \circ \chi_{\mathbf{u}'+U_6}$:	1	1	0	0	0	0	1	
	μ^0 (		1	0	0	0	0	1	) = 0
	SW $\mathtt{E} \circ \chi_{\mathbf{u}'+U_6}$:	1	1	0	0	0	0	1	

Insgesamt berechnen wir $49 = 7 \cdot 7$ Checksummen, um anschließend mit Hilfe von sieben 6-Mehrheitsentscheidungen und $49 = 7 \cdot 7$ Additionen die Fehlersummen zu den zweidimensionalen affinen Räumen zu bestimmen. Die Fehlersymbole erhaltend wir mittels 31 Mehrheitsentscheidungen.

$$\text{SW } \mathtt{E}_0 \coloneqq \mu^0\left(\text{SW } \mathtt{E} \circ \chi_{1+U_i} \mid i \in \mathbb{Z}_7 \setminus \{0\}\right) = \mu^0\left(0,0,1,1,1,1\right) = 1$$
$$\text{SW } \mathtt{E}_1 \coloneqq \mu^0\left(\text{SW } \mathtt{E} \circ \chi_{2+U_i} \mid i \in \mathbb{Z}_7 \setminus \{1\}\right) = \mu^0\left(0,0,0,0,1,1\right) = 0$$
$$\text{SW } \mathtt{E}_2 \coloneqq \mu^0\left(\text{SW } \mathtt{E} \circ \chi_{4+U_i} \mid i \in \mathbb{Z}_7 \setminus \{3\}\right) = \mu^0\left(0,0,0,1,1,0\right) = 0$$
$$\text{SW } \mathtt{E}_3 \coloneqq \mu^0\left(\text{SW } \mathtt{E} \circ \chi_{8+U_i} \mid i \in \mathbb{Z}_7 \setminus \{6\}\right) = \mu^0\left(0,0,1,0,0,0\right) = 0$$
$$\text{SW } \mathtt{E}_4 \coloneqq \mu^0\left(\text{SW } \mathtt{E} \circ \chi_{16+U_i} \mid i \in \mathbb{Z}_7 \setminus \{6\}\right) = \mu^0\left(0,0,0,0,0,0\right) = 0$$
$$\text{SW } \mathtt{E}_5 \coloneqq \mu^0\left(\text{SW } \mathtt{E} \circ \chi_{5+U_i} \mid i \in \mathbb{Z}_7 \setminus \{4\}\right) = \mu^0\left(0,0,1,1,0,0\right) = 0$$

$$\begin{aligned}
\text{SW } \mathtt{E}_6 &\coloneqq \mu^0\left(\text{SW } \mathtt{E} \circ \chi_{10+U_i} \mid i \in \mathbb{Z}_7 \backslash \{6\}\right) = \mu^0(0,0,0,0,0,0) = 0\\
\text{SW } \mathtt{E}_7 &\coloneqq \mu^0\left(\text{SW } \mathtt{E} \circ \chi_{20+U_i} \mid i \in \mathbb{Z}_7 \backslash \{2\}\right) = \mu^0(0,0,0,0,0,0) = 0\\
\text{SW } \mathtt{E}_8 &\coloneqq \mu^0\left(\text{SW } \mathtt{E} \circ \chi_{13+U_i} \mid i \in \mathbb{Z}_7 \backslash \{6\}\right) = \mu^0(1,1,0,1,0,0) = 0\\
\text{SW } \mathtt{E}_9 &\coloneqq \mu^0\left(\text{SW } \mathtt{E} \circ \chi_{26+U_i} \mid i \in \mathbb{Z}_7 \backslash \{6\}\right) = \mu^0(0,0,0,0,1,1) = 0\\
\text{SW } \mathtt{E}_{10} &\coloneqq \mu^0\left(\text{SW } \mathtt{E} \circ \chi_{17+U_i} \mid i \in \mathbb{Z}_7 \backslash \{1\}\right) = \mu^0(0,1,0,0,0,1) = 0\\
\text{SW } \mathtt{E}_{11} &\coloneqq \mu^0\left(\text{SW } \mathtt{E} \circ \chi_{7+U_i} \mid i \in \mathbb{Z}_7 \backslash \{6\}\right) = \mu^0(0,0,0,0,0,1) = 0\\
\text{SW } \mathtt{E}_{12} &\coloneqq \mu^0\left(\text{SW } \mathtt{E} \circ \chi_{14+U_i} \mid i \in \mathbb{Z}_7 \backslash \{6\}\right) = \mu^0(0,1,0,0,1,0) = 0\\
\text{SW } \mathtt{E}_{13} &\coloneqq \mu^0\left(\text{SW } \mathtt{E} \circ \chi_{28+U_i} \mid i \in \mathbb{Z}_7 \backslash \{4\}\right) = \mu^0(0,1,1,0,1,0) = 0\\
\text{SW } \mathtt{E}_{14} &\coloneqq \mu^0\left(\text{SW } \mathtt{E} \circ \chi_{29+U_i} \mid i \in \mathbb{Z}_7 \backslash \{5\}\right) = \mu^0(0,1,0,0,1,0) = 0\\
\text{SW } \mathtt{E}_{15} &\coloneqq \mu^0\left(\text{SW } \mathtt{E} \circ \chi_{31+U_i} \mid i \in \mathbb{Z}_7 \backslash \{0\}\right) = \mu^0(1,1,1,1,1,1) = 1\\
\text{SW } \mathtt{E}_{16} &\coloneqq \mu^0\left(\text{SW } \mathtt{E} \circ \chi_{27+U_i} \mid i \in \mathbb{Z}_7 \backslash \{5\}\right) = \mu^0(0,0,0,1,0,0) = 0\\
\text{SW } \mathtt{E}_{17} &\coloneqq \mu^0\left(\text{SW } \mathtt{E} \circ \chi_{19+U_i} \mid i \in \mathbb{Z}_7 \backslash \{1\}\right) = \mu^0(1,0,1,0,0,0) = 0\\
\text{SW } \mathtt{E}_{18} &\coloneqq \mu^0\left(\text{SW } \mathtt{E} \circ \chi_{3+U_i} \mid i \in \mathbb{Z}_7 \backslash \{2\}\right) = \mu^0(0,0,0,1,0,0) = 0\\
\text{SW } \mathtt{E}_{19} &\coloneqq \mu^0\left(\text{SW } \mathtt{E} \circ \chi_{6+U_i} \mid i \in \mathbb{Z}_7 \backslash \{5\}\right) = \mu^0(0,0,1,0,1,0) = 0\\
\text{SW } \mathtt{E}_{20} &\coloneqq \mu^0\left(\text{SW } \mathtt{E} \circ \chi_{12+U_i} \mid i \in \mathbb{Z}_7 \backslash \{6\}\right) = \mu^0(1,1,0,0,0,0) = 0\\
\text{SW } \mathtt{E}_{21} &\coloneqq \mu^0\left(\text{SW } \mathtt{E} \circ \chi_{24+U_i} \mid i \in \mathbb{Z}_7 \backslash \{6\}\right) = \mu^0(0,0,0,0,1,0) = 0\\
\text{SW } \mathtt{E}_{22} &\coloneqq \mu^0\left(\text{SW } \mathtt{E} \circ \chi_{21+U_i} \mid i \in \mathbb{Z}_7 \backslash \{6\}\right) = \mu^0(0,0,0,0,0,0) = 0\\
\text{SW } \mathtt{E}_{23} &\coloneqq \mu^0\left(\text{SW } \mathtt{E} \circ \chi_{15+U_i} \mid i \in \mathbb{Z}_7 \backslash \{6\}\right) = \mu^0(0,1,0,0,0,0) = 0\\
\text{SW } \mathtt{E}_{24} &\coloneqq \mu^0\left(\text{SW } \mathtt{E} \circ \chi_{30+U_i} \mid i \in \mathbb{Z}_7 \backslash \{0\}\right) = \mu^0(1,0,0,0,1,0) = 0\\
\text{SW } \mathtt{E}_{25} &\coloneqq \mu^0\left(\text{SW } \mathtt{E} \circ \chi_{25+U_i} \mid i \in \mathbb{Z}_7 \backslash \{4\}\right) = \mu^0(0,0,0,0,1,1) = 0\\
\text{SW } \mathtt{E}_{26} &\coloneqq \mu^0\left(\text{SW } \mathtt{E} \circ \chi_{23+U_i} \mid i \in \mathbb{Z}_7 \backslash \{2\}\right) = \mu^0(0,0,1,1,0,1) = 0\\
\text{SW } \mathtt{E}_{27} &\coloneqq \mu^0\left(\text{SW } \mathtt{E} \circ \chi_{11+U_i} \mid i \in \mathbb{Z}_7 \backslash \{6\}\right) = \mu^0(0,0,1,0,1,0) = 0\\
\text{SW } \mathtt{E}_{28} &\coloneqq \mu^0\left(\text{SW } \mathtt{E} \circ \chi_{22+U_i} \mid i \in \mathbb{Z}_7 \backslash \{3\}\right) = \mu^0(0,0,0,0,0,0) = 0\\
\text{SW } \mathtt{E}_{29} &\coloneqq \mu^0\left(\text{SW } \mathtt{E} \circ \chi_{9+U_i} \mid i \in \mathbb{Z}_7 \backslash \{6\}\right) = \mu^0(0,0,0,1,0,0) = 0\\
\text{SW } \mathtt{E}_{30} &\coloneqq \mu^0\left(\text{SW } \mathtt{E} \circ \chi_{18+U_i} \mid i \in \mathbb{Z}_7 \backslash \{3\}\right) = \mu^0(1,0,1,1,0,1) = 1
\end{aligned}$$

◁

Der Decoder hat an den Positionen $\{0, 15, 30\}$ einen Fehler registriert und gibt $\mathtt{z} - \mathbf{e}_0 - \mathbf{e}_{15} - \mathbf{e}_{30}$ aus, das tatsächlich das ursprünglich übertragene Codewort $\mathtt{c}$ ist.

Decodieraufwand

Umfasst die erste Majority-Logic-Stufe nur Mehrheitsentscheidungen und wird anschließend mit dem verbesserten Verfahren decodiert, so berechnen wir zunächst höchstens

$$\mathcal{V}_{\mathbf{E}}^{2\cdot\mathrm{EG}-1} \coloneqq (\eta+1)^{\varsigma-1} \cdot \begin{cases} (q^{m-r}-1)\cdot(q^{m-r}-3) & p \neq 2, \\ \binom{q^{m-r}-1}{2} & p = 2 \end{cases}$$

Checksummen und bestimmen basierend auf diesen die Fehlersummen zu

$$(\eta+1)^{\varsigma-1} \cdot \left(q^{m-r}-1\right)$$

affinen Räumen aus $\mathcal{A}^*_{r,m,q}$ mit Hilfe jeweils einer η-Mehrheitsentscheidung. Die Decodierung setzen wir mit dem verbesserten Verfahren wie oben beschrieben fort, so dass wir zur vollständigen Decodierung insgesamt

$$\mathcal{V}_{\mu}^{2\cdot\mathrm{EG}-1} \coloneqq (\eta+1)^{\varsigma-1} \cdot \left(q^{m-r}-1\right) + \sum_{t=0}^{\varsigma-2} (\eta+1)^t \left(q^{m-D_t}-1\right)$$

η-Mehrheitsentscheidungen benötigen.

Modifizieren wir hingegen die erste Majority-Logic-Stufe gemäß Proposition 4.4.1, so sind insgesamt

$$\mathcal{H}_{\mathbf{E}}^{2\cdot\mathrm{EG}-1} \coloneqq (\eta+1)^{\varsigma-1} \cdot \left(q^{m-r}-1\right)$$

Checksummen aus jeweils q^r Summanden zu berechnen. Auf deren Basis werden $(\eta+1)^{\varsigma-1}$ Mehrheitsentscheidungen getroffen und anschließend $(\eta+1)^{\varsigma-1}\cdot(q^{m-r}-1)$ Additionen/Subtraktionen durchgeführt. Dann wird die Decodierung entsprechend dem hybriden Verfahren mit $\dot{D} = m-r$ fortgeführt. Insgesamt benötigen wir zur vollständigen Decodierung

$$\mathcal{H}_{\mu}^{2\cdot\mathrm{EG}-1} \coloneqq (\eta+1)^{\varsigma-1} + \sum_{0\leq t\leq\varsigma-2} (\eta+1)^t \cdot \begin{cases} \left(q^{m-D_t}-1\right) & t \notin T, \\ \mathcal{H}_{\mu}\left(D_t, \dot{D}\right) & t \in T \end{cases}$$

η-Mehrheitsentscheidungen und

$$\mathcal{H}_{\pm}^{2\cdot\mathrm{EG}-1} \coloneqq (\eta+1)^{\varsigma-1} \cdot \left(q^{m-r}-1\right) + \sum_{\substack{0\leq t\leq\varsigma-2 \\ t\in T}} (\eta+1)^t \cdot \mathcal{H}_{\pm}\left(D_t, \dot{D}\right)$$

Additionen/Subtraktionen.

In Tabelle 7.5 listen wir für einige zweifache Typ-1-EG(m, q)-Codes die Parameter Länge n, k und τ_{MLG} sowie den Aufwand sowohl der verbesserten als auch der hybriden Decodierung eines Codeworts unter der invertierten Abstufung. Die sequenzielle bzw. die parallele Laufzeit bestimmt sich durch die Angaben im zweiten bzw. dritten Block der Tabelle 7.5.

Tabelle 7.5: Parameter und Decodieraufwand unter der invertierten Abstufung beim zweifachen Typ-1-EG(m, q)-Code der Ordnung r mit $\tau_{\max} = \tau_{\mathrm{MLG}}$.

n	k	$\frac{k}{n}$	$\tau_{\max}$	$\mathcal{V}_{\mu,I}^{2\cdot\mathrm{EG}-1}$	$\mathcal{H}_{\mu,I}^{2\cdot\mathrm{EG}-1}$	$\mathcal{H}_{\pm,I}^{2\cdot\mathrm{EG}-1}$	$\mathcal{V}_{\mathbf{E},I}^{2\cdot\mathrm{EG}-1}$	$\mathcal{H}_{\mathbf{E},I}^{2\cdot\mathrm{EG}-1}$	ς	$\lvert T_I\rvert$	η	q	r
3	1	0,33	1	3	1	3	3	3	1	1	2	2	0
7	1	0,14	3	7	1	7	21	7	1	1	6	2	0
7	4	0,57	1	16	5	14	9	9	2	2	2	2	1
15	5	0,33	3	64	9	62	147	49	2	2	6	2	1
15	11	0,73	1	24	18	9	9	9	2	1	2	2	2
15	11	0,73	1	24	15	18	9	9	2	2	2	4	1
31	6	0,19	7	256	17	254	1.575	225	2	2	14	2	1
31	16	0,52	3	80	38	49	147	49	2	1	6	2	2
31	26	0,84	1	79	46	42	27	27	3	2	2	2	3
63	7	0,11	15	1.024	33	1.022	14.415	961	2	2	30	2	1
63	22	0,35	7	288	78	225	1.575	225	2	1	14	2	2
63	24	0,38	7	288	59	282	1.575	225	2	2	14	4	1
63	42	0,67	3	112	70	49	147	49	2	1	6	2	3
63	45	0,71	3	112	63	98	147	49	2	2	6	8	1
63	57	0,9	1	111	78	42	27	27	3	2	2	2	4
63	57	0,9	1	135	108	54	27	27	3	2	2	4	2
127	29	0,23	15	1.088	158	961	14.415	961	2	1	30	2	2
127	64	0,5	7	352	142	225	1.575	225	2	1	14	2	3
127	99	0,78	3	575	190	434	1.029	343	3	2	6	2	4
127	120	0,94	1	199	181	27	27	27	3	1	2	2	5
255	37	0,15	31	4.224	318	3.969	123.039	3.969	2	1	62	2	2
255	45	0,18	31	4.224	235	4.218	123.039	3.969	2	2	62	4	1
255	93	0,36	15	1.216	286	961	14.415	961	2	1	30	2	3

Fortsetzung der Tabelle auf der nächsten Seite

Tabelle 7.5 – Fortsetzung der Tabelle

n	k	$\frac{k}{n}$	$\tau_{\max}$	$\mathcal{V}_{\mu,I}^{2\cdot\mathrm{EG}-1}$	$\mathcal{H}_{\mu,I}^{2\cdot\mathrm{EG}-1}$	$\mathcal{H}_{\pm,I}^{2\cdot\mathrm{EG}-1}$	$\mathcal{V}_{\mathtt{E},I}^{2\cdot\mathrm{EG}-1}$	$\mathcal{H}_{\mathtt{E},I}^{2\cdot\mathrm{EG}-1}$	ς	$\lvert T_I\rvert$	η	q	r
255	163	0,64	7	480	270	225	1.575	225	2	1	14	2	4
255	171	0,67	7	480	270	225	1.575	225	2	1	14	4	2
255	191	0,75	7	480	255	450	1.575	225	2	2	14	16	1
255	219	0,86	3	703	318	434	1.029	343	3	2	6	2	5
255	247	0,97	1	327	309	27	27	27	3	1	2	2	6
255	247	0,97	1	327	300	54	27	27	3	2	2	4	3
511	130	0,25	31	4.480	574	3.969	123.039	3.969	2	1	62	2	3
511	184	0,36	31	4.480	503	4.466	123.039	3.969	2	2	62	8	1
511	256	0,5	15	1.472	542	961	14.415	961	2	1	30	2	4
511	382	0,75	7	4.351	766	3.810	23.625	3.375	3	2	14	2	5
511	466	0,91	3	1.071	777	343	1.029	343	3	1	6	2	6
511	475	0,93	3	1.295	952	686	1.029	343	3	2	6	8	2
511	502	0,98	1	748	649	126	81	81	4	2	2	2	7

7.3.2 Zweifache Typ-0-EG(m, q)-Codes

Wir betrachten den Code

$$\left\langle \chi_{\mathbf{v}+U} - \chi_{\mathbf{w}+U} \mid U \leq \mathbb{F}_q^m, \dim U = r, \mathbf{v}, \mathbf{w} \in \mathbb{F}_q^m, \mathbf{v} - \mathbf{w} \notin U \right\rangle_{\mathbb{F}_{q_{\mathcal{C}}}},$$

der $\bar{\mathcal{C}}_{r,m,q,q_{\mathcal{C}}}$ als Subcode enthält. Dieser Code ist äquivalent zum (zyklischen) primitiven Polynomcode über $\mathbb{F}_p$ der Ordnung $(m-r)(q-1)-1$ [18], [21, Theorem 3, Theorem 11]. Das Erzeugerpolynom dieses Codes besitzt also (ausschließlich) die Nullstellen γ^i, $i \in I$ mit

$$I \coloneqq \left\{ i \in \mathbb{Z}_n \;\middle|\; 0 < \min_{0 \leq j < \log_p(q)} \omega_q\left(i \cdot p^j \bmod n\right) < r(q-1)+1 \right\},$$

[18, Theorem 6, Corollary 7].

Es sei $\mathcal{C}$ sein Dualcode. Der Code $\mathcal{C}$ ist äquivalent zum sogenannten *zweifachen Typ-0-EG(m, q)-Code* der Ordnung r (engl. Type-0 twofold $(r, \log_p(q))$th-order

EG code), [21, Theorem 11]. Dessen Erzeugerpolynom besitzt (ausschließlich) die Nullstellen γ^i, $i \in \bar{I}$ mit

$$\bar{I} \coloneqq \left\{ i \in \mathbb{Z}_n \;\middle|\; 0 \leq \max_{0 \leq j < \log_p(q)} \omega_q\left(i \cdot p^j \bmod n\right) \leq (m-r)(q-1) - 1 \right\},$$

[21, §IV. Theorem 3]. Die Minimaldistanz d von $\mathcal{C}$ beträgt

$$d \coloneqq q^{m-r}, \tag{7.11}$$

siehe [21, Theorem 5]. Also ist $\mathcal{C}$ ein $\tau_{\max}$-fehlerkorrigierender Code mit

$$\tau_{\max} \coloneqq \left\lfloor \frac{q^{m-r} - 1}{2} \right\rfloor.$$

Da $\mathcal{C}$ ein Subcode des zweifachen Typ-1-EG(m, q)-Codes der Ordnung r ist, können Codewörter aus $\mathcal{C}$ selbstverständlich wie für den zweifachen Typ-1-EG(m, q)-Codes gezeigt mit

$$\eta \coloneqq \begin{cases} q^{m-r} - 3 & p \neq 2, \\ q^{m-r} - 2 & p = 2 \end{cases}.$$

rekonstruiert werden, sofern nicht mehr als

$$\left\lfloor \frac{q^{m-r} - 2}{2} \right\rfloor$$

Fehler auftreten. Darüber hinaus gibt es die Möglichkeit, die Decodierung auf den zweifachen Typ-0-EG(m, q)-Code anzupassen.

Anpassung der Decodierung an den zweifachen Typ-0-EG(m, q)-Code

Wir orientieren uns eng an der Decodierung für zweifache Typ-1-EG(m, q)-Codes. Im Gegensatz zum zweifachen Typ-1-EG(m, q)-Code liegen auch Inzidenzvektoren der Form $\chi_U - \chi_{\mathbf{v}+U}$ im Dualcode und können für die Decodierung herangezogen werden. Wir setzen daher

$$\eta \coloneqq \begin{cases} q^{m-r} - 1 & p \neq 2, \\ q^{m-r} - 2 & p = 2 \end{cases}.$$

Es gilt

$$\tau_{\mathrm{MLG}} \coloneqq \lfloor \eta/2 \rfloor = \tau_{\max}. \qquad [7.12]$$

Falls $p \neq 2$, ist es also möglich im Vergleich zum zweifachen Typ-1-EG(m, q)-Code einen zusätzlichen Fehler zu korrigieren.

Die für die verbesserte und hybride Decodierung benötigten Decodierbäume können kleiner als beim zweifachen Typ-1-EG(m, q)-Code konstruiert werden, indem jeder Elternknoten statt $\eta+1$ nur noch η Kindknoten besitzt. Beim Decodieren müssen allerdings nicht nur die Fehlersummen von echten affinen Räumen sondern auch von Unterräumen (bis auf den Nullraum) bestimmt werden. Dies führt zu einem veränderten Decodieraufwand gegenüber dem zweifachen Typ-1-EG(m, q)-Code. Der Parameter $\dot{D}$ wird wie folgt festgesetzt,

$$\dot{D} \coloneqq \left\lfloor \log_q \left((q-1) \cdot \eta + 1 \right) \right\rfloor = \begin{cases} m-r & q \neq 2, \\ m-r-1 & q = 2 \end{cases}.$$

Den Decodieraufwand geben wir wie folgt an:

$$\begin{aligned}
\mathcal{V}_{\mathtt{E}}^{2\cdot\mathrm{EG}-0} &\coloneqq \eta^{\varsigma-1} \cdot q^{m-r} \cdot \begin{cases} (q^{m-r}-1) & p \neq 2, \\ (q^{m-r}-2)/2 & p = 2 \end{cases}, \\
\mathcal{V}_{\mu}^{2\cdot\mathrm{EG}-0} &\coloneqq \eta^{\varsigma-1} \cdot q^{m-r} - 1 + \sum_{t=0}^{\varsigma-2} \eta^t q^{m-D_t}, \\
\mathcal{H}_{\mathtt{E}}^{2\cdot\mathrm{EG}-0} &\coloneqq \eta^{\varsigma-1} \cdot q^{m-r} - \mathbb{1}_{\varsigma=1}, \\
\mathcal{H}_{\mu}^{2\cdot\mathrm{EG}-0} &\coloneqq \eta^{\varsigma-1} - \mathbb{1}_{\substack{0\notin T, \\ \varsigma>1}} + \sum_{0\leq t\leq\varsigma-2} \eta^t \cdot \begin{cases} q^{m-D_t} & t \notin T, \\ \mathcal{H}_{\mu}\left(D_t, \dot{D}\right) & t \in T \end{cases}, \\
\mathcal{H}_{\pm}^{2\cdot\mathrm{EG}-0} &\coloneqq \eta^{\varsigma-1} \cdot q^{m-r} - \mathbb{1}_{\varsigma=1} \\
&\quad + \mathbb{1}_{\substack{0\in T, \\ \varsigma>1}} \cdot \mathcal{H}_{\pm}\left(0, \dot{D}\right) + \sum_{\substack{1\leq t\leq\varsigma-2 \\ t\in T}} \eta^t \cdot \left(q - 1 + \mathcal{H}_{\pm}\left(D_t, \dot{D}\right)\right).
\end{aligned}$$

In Tabelle 7.6 listen wir für einige zweifache Typ-0-EG(m, q)-Codes die Parameter Länge n, k und τ_{MLG} sowie den Aufwand sowohl der verbesserten als auch der hybriden Decodierung eines Codeworts unter der invertierten Abstufung. Die Angaben im zweiten bzw. dritten Block der Tabelle 7.6 geben wie zuvor Aufschluss über die sequenzielle bzw. die parallele Laufzeit.

Tabelle 7.6: Parameter und Decodieraufwand unter der invertierten Abstufung beim zweifachen Typ-0-EG(m, q)-Code der Ordnung r mit $\tau_{\max} = \tau_{\mathrm{MLG}}$.

n	k	$\frac{k}{n}$	$\tau_{\max}$	$\mathcal{V}_{\mu,I}^{2\cdot\mathrm{EG}-0}$	$\mathcal{H}_{\mu,I}^{2\cdot\mathrm{EG}-0}$	$\mathcal{H}_{\pm,I}^{2\cdot\mathrm{EG}-0}$	$\mathcal{V}_{\mathbf{E},I}^{2\cdot\mathrm{EG}-0}$	$\mathcal{H}_{\mathbf{E},I}^{2\cdot\mathrm{EG}-0}$	ς	$\lvert T_I\rvert$	η	q	r
7	3	0,43	1	15	6	11	8	8	2	2	2	2	1
15	4	0,27	3	63	10	59	144	48	2	2	6	2	1
15	10	0,67	1	23	17	8	8	8	2	1	2	2	2
15	10	0,67	1	23	14	17	8	8	2	2	2	4	1
31	5	0,16	7	255	18	251	1.568	224	2	2	14	2	1
31	15	0,48	3	79	37	48	144	48	2	1	6	2	2
31	25	0,81	1	63	43	24	16	16	3	2	2	2	3
63	6	0,1	15	1.023	34	1.019	14.400	960	2	2	30	2	1
63	21	0,33	7	287	77	224	1.568	224	2	1	14	2	2
63	23	0,37	7	287	58	281	1.568	224	2	2	14	4	1
63	41	0,65	3	111	69	48	144	48	2	1	6	2	3
63	44	0,7	3	111	62	97	144	48	2	2	6	8	1
63	56	0,89	1	95	75	24	16	16	3	2	2	2	4
63	56	0,89	1	111	91	40	16	16	3	2	2	4	2
127	28	0,22	15	1.087	157	960	14.400	960	2	1	30	2	2
127	63	0,5	7	351	141	224	1.568	224	2	1	14	2	3
127	98	0,77	3	511	187	360	864	288	3	2	6	2	4
127	119	0,94	1	175	163	16	16	16	3	1	2	2	5
255	36	0,14	31	4.223	317	3.968	123.008	3.968	2	1	62	2	2
255	44	0,17	31	4.223	234	4.217	123.008	3.968	2	2	62	4	1
255	92	0,36	15	1.215	285	960	14.400	960	2	1	30	2	3
255	162	0,64	7	479	269	224	1.568	224	2	1	14	2	4
255	170	0,67	7	479	269	224	1.568	224	2	1	14	4	2
255	190	0,75	7	479	254	449	1.568	224	2	2	14	16	1
255	218	0,85	3	639	315	360	864	288	3	2	6	2	5
255	246	0,96	1	303	291	16	16	16	3	1	2	2	6
255	246	0,96	1	303	283	40	16	16	3	2	2	4	3
511	129	0,25	31	4.479	573	3.968	123.008	3.968	2	1	62	2	3
511	183	0,36	31	4.479	502	4.465	123.008	3.968	2	2	62	8	1

Fortsetzung der Tabelle auf der nächsten Seite

Tabelle 7.6 – Fortsetzung der Tabelle

n	k	$\frac{k}{n}$	$\tau_{\max}$	$\mathcal{V}_{\mu,I}^{2\cdot\mathrm{EG}-0}$	$\mathcal{H}_{\mu,I}^{2\cdot\mathrm{EG}-0}$	$\mathcal{H}_{\pm,I}^{2\cdot\mathrm{EG}-0}$	$\mathcal{V}_{\mathtt{E},I}^{2\cdot\mathrm{EG}-0}$	$\mathcal{H}_{\mathtt{E},I}^{2\cdot\mathrm{EG}-0}$	ς	$\lvert T_I\rvert$	η	q	r
511	255	0,5	15	1.471	541	960	14.400	960	2	1	30	2	4
511	381	0,75	7	4.095	763	3.528	21.952	3.136	3	2	14	2	5
511	465	0,91	3	991	739	288	864	288	3	1	6	2	6
511	474	0,93	3	1.183	883	624	864	288	3	2	6	8	2
511	501	0,98	1	639	599	48	32	32	4	2	2	2	7

7.4 Zyklische respektive punktierte Reed-Muller-Codes

Sei $r+1 < m$ und $q = q_{\mathcal{C}} = p = 2$. Dann fallen der Typ-1-EG(m,q)-Code und der zweifache Typ-1-EG(m,q)-Code jeweils der Ordnung r zusammen. Dies lässt sich unmittelbar mit den Nullstellen des Erzeugerpolynoms, die identisch sind, oder auch mit $\mathcal{C}_{r+1,m,2,2} = \bar{\mathcal{C}}_{r,m,2,2}$ aus Proposition 7.1.3 begründen.

Dieser Code, in diesem Abschnitt mit $\mathcal{C}$ bezeichnet, ist in der Literatur als *zyklischer respektive punktierter*[7] *Reed-Muller-Code der Ordnung* r, $\mathrm{RM}^*(r,m)$, bekannt [4, §6.7] [22, §8.6, S. 315 ff.]

Dessen Erzeugerpolynom besitzt (ausschließlich) die Nullstellen γ^i, $i \in \bar{I}$ mit

$$\bar{I} \coloneqq \{i \in \mathbb{Z}_n \mid 0 < \omega_2(i) \leq m-r-1\},$$

wobei γ ein multiplikatives Erzeugendes von $\mathbb{F}_{2^m}^*$ ist, vgl. Abschnitt 7.2, Abschnitt 7.3 alternativ [4, §6.7], [22, (8.34)], [24, Ch. 13. §5. Theorem 11.]. Die Dimension k des Codes $\mathcal{C}$ lässt sich also explizit angeben durch $k = \sum_{i=0}^{r} \binom{m}{i}$, denn

$$\begin{aligned} k &= n - \left|\bar{I}\right| \\ &= 2^m - 1 - \sum_{i=1}^{m-r-1} \binom{m}{i} \end{aligned}$$

[7] Die Klasse der Reed-Muller-Codes im eigentlichen Sinn umfasst Codes der Länge 2^m. Punktiert man diese an der zum Nullvektor korrespondierenden Position, so erhält man den zyklischen oder eben punktierten Reed-Muller-Code.

$$= \sum_{i=0}^{m} \binom{m}{i} - 1 - \sum_{i=1}^{m-r-1} \binom{m}{i}$$
$$= \sum_{i=0}^{r} \binom{m}{i}.$$

Die Minimaldistanz d von $\mathcal{C}$ beträgt $2^{m-r}-1$, vgl. Gleichung [7.6] auf Seite 129 oder [22, §8.6, S. 315]. $\mathcal{C}$ ist damit ein $\tau_{\max}$-fehlerkorrigierender Code mit

$$\tau_{\max} = \tau_{\mathrm{BCH}} = \tau_{\mathrm{MLG}} = 2^{m-r-1} - 1.$$

Wir weisen daraufhin, dass die Hybriddecodierung für Typ-1-EG$(m,2)$-Codes sich in der ersten Majority-Logic-Stufe von der für zweifache Typ-1-EG$(m,2)$-Codes unterscheidet. Erwartungsgemäß ist der Aufwand der Hybriddecodierung geringer, wenn man $\mathcal{C}$ als zweifachen Typ-1-EG$(m,2)$-Code auffasst und sich an dessen Decodierung orientiert. Genau genommen ändern sich dann die Anzahl der benötigten Fehlersummen und die Anzahl der ausgeführten Additionen/Subtraktionen. Bei der klassischen und verbesserten Decodierung hingegen gibt es keinen Unterschied. In jedem Fall ist

$$\eta = 2^{m-r} - 2,$$
$$\dot{D} = m - r.$$

Der Decodieraufwand ergibt sich dann wie folgt:

$$\mathcal{V}_{\mathtt{E}}^{RM*} \coloneqq (\eta+1)^{\varsigma} \left(2^{m-r-1}-1\right),$$
$$\mathcal{V}_{\mu}^{RM*} \coloneqq \sum_{t=0}^{\varsigma-1} (\eta+1)^{t} \left(2^{m-D_t}-1\right),$$
$$\mathcal{H}_{\mathtt{E}}^{RM*} \coloneqq \begin{cases} (\eta+1)^{\varsigma} & \text{als zweifacher EG-Typ-1-Code aufgefasst,} \\ (\eta+1)^{\varsigma}\left(2^{m-r-1}-1\right) & \text{als EG-Typ-1-Code aufgefasst,} \end{cases},$$
$$\mathcal{H}_{\mu}^{RM*} \coloneqq \sum_{0 \le t \le \varsigma-1} (\eta+1)^{t} \cdot \begin{cases} \left(2^{m-D_t}-1\right) & t \notin T, \\ \mathcal{H}_{\mu}\left(D_t, \dot{D}\right) & t \in T \end{cases},$$
$$\mathcal{H}_{\pm}^{RM*} \coloneqq \sum_{\substack{0 \le t \le \varsigma-2 \\ t \in T}} (\eta+1)^{t} \cdot \mathcal{H}_{\pm}\left(D_t, \dot{D}\right)$$
$$+ (\eta+1)^{\varsigma-1} \cdot \begin{cases} \eta+1 & \text{als zweifacher EG-Typ-1-Code aufgefasst,} \\ \eta & \text{als EG-Typ-1-Code aufgefasst} \end{cases}.$$

Zu beachten ist, dass der Aufwand, eine einzelne Fehlersumme zu bestimmen, nicht konstant ist. Das Zusammenfassen zweier Werte in $\mathcal{H}_{\mathbf{E}}^{RM*}$ könnte darüber hinwegtäuschen. Tatsächlich werden die Fehlersummen beim zweifachen EG-Typ-1-Code aus jeweils q^r Summanden, diejenigen beim EG-Typ-1-Code aber aus q^{r+1} Summanden gebildet. Der Vorteil, den zyklischen Reed-Muller-Code als zweifachen EG-Typ-1-Code aufzufassen, verstärkt sich dadurch zusätzlich.

Der konkrete Decodieraufwand unter der invertierten Abstufung ist in den Zeilen von Tabelle 7.1 und Tabelle 7.5 mit $q = 2$ abzulesen.

7.5 Reed-Muller-Codes

Sei $r + 1 < m$ und $q = q_{\mathcal{C}} = p = 2$. Dann fallen die Codes $\mathcal{C}_{r+1,m,q,q_{\mathcal{C}}}$ und $\bar{\mathcal{C}}_{r,m,q,q_{\mathcal{C}}}$ und somit auch der $\mathrm{EG}(m, q)$-Code und der zweifache $\mathrm{EG}(m, q)$-Code jeweils der Ordnung r zusammen.

Sei

$$\mathcal{C}_0 \coloneqq \langle \chi_{\mathbf{v}+U} \mid U \leq \mathbb{F}_2^m, \dim U = m - r, \mathbf{v} \in \mathbb{F}_2^m \rangle_{\mathbb{F}_2} .$$

Wir betrachten die Erweiterung von $\mathcal{C}_0$ durch ein Parity-Check-Symbol,

$$\mathcal{C} \coloneqq \left\langle \hat{\chi}_{\mathbf{v}+U} \in \mathbb{F}_2^{2^m} \mid U \leq \mathbb{F}_2^m, \dim U = m - r, \mathbf{v} \in \mathbb{F}_2^m \right\rangle_{\mathbb{F}_2}$$

mit

$$\hat{\chi}_{\mathbf{v}+U} \coloneqq \begin{cases} (\chi_{\mathbf{v}+U}, 0) & \mathbf{v} \notin U \\ (\chi_{\mathbf{v}+U}, 1) & \mathbf{v} \in U \end{cases} .$$

Der aus der Literatur [29, §5.5], [22, §4.3], [24, Ch. 13.], [40, S. 21 ff.] bekannte *Reed-Muller-Code der Ordnung* r, bezeichnet mit $\mathrm{RM}(r, m)$, ist äquivalent zum Code $\mathcal{C}$ [24, Ch. 13. §6., Theorem 12].

Die Dimension k des Codes $\mathcal{C}$ ist $k = \sum_{i=0}^{r} \binom{m}{i}$, [24, Ch. 13. §3., S. 373 f.]. Die Minimaldistanz d beträgt 2^{m-r}, [24, Ch. 13. §3., Theorem 3]. $\mathcal{C}$ ist damit ein $\tau_{\max}$-fehlerkorrigierender Code mit

$$\tau_{\max} = 2^{m-r-1} - 1.$$

Sein Dualcode ist äquivalent zum Reed-Muller-Code der Ordnung $m - r - 1$, [24, Ch. 13. §3., Theorem 4]. Mit anderen Worten, der Dualcode von $\mathcal{C}$ wird erzeugt durch die Inzidenzvektoren der Länge 2^m zu allen affinen Unterräumen der Dimension $r+1$. Diese Aussage ist leicht einsichtig, wenn man sich bewusst macht, dass zwei affine Räume $\mathbf{v}_0 + U_0$, $\mathbf{v}_1 + U_1$ entweder kein gemeinsames Element besitzen oder sich in einem affinen Unterraum schneiden. Dieser affine Schnittraum besitzt eine Dimension von mindestens eins und damit eine gerade Anzahl von Elementen, falls die Dimension von U_0 und U_1 gerade $m - r$ bzw. $r + 1$ ist.

Für Reed-Muller-Codes RM(r, m) mit $2 < 2r \leq m$, wird die zweistufige Hybriddecodierung in [2] und [3] detailliert erklärt. Der Reed-Muller-Code wird dabei in beiden Schriften als zweifacher EG-Code aufgefasst. Beispielhaft wird der Decoder in [2] für den Reed-Muller-Code RM(2,5) sowie in [3] für den Reed-Muller-Code RM(2,4) konstruiert und mittels Schaltbilder veranschaulicht, siehe Abbildung 7.1, Abbildung 7.2 und Abbildung 7.3.

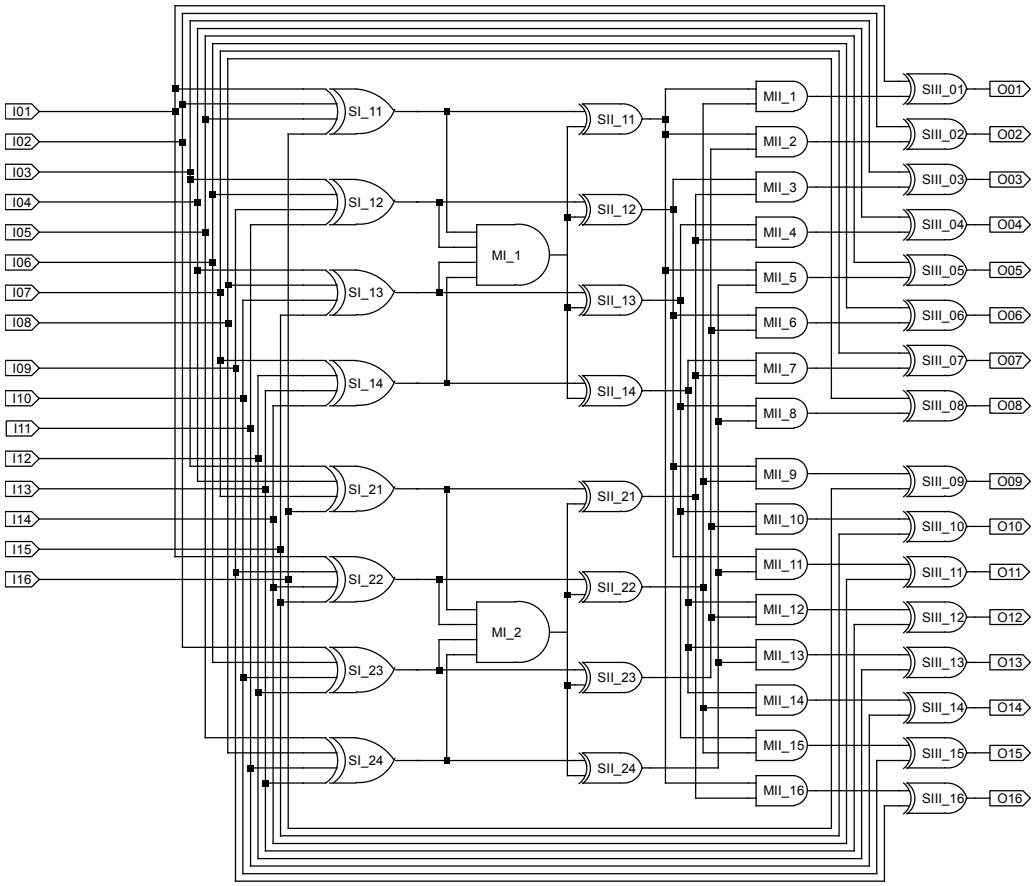


Abbildung 7.1: Decoder für den Reed-Muller-Code RM$(2, 4)$, entnommen aus [3, Fig. 3], bestehend aus acht Paritätserzeugungsmodulen (beschriftet mit „SI_{11}“ bis „SI_{24}“), zwei 4-Majoritätsgattern (beschriftet mit MI_1 und MI_2), 24 XOR-Gattern (beschriftet mit SII_{11} bis SII_{24} und SIII_{01} bis SII_{16}) und 16 AND-Gattern (beschriftet mit MI_1 bis MI_{16}). Die Paritätserzeugungsmodule berechnen die Checksummen der zweidimensionalen affinen Räume. Mit AND-Gattern werden die 2-Mehrheitsentscheidungen realisiert.

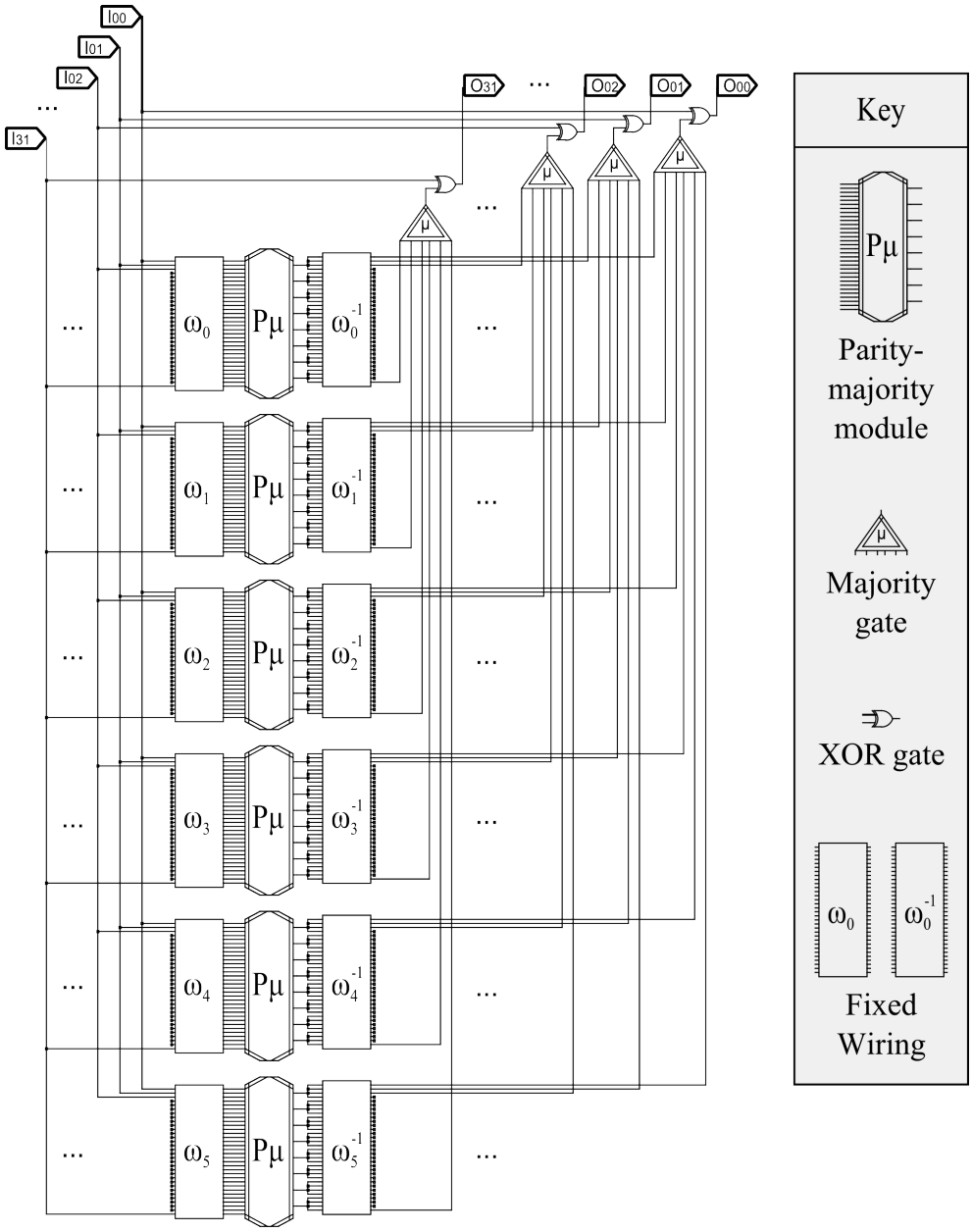


Abbildung 7.2: Decoder für den Reed-Muller-Code RM$(2, 5)$, entnommen aus [2, Fig. 1], bestehend aus sechs Paritätsmajoritätsmodulen (*parity-majority modules*), 32 Majoritätsgattern (*majority gates*), 32 XOR-Gattern (*XOR gates*) und fester Verdrahtung (*fixed wiring*).

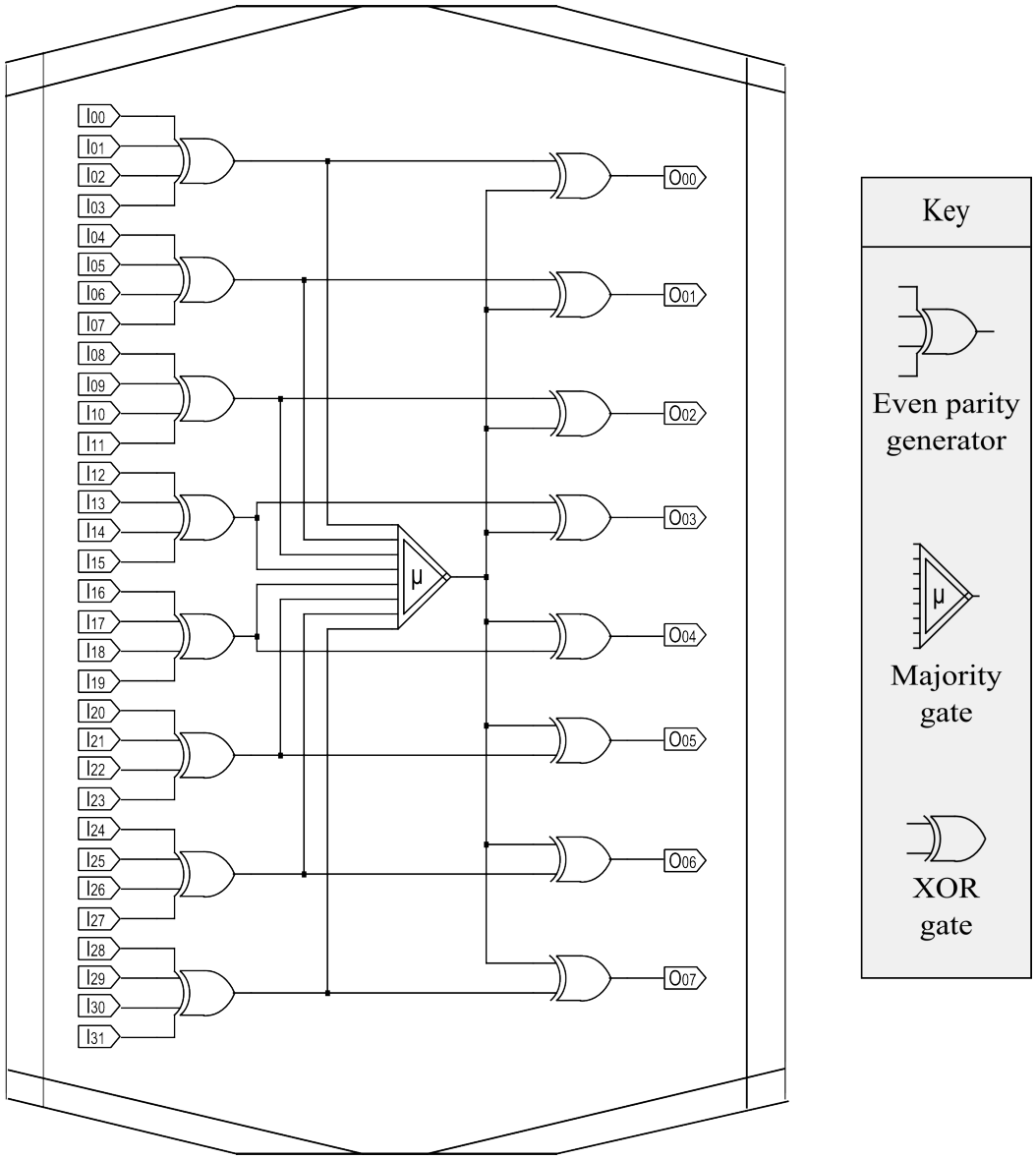


Abbildung 7.3: Paritätsmajoritätsmodul als Bestandteil des Decoders für den Reed-Muller-Code RM(2,5), entnommen aus [2, Fig. 2], bestehend aus acht Paritätserzeugungsmodulen (*even parity generator*), einem Majoritätsgatter (*majority gate*) und acht XOR-Gattern (*XOR gates*). Die Paritätserzeugungsmodule berechnen die Checksummen der zweidimensionalen affinen Räume.

Decodieraufwand

Für die Typ-0-Codes haben wir bereits gesehen, wie man ausnutzen kann, dass im Dualcode auch die Inzidenzvektoren zu Unterräumen liegen. Wir hatten bei diesen Codes nicht nur die Fehlersummen von echten affinen Räumen sondern auch die von Unterräumen (bis auf den Nullraum) bestimmt. Wenn wir den dort präsentierten Aufwand um den Aufwand, die Fehlersumme des Nullraums zu bestimmen, ergänzen, erhalten wir den Decodieraufwand für den Reed-Muller-Code. Es gilt,

$$\begin{aligned}\eta &= 2^{m-r} - 2,\\ \dot{D} &= m - r - 1.\end{aligned}$$

Der Decodieraufwand bei der verbesserten Decodierung liegt bei

$$\begin{aligned}\mathcal{V}_{\mathtt{E}}^{\mathrm{RM}} &\coloneqq \eta^{\varsigma-1} \cdot 2^{m-r} \cdot \left(2^{m-r} - 2\right)/2 = \eta^{\varsigma} \cdot 2^{m-r-1},\\ \mathcal{V}_{\mu}^{\mathrm{RM}} &\coloneqq \sum_{t=0}^{\varsigma-1} \eta^t \cdot 2^{m-D_t}.\end{aligned}$$

Wie wir bereits vom zyklischen Reed-Muller-Code wissen, variiert der Decodieraufwand bei der Hybriddecodierung, je nachdem, ob man sich bei der Decodierung am EG(m,2)-Code,

$$\begin{aligned}\mathcal{H}_{\mathtt{E}}^{\mathrm{RM}} &\coloneqq \eta^{\varsigma} \cdot 2^{m-r-1},\\ \mathcal{H}_{\mu}^{\mathrm{RM}} &\coloneqq 2 \cdot \eta^{\varsigma-1} + \sum_{0 \le t \le \varsigma-2} \eta^t \cdot \begin{cases} 2^{m-D_t} & t \notin T,\\ \mathcal{H}_{\mu}\left(D_t, \dot{D}\right) & t \in T \end{cases},\\ \mathcal{H}_{\pm}^{\mathrm{RM}} &\coloneqq \eta^{\varsigma-1} \cdot (2^{m-r} - 2) + \sum_{\substack{0 \le t \le \varsigma-2\\ t \in T}} \eta^t \cdot \left(1 + \mathcal{H}_{\pm}\left(D_t, \dot{D}\right)\right),\end{aligned}$$

oder am zweifachen EG(m,2)-Code orientiert,

$$\begin{aligned}\mathcal{H}_{\mathtt{E}}^{\mathrm{RM}} &\coloneqq \eta^{\varsigma-1} \cdot 2^{m-r},\\ \mathcal{H}_{\mu}^{\mathrm{RM}} &\coloneqq \eta^{\varsigma-1} + \sum_{0 \le t \le \varsigma-2} \eta^t \cdot \begin{cases} 2^{m-D_t} & t \notin T,\\ \mathcal{H}_{\mu}\left(D_t, \dot{D}\right) & t \in T \end{cases},\\ \mathcal{H}_{\pm}^{\mathrm{RM}} &\coloneqq \eta^{\varsigma-1} \cdot 2^{m-r} + \sum_{\substack{0 \le t \le \varsigma-2\\ t \in T}} \eta^t \cdot \left(1 + \mathcal{H}_{\pm}\left(D_t, \dot{D}\right)\right).\end{aligned}$$

Wir weisen noch einmal explizit daraufhin, dass der Aufwand, eine einzelne Fehlersumme zu bestimmen, nicht konstant ist. Vielmehr hängt er davon ab, wie wir die erste Majority-Logic-Stufe beim Hybridverfahren ausführen. Im direkten Vergleich ist es von Vorteil, sich am Decodierverfahren für den zweifachen EG-Code zu orientieren. Für den konkreten Decodieraufwand verweisen wir auf Tabelle 7.7.

Tabelle 7.7: Parameter und Decodieraufwand des verbesserten bzw. hybriden Verfahrens (in Anlehnung an den zweifachen EG-Code) unter der invertierten Abstufung beim Reed-Muller-Code der Ordnung r mit $\tau_{\max} = \tau_{\mathrm{MLG}}$.

n	k	$\frac{k}{n}$	$\tau_{\max}$	$\mathcal{V}_{\mu,I}^{\mathrm{RM}}$	$\mathcal{H}_{\mu,I}^{\mathrm{RM}}$	$\mathcal{H}_{\pm,I}^{\mathrm{RM}}$	$\mathcal{V}_{\mathbf{E},I}^{\mathrm{RM}}$	$\mathcal{H}_{\mathbf{E},I}^{\mathrm{RM}}$	ς	$\lvert T_I\rvert$	η	r
4	1	0.25	1	4	1	4	4	4	1	1	2	0
8	1	0.13	3	8	1	8	24	8	1	1	6	0
8	4	0.5	1	16	6	12	8	8	2	2	2	1
16	5	0,31	3	64	10	60	144	48	2	2	6	1
16	11	0,69	1	24	18	8	8	8	2	1	2	2
32	6	0,19	7	256	18	252	1.568	224	2	2	14	1
32	16	0,5	3	80	38	48	144	48	2	1	6	2
32	26	0,81	1	64	44	24	16	16	3	2	2	3
64	7	0,11	15	1.024	34	1.020	14.400	960	2	2	30	1
64	22	0,34	7	288	78	224	1.568	224	2	1	14	2
64	42	0,66	3	112	70	48	144	48	2	1	6	3
64	57	0,89	1	96	76	24	16	16	3	2	2	4
128	29	0,23	15	1.088	158	960	14.400	960	2	1	30	2
128	64	0,5	7	352	142	224	1.568	224	2	1	14	3
128	99	0,77	3	512	188	360	864	288	3	2	6	4
128	120	0,94	1	176	164	16	16	16	3	1	2	5
256	37	0,14	31	4.224	318	3.968	123.008	3.968	2	1	62	2
256	93	0,36	15	1.216	286	960	14.400	960	2	1	30	3
256	163	0,64	7	480	270	224	1.568	224	2	1	14	4
256	219	0,86	3	640	316	360	864	288	3	2	6	5
256	247	0,96	1	304	292	16	16	16	3	1	2	6
512	1	0	255	512	1	512	130.560	512	1	1	510	0

Fortsetzung der Tabelle auf der nächsten Seite

Tabelle 7.7 – Fortsetzung der Tabelle

n	k	$\frac{k}{n}$	$\tau_{\max}$	$\mathcal{V}_{\mu,I}^{\mathrm{RM}}$	$\mathcal{H}_{\mu,I}^{\mathrm{RM}}$	$\mathcal{H}_{\pm,I}^{\mathrm{RM}}$	$\mathcal{V}_{\mathtt{E},I}^{\mathrm{RM}}$	$\mathcal{H}_{\mathtt{E},I}^{\mathrm{RM}}$	ς	$\lvert T_I\rvert$	η	r
512	130	0,25	31	4.480	574	3.968	123.008	3.968	2	1	62	3
512	256	0,5	15	1.472	542	960	14.400	960	2	1	30	4
512	382	0,75	7	4.096	764	3.528	21.952	3.136	3	2	14	5
512	466	0,91	3	992	740	288	864	288	3	1	6	6
512	502	0,98	1	640	600	48	32	32	4	2	2	7
1.024	176	0,17	63	17.152	1.150	16.128	1.016.064	16.128	2	1	126	3
1.024	386	0,38	31	4.992	1.086	3.968	123.008	3.968	2	1	62	4
1.024	638	0,62	15	1.984	1.054	960	14.400	960	2	1	30	5
1.024	848	0,83	7	4.608	1.276	3.528	21.952	3.136	3	2	14	6
1.024	968	0,95	3	1.504	1.252	288	864	288	3	1	6	7
1.024	1.013	0,99	1	1.152	1.112	48	32	32	4	2	2	8

7.6 Binäre Hamming-Codes

Definition 7.6.1. Sei $q = q_{\mathcal{C}} = p = 2$ und $m \geq 3$. Es gilt $n = 2^m - 1$ und

$$\mathbb{F}_2^m \setminus \{0\} = \{\mathbf{w}_0, \mathbf{w}_1, \ldots, \mathbf{w}_{n-1}\}\,.$$

Ein Code, dessen Kontrollmatrix $\left(\mathbf{w}_0^\top, \mathbf{w}_1^\top, \ldots, \mathbf{w}_{n-1}^\top\right)$ ist, wird als *binärer Hamming-Code* bezeichnet [39, Beispiele 1.2.9].

Ein Hamming-Code $\mathcal{C}$ ist ein perfekter $[n, n-m, 3]_2$-Code [39, Beispiele 1.2.9, Bemerkung 1.1.21.], denn mit disjunkten Kugeln vom Radius eins um die Codeworte aus $\mathcal{C}$ lässt sich ganz $\mathbb{F}_2^n$ überdecken [39, Definition 1.1.18.]:

$$\mathbb{F}_2^n = \dot{\bigcup_{\mathtt{c}\in\mathcal{C}}} \{\mathbf{v} \in \mathbb{F}_2^n \;\mid\; \delta(\mathbf{v}, \mathtt{c}) \leq 1\}\,.$$

Der binäre Hamming-Code ist äquivalent zum zyklischen Reed-Muller-Code der Ordnung $r := m-2$, [29, §10.2 Euclidean Geometry Codes] oder [4, §6.7]. Wir verweisen auf Abschnitt 7.4 für die Decodierung des binären Hamming-Codes. Der konkrete Decodieraufwand unter der invertierten Abstufung ist

in Tabelle 7.1 und in Tabelle 7.5 in den Zeilen $q = 2$, $r = \log_2(n+1) - 2$ abzulesen.

Wir möchten noch einmal auf die Beispiele 5.2.3, 5.3.5 und 5.4.8 hinweisen, die die Decodierung mit dem klassischen, verbesserten und hybriden Verfahren beim binären Hamming-Code veranschaulichen.

7.7 Verallgemeinerte Euklidische-Geometrie- und verallgemeinerte Reed-Muller-Codes

Bislang haben wir nur Codes betrachtet, deren Symbole im Primkörper $\mathbb{F}_p$ liegen ($q_{\mathcal{C}} = p$ prim). Ohne Weiteres lassen sich die Decodierverfahren aus Kapitel 5 für die Dualcodes von $\mathcal{C}_{r,m,q,q_{\mathcal{C}}}$ oder $\bar{\mathcal{C}}_{r,m,q,q_{\mathcal{C}}}$ verwenden, wenn $q_{\mathcal{C}}$ nicht prim ist. Zu den Subcodes dieser Dualcodes gehören die *verallgemeinerten EG(m,q)-Codes* der Ordnung $(m-r-1)(q-1)$ über $\mathbb{F}_{q_{\mathcal{C}}}$, [18, S. 812 f.]. Im Fall $q = q_{\mathcal{C}}$ ist der verallgemeinerte Typ-1-EG(m,q)-Code der Ordnung $(m-r-1)(q-1)$ über $\mathbb{F}_{q_{\mathcal{C}}}$ ein verallgemeinerter Reed-Muller-Code (GRM-Code) der Ordnung $(r+1)(q-1)-1$ [17], [29, §10.5], [18, S. 811].

Dabei gilt, dass Länge und Dimension der Codes $\mathcal{C}_{r,m,q,q_{\mathcal{C}}}$ und $\mathcal{C}_{r,m,q,p}$ sowie der Codes $\bar{\mathcal{C}}_{r,m,q,q_{\mathcal{C}}}$ und $\bar{\mathcal{C}}_{r,m,q,p}$ identisch ist. Die Majority-Logic-Fehlerkorrekturfähigkeit der Dualcodes ist ebenfalls identisch, da τ_{MLG} nur von q nicht jedoch von $q_{\mathcal{C}}$ abhängt. Wir werden diese Codes daher nicht weiter untersuchen.

7.8 Vergleich der Codes hinsichtlich ihrer Parameter und des Decodieraufwands

Die verschiedenen Decodierverfahren haben wir miteinander verglichen. Nun wollen wir auch die zuvor betrachteten Codes, für welche die Verfahren angewandt werden können, miteinander vergleichen. Wir werfen die Frage auf,

welcher Code über $\mathbb{F}_p$ bei gegebenen Parametern n und τ_{MLG} zu wählen ist. In einigen Fällen definieren diese Parameter bereits den Code.

Proposition 7.8.1. *Gegeben zwei verschiedene Codes, jeder entweder ein EG-Code oder ein zweifacher EG-Code über $\mathbb{F}_p$. Die beiden Codes besitzen dieselben Parameter n und τ_{MLG}, wenn es sich bei den beiden Codes um eine der folgenden Kombinationen handelt,*

(a) *einen zweifachen EG$(s{\cdot}m, q)$-Code der Ordnung $s{\cdot}r$ und einen zweifachen EG(m, q^s)-Code der Ordnung r, wobei*

  (i) entweder beide jeweils vom gleichen Typ mit $s > 1$

  (ii) oder – vorausgesetzt $p = 2$ – von verschiedenen Typen;

(b) *einen zweifachen EG$(m, 2^s)$-Code der Ordnung r und einen EG$(s \cdot m, 2)$-Code der Ordnung $s \cdot r$ mit $s > 1$, $s(m - r) > 1$;*

(c) *einen Typ-1-EG$(\frac{m}{m-r}, q^{m-r})$-Code der Ordnung $\frac{m}{m-r} - 2$ und einen zweifachen Typ-0-EG(m, q)-Code der Ordnung r mit $(m - r) \,|\, m$ und*

  (i) entweder $p = 3$, 3 teilt τ_{MLG} nicht

  (ii) oder $p > 3$.

Dabei ist $m \in \mathbb{N}$, $r \in \mathbb{N}_0$, $s \in \mathbb{N}$, $m > r$.

Beweis. Seien $\mathcal{C}_1, \mathcal{C}_2$ zwei verschiedene Codes mit identischen Parametern n und τ_{MLG}. Die zum Code $\mathcal{C}_i$ korrespondierenden Parameter m, q werden mit m_i, q_i bezeichnet, $i = 1, 2$.

Mit Blick auf die Majority-Logic-Fehlerkorrekturfähigkeit in Gleichung [7.2], Gleichung [7.4] und Gleichung [7.5] wissen wir, dass $\mathcal{C}_1$ und $\mathcal{C}_2$ nicht beide reguläre EG-Codes sein können. Hingegen ist es möglich, dass sowohl $\mathcal{C}_1$ als auch $\mathcal{C}_2$ ein zweifacher EG-Code ist. Dazu muss es p-Potenzen q_1, q_2 und $m_1, m_2 \in \mathbb{N}$, $r_1, r_2 \in \mathbb{N}_0$ geben, so dass

$$q_1^{m_1} = q_2^{m_2}, \quad q_1^{m_1 - r_1} = q_2^{m_2 - r_2}$$

erfüllt ist. Weiterhin müssen beide Codes vom selben Typ sein, sofern $p \neq 2$. Der Fall (a) folgt.

Nehmen wir an, $\mathcal{C}_1$ ist ein allgemeiner EG(m_1, q_1)-Code der Ordnung r_1, während $\mathcal{C}_2$ ein zweifacher EG(m_2, q_2)-Code der Ordnung r_2 ist. Für $\mathcal{C}_1$ gilt

$$2 \cdot \tau_{\mathrm{MLG}} \equiv \begin{cases} p-1 \bmod p & \mathcal{C}_1 \text{ vom Typ-1, } p \neq 2, m_1 - r_1 \text{ gerade,} \\ 1 \bmod p & \mathcal{C}_1 \text{ vom Typ-0, } p \neq 2, m_1 - r_1 \text{ gerade, ,} \\ 0 \bmod p & \text{sonst} \end{cases} \quad [7.13]$$

siehe Gleichung [7.2], Gleichung [7.4] und Gleichung [7.5]. Wohingegen bei $\mathcal{C}_2$ gilt, dass

$$2 \cdot \tau_{\mathrm{MLG}} \equiv \begin{cases} p-1 \bmod p & \text{zweifacher Typ-0-EG}(m_2, q_2)\text{-Code}, p \neq 2, \\ p-3 \bmod p & \text{zweifacher Typ-1-EG}(m_2, q_2)\text{-Code}, p \neq 2, , \\ p-2 \bmod p & p = 2 \end{cases} \quad [7.14]$$

siehe Gleichung [7.8] und Gleichung [7.12].

Ist p ein Teiler von $2 \cdot \tau_{\mathrm{MLG}}$, so ist

$$\frac{2 \cdot \tau_{\mathrm{MLG}}}{p} = \frac{q_1^{m_1 - r_1} - q_1}{p \cdot (q_1 - 1)} \equiv \begin{cases} 0 \bmod p & q_1 > p, \\ 1 \bmod p & q_1 = p \end{cases}. \quad [7.15]$$

Ist $p = 2$, so folgt mit Gleichung [7.15] und

$$\frac{2 \cdot \tau_{\mathrm{MLG}}}{p} = \frac{q^{m_2 - r_2} - 2}{2} \equiv 1 \bmod p,$$

dass $q_1 = 2$ sein muss. Damit ist $\mathcal{C}_1$ gleichzeitig auch ein zweifacher EG-Code. Das Codepaar in (b) ergibt sich als Spezialfall aus (a).

Ist $p = 3$ und 3 teilt τ_{MLG}, so ist

$$0 < \frac{2 \cdot \tau_{\mathrm{MLG}}}{p} = \frac{q^{m_2 - r_2} - 3}{3} \equiv 2 \bmod 3.$$

Widerspruch zu Gleichung [7.15].

Ist entweder $p = 3$ kein Teiler von τ_{MLG} oder $p > 3$, so ist wegen Gleichung [7.13] und Gleichung [7.14]

$$2 \cdot \tau_{\mathrm{MLG}} \equiv p - 1 \bmod p$$

und $m_1 - r_1$ gerade. Der reguläre EG-Code $\mathcal{C}_1$ ist vom Typ-1. Der zweifache EG-Code $\mathcal{C}_2$ ist vom Typ-0, so dass einerseits

$$2 \cdot \tau_{\mathrm{MLG}} = \frac{q_1^{m_1 - r_1} - q_1}{q_1 - 1} - 1$$

und andererseits

$$2 \cdot \tau_{\mathrm{MLG}} = q_2^{m_2 - r_2} - 1$$

gilt. Folglich ist $q_2^{m_2 - r_2} = q_1$, $m_1 - r_1 = 2$. Wegen

$$n = q_2^{(m_2 - r_2) m_1} = q_2^{m_2}$$

ist $m_2 - r_2$ ein Teiler von m_2. Die Kombination (c) folgt. □

Die für die Praxis wichtigsten Codes sind die binären Codes. Wir vergleichen die Dimensionen zweier Codes mit identischen Parametern n und τ_{MLG}.

Proposition 7.8.2. *Seien $\mathcal{C}_1, \mathcal{C}_2$ zwei verschiedene Codes, jeder entweder ein EG-Code oder ein zweifacher EG-Code über $\mathbb{F}_2$ mit identischen Parametern n und τ_{MLG}. Die zum Code $\mathcal{C}_i$ korrespondierenden Parameter m, q seien mit m_i, q_i bezeichnet, $i = 1, 2$, wobei ohne Beschränkung der Allgemeinheit $q_1 \leq q_2$. Entweder ist $\mathcal{C}_2$ vom Typ-0, $\mathcal{C}_1$ vom Typ-1 und*

$$\dim(\mathcal{C}_2) = \dim(\mathcal{C}_1) - 1 \qquad [7.16]$$

oder es ist

$$\dim(\mathcal{C}_2) \geq \dim(\mathcal{C}_1). \qquad [7.17]$$

Beweis. Wir betrachten die Kombinationen aus Proposition 7.8.1.

(a) Sei $\mathcal{C}_1$ ein zweifacher EG$(s \cdot m, q)$-Code der Ordnung $s \cdot r$ und sei $\mathcal{C}_2$ ein zweifacher EG(m, q^s)-Code der Ordnung r,

(i) entweder beide jeweils vom gleichen Typ mit $s > 1$

(ii) oder – vorausgesetzt $p = 2$ – von verschiedenen Typen;

Lin hat in [20, Theorem 7] gezeigt,

$$(q^s - 1)\, \omega_q\left(i \cdot p^j \bmod n\right) = (q - 1) \cdot \sum_{l=0}^{s-1} \omega_{q^s}\left(i \cdot p^j \cdot q^l \bmod n\right).$$

Die Ungleichung

$$0 < \max_{0 \le j < \log_p(q^s)} \omega_{q^s}\left(i \cdot p^j \bmod n\right) < (m-r)(q^s-1)$$

impliziert damit

$$\begin{aligned} 0 < \max_{0 \le j < \log_p(q)} \omega_q\left(i \cdot p^j \bmod n\right) &= \max_{0 \le j < \log_p(q)} \frac{q-1}{q^s-1} \cdot \sum_{l=0}^{s-1} \omega_{q^s}\left(i \cdot p^j \cdot q^l \bmod n\right) \\ &< \frac{q-1}{q^s-1} \cdot s \cdot (m-r)(q^s-1) \\ &= (m-r)s(q-1) \end{aligned}$$

Also ist jede Nullstelle ungleich eins des Erzeugerpolynoms von $\mathcal{C}_2$ auch eine Nullstelle des Erzeugerpolynoms von $\mathcal{C}_1$. Ist $\mathcal{C}_2$ vom Typ-0 und $\mathcal{C}_1$ vom Typ-1, so ist 1 eine Nullstelle des Erzeugerpolynoms von $\mathcal{C}_2$, nicht aber von $\mathcal{C}_1$. Es folgen Ungleichung [7.17] auf der vorherigen Seite und Ungleichung [7.16].

(b) Ein EG$(s \cdot m, 2)$-Code der Ordnung $s \cdot r$ ist gleichzeitig ein zweifacher EG$(s \cdot m, 2)$-Code der Ordnung $s \cdot r$. Die Behauptung ergibt sich aus (a).

(c) Der Fall entfällt, da $p = 2$. □

Für die binären Codes können wir zusammenfassend festhalten: Die zweifachen Typ-1-EG(m, q)-Codes mit möglichst großem q bieten die beste Informationsrate.

Zu den binären Codes kurzer Länge möchten wir konkrete Empfehlungen aussprechen. Dazu verweisen wir auf Tabelle 7.8. In dieser geben wir die Länge n und Minimaldistanz d vor und schauen, welcher Code die bessere Informationsrate zu welchem Aufwand beim Hybridverfahren unter der invertierten Abstufung bietet.

Für eine bessere Vergleichbarkeit des Aufwands bilden wir aus den beiden Werten, die unter der Hybriddecodierung die Anzahl der Fehlersummen $\mathcal{H}_{\mathtt{E}}$ und die Anzahl der Additionen/Subtraktionen $\mathcal{H}_{\pm}$ angeben, einen Gesamtwert $\mathcal{H}_{\mathrm{Op}}$. Ganz konkret gehen wir davon aus, dass unter dem EG-Code der Ordnung r maximal $q^{r+1}-1$ Operationen und unter dem zweifachen EG-Code

der Ordnung r maximal $q^r - 1$ Operationen benötigt werden, um eine einzige Fehlersumme zu berechnen. Der Wert $\mathcal{H}_{\mathrm{Op}}$ definiert sich also wie folgt,

$$\mathcal{H}_{\mathrm{Op}} \coloneqq \mathcal{H}_{\pm} + \mathcal{H}_{\mathtt{E}} \cdot \begin{cases} q^{r+1} - 1 & \mathcal{C} \text{ ist EG-Code,} \\ q^r - 1 & \mathcal{C} \text{ ist zweifacher EG-Code} \end{cases}.$$

Mehrere Zeilen der Tabelle werden zu einem Block zusammengefasst, wenn die Codes dieselben Parameter n und τ_{MLG} besitzen. Innerhalb eines jeden Blocks haben wir eine Tabellenzeile dann ausgegraut, wenn die Decodierung zum jeweiligen Code weniger performant ist als zu einem anderen Code des Blocks mit gleicher oder höherer k und der Code daher nicht zu empfehlen ist. Zusammenfassend halten wir mit Blick auf Tabelle Tabelle 7.8 folgende Aussagen für binäre Codes der Länge nicht größer als 511 fest:

(a) Einen regulären EG-Code zu verwenden, ist nur dann empfehlenswert, wenn es keinen zweifachen EG-Code mit den gleichen Parametern n und τ_{MLG} gibt. Ansonsten ist der zweifache EG-Code performanter als der allgemeine EG-Code gleichen Typs und daher diesem vorzuziehen.

(b) Ein Typ-1-Code hat gegenüber dem korrespondierenden Typ-0-Code eine um eins höhere Dimension und damit eine bessere Informationsrate; der Decodieraufwand ist wenn überhaupt nur geringfügig höher.

(c) Bei einem 1-Fehler-korrigierenden zweifachen EG-Code – unerheblich welchen Typs – ist stets $q = 2$ zu präferieren, um den Decodieraufwand bei gleichen Parametern n, $\tau_{\mathrm{MLG}} = 1$, k zu minimieren. Der geringere Aufwand ist dadurch begründet, dass bei $\tau_{\mathrm{MLG}} = 1$ stets 2-Mehrheitsentscheidungen getroffen werden und jede dieser einer gewöhnlichen Und-Verknüpfung entspricht.

(d) Sollen mehrere Fehler korrigiert werden können, so ist beim zweifachen EG(m, q)-Code die Wahl des Typs und des Parameters q hinsichtlich der Informationsrate und der Performanz abzuwägen: Bei gegebenem Typ wächst mit größerem q sowohl die Informationsrate als auch der Wert $\mathcal{H}_{\mathrm{Op}}$ monoton, wohingegen $\mathcal{H}_{\mu}$ häufig fällt. Es ist nicht auszuschließen, dass mit größerem q die Informationsrate und die Performanz steigen.

Tabelle 7.8: Vergleich des EG-Codes und des zweifachen EG-Codes bei gegebener Länge n und gegebener Minimaldistanz d.

Code	n	τ_{MLG}	k	$\frac{k}{n}$	$\mathcal{H}_{\mu,I}$	$\mathcal{H}_{\mathrm{Op},I}$	ς	$\lvert T_I\rvert$	η	q	r
2-f. Typ-1-EG	3	1	1	0,33	1	3	1	1	2	2	0
Typ-1-EG	3	1	1	0,33	1	5	1	1	2	2	0
2-f. Typ-1-EG	7	3	1	0,14	1	7	1	1	6	2	0
Typ-1-EG	7	3	1	0,14	1	27	1	1	6	2	0
2-f. Typ-0-EG	7	1	3	0,43	6	19	2	2	2	2	1
Typ-0-EG	7	1	3	0,43	8	31	2	2	2	2	1
2-f. Typ-1-EG	7	1	4	0,57	5	23	2	2	2	2	1
Typ-1-EG	7	1	4	0,57	5	38	2	2	2	2	1
2-f. Typ-0-EG	15	3	4	0,27	10	107	2	2	6	2	1
Typ-0-EG	15	3	4	0,27	16	479	2	2	6	2	1
2-f. Typ-1-EG	15	3	5	0,33	9	111	2	2	6	2	1
Typ-1-EG	15	3	5	0,33	9	496	2	2	6	2	1
Typ-1-EG	15	2	7	0,47	11	57	1	1	4	4	0
2-f. Typ-0-EG	15	1	10	0,67	14	41	2	2	2	4	1
2-f. Typ-0-EG	15	1	10	0,67	17	32	2	1	2	2	2
Typ-0-EG	15	1	10	0,67	19	60	2	1	2	2	2
2-f. Typ-1-EG	15	1	11	0,73	15	45	2	2	2	4	1
2-f. Typ-1-EG	15	1	11	0,73	18	36	2	1	2	2	2
Typ-1-EG	15	1	11	0,73	18	69	2	1	2	2	2
2-f. Typ-0-EG	31	7	5	0,16	18	475	2	2	14	2	1
Typ-0-EG	31	7	5	0,16	32	4.927	2	2	14	2	1
2-f. Typ-1-EG	31	7	6	0,19	17	479	2	2	14	2	1
Typ-1-EG	31	7	6	0,19	17	4.964	2	2	14	2	1
2-f. Typ-0-EG	31	3	15	0,48	37	192	2	1	6	2	2
Typ-0-EG	31	3	15	0,48	43	1.044	2	1	6	2	2
2-f. Typ-1-EG	31	3	16	0,52	38	196	2	1	6	2	2
Typ-1-EG	31	3	16	0,52	38	1.071	2	1	6	2	2
2-f. Typ-0-EG	31	1	25	0,81	43	136	3	2	2	2	3
Typ-0-EG	31	1	25	0,81	47	256	3	2	2	2	3

Fortsetzung der Tabelle auf der nächsten Seite

Tabelle 7.8 – Fortsetzung der Tabelle

Code	n	τ_{MLG}	k	$\frac{k}{n}$	$\mathcal{H}_{\mu,I}$	$\mathcal{H}_{\mathrm{Op},I}$	ς	$\lvert T_I\rvert$	η	q	r
2-f. Typ-1-EG	31	1	26	0,84	46	231	3	2	2	2	3
Typ-1-EG	31	1	26	0,84	46	438	3	2	2	2	3
2-f. Typ-0-EG	63	15	6	0,1	34	1.979	2	2	30	2	1
Typ-0-EG	63	15	6	0,1	64	44.159	2	2	30	2	1
2-f. Typ-1-EG	63	15	7	0,11	33	1.983	2	2	30	2	1
Typ-1-EG	63	15	7	0,11	33	44.236	2	2	30	2	1
Typ-1-EG	63	10	13	0,21	43	1.005	1	1	20	4	0
2-f. Typ-0-EG	63	7	21	0,33	77	896	2	1	14	2	2
Typ-0-EG	63	7	21	0,33	91	11.172	2	1	14	2	2
2-f. Typ-1-EG	63	7	22	0,35	78	900	2	1	14	2	2
Typ-1-EG	63	7	22	0,35	78	11.235	2	1	14	2	2
2-f. Typ-0-EG	63	7	23	0,37	58	953	2	2	14	4	1
2-f. Typ-1-EG	63	7	24	0,38	59	957	2	2	14	4	1
Typ-1-EG	63	4	37	0,59	55	497	1	1	8	8	0
2-f. Typ-0-EG	63	3	41	0,65	69	384	2	1	6	2	3
Typ-0-EG	63	3	41	0,65	75	2.196	2	1	6	2	3
2-f. Typ-1-EG	63	3	42	0,67	70	392	2	1	6	2	3
Typ-1-EG	63	3	42	0,67	70	2.247	2	1	6	2	3
2-f. Typ-0-EG	63	3	44	0,7	62	433	2	2	6	8	1
2-f. Typ-1-EG	63	3	45	0,71	63	441	2	2	6	8	1
Typ-0-EG	63	2	47	0,75	96	1.053	2	2	4	4	1
Typ-1-EG	63	2	48	0,76	99	1.242	2	2	4	4	1
2-f. Typ-0-EG	63	1	56	0,89	75	264	3	2	2	2	4
Typ-0-EG	63	1	56	0,89	79	512	3	2	2	2	4
2-f. Typ-0-EG	63	1	56	0,89	91	280	3	2	2	4	2
2-f. Typ-1-EG	63	1	57	0,9	78	447	3	2	2	2	4
Typ-1-EG	63	1	57	0,9	78	870	3	2	2	2	4
2-f. Typ-1-EG	63	1	57	0,9	108	459	3	2	2	4	2
2-f. Typ-0-EG	127	15	28	0,22	157	3.840	2	1	30	2	2
Typ-0-EG	127	15	28	0,22	187	101.700	2	1	30	2	2
2-f. Typ-1-EG	127	15	29	0,23	158	3.844	2	1	30	2	2

Fortsetzung der Tabelle auf der nächsten Seite

Tabelle 7.8 – Fortsetzung der Tabelle

Code	n	τ_{MLG}	k	$\frac{k}{n}$	$\mathcal{H}_{\mu,I}$	$\mathcal{H}_{\mathrm{Op},I}$	ς	$\lvert T_I\rvert$	η	q	r
Typ-1-EG	127	15	29	0,23	158	101.835	2	1	30	2	2
2-f. Typ-0-EG	127	7	63	0,5	141	1.792	2	1	14	2	3
Typ-0-EG	127	7	63	0,5	155	23.716	2	1	14	2	3
2-f. Typ-1-EG	127	7	64	0,5	142	1.800	2	1	14	2	3
Typ-1-EG	127	7	64	0,5	142	23.835	2	1	14	2	3
2-f. Typ-0-EG	127	3	98	0,77	187	4.680	3	2	6	2	4
Typ-0-EG	127	3	98	0,77	223	27.072	3	2	6	2	4
2-f. Typ-1-EG	127	3	99	0,78	190	5.579	3	2	6	2	4
Typ-1-EG	127	3	99	0,78	190	32.284	3	2	6	2	4
2-f. Typ-0-EG	127	1	119	0,94	163	512	3	1	2	2	5
Typ-0-EG	127	1	119	0,94	167	1.016	3	1	2	2	5
2-f. Typ-1-EG	127	1	120	0,94	181	864	3	1	2	2	5
Typ-1-EG	127	1	120	0,94	181	1.719	3	1	2	2	5
2-f. Typ-0-EG	255	31	36	0,14	317	15.872	2	1	62	2	2
Typ-0-EG	255	31	36	0,14	379	864.900	2	1	62	2	2
2-f. Typ-1-EG	255	31	37	0,15	318	15.876	2	1	62	2	2
Typ-1-EG	255	31	37	0,15	318	865.179	2	1	62	2	2
2-f. Typ-0-EG	255	31	44	0,17	234	16.121	2	2	62	4	1
2-f. Typ-1-EG	255	31	45	0,18	235	16.125	2	2	62	4	1
2-f. Typ-0-EG	255	15	92	0,36	285	7.680	2	1	30	2	3
Typ-0-EG	255	15	92	0,36	315	216.900	2	1	30	2	3
2-f. Typ-1-EG	255	15	93	0,36	286	7.688	2	1	30	2	3
Typ-1-EG	255	15	93	0,36	286	217.155	2	1	30	2	3
Typ-0-EG	255	10	126	0,49	1.056	97.437	2	2	20	4	1
Typ-1-EG	255	10	127	0,5	1.075	100.734	2	2	20	4	1
Typ-1-EG	255	8	175	0,69	239	4.065	1	1	16	16	0
2-f. Typ-0-EG	255	7	162	0,64	269	3.584	2	1	14	2	4
Typ-0-EG	255	7	162	0,64	283	48.804	2	1	14	2	4
2-f. Typ-1-EG	255	7	163	0,64	270	3.600	2	1	14	2	4
Typ-1-EG	255	7	163	0,64	270	49.035	2	1	14	2	4
2-f. Typ-0-EG	255	7	170	0,67	269	3.584	2	1	14	4	2

Fortsetzung der Tabelle auf der nächsten Seite

Tabelle 7.8 – Fortsetzung der Tabelle

Code	n	τ_{MLG}	k	$\frac{k}{n}$	$\mathcal{H}_{\mu,I}$	$\mathcal{H}_{\mathrm{Op},I}$	ς	$\lvert T_I\rvert$	η	q	r
2-f. Typ-1-EG	255	7	171	0,67	270	3.600	2	1	14	4	2
2-f. Typ-0-EG	255	7	190	0,75	254	3.809	2	2	14	16	1
2-f. Typ-1-EG	255	7	191	0,75	255	3.825	2	2	14	16	1
2-f. Typ-0-EG	255	3	218	0,85	315	9.288	3	2	6	2	5
Typ-0-EG	255	3	218	0,85	351	54.720	3	2	6	2	5
2-f. Typ-1-EG	255	3	219	0,86	318	11.067	3	2	6	2	5
Typ-1-EG	255	3	219	0,86	318	65.212	3	2	6	2	5
Typ-0-EG	255	2	230	0,9	303	4.080	2	1	4	4	2
Typ-1-EG	255	2	231	0,91	310	4.785	2	1	4	4	2
2-f. Typ-0-EG	255	1	246	0,96	283	1.048	3	2	2	4	3
2-f. Typ-0-EG	255	1	246	0,96	291	1.024	3	1	2	2	6
Typ-0-EG	255	1	246	0,96	295	2.040	3	1	2	2	6
2-f. Typ-1-EG	255	1	247	0,97	300	1.755	3	2	2	4	3
2-f. Typ-1-EG	255	1	247	0,97	309	1.728	3	1	2	2	6
Typ-1-EG	255	1	247	0,97	309	3.447	3	1	2	2	6
Typ-1-EG	511	36	139	0,27	439	32.697	1	1	72	8	0
2-f. Typ-0-EG	511	31	129	0,25	573	31.744	2	1	62	2	3
Typ-0-EG	511	31	129	0,25	635	1.848.964	2	1	62	2	3
2-f. Typ-1-EG	511	31	130	0,25	574	31.752	2	1	62	2	3
Typ-1-EG	511	31	130	0,25	574	1.849.491	2	1	62	2	3
2-f. Typ-0-EG	511	31	183	0,36	502	32.241	2	2	62	8	1
2-f. Typ-1-EG	511	31	184	0,36	503	32.249	2	2	62	8	1
2-f. Typ-0-EG	511	15	255	0,5	541	15.360	2	1	30	2	4
Typ-0-EG	511	15	255	0,5	571	447.300	2	1	30	2	4
2-f. Typ-1-EG	511	15	256	0,5	542	15.376	2	1	30	2	4
Typ-1-EG	511	15	256	0,5	542	447.795	2	1	30	2	4
2-f. Typ-0-EG	511	7	381	0,75	763	100.744	3	2	14	2	5
Typ-0-EG	511	7	381	0,75	959	1.386.112	3	2	14	2	5
2-f. Typ-1-EG	511	7	382	0,75	766	108.435	3	2	14	2	5
Typ-1-EG	511	7	382	0,75	766	1.491.960	3	2	14	2	5
Typ-0-EG	511	4	447	0,87	896	33.145	2	2	8	8	1

Fortsetzung der Tabelle auf der nächsten Seite

Tabelle 7.8 – Fortsetzung der Tabelle

Code	n	τ_{MLG}	k	$\frac{k}{n}$	$\mathcal{H}_{\mu,I}$	$\mathcal{H}_{\mathrm{Op},I}$	ς	$\lvert T_I\rvert$	η	q	r
Typ-1-EG	511	4	448	0,88	935	36.722	2	2	8	8	1
2-f. Typ-0-EG	511	3	465	0,91	739	18.432	3	1	6	2	6
Typ-0-EG	511	3	465	0,91	775	109.944	3	1	6	2	6
2-f. Typ-1-EG	511	3	466	0,91	777	21.952	3	1	6	2	6
Typ-1-EG	511	3	466	0,91	777	130.977	3	1	6	2	6
2-f. Typ-0-EG	511	3	474	0,93	883	18.768	3	2	6	8	2
2-f. Typ-1-EG	511	3	475	0,93	952	22.295	3	2	6	8	2
2-f. Typ-0-EG	511	1	501	0,98	599	4.112	4	2	2	2	7
Typ-0-EG	511	1	501	0,98	607	8.192	4	2	2	2	7
2-f. Typ-1-EG	511	1	502	0,98	649	10.413	4	2	2	2	7
Typ-1-EG	511	1	502	0,98	649	20.754	4	2	2	2	7

Kapitel 8

Zusammenfassung und Ausblick

8.1 Zusammenfassung

Diese Arbeit befasst sich mit der Majority-Logic-Decodierung für Euklidische-Geometrie-Codes. Zunächst haben wir daher Eigenschaften der Majoritätsfunktion aufgezeigt, anschließend diskutierten wir verschiedene Lösungen, Mehrheitsentscheidungen in Soft- und Hardware zu realisieren.

Drei Majority-Logic-Decodierverfahren haben wir vorgestellt, die klassische, die verbesserte und die hybride Decodierung. Die letzteren beiden haben wir selbst entwickelt. Die Korrektheit aller drei Verfahren haben wir mathematisch bewiesen. Veranschaulicht haben wir die Verfahren durch mehrere Beispiele. Für alle drei Verfahren haben wir den genauen Decodieraufwand aufgeschlüsselt und offengelegt, dass die von uns entwickelten Verfahren eines wesentlich geringeren Rechenaufwands als das klassische Verfahren bedürfen. Darüber hinaus haben wir weitere Optimierungsmöglichkeiten für die verbesserte und die hybride Decodierung aufgezeigt.

Für jedes dieser drei Verfahren muss eine konkrete Abstufung der Majority-Logic-Stufen festgelegt werden. Daher haben wir uns die beiden aus der Literatur bekannten Abstufungen von Reed und Chen genauer angesehen und eine

weitere Abstufung entwickelt, die wir als invertiert bezeichnet haben, siehe Kapitel 6.

Die invertierte Abstufung zeichnet sich dadurch aus, dass die Stufen zunächst unabhängig von dem durch den Code definierten Parameter $D_{\mathcal{C}}$ und nur in Abhängigkeit von m definiert werden, so dass Teilbäume der zu konstruierenden Decodierbäume bei festem m und variablem $D_{\mathcal{C}}$ bis zu einer gewissen Stufe wiederverwendet werden können. Damit lässt sich der Speicherbedarf verringern, wenn man die Decodierer für verschiedene Codes einsetzen möchte. Ein weiterer entscheidender Vorteil der invertierten Abstufung ist, dass der Decodieraufwand hinsichtlich der Anzahl der heranzuziehenden Fehlersummen und der zu treffenden Mehrheitsentscheidungen kleiner oder maximal genauso hoch ist wie unter den anderen beiden Abstufungen. Gleichzeitig sind im Vergleich zu den Abstufungen von Reed und Chen die Fehlerkorrektureigenschaften genauso gut und die Anzahl der Majority-Logic-Stufen, die Aufschluss über die parallele Laufzeit gibt, nicht höher.

Wir haben den Decodieraufwand der einzelnen Verfahren und der einzelnen Abstufungen in Abschnitt 6.3 in Beziehung gesetzt und untereinander verglichen. Relationen haben wir mathematisch bewiesen und veranschaulicht, siehe beispielsweise Abbildung 6.4. Zusammenfassend halten wir fest, dass die wenigsten Mehrheitsentscheidungen und die wenigsten Fehlersummen bei der hybriden Decodierung unter der invertierten Abstufung benötigt werden.

In Kapitel 7 haben wir gezeigt, warum und wie die drei Verfahren bei verschiedenen Euklidische-Geometrie-Codes wie den regulären EG-Codes, zweifachen EG-Codes, zyklischen Reed-Muller-Codes, Reed-Muller-Codes, binären Hamming-Codes, verallgemeinerten EG-Codes anwendbar sind. Für eine Auswahl verschiedener Codes haben wir den Decodieraufwand der verbesserten und der hybriden Decodierung unter der invertierten Abstufung explizit angegeben. Darauf aufbauend haben wir Empfehlungen ausgesprochen, welcher Code bei gegebenen Parametern n und τ_{MLG} vorzugsweise zu wählen ist.

8.2 Ausblick

8.2.1 Decodierung an Informationspositionen

Wie wir bereits in Abschnitt 4.4 erwähnt haben, lässt sich die Decodierung möglicherweise noch effizienter gestalten, wenn nur an den Informationspositionen (IP) decodiert wird. Ohne Weiteres ist es möglich, die vorgestellten Decodierverfahren so anzupassen, dass im letzten Majority-Logic-Schritt, nur die Fehlersymbole an Informationspositionen berechnet werden. In welchem Maß durch die Einschränkung auf Informationspositionen in diesem letzten Schritt auch in vorhergehenden Majority-Logic-Schritten Mehrheitsentscheidungen eingespart werden können, bleibt an dieser Stelle offen.

In [13] wurde für einige Reed-Muller-Codes $\mathrm{RM}(r, m)$ kurzer Länge, $2r \leq m$, untersucht, wie viele Mehrheitsentscheidungen mindestens und höchstens benötigt werden, wenn nur an Informationspositionen mit Hilfe des verbesserten Verfahrens decodiert wird. Dazu wurde nicht nur der letzte Majority-Logic-Schritt berücksichtigt sondern auch der erste der beiden. Durch eine aufwändige, zum Teil computergestützte Analyse wurden möglichst kleine Familien von r-dimensionalen affinen Räumen ermittelt, mit Hilfe derer die gesamte Information `I` rekonstruiert werden kann. Die Ergebnisse in [13, Table V] beweisen, dass sich der Bedarf an Mehrheitsentscheidungen signifikant im Vergleich zum Decodieren an allen Positionen reduziert.

Jedoch zeigt sich für die in [13, Table V] gelisteten Reed-Muller-Codes auch, dass mit Hilfe der Hybriddecodierung unter der invertierten Abstufung die Anzahl der Mehrheitsentscheidungen mindestens in gleichem Maß, häufig stärker reduziert werden kann als bei einer Einschränkung auf Informationspositionen in allen Majority-Logic-Schritten mit verbesserter Decodierung wie in [13], siehe in Tabelle 8.1. Der Effekt, den die Hybriddecodierung bietet, ist demnach häufig größer als der durch die Reduktion auf Informationspositionen erzielte. Verstärkt wird der Vorteil der Hybriddecodierung zusätzlich, wenn man sich im letzten Majority-Logic-Schritt – sofern $0 \notin T$ – auf die Decodierung

an Informationspositionen beschränkt, siehe Zahlen in Klammern in Spalte „Hybridverfahren“ in Tabelle 8.1.

Tabelle 8.1: Anzahl der Mehrheitsentscheidungen beim Decodieren an Informationspositionen: Vergleich zwischen Hybridverfahren und Ergebnissen aus [13, Table I; §IV]

r	m	Hybridverfahren $\mathcal{H}_{\mu,I}^{\mathrm{RM}}$	optimierte Decodierung an IP [13, Table I; Table V; §IV] $\mu(r,m)+k$
1	3	6 (6)	$8 = 4 + 4$
1	≥ 4	$2^{m-1}+2$ $(2^{m-1}+2)$	$\frac{(m+1)(2^m-m-4)}{2} + m + 1$
2	4	18 (13)	$18 = 7 + 11$
2	5	38 (22)	$46 = 30 + 16$
2	6	78 (36)	$\geq 151 = 129 + 22$
2	7	158 (59)	$\geq 493 = 464 + 29$
3	6	70 (48)	$\geq 75 = 33 + 42$
3	7	142 (78)	$\geq 229 = 165 + 64$

8.2.2 Differenz zwischen τ_{MLG} und $\tau_{\max}$

Wir wollen die Differenz zwischen den Fehlerkorrekturgrenzen angeben, die durch Code, $\tau_{\max}$, und durch Decodierverfahren, τ_{MLG}, definiert werden. In Abschnitt 7.3 haben wir gesehen, dass τ_{MLG} und $\tau_{\max}$ beim zweifachen EG-Code zusammenfallen. Beim Euklidischen-Geometrie-Code jedoch ist τ_{MLG} bestenfalls gleich $\tau_{\max}$, häufig jedoch kleiner. Die Differenz $\tau_{\max} - \tau_{\mathrm{MLG}}$ beträgt in jedem der drei Fälle

- $p \neq 2$, $m - r$ gerade, Typ 1
- $p = 2$, Typ 0
- $m - r$ ungerade, Typ 0

mindestens

$$\left\lfloor (p-1)/2 \cdot q^{m-r-2} - 1/2 \cdot \sum_{i=0}^{m-r-3} q^i \right\rfloor$$

und ansonsten mindestens

$$\left\lfloor (p-1)/2 \cdot q^{m-r-2} - 1/2 - 1/2 \cdot \sum_{i=0}^{m-r-3} q^i \right\rfloor,$$

siehe Ungleichung [7.1] auf Seite 121 und Gleichung [7.2] auf Seite 121 für den Typ-1-Code beziehungsweise Ungleichung [7.3] auf Seite 124, Gleichung [7.4] auf Seite 126 und Gleichung [7.5] auf Seite 126 für den Typ-0-Code.

Die Differenz wächst, wenn q oder $m-r$ wächst. Wenn keiner der drei folgenden Fälle eintritt,

- $p = 2$, $m - r = 2$;
- $q = p = 2$;
- $p = 3$, $m - r = 2$ beim Typ-0-EG-Code,

ist die Differenz $\tau_{\max} - \tau_{\mathrm{MLG}}$ größer null. Demzufolge bieten die EG-Codes in diesem Fall bessere Fehlerkorrektureigenschaften, als die präsentierten Majority-Logic-Decodierverfahren fähig sind auszunutzen. Dann könnte ein Decodierer von Vorteil sein, der nach dem Prinzip der Hamming-Decodierung bis zur Grenze von $\tau_{\max}$ Fehlern korrekt arbeitet. Die Frage ist, ob es für diesen Fall, schnelle Hamming-Decodierverfahren gibt und wie diese aussehen. Statt auf ein anderes Decodierverfahren auszuweichen, ist das Verwenden eines zweifachen EG-Codes zu erwägen.

8.2.3 Überschreitung der Fehlerkorrekturgrenze τ_{MLG}

Wir fragen uns, was passiert, wenn mehr als τ_{MLG} Positionen nach der Übertragung fehlerbehaftet sind. Es kann der glückliche Fall eintreten, dass mit den Majority-Logic-Decodierverfahren aus Kapitel 5 auf Seite 45 das ursprüngliche Codewort rekonstruiert wird, selbst wenn mehr als τ_{MLG} Fehler auftreten (im Gegensatz zu sogenannten *Bounded-Distance-Decodern*).

Werfen wir dazu noch einmal einen Blick auf Beispiel 7.3.1.

Beispiel 8.2.1. Wir übernehmen die Notationen aus Beispiel 7.3.1 und betrachten erneut den den zweifachen Typ-1-EG$(5, 2)$-Code respektive den zyklischen Reed-Muller-Code der Ordnung zwei, RM$^*(2, 5)$. Angenommen, das gleiche Codewort wie in Beispiel 7.3.1 wurde übertragen,

$$\mathsf{c} \coloneqq (11110101111100010000000000000000) \cong g(X).$$

Wir erinnern uns, dass in Beispiel 7.3.1 drei Fehler an den Positionen $\{0, 15, 30\}$ aufgetreten sind. Von Proposition 4.3.1 auf Seite 39 wissen wir, dass die dortigen Schätzwerte der Fehlersummen tatsächlich die richtigen Fehlersummen sind, da $\tau_{\mathrm{MLG}} = 3$.

Angenommen, es tritt ein weiterer Fehler an Position $i \in \mathbb{Z}_{31}$, $i \neq 0, 15, 30$, auf. Sechs der sieben Mehrheitsentscheidungen auf Seite 134 liefern den gleichen Mehrheitswert. Nur die Mehrheitsentscheidung, die dem Unterraum U_1 zugewiesen ist, gibt gegebenenfalls einen anderen Mehrheitswert aus, nämlich dann, wenn γ^i in

$$\bigcup_{\mathbf{u}' \in \{4,5,8,9\}} \mathbf{u}' + \langle 2, 7 \rangle_{\mathbb{F}_2}$$

liegt. Ist jedoch

$$\gamma^i \in \bigcup_{\mathbf{u}' \in \{0,1,12,13\}} \mathbf{u}' + \langle 2, 7 \rangle_{\mathbb{F}_2},$$

was mit Blick auf Tabelle 7.3 äquivalent ist zu

$$i \in \{1, 4, 8, 10, 12, 13, 14, 17, 18, 20, 23, 24\},$$

dann bleiben alle sieben Mehrheitswerte unverändert. In diesem Fall ändern sich nur die Schätzwerte von Fehlersummen zu den zweidimensionalen affinen Räumen, die γ^i enthalten. Folglich sind alle diese Schätzwerte der Fehlersummen zu den betrachteten zweidimensionalen affinen Räumen tatsächlich die korrekten Werte.

Dies bedeutet allerdings noch nicht, dass auch die einzelnen Fehlersymbole korrekt geschätzt werden, da auch in der nächsten Majority-Logic-Stufe die Mehrheitsentscheidungen auf nur sechs Werten basieren.

Angenommen,

$$\mathtt{z} \coloneqq (01110101011100000000000000000001)$$

mit Fehlern an den Positionen $\{0, 8, 15, 30\}$ wurde empfangen.

Wir greifen auf den Decodierbaum aus Beispiel 7.3.1 zurück und berechnen die benötigten $49 = 7 \cdot 7$ Checksummen. Anhand dieser treffen wir sieben 6-Mehrheitsentscheidungen und addieren die Mehrheitswerte zu den entsprechenden Checksummen. Auf diese Art erhalten wir Schätzwerte (SW) für die Fehlersummen zu den zweidimensionalen affinen Räumen. Wir haben jene Werte eingekreist, die sich im Vergleich zu Beispiel 7.3.1 verändern.

$U_0 = \langle 1, 30 \rangle_{\mathbb{F}_2}$	$\mathbf{u}' \in U' \setminus \{0\}$:	2	4	6	8	10	12	14	
	$\mathtt{z} \circ \chi_{\mathbf{u}'+U_0}$:	1	1	1	1	1	(1)	1	
	μ^0 (		1	1	1	1	(1)	1	) = 1
	SW $\mathtt{E} \circ \chi_{\mathbf{u}'+U_0}$:	0	0	0	0	0	(0)	0	
$U_1 = \langle 2, 17 \rangle_{\mathbb{F}_2}$	$\mathbf{u}' \in U' \setminus \{0\}$:	1	4	5	8	9	12	13	
	$\mathtt{z} \circ \chi_{\mathbf{u}'+U_1}$:	1	1	1	1	1	0	(1)	
	μ^0 (		1	1	1	1	0	(1)	) = 1
	SW $\mathtt{E} \circ \chi_{\mathbf{u}'+U_1}$:	0	0	0	0	0	1	(0)	
$U_2 = \langle 3, 20 \rangle_{\mathbb{F}_2}$	$\mathbf{u}' \in U' \setminus \{0\}$:	1	4	5	8	9	12	13	
	$\mathtt{z} \circ \chi_{\mathbf{u}'+U_2}$:	0	0	1	1	0	0	(1)	
	μ^0 (		0	1	1	0	0	(1)	) = 0
	SW $\mathtt{E} \circ \chi_{\mathbf{u}'+U_2}$:	0	0	1	1	0	0	(1)	
$U_3 = \langle 4, 18 \rangle_{\mathbb{F}_2}$	$\mathbf{u}' \in U' \setminus \{0\}$:	1	2	3	8	9	10	11	
	$\mathtt{z} \circ \chi_{\mathbf{u}'+U_3}$:	1	0	0	0	(0)	0	0	
	μ^0 (		0	0	0	(0)	0	0	) = 0
	SW $\mathtt{E} \circ \chi_{\mathbf{u}'+U_3}$:	1	0	0	0	(0)	0	0	
$U_4 = \langle 5, 25 \rangle_{\mathbb{F}_2}$	$\mathbf{u}' \in U' \setminus \{0\}$:	1	2	3	8	9	10	11	
	$\mathtt{z} \circ \chi_{\mathbf{u}'+U_4}$:	1	0	1	(1)	0	0	1	
	μ^0 (		0	1	(1)	0	0	1	) = 0
	SW $\mathtt{E} \circ \chi_{\mathbf{u}'+U_4}$:	1	0	1	(1)	0	0	1	

$U_5 = \langle 6, 27 \rangle_{\mathbb{F}_2}$	$\mathbf{u}' \in U' \setminus \{0\}$:	1	2	3	8	9	10	11	
	$\mathtt{Z} \circ \chi_{\mathbf{u}'+U_5}$:	0	0	1	1	1	1	(0)	
	μ^0 (		0	1	1	1	1	(0)	) = 1
	SW $\mathtt{E} \circ \chi_{\mathbf{u}'+U_5}$:	1	1	0	0	0	0	(1)	
$U_6 = \langle 8, 16 \rangle_{\mathbb{F}_2}$	$\mathbf{u}' \in U' \setminus \{0\}$:	1	2	3	4	5	6	7	
	$\mathtt{Z} \circ \chi_{\mathbf{u}'+U_6}$:	1	1	0	0	(1)	0	1	
	μ^0 (		1	0	0	(1)	0	1	) = 0
	SW $\mathtt{E} \circ \chi_{\mathbf{u}'+U_6}$:	1	1	0	0	(1)	0	1	

Die Fehlersymbole erhaltend wir nun mittels 31 Mehrheitsentscheidungen. Wieder haben wir die Werte eingekreist, die sich gegenüber Beispiel 7.3.1 verändern.

$$\text{SW } \mathtt{E}_0 \coloneqq \mu^0 \left(\text{SW } \mathtt{E} \circ \chi_{1+U_i} \mid i \in \mathbb{Z}_7 \setminus \{0\}\right) = \mu^0 \left(0,0,1,1,1,1\right) = 1$$

$$\text{SW } \mathtt{E}_1 \coloneqq \mu^0 \left(\text{SW } \mathtt{E} \circ \chi_{2+U_i} \mid i \in \mathbb{Z}_7 \setminus \{1\}\right) = \mu^0 \left(0,0,0,0,1,1\right) = 0$$

$$\text{SW } \mathtt{E}_2 \coloneqq \mu^0 \left(\text{SW } \mathtt{E} \circ \chi_{4+U_i} \mid i \in \mathbb{Z}_7 \setminus \{3\}\right) = \mu^0 \left(0,0,0,1,1,0\right) = 0$$

$$\text{SW } \mathtt{E}_3 \coloneqq \mu^0 \left(\text{SW } \mathtt{E} \circ \chi_{8+U_i} \mid i \in \mathbb{Z}_7 \setminus \{6\}\right) = \mu^0 \left(0,0,1,0,(1),0\right) = 0$$

$$\text{SW } \mathtt{E}_4 \coloneqq \mu^0 \left(\text{SW } \mathtt{E} \circ \chi_{16+U_i} \mid i \in \mathbb{Z}_7 \setminus \{6\}\right) = \mu^0 \left(0,0,0,0,0,(1)\right) = 0$$

$$\text{SW } \mathtt{E}_5 \coloneqq \mu^0 \left(\text{SW } \mathtt{E} \circ \chi_{5+U_i} \mid i \in \mathbb{Z}_7 \setminus \{4\}\right) = \mu^0 \left(0,0,1,1,0,(1)\right) = 0$$

$$\text{SW } \mathtt{E}_6 \coloneqq \mu^0 \left(\text{SW } \mathtt{E} \circ \chi_{10+U_i} \mid i \in \mathbb{Z}_7 \setminus \{6\}\right) = \mu^0 \left(0,0,0,0,0,0\right) = 0$$

$$\text{SW } \mathtt{E}_7 \coloneqq \mu^0 \left(\text{SW } \mathtt{E} \circ \chi_{20+U_i} \mid i \in \mathbb{Z}_7 \setminus \{2\}\right) = \mu^0 \left(0,0,0,(1),0,0\right) = 0$$

$$\begin{aligned}\text{SW } \mathtt{E}_8 &\coloneqq \mu^0 \left(\text{SW } \mathtt{E} \circ \chi_{13+U_i} \mid i \in \mathbb{Z}_7 \setminus \{6\}\right) \\ &= \mu^0 \left((0),(0),(1),(0),(1),(1)\right) = 0\end{aligned}$$

$$\text{SW } \mathtt{E}_9 \coloneqq \mu^0 \left(\text{SW } \mathtt{E} \circ \chi_{26+U_i} \mid i \in \mathbb{Z}_7 \setminus \{6\}\right) = \mu^0 \left(0,0,(1),0,1,1\right) = 0$$

$$\text{SW } \mathtt{E}_{10} \coloneqq \mu^0 \left(\text{SW } \mathtt{E} \circ \chi_{17+U_i} \mid i \in \mathbb{Z}_7 \setminus \{1\}\right) = \mu^0 \left(0,1,0,(1),0,1\right) = 0$$

$$\text{SW } \mathtt{E}_{11} \coloneqq \mu^0 \left(\text{SW } \mathtt{E} \circ \chi_{7+U_i} \mid i \in \mathbb{Z}_7 \setminus \{6\}\right) = \mu^0 \left(0,0,0,0,0,1\right) = 0$$

$$\text{SW } \mathtt{E}_{12} \coloneqq \mu^0 \left(\text{SW } \mathtt{E} \circ \chi_{14+U_i} \mid i \in \mathbb{Z}_7 \setminus \{6\}\right) = \mu^0 \left(0,1,(1),0,1,0\right) = 0$$

$$\text{SW } \mathtt{E}_{13} \coloneqq \mu^0 \left(\text{SW } \mathtt{E} \circ \chi_{28+U_i} \mid i \in \mathbb{Z}_7 \setminus \{4\}\right) = \mu^0 \left(0,(0),1,0,1,0\right) = 0$$

$$\text{SW } \mathtt{E}_{14} \coloneqq \mu^0 \left(\text{SW } \mathtt{E} \circ \chi_{29+U_i} \mid i \in \mathbb{Z}_7 \setminus \{5\}\right) = \mu^0 \left(0,1,0,0,1,(1)\right) = 0$$

$$\text{SW } \mathtt{E}_{15} \coloneqq \mu^0\left(\text{SW } \mathtt{E} \circ \chi_{31+U_i} \mid i \in \mathbb{Z}_7 \backslash \{0\}\right) = \mu^0\left(1, 1, \textcircled{0}, 1, 1, 1\right) = 1$$

$$\text{SW } \mathtt{E}_{16} \coloneqq \mu^0\left(\text{SW } \mathtt{E} \circ \chi_{27+U_i} \mid i \in \mathbb{Z}_7 \backslash \{5\}\right) = \mu^0\left(0, 0, 0, \textcircled{0}, 0, 0\right) = 0$$

$$\text{SW } \mathtt{E}_{17} \coloneqq \mu^0\left(\text{SW } \mathtt{E} \circ \chi_{19+U_i} \mid i \in \mathbb{Z}_7 \backslash \{1\}\right) = \mu^0\left(\textcircled{0}, 0, 1, 0, 0, 0\right) = 0$$

$$\text{SW } \mathtt{E}_{18} \coloneqq \mu^0\left(\text{SW } \mathtt{E} \circ \chi_{3+U_i} \mid i \in \mathbb{Z}_7 \backslash \{2\}\right) = \mu^0\left(0, 0, 0, 1, 0, 0\right) = 0$$

$$\text{SW } \mathtt{E}_{19} \coloneqq \mu^0\left(\text{SW } \mathtt{E} \circ \chi_{6+U_i} \mid i \in \mathbb{Z}_7 \backslash \{5\}\right) = \mu^0\left(0, 0, 1, 0, 1, 0\right) = 0$$

$$\text{SW } \mathtt{E}_{20} \coloneqq \mu^0\left(\text{SW } \mathtt{E} \circ \chi_{12+U_i} \mid i \in \mathbb{Z}_7 \backslash \{6\}\right) = \mu^0\left(\textcircled{0}, 1, 0, 0, 0, 0\right) = 0$$

$$\text{SW } \mathtt{E}_{21} \coloneqq \mu^0\left(\text{SW } \mathtt{E} \circ \chi_{24+U_i} \mid i \in \mathbb{Z}_7 \backslash \{6\}\right) = \mu^0\left(0, 0, 0, 0, 1, 0\right) = 0$$

$$\text{SW } \mathtt{E}_{22} \coloneqq \mu^0\left(\text{SW } \mathtt{E} \circ \chi_{21+U_i} \mid i \in \mathbb{Z}_7 \backslash \{6\}\right) = \mu^0\left(0, 0, 0, 0, 0, 0\right) = 0$$

$$\text{SW } \mathtt{E}_{23} \coloneqq \mu^0\left(\text{SW } \mathtt{E} \circ \chi_{15+U_i} \mid i \in \mathbb{Z}_7 \backslash \{6\}\right) = \mu^0\left(0, \textcircled{0}, 0, 0, 0, 0\right) = 0$$

$$\text{SW } \mathtt{E}_{24} \coloneqq \mu^0\left(\text{SW } \mathtt{E} \circ \chi_{30+U_i} \mid i \in \mathbb{Z}_7 \backslash \{0\}\right) = \mu^0\left(\textcircled{0}, 0, 0, 0, 1, 0\right) = 0$$

$$\text{SW } \mathtt{E}_{25} \coloneqq \mu^0\left(\text{SW } \mathtt{E} \circ \chi_{25+U_i} \mid i \in \mathbb{Z}_7 \backslash \{4\}\right) = \mu^0\left(0, 0, \textcircled{1}, 0, 1, 1\right) = 0$$

$$\text{SW } \mathtt{E}_{26} \coloneqq \mu^0\left(\text{SW } \mathtt{E} \circ \chi_{23+U_i} \mid i \in \mathbb{Z}_7 \backslash \{2\}\right) = \mu^0\left(0, 0, 1, 1, 0, 1\right) = 0$$

$$\text{SW } \mathtt{E}_{27} \coloneqq \mu^0\left(\text{SW } \mathtt{E} \circ \chi_{11+U_i} \mid i \in \mathbb{Z}_7 \backslash \{6\}\right) = \mu^0\left(0, 0, 1, 0, 1, \textcircled{1}\right) = 0$$

$$\text{SW } \mathtt{E}_{28} \coloneqq \mu^0\left(\text{SW } \mathtt{E} \circ \chi_{22+U_i} \mid i \in \mathbb{Z}_7 \backslash \{3\}\right) = \mu^0\left(0, 0, 0, 0, \textcircled{1}, 0\right) = 0$$

$$\text{SW } \mathtt{E}_{29} \coloneqq \mu^0\left(\text{SW } \mathtt{E} \circ \chi_{9+U_i} \mid i \in \mathbb{Z}_7 \backslash \{6\}\right) = \mu^0\left(0, 0, 0, \textcircled{0}, 0, 0\right) = 0$$

$$\text{SW } \mathtt{E}_{30} \coloneqq \mu^0\left(\text{SW } \mathtt{E} \circ \chi_{18+U_i} \mid i \in \mathbb{Z}_7 \backslash \{3\}\right) = \mu^0\left(\textcircled{0}, 0, 1, 1, 0, 1\right) = 0$$

Die Fehler an den Positionen 0 und 15 wurden also vom Decoder erkannt, die an Position 8 und 30 nicht.

Besondere Beachtung verdient die Position 8. Die zu Position 8 korrespondierende Binärzahl 13 liegt in keinem der Unterräume $U_0, U_1, \ldots, U_6$, so dass nicht nur sechs sondern sieben Fehlersummen für die Mehrheitsentscheidung zur Verfügung stehen. Im Fall der Position 8 ist dies sogar entscheidend,

$$\begin{aligned}\text{SW } \mathtt{E}_8 &\coloneqq \mu^0\left(\text{SW } \mathtt{E} \circ \chi_{13+U_i} \mid i \in \mathbb{Z}_7 \backslash \{6\}\right) \\ &= \mu^0\left(\textcircled{0}, \textcircled{0}, \textcircled{1}, \textcircled{0}, \textcircled{1}, \textcircled{1}\right) = 0\end{aligned}$$

gegenüber

$$\text{SW } \mathtt{E}_8 \coloneqq \mu^0\left(\text{SW } \mathtt{E} \circ \chi_{13+U_i} \mid i \in \mathbb{Z}_7\right)$$

$$= \mu^0 \left(\textcircled{0}, \textcircled{0}, \textcircled{1}, \textcircled{0}, \textcircled{1}, \textcircled{1}, \textcircled{1} \right) = 1.$$

Der Fehler an Position 8 wird also erkannt, falls alle sieben Fehlersummen für die Mehrheitsentscheidung herangezogen werden. Unabhängig davon bleibt der Fehler an der Position 30 unerkannt, so dass das ursprüngliche Codewort $\mathtt{c}$ nicht rekonstruiert wird.

Gibt es eine Möglichkeit, doch noch das $\mathtt{c}$ zu erhalten? Ein Vorschlag ist, das Verfahren ein zweites Mal mit dem neuen Wort $\mathtt{z}'$

$$\mathtt{z}' \coloneqq \mathtt{z} - \mathbf{e}_0 - \mathbf{e}15$$

anzuwenden. Da der Hammingabstand zwischen $\mathtt{z}'$ und $\mathtt{c}$ zwei ist und damit die Majority-Logic-Fehlerkorrekturgrenze von $\tau_{\mathrm{MLG}} = 3$ nicht übersteigt, erhalten wir nach einem zweiten Durchlauf schlussendlich $\mathtt{c}$. ◁

Zwei Punkte sind anzumerken zur Idee, die Decodierung solange zu wiederholen, bis der Decoder ein Codewort statt eines Worts außerhalb des Codes zurückgibt. Zum einen ist offen, ob dieser Prozess terminiert. Zum anderen ist nicht garantiert, dass ein ermitteltes Codewort auch das ursprünglich gesendete ist.

Als einfaches Gegenbeispiel stelle man sich vor, ein Codewort $\mathtt{c}$ eines zyklischen Reed-Muller-Codes $\mathrm{RM}^*(r, m)$ wird gesendet und

$$\mathtt{z} \coloneqq \mathtt{c} + \sum_{i \in S} \mathbf{e}_i$$

wird empfangen, wobei $\mathtt{c}_{\min}$ ein Codewort in $\mathrm{RM}^*(r, m)$ mit Minimalgewicht $2^{m-r} - 1$ und S eine Teilmenge des Trägers von $\mathtt{c}_{\min}$ der Kardinalität 2^{m-r-1} ist. Es sind also

$$|S| = 2^{m-r-1} = \tau_{\mathrm{MLG}} + 1$$

Fehler aufgetreten. Die Majority-Logic-Decodier-Verfahren aus Kapitel 5 geben dann statt des gewünschten Codeworts $\mathtt{c}$ das Codewort $\mathtt{c} + \mathtt{c}_{\min}$ aus, dessen Abstand zu $\mathtt{z}$ gerade

$$2^{m-r-1} - 1 = \tau_{\mathrm{MLG}}$$

ist.

Zusammenfassend halten wir fest: Mit den Majority-Logic-Decodier-Verfahren aus Kapitel 5 können in ein oder mehreren Durchläufen mitunter auch Codewörter rekonstruiert werden, wenn mehr als τ_{MLG} Fehler aufgetreten sind. In diesem Fall ist jedoch nicht gewährleistet, dass ein ermitteltes Codewort auch das ursprünglich übertragene ist.

Lennard Schulz untersucht im Rahmen seiner Masterarbeit die Fehlerkorrekturfähigkeit bei Reed-Muller-Codes erster Ordnung, $\mathrm{RM}(1, m)$ [33]. Wie wir in Abschnitt 7.5 gesehen haben, gilt für Reed-Muller-Codes erster Ordnung $\tau_{\max} = 2^{m-2} - 1$ (vgl. Gleichung [7.5] auf Seite 145). Schulz präsentiert eine Bedingung, die von der zu decodierenden Position i, von einer Konstante $\kappa \in \mathbb{Z}$[1] und von der Menge der Positionen, an denen beim Übertragen ein Fehler auftrat, *Fehlermenge* (engl. *error set*) genannt, abhängt. Ist diese Bedingung erfüllt, so kann das Codewortsymbol an dieser Position i auch bei insgesamt bis zu $\tau_{\max} + \kappa$ Fehlern korrekt rekonstruiert werden [33, Theorem 3.1.3.]. Bei dem von ihm eingesetzten Decodierverfahren, das ebenfalls auf dem Prinzip der Majority-Logic-Decodierung (vgl. Proposition 4.3.1) basiert, werden stets alle der anwendbaren affinen Unterräume verwendet [33, Example 2.3.7]. Für verschiedene Parameter m und $\kappa = 1$ gibt er an bzw. schätzt er über Stichproben ab, wie groß der Anteil der Fehlermengen der Kardinalität $\tau_{\max} + 1$ ist, bei denen $\mathtt{z}$ inkorrekt decodiert wird [33, Example 3.2.1, Example 3.2.2, Example 3.2.3]. Darüber hinaus leitet er eine obere Schranke her für den Anteil der Fehlermengen der Kardinalität $\tau_{\max} + \kappa$, bei denen $\mathtt{z}$ falsch decodiert wird. Diese obere Schranke ist gegeben durch

$$2^{m \cdot \kappa} \cdot (6/7)^{2^{m-3}}$$

und konvergiert für ein festes $\kappa \geq 1$ und steigendem m gegen 0 [33, Corollary 3.4.5].

[1] Schulz bezeichnet die Konstante mit q.

Anhang A

Existenz und Konstruktion spezieller affiner Unterräume

In diesem Abschnitt möchten wir den ausstehenden Beweis von Proposition 5.1.4 führen. Er stützt sich maßgeblich auf die theoretischen Ausführungen in [10] zu projektiven Räumen. Wir benötigen mehrere aufeinander aufbauende Lemmata und Korollare, die gleichzeitig die Idee liefern, wie affine Unterräume, die sich paarweise in einem gegebenen affinen Raum schneiden, konstruiert werden können.

Wir setzen voraus, dass q eine Primzahlpotenz und $m \in \mathbb{N}_0$ ist.

Lemma A.0.1. *Gegeben $s \in \mathbb{N}$. Sei W ein Vektorraum der Dimension $2s$ über $\mathbb{F}_q$ und sei W_0 ein beliebiger s-dimensionaler Unterraum von W. Es existieren weitere s-dimensionale Unterräume $W_1, \ldots, W_{q^s}$ von $\mathbb{F}_q^{2s}$, so dass jeder Vektor $\mathbf{w} \in \mathbb{F}_q^{2s}$, $\mathbf{w} \neq 0$, in genau einem der Unterräume W_j, $0 \leq j \leq q^s$, enthalten ist.*

Beweis vgl. [10, Construction 1]. Fassen wir $\mathbb{F}_{q^{2s}}$ als $(2s)$-dimensionalen Vektorraum über $\mathbb{F}_q$ auf, so ist dieser isomorph zu W. Der Unterraum W_0 werde identifiziert mit der additiven Gruppe von $\mathbb{F}_{q^s} \subset \mathbb{F}_{q^{2s}}$. Nach dem Satz von Lagrange gibt es

$$\left|\mathbb{F}_{q^{2s}}^*\right| / \left|\mathbb{F}_{q^s}^*\right| = q^s + 1$$

Nebenklassen von $\mathbb{F}_{q^s}^*$ in $\mathbb{F}_{q^{2s}}^*$. Jede dieser Nebenklassen zusammen mit dem Nullelement ist eine additive Untergruppe von $\mathbb{F}_{q^{2s}}$ der Kardinalität q^s. Diese additiven Untergruppen definieren Unterräume

$$W_0, W_1, \ldots, W_{q^s} \leq W$$

der Dimension s, so dass jeder Vektor $\mathbf{w} \in \mathbb{F}_q^{2s}$, $\mathbf{w} \neq 0$, in genau einem dieser Unterräume W_j, $0 \leq j \leq q^s$, enthalten ist. □

Lemma A.0.2 (vgl. [10, Lemma 2.2]). *Seien $s \in \mathbb{N}$ und $t \in \mathbb{N}_0$ mit $s \geq t$. Sei V ein $(s+t)$-dimensionaler Vektorraum über $\mathbb{F}_q$ und V_0 ein s-dimensionaler Unterraum von V. Es existieren t-dimensionale Unterräume $V_1, V_2, \ldots, V_{q^s}$ von V, so dass jeder Vektor $\mathbf{v} \in V$, $\mathbf{v} \neq 0$, in genau einem der Unterräume V_j, $0 \leq j \leq q^s$, enthalten ist.*

Beweis. Die äußere direkte Summe $W \coloneqq V \oplus \mathbb{F}_q^{s-t}$ ist ein Vektorraum der Dimension $2s$ über $\mathbb{F}_q$. Sei $W_0 \coloneqq V_0$. Nach Lemma A.0.1 existieren weitere s-dimensionale Unterräume $W_1, \ldots, W_{q^s}$ von W, so dass jeder Vektor $\mathbf{w} \in W$, $\mathbf{w} \neq 0$, in genau einem dieser Unterräume W_j, $0 \leq j \leq q^s$, enthalten ist,

$$W \setminus \{0\} = \dot{\bigcup_{0 \leq j \leq q^s}} W_j \setminus \{0\}.$$

Wir definieren für alle $1 \leq j \leq q^s$

$$V_j \coloneqq V \cap W_j.$$

Darüber hinaus gilt wegen $W_0 \coloneqq V_0 \leq V$ ebenfalls

$$V_0 = V \cap W_0.$$

Dann ist

$$V \setminus \{0\} = V \cap W \setminus \{0\} = \dot{\bigcup_{0 \leq j \leq q^s}} V \cap W_j \setminus \{0\} = \dot{\bigcup_{0 \leq j \leq q^s}} V_j \setminus \{0\}. \qquad [A.1]$$

Bleibt zu zeigen, dass die Unterräume V_j, $1 \leq j \leq q^s$, Dimension t haben. Sei i, $1 \leq i \leq q^s$, beliebig. Da $V_i \cap V_0 = \{0\}$, ist

$$\dim(V_i) \leq \dim(V) - \dim(V_0) = t.$$

Angenommen, $\dim(V_i) < t$, dann liefert Gleichung [A.1]

$$q^{s+t} - 1 = |V \setminus \{0\}| = \sum_{0 \le j \le q^s} |V_j \setminus \{0\}| < (q^s - 1) + q^s \cdot (q^t - 1) = q^{s+t} - 1.$$

Widerspruch. Der Unterraum V_i hat Dimension t. □

Lemma A.0.3 (vgl. [10, Corollary 2.3]). *Seien $D, t \in \mathbb{N}$ mit $D \ge t$ beliebig. Es existieren $Q \in \mathbb{N}$, $R \in \mathbb{N}_0$ mit $R < t$, so dass $D := Q \cdot t + R$. Wir setzen*

$$\ell := \frac{q^D - q^{t+R}}{q^t - 1} = q^{t+R} \cdot \frac{q^{(Q-1)\cdot t} - 1}{q^t - 1} \in \mathbb{N}_0.$$

(Beachte, $\ell = 0$, falls $Q = 1$.)

Sei U ein D-dimensionaler Vektorraum und sei U_0 ein $(t+R)$-dimensionaler Unterraum von U. Es existieren ℓ Unterräume $U_1, U_2, \ldots, U_\ell \le U$ der Dimension t, so dass jeder Vektor $\mathbf{u} \in U$, $\mathbf{u} \neq 0$, in genau einem der Unterräume U_j, $0 \le j \le \ell$, enthalten ist.

Beweis. Induktion über Q.
Ist $Q = 1$ und damit $D = t + R$, so ist $U = U_0$ und $\ell = 0$.
Ist $Q = 2$, so ist $\ell = q^{t+R}$. Indem wir $s = t + R$ setzen, folgt die Behauptung direkt aus Lemma A.0.2.
Sei $Q > 2$. Sei V ein $(D-t)$-dimensionaler Unterraum von U, der U_0 enthält,

$$U_0 \le V \le U.$$

Wir setzen

$$L := \frac{q^{D-t} - q^{t+R}}{q^t - 1} == q^{t+R} \cdot \frac{q^{(Q-2)\cdot t} - 1}{q^t - 1}.$$

Nach Induktionsannahme gibt es t-dimensionale Unterräume $V_1, V_2, \ldots, V_L$ von V, so dass jeder Vektor $\mathbf{v} \in V$, $\mathbf{v} \neq 0$, in genau einem der Unterräume $U_0, V_1, V_2, \ldots, V_L$, enthalten ist,

$$V \setminus \{0\} = U_0 \setminus \{0\} \,\dot\cup \bigdot_{1 \le j \le L} V_j \setminus \{0\}\,. \qquad [A.2]$$

Des weiteren gibt es gemäß Lemma A.0.2 Unterräume $U_1, U_2, \ldots, U_{q^{D-t}} \le U$ der Dimension t, so dass jeder Vektor $\mathbf{u} \in U$, $\mathbf{u} \neq 0$, in genau einem dieser Unterräume $V, U_1, U_2, \ldots, U_{q^{D-t}}$, enthalten ist,

$$U \setminus \{0\} = V \setminus \{0\} \,\dot\cup \bigdot_{1 \le j \le q^{D-t}} U_j \setminus \{0\}\,. \qquad [A.3]$$

Die Behauptung folgt, wenn man Gleichung [A.2] in Gleichung [A.3] einsetzt, wobei

$$L + q^{D-t} = \frac{q^{D-t} - q^{t+R}}{q^t - 1} + q^{D-t} = \frac{q^D - q^{t+R}}{q^t - 1} = \ell. \qquad \square$$

Lemma A.0.4 (vgl. [10, Corollary 2.4, Theorem 2.7]). *Seien $D, t \in \mathbb{N}$ mit $D \geq t$ beliebig. Wir definieren*

$$R \coloneqq D \bmod t \in \mathbb{N}_0$$

als den ganzzahligen Rest der Division D durch t und setzen

$$\ell \coloneqq \left(q^D - q^{t+R}\right) / \left(q^t - 1\right).$$

Sei U ein D-dimensionaler Vektorraum und sei $\mathbf{w} \in U$, $\mathbf{w} \neq 0$ beliebig.

(a) *Falls $R = 0$, so gibt es $\ell + 1$, jedoch keine $\ell + 2$ Unterräume von U der Dimension t, die sich paarweise in $\{0\}$ schneiden. Es ist nicht möglich diese $\ell + 1$ Unterräume so zu konstruieren, dass der Vektor $\mathbf{w}$ in keinem enthalten ist.*

(b) *Falls $R > 0$, so gibt es mindestens $\ell + 1$ und höchstens $\ell + q^R - q + 1$ Unterräume von U der Dimension t, die sich paarweise in $\{0\}$ schneiden. Es ist möglich diese $\ell + 1$ Unterräume so zu konstruieren, dass der Vektor $\mathbf{w}$ in keinem enthalten ist.*

Beweis.

(a) Angenommen, $R = 0$. Die Anzahl der t-dimensionalen Unterräume von U, die sich paarweise in $\{0\}$ schneiden, ist ganz offensichtlich nach oben beschränkt durch $\ell + 1$, da

$$\frac{q^D - 1}{q^t - 1} = \ell + 1. \qquad [A.4]$$

Wir wissen aus Lemma A.0.3, dass wir die Menge der Vektoren aus U ungleich null tatsächlich in $\ell + 1$ Teilmengen partitionieren können, so dass jede dieser Teilmengen zusammen mit dem Nullvektor einen t-dimensionalen Unterraum von U bildet.

(b) Angenommen, $R > 0$. Sei U_0 ein beliebiger $(t+R)$-dimensionaler Unterraum von U, der $\mathbf{w}$ enthält. Nach Lemma A.0.3 existieren t-dimensionale Unterräume $U_1, U_2, \ldots, U_\ell$ von U, so dass jeder Vektor $\mathbf{u} \in U$, $\mathbf{u} \neq 0$, in genau einem der Unterräume U_j, $0 \leq j \leq \ell$, enthalten ist. Diese $U_1, U_2, \ldots, U_\ell$ zusammen mit einem beliebigen t-dimensionalen Unterraum von U_0, der $\mathbf{w}$ nicht enthält, bilden die gesuchte Menge.

Die Anzahl der t-dimensionalen Unterräume von U, die sich paarweise in $\{0\}$ schneiden, ist nach oben beschränkt durch $\ell + q^R$, da

$$\left\lfloor (q^D-1)/(q^t-1) \right\rfloor = \ell + 1 + \left\lfloor q^R \frac{\left(q^t - q^{t-R}\right)}{q^t-1} \right\rfloor < \ell + 1 + q^R.$$

Sei $b \in \mathbb{N}_0$ beliebig. Angenommen, es gibt $L := \ell + q^R - b$ Unterräume $U_0, U_1, \ldots, U_{L-1}$ von U der Dimension t, die sich paarweise in $\{0\}$ schneiden. Wir werden zeigen, dass $b \geq q-1$ gilt.

(i) Die Anzahl der Vektoren in U, die in keinem dieser Unterräume liegen, ist gegeben durch

$$\begin{aligned}
&\{\mathbf{v} \in U \mid \mathbf{v} \notin U_0, U_1, \ldots, U_{L-1}\} \\
&= q^D - 1 - L \cdot (q^t - 1) \\
&= q^D - 1 - \left(\ell + q^R - b\right) \cdot (q^t - 1) \\
&= q^D - 1 - (q^D - q^{t+R}) - q^{t+R} + q^R + b \cdot (q^t - 1) \\
&= q^R - 1 + b \cdot (q^t - 1).
\end{aligned}$$

(ii) Sei H ein beliebiger $(D-1)$-dimensionaler Unterraum von U. Sei $j \in \mathbb{Z}_L$ beliebig. Die Dimension des Unterraums $H \cap U_j$ ist mindestens $t-1$ und höchstens t,

$$\begin{aligned}
t - 1 &= \dim(H) + \dim(U_j) - \dim(U) \\
&\leq \dim\left(H \cap U_j\right) \\
&\leq \dim(U_j) = t
\end{aligned}$$

Damit ist

$$\begin{aligned}
|H \cap U_j \setminus \{0\}| &= q^{t-1} - 1 + a \cdot (q-1)q^{t-1} \\
&\equiv q^{t-1} - 1 \bmod (q-1)q^{t-1}.
\end{aligned}$$

für ein $a \in \{0,1\}$. Die Anzahl der Vektoren in H, die in keinem der Unterräume $U_0, U_1, \ldots, U_{L-1}$ liegen, ist gegeben durch

$$\begin{aligned}
&|\{\mathbf{v} \in H \mid \mathbf{v} \notin U_0, U_1, \ldots, U_{L-1}\}| \\
&\equiv q^{D-1} - 1 - L \cdot (q^{t-1} - 1) \\
&= q^{D-1} - 1 - \ell(q^{t-1} - q^t + q^t - 1) - q^R(q^{t-1} - 1) + b(q^{t-1} - 1) \\
&\equiv q^{D-1} - 1 - (q^D - q^{t+R}) - q^{t+R-1} + q^R + b(q^{t-1} - 1) \\
&\equiv -1 + q^R + b(q^{t-1} - 1) \bmod q^{t-1}(q-1).
\end{aligned}$$

(iii) Angenommen, $b \leq q-2$. Dann ist

$$\begin{aligned}
-1 + q^R + b(q^{t-1} - 1) &\leq -1 + q^R + (q-2)(q^{t-1} - 1) \\
&= \underbrace{q^R - q^{t-1}}_{\leq 0} \underbrace{-q+1}_{<0} + q^{t-1}(q-1) \\
&< q^{t-1}(q-1).
\end{aligned}$$

Wegen (ii) ist

$$|\{\mathbf{v} \in H \mid \mathbf{v} \notin U_0, U_1, \ldots, U_{L-1}\}| \geq -1 + q^R + b(q^{t-1} - 1). \quad [A.5]$$

Wir betrachten

$$\left| \left\{ (\mathbf{v}, H) \;\middle|\; H \leq U, \dim(H) = D-1, \mathbf{v} \in H, \mathbf{v} \notin \bigcup_{j \in \mathbb{Z}_L} U_j \right\} \right|.$$

Zum einen gilt wegen (i)

$$\begin{aligned}
&\left| \left\{ (\mathbf{v}, H) \;\middle|\; H \leq U, \dim(H) = D-1, \mathbf{v} \in H, \mathbf{v} \notin \bigcup_{0 \leq j \leq L-1} U_j \right\} \right| \\
&= \left| \left\{ \left(\mathbf{v}, \tilde{H}\right) \;\middle|\; \mathbf{v} \in U \setminus \bigcup_{0 \leq j \leq L-1} U_j, \tilde{H} \leq U/\langle \mathbf{v} \rangle, \dim(\tilde{H}) = D-2 \right\} \right| \\
&= \left(q^R - 1 + b \cdot (q^t - 1)\right) \cdot (q^{D-1} - 1)/(q-1),
\end{aligned}$$

wobei die Anzahl der Hyperebenen gerade der Anzahl der eindimensionalen Unterräume entspricht (*Dualitätsprinzip*). Zum anderen ist wegen Ungleichung [A.5]

$$\begin{aligned}
&\left| \left\{ (\mathbf{v}, H) \;\middle|\; H \leq U, \dim(H) = D-1, \mathbf{v} \in H, \mathbf{v} \notin \bigcup_{0 \leq j \leq L-1} U_j \right\} \right| \\
&\geq |\{H \leq U \mid \dim(H) = D-1\}| \cdot \left(q^R - 1 + b(q^{t-1} - 1)\right) \\
&= (q^D - 1)/(q-1) \cdot \left(-1 + q^R + b(q^{t-1} - 1)\right).
\end{aligned}$$

Unmittelbar folgt die Ungleichung,

$$\begin{aligned}&\left(q^R - 1 + b \cdot (q^t - 1)\right) \cdot (q^{D-1} - 1)\\ \geq\ &(q^D - 1) \cdot \left(-1 + q^R + b(q^{t-1} - 1)\right),\end{aligned}$$

die sich vereinfachen lässt zu

$$b(q^{D-t} - 1) \geq q^{D-t}\left(q^R - 1\right).$$

Da wir $b < q - 1$ angenommen hatten, ergibt sich

$$(q-1)(q^{D-t}-1) > b(q^{D-t}-1) \geq \underbrace{q^{D-t}}_{>q^{D-t}-1}\ \underbrace{\left(q^R - 1\right)}_{>q-1}.$$

Widerspruch. Es gilt daher $b \geq q - 1$. □

Schließlich können wir Proposition 5.1.4 beweisen, die wir der Übersicht wegen an dieser Stelle noch einmal anführen – einschließlich der verwendeten Definitionen 5.1.1 und 5.1.2.

Definition 5.1.1 Für $D_0 \in \mathbb{N}_0, D_1 \in \mathbb{N}$ mit $D_0 < D_1 \leq m$ und für einen beliebigen D_0-dimensionalen affinen Unterraum $A \in \mathcal{A}_{D_0,m,q}$ setzen wir

$$\ell_{D_0,D_1,m,q} := \max\left(\left\{ l \in \mathbb{N} \;\middle|\; \begin{array}{c} \exists\ A_0, A_1, \ldots, A_{l-1} \in \mathcal{A}_{D_1,m,q},\\ A \subseteq A_i \text{ für alle } 0 \leq i \leq l-1,\\ A_i \cap A_j = A \text{ für alle } 0 \leq i < j \leq l-1 \end{array} \right\}\right).$$

Definition 5.1.2 Für $D_0 \in \mathbb{N}_0, D_1 \in \mathbb{N}$ mit $D_0 < D_1 < m$ und für einen beliebigen D_0-dimensionalen affinen Unterraum $A \in \mathcal{A}^*_{D_0,m,q}$ setzen wir

$$\ell^*_{D_0,D_1,m,q} := \max\left(\left\{ l \in \mathbb{N} \;\middle|\; \begin{array}{c} \exists\ A_0, A_1, \ldots, A_{l-1} \in \mathcal{A}^*_{D_1,m,q},\\ A \subseteq A_i \text{ für alle } 0 \leq i \leq l-1,\\ A_i \cap A_j = A \text{ für alle } 0 \leq i < j \leq l-1 \end{array} \right\}\right).$$

Proposition 5.1.4 Seien $D_0, D_1 \in \mathbb{N}_0$ mit $D_0 < D_1 \leq m$. Wir definieren

$$R \coloneqq (m - D_0) \bmod (D_1 - D_0) \in \mathbb{N}_0$$

als den ganzzahligen Rest der Division $(m - D_0)$ durch $(D_1 - D_0)$ und setzen

$$\ell \coloneqq \frac{q^{m-D_0} - q^{D_1-D_0+R}}{q^{D_1-D_0} - 1}.$$

(Insbesondere ist genau dann $\ell = 0$, wenn $m - D_0 < 2(D_1 - D_0)$.)

Dann ist

(a) im Fall $R = 0$,

$$\ell_{D_0,D_1,m,q} = \ell + 1,$$

(b) im Fall $R > 0$,

$$\ell + 1 \leq \ell_{D_0,D_1,m,q} \leq \ell + q^R - q + 1.$$

Gilt darüber hinaus $D_1 < m$, so ist

(c) stets

$$\ell_{D_0,D_1,m,q} - 1 \leq \ell^*_{D_0,D_1,m,q} \leq \ell_{D_0,D_1,m,q}.$$

(d) im Fall $R = 0$,

$$\ell^*_{D_0,D_1,m,q} = \ell.$$

(e) im Fall $R > 0$,

$$\ell + 1 \leq \ell^*_{D_0,D_1,m,q} \leq \ell + q^R - q + 1$$

Beweis. Sei $A \in \mathcal{A}_{D_0,m,q}$ ein beliebiger D_0-dimensionaler affiner Unterraum des $\mathbb{F}_q^m$. Es existieren $U, U' \leq \mathbb{F}_q^m$, $\mathbf{v} \in U'$ mit $U \oplus U' = \mathbb{F}_q^m$, $\dim U = D_0$ und

$$A = \mathbf{v} + U.$$

Wir zeigen zunächst direkt die unteren Schranken und anschließend durch Widerspruchsbeweis die oberen Schranken.

1. Nach Lemma A.0.4 gibt es $\ell + 1$ Unterräume $U_0', U_1', \ldots, U_\ell' \leq U'$ der Dimension $D_1 - D_0$ gibt, die sich paarweise in $\{0\}$ schneiden. Dann bilden

$$\mathbf{v} + \left(U \oplus U_0'\right), \mathbf{v} + \left(U \oplus U_1'\right), \ldots \mathbf{v} + \left(U \oplus U_\ell'\right)$$

die gesuchte Menge der D_1-dimensionalen affinen Unterräume des $\mathbb{F}_q^m$ mit paarweisem Schnitt in A, so dass sowohl bei $R = 0$ als auch bei $R > 0$ gilt,

$$\ell_{D_0,D_1,m,q} \geq \ell + 1.$$

Gehen wir davon aus, dass $A \in \mathcal{A}^*_{D_0,m,q}$ und $D_1 < m$. Dann ist $\mathbf{v} \neq 0$. Betrachten wir die zuvor konstruierten affinen Räume

$$\mathbf{v} + \left(U \oplus U_0'\right), \mathbf{v} + \left(U \oplus U_1'\right), \ldots \mathbf{v} + \left(U \oplus U_\ell'\right).$$

Mindestens ℓ dieser liegen in $\mathcal{A}^*_{D_1,m,q}$, da $\mathbf{v} \in U'$ und die Unterräume $U_0', U_1', \ldots, U_\ell'$ sich paarweise in $\{0\}$ schneiden. Also stets

$$\ell^*_{D_0,D_1,m,q} \geq \ell_{D_0,D_1,m,q} - 1 \geq \ell.$$

Allerdings wissen wir aus Lemma A.0.4, dass es im Fall $R > 0$ möglich ist, die $\ell + 1$ Unterräume $U_0', U_1', \ldots, U_\ell' \leq U'$ so zu konstruieren, dass sie sich paarweise in $\{0\}$ schneiden und kein einziger den Vektor $\mathbf{v}$ enthält. Dann liegen die $\ell + 1$ affinen Räume

$$\mathbf{v} + \left(U \oplus U_0'\right), \mathbf{v} + \left(U \oplus U_1'\right), \ldots \mathbf{v} + \left(U \oplus U_\ell'\right)$$

allesamt in $\mathcal{A}^*_{D_1,m,q}$. Im Fall $R > 0$ ist also

$$\ell^*_{D_0,D_1,m,q} \geq \ell + 1.$$

2. Sei

$$L \coloneqq \begin{cases} \ell + 1 & R = 0, \\ \ell + q^R - q + 1 & R > 0. \end{cases}$$

Angenommen, es existieren $L + 1$ affine Räume $A_0, A_1, \ldots, A_L \in \mathcal{A}_{D_1,m,q}$ mit paarweisem Schnitt in A. Dann gibt es für jedes A_j, $0 \leq j \leq L$, einen $(D_1 - D_0)$-dimensionalen Unterraum $U_j' \leq U'$, so dass $A_j = \mathbf{v} + (U \oplus U_j')$.

Darüber hinaus schneiden sich die Unterräume $U_0', U_1', \ldots, U_L'$ paarweise in $\{0\}$. Widerspruch zu Lemma A.0.4. Also,

$$\ell_{D_0,D_1,m,q} \leq L.$$

Setzen wir voraus, dass $D_1 < m$ und $A \in \mathcal{A}^*_{D_0,m,q}$. Dann ist $\mathbf{v} \neq 0$. Da

$$\mathcal{A}^*_{D_1,m,q} \subseteq \mathcal{A}_{D_1,m,q},$$

ist

$$\ell^*_{D_0,D_1,m,q} \leq \ell_{D_0,D_1,m,q} \leq L.$$

Bleibt die obere Schranke von $\ell^*_{D_0,D_1,m,q}$ für den Fall $R = 0$ zu zeigen. Angenommen, es existieren $\ell + 1$ affine Räume $A_0, A_1, \ldots, A_\ell \in \mathcal{A}^*_{D_1,m,q}$ mit paarweisem Schnitt in A. Dann gibt es für jedes A_j, $0 \leq j \leq l$, einen $(D_1 - D_0)$-dimensionalen Unterraum $U_j' \leq U'$, so dass $A_j = \mathbf{v} + (U \oplus U_j')$. Darüber hinaus schneiden sich die Unterräume $U_0', U_1', \ldots, U_\ell'$ paarweise in $\{0\}$. Aus Lemma A.0.4 wissen wir, dass der Vektor $\mathbf{v}$ in einem dieser Unterräume enthalten ist, sagen wir ohne Beschränkung der Allgemeinheit $\mathbf{v} \in U_0'$. Dann aber $A_0 = U \oplus U_0'$. Widerspruch zu $A_0 \in \mathcal{A}^*_{D_1,m,q}$. Folglich im Fall $R = 0$,

$$\ell^*_{D_0,D_1,m,q} \leq \ell \qquad \square$$

Proposition A.0.5. *Seien $D_0, D_1 \in \mathbb{N}_0$ mit $D_0 < D_1 < m - 1$. Es ist*

$$\ell^*_{D_0,D_1,m,q} \geq \ell^*_{D_0+1,D_1+1,m,q}.$$

Beweis. Folgt aus der Definition von $\ell^*_{D_0,D_1,m,q}$. (Sei $A \coloneqq \mathbf{v} + U \in \mathcal{A}^*_{D_0,m,q}$ beliebig. Wähle einen beliebigen Vektor $\mathbf{w} \in \mathbb{F}_q^m \backslash U$, so dass $\mathbf{v} \notin \langle \mathbf{w} \rangle_{\mathbb{F}_q} \oplus U$. Dann gibt es $N \coloneqq \ell^*_{D_0+1,D_1+1,m,q}$ affine Unterräume $A_0, A_1, \ldots, A_{N-1} \in \mathcal{A}^*_{D_1+1,m,q}$, die sich paarweise in $\mathbf{v} + \langle \mathbf{w} \rangle_{\mathbb{F}_q} \oplus U$ schneiden. Jedes A_i, $i \in \mathbb{Z}_N$, hat die Form

$$A_i = \mathbf{v} + \langle \mathbf{w} \rangle_{\mathbb{F}_q} \oplus U \oplus U_i$$

für ein $U_i \leq \mathbb{F}_q^m$. Betrachten wir die D_1-dimensionalen affinen Räume $A_i' \coloneqq \mathbf{v} + U \oplus U_i$, $i \in \mathbb{Z}_N$, so schneiden diese sich paarweise in A. Die Behauptung folgt.) $\square$

Korollar A.0.6. *Seien $D_0, D_1 \in \mathbb{N}_0$ mit*

$$D_0 < D_1 - 1 < D_1 < m$$

beliebig. Es ist

$$\ell^*_{D_1-1,D_1,m,q} > \ell^*_{D_0,D_1,m,q}.$$

Ist $q > 2$ oder $D_1 < m - 1$, so gilt sogar

$$\ell^*_{D_1-1,D_1,m,q} > \ell^*_{D_0,D_1,m,q} + 1.$$

Beweis. Sei $D_0 \in \mathbb{N}_0$, $D_0 < D_1 - 1$, beliebig. Wir definieren R als den ganzzahligen Rest der Division $(m - D_0)$ durch $(D_1 - D_0)$,

$$R \coloneqq (m - D_0) \bmod (D_1 - D_0) \in \mathbb{N}_0.$$

Angenommen, $R = 0$. Wegen $m > D_1$ ist dann

$$m - D_0 \geq 2(D_1 - D_0)$$

und damit

$$m - D_1 \geq D_1 - D_0 \geq 2.$$

Es folgt mit Proposition 5.1.4

$$\begin{aligned}
\ell^*_{D_0,D_1,m,q} + 1 &= \frac{q^{m-D_0} - q^{D_1-D_0}}{q^{D_1-D_0} - 1} + 1 \\
&= 1 + \sum_{\substack{i=2, \\ (D_1-D_0) \mid i}}^{m-D_1} q^i \\
&< q + \sum_{i=2}^{m-D_1} q^i \\
&= \sum_{i=1}^{m-D_1} q^i.
\end{aligned}$$

Angenommen, $R > 0$. Es ist

$$m - D_0 \geq D_1 - D_0 + R$$

und deshalb

$$m - D_1 \geq R.$$

Erneut wenden wir Proposition 5.1.4 an und erhalten

$$\begin{aligned}\ell^*_{D_0,D_1,m,q} + 1 &\leq \frac{q^{m-D_0} - q^{D_1-D_0+R}}{q^{D_1-D_0}-1} + q^r \underbrace{-q+2}_{\leq 0} \\ &\leq \sum_{\substack{i=R,\\ (D_1-D_0)\,|\,(i-R)}}^{m-D_1} q^i \\ &\leq \sum_{i=1}^{m-D_1} q^i,\end{aligned}$$

wobei die gesamte Ungleichung strikt ist, wenn $q > 2$ oder $m - D_1 > 1$.

Die Behauptungen folgen mit

$$\sum_{i=1}^{m-D_1} q^i = \frac{q^{m-D_1+1} - q}{q-1} = \ell^*_{D_1-1,D_1,m,q}. \qquad \square$$

Korollar A.0.7. *Seien $D_0, D_1 \in \mathbb{N}_0$ mit $D_0 < D_1 < m$.*

(a) *Es ist stets*

$$q^{m-D_1+1} > \ell^*_{D_0,D_1,m,q}.$$

(b) *Es ist $m \geq 2D_1 - D_0$ genau dann, wenn*

$$\ell^*_{D_0,D_1,m,q} \geq q^{m-D_1}$$

beziehungsweise

$$\ell_{D_0,D_1,m,q} - 1 \geq q^{m-D_1}.$$

Beweis. Sei R der ganzzahlige Rest der Division $(m - D_0)$ durch $(D_1 - D_0)$,

$$R \coloneqq (m - D_0) \bmod (D_1 - D_0) \in \mathbb{N}_0.$$

Weiterhin sei

$$\ell \coloneqq \frac{q^{m-D_0} - q^{D_1-D_0+R}}{q^{D_1-D_0}-1}.$$

(a) Es ist wegen Korollar A.0.6

$$\ell^*_{D_0,D_1,m,q} \leq \ell^*_{D_1-1,D_1,m,q} = \frac{q^{m-D_1+1} - q}{q-1} < q^{m-D_1+1}.$$

(b) Angenommen, $m \geq 2D_1 - D_0$. Dann ist $m - D_0 \geq 2(D_1 - D_0) + R$, so dass

$$q^{m-D_0} - q^{D_1-D_0+R} \geq q^{m-D_0} - q^{m-D_1} = q^{m-D_1}\left(q^{D_1-D_0} - 1\right)$$

und damit

$$\ell = \frac{q^{m-D_0} - q^{D_1-D_0+R}}{q^{D_1-D_0} - 1} \geq q^{m-D_1}.$$

Angenommen, $m < 2D_1 - D_0$. Dann ist $m - D_0 = D_1 - D_0 + R$, also $R = m - D_1 > 0$. Es folgt $\ell = 0$ und damit

$$\ell + q^R - q + 1 < q^R = q^{m-D_1}.$$

Die Behauptungen folgen mit Proposition 5.1.4. □

Anhang B

Vergleich des Aufwands der Hybriddecodierung unter drei Abstufungen

Wir erinnern uns daran, dass wir in Tabelle 6.2 auf Seite 111 die drei vorgestellten Abstufungen, jene nach Chen, nach Reed und die von uns entwickelte invertierte Abstufung, hinsichtlich des Aufwands der Hybriddecodierung miteinander verglichen haben. Allerdings haben wir die Behauptungen bislang nicht formal bewiesen. Das holen wir nun nach.

Wir übernehmen die folgenden Notationen und Definitionen aus Abschnitt 6.3:

$$\eta = 2 \cdot \left\lfloor \frac{q^{m+1-D_\mathcal{C}} - q}{2(q-1)} \right\rfloor = \frac{q^{m+1-D_\mathcal{C}} - q}{q-1} - \begin{cases} 1 & p \neq 2, m - D_\mathcal{C} \text{ ungerade,} \\ 0 & \text{sonst} \end{cases},$$

$$\varsigma = 1 + \lceil \log_2 \left(m/(m+1-D_\mathcal{C}) \right) \rceil ,$$

$$\dot{D} = \left\lfloor \log_q((q-1)(\eta+1)+1) \right\rfloor = m - D_\mathcal{C} + \begin{cases} 0 & p \neq 2, m - D_\mathcal{C} \text{ ungerade,} \\ 1 & \text{sonst} \end{cases}.$$

Wir wiederholen den Hinweis, dass die Anzahl der Majority-Logic-Stufen unter Reeds Abstufung $D_\mathcal{C}$ und nicht ς ist.

Die Operationen μ und $\pm$ symbolisieren wie gehabt jeweils eine Mehrheitsentscheidung beziehungsweise eine Addition/Subtraktion über $\mathbb{F}_{q_\mathcal{C}}$. Die Buchstaben R, C und I repräsentieren jeweils Reeds, Chens und die invertierte Abstufung. Es bezeichne $\mathcal{K}_{\sim,A}$ bzw. $\mathcal{V}_{\sim,A}$ bzw. $\mathcal{H}_{\sim,A}$ die Anzahl der zur klassischen bzw. verbesserten bzw. hybriden Decodierung benötigten Operationen $\sim$ unter der Abstufung A, wobei $\sim \in \{\mu, \pm\}$ und $A \in \{R, C, I\}$. Weiterhin werde für alle $A \in \{R, C, I\}$ der Parameter D_t unter der Abstufung A bezeichnet durch $D_{t,A}$.

Wie in Theorem 5.4.4 gesehen, ist zum einen wegen $T_R = \mathbb{Z}_{D_\mathcal{C}}$

$$\begin{aligned}
\mathcal{H}_{\mu,R} &= \sum_{0 \le t \le D_\mathcal{C}-1} (\eta+1)^t \cdot \mathcal{H}_\mu\left(D_{t,R}, \dot{D}\right), \\
\mathcal{H}_{\pm,R} &= \sum_{0 \le t \le D_\mathcal{C}-1} (\eta+1)^t \cdot \mathcal{H}_\pm\left(D_{t,R}, \dot{D}\right), \\
\mathcal{H}_{\mathtt{E},R} &= (\eta+1)^{D_\mathcal{C}} \cdot \left(q^{m-D_\mathcal{C}} - 1\right) = \mathcal{V}_{\mathtt{E},R},
\end{aligned}$$

und zum anderen für $A \in \{C, I\}$

$$\begin{aligned}
\mathcal{H}_{\mu,A} &= \sum_{\substack{0 \le t \le \varsigma-1 \\ t \notin T_A}} (\eta+1)^t \cdot \left(q^{m-D_{t,A}} - 1\right) + \sum_{\substack{0 \le t \le \varsigma-1 \\ t \in T_A}} (\eta+1)^t \cdot \mathcal{H}_\mu\left(D_{t,A}, \dot{D}\right), \\
\mathcal{H}_{\pm,A} &= \sum_{\substack{0 \le t \le \varsigma-1 \\ t \in T_A}} (\eta+1)^t \cdot \mathcal{H}_\pm\left(D_{t,A}, \dot{D}\right), \\
\mathcal{H}_{\mathtt{E},A} &= (\eta+1)^\varsigma \cdot \left(q^{m-D_\mathcal{C}} - 1\right) = \mathcal{V}_{\mathtt{E},A},
\end{aligned}$$

wobei mit Lemma 6.3.3

$$T_C = \begin{cases} \{0, \varsigma-1\} & \varsigma = 2 = D_\mathcal{C} \text{ oder} \\ & \varsigma \ge 3, m = 1 + 2^{\varsigma-2}(m+1-D_\mathcal{C}) \\ \{\varsigma-1\} & \text{sonst} \end{cases},$$

$$T_I = \begin{cases} \{\varsigma-2, \varsigma-1\} & \varsigma = 2 = D_\mathcal{C} \text{ oder} \\ & \varsigma \ge 3, m+1-D_\mathcal{C} = \lceil m/2^{\varsigma-2} \rceil - 1 \\ \{\varsigma-1\} & \text{sonst} \end{cases}.$$

Das folgende Lemma beleuchtet die Abhängigkeiten zwischen T_C und T_I.

Lemma B.0.1. *Vorausgesetzt,* $\varsigma \geq 3$*. Dann ist*

$$T_I = \{\varsigma - 2, \varsigma - 1\} \qquad mit \qquad m \bmod 2^{\varsigma-2} = 1$$

genau dann, wenn $T_C = \{0, \varsigma - 1\}$*. Insbesondere ist* T_I *keine echte Teilmenge von* T_C*, d.h.,* $T_I \not\subset T_C$*.*

Beweis. Lemma 6.3.3 betrachtend ist Bedingung (ii) äquivalent zu Bedingung (iii) zusammen mit $m \bmod 2^{\varsigma-2} = 1$. Die Hinrichtung ist einfach gezeigt. Angenommen, es gilt (ii). Dann ist

$$\left\lceil m/2^{\varsigma-2} \right\rceil - 1 = \left\lceil \frac{1 + 2^{\varsigma-2}\,(m + 1 - D_{\mathcal{C}})}{2^{\varsigma-2}} \right\rceil - 1 = m + 1 - D_{\mathcal{C}}.$$

Für die Rückrichtung nehmen wir Bedingung (iii) sowie $m \bmod 2^{\varsigma-2} = 1$ an. Dann ist

$$m + 1 - D_{\mathcal{C}} = \left\lceil m/2^{\varsigma-2} \right\rceil - 1 = (m-1)/2^{\varsigma-2}. \qquad \square$$

Proposition B.0.2. *Es gelten die Gleichungen und Ungleichungen aus Tabelle 6.2:*

		Reed		*Chen*		*Invertiert*		*Reed*
$\mathcal{H}_\mu$		$\mathcal{H}_{\mu,R}$	$\geq$	$\mathcal{H}_{\mu,C}$	$\geq$	$\mathcal{H}_{\mu,I}$		
$\mathcal{H}_{\mathbf{E}}$		$\mathcal{H}_{\mathbf{E},R}$	$\geq$	$\mathcal{H}_{\mathbf{E},C}$	$=$	$\mathcal{H}_{\mathbf{E},I}$		
$\mathcal{H}_\pm$	$T_C = T_I$	$\mathcal{H}_{\pm,R}$	$\geq$	$\mathcal{H}_{\pm,C}$	$=$	$\mathcal{H}_{\pm,I}$		
	$T_C \subset T_I$			$\mathcal{H}_{\pm,C}$	$<$	$\mathcal{H}_{\pm,I}$	$\leq$	$\mathcal{H}_{\pm,R}$
	$T_C \not\subseteq T_I$	$\mathcal{H}_{\pm,R}$	$\geq$	$\mathcal{H}_{\pm,C}$	$>$	$\mathcal{H}_{\pm,I}$		

Beweis.

Wir erinnern uns an die Definitionen der Funktionen $\mathcal{H}_\mu(D, \dot{D})$ und $\mathcal{H}_\pm(D, \dot{D})$ (siehe Gleichung [5.11], Gleichung [5.12] auf Seite 73)

$$\begin{aligned} \mathcal{H}_\mu\left(D, \dot{D}\right) &\coloneqq q^{m-D-\dot{D}} \cdot \left(1 + (q-2) \cdot \frac{q^{\dot{D}} - 1}{q-1}\right), \\ \mathcal{H}_\pm\left(D, \dot{D}\right) &\coloneqq q^{m-D-\dot{D}} \cdot \left(q^{\dot{D}} - 1\right) - q + 1., \end{aligned} \qquad \text{[B.1]}$$

wobei $D \in \mathbb{N}_0$, $D \leq D_{\mathcal{C}} - 1$, so dass $m - D - \dot{D} \geq 0$. Damit ist für alle $D \leq D_{\mathcal{C}} - 1$ wegen $\dot{D} > 0$

$$\mathcal{H}_\mu(D, \dot{D}) = q^{m-D} - q^{m-D-\dot{D}} \cdot \frac{q^{\dot{D}} - 1}{q - 1} \leq q^{m-D} - 1. \qquad \text{[B.2]}$$

Gleichheit bestünde nur im Fall $\dot{D} = 1$, $D = m - 1$, was äquivalent ist zu $D = D_{\mathcal{C}} = m - 1$, $p \neq 2$. Wir hatten jedoch angenommen, $D < D_{\mathcal{C}}$.

Wir definieren für $A \in \{C, I\}$

$$f_A \coloneqq \mathbb{Z}_\varsigma \to \mathbb{N},$$
$$t \mapsto f_A(t) \coloneqq \begin{cases} \mathcal{H}_\mu\left(D_{t,A}, \dot{D}\right) & t \in T_A, \\ q^{m-D_{t,A}} - 1 & t \notin T_A \end{cases},$$

so dass für die Anzahl der Mehrheitsentscheidungen bei der Hybriddecodierung unter der Abstufung $A \in \{C, I\}$

$$\mathcal{H}_{\mu,A} = \sum_{t=0}^{\varsigma-1} (\eta + 1)^t f_A(t).$$

Beweisteil A: Fehlersummen. Die Aussagen über die Anzahl der zu Beginn benötigten Fehlersummen $\mathcal{H}_{\mathtt{E}}$ folgen sofort aus der Tatsache, dass $D_{\mathcal{C}} \geq \varsigma$.

Beweisteil B: Vergleich zwischen Chens und der invertierten Abstufung hinsichtlich der Mehrheitsentscheidungen. Sei $0 \leq t \leq \varsigma$ beliebig.

Wir unterscheiden vier Fälle:

(a) $\mathbf{t \in T_C \cap T_I}$**:** Ist $t \in T_C$, dann ist $D_{t,C} = D_{t,I}$. Damit ist im Fall $t \in T_C \cap T_I$

$$f_C(t) = \mathcal{H}_\mu\left(D_{t,C}, \dot{D}\right) = \mathcal{H}_\mu\left(D_{t,I}, \dot{D}\right) = f_I(t), .$$

(b) $\mathbf{t \notin T_C, T_I}$**:** Nach Proposition 6.3.1 ist $D_{t,C} \leq D_{t,I}$. Ist $t \notin T_C, T_I$, so ist also

$$f_C(t) - f_I(t) = \left(q^{m-D_{t,C}} - q^{m-D_{t,I}}\right) \geq 0,$$

wobei Gleichheit nur im Fall $D_{t,C} = D_{t,I}$ besteht.

(c) $\mathbf{t \notin T_C, t \in T_I}$**:** Mit vorherigem Argument $D_{t,C} \leq D_{t,I}$ ist

$$\begin{aligned} f_C(t) - f_I(t) &= q^{m-D_{t,C}} - 1 - \mathcal{H}_\mu\left(D_{t,I}, \dot{D}\right) \\ &\underset{[B.2]}{>} q^{m-D_{t,C}} - 1 - (q^{m-D_{t,I}} - 1) \\ &\geq 0. \end{aligned}$$

Gleichheit liegt hier nur dann vor, wenn $D_{t,C} = D_{t,I} = D_{\mathcal{C}} = m - 1$, $p \neq 2$ gilt.

(d) $\mathbf{t \in T_C, t \notin T_I}$**:** Dann ist $t = 0$ und $\varsigma \geq 3, m = 1 + 2^{\varsigma-2}(m + 1 - D_{\mathcal{C}})$.

Zusammenfassend gibt es drei Fälle.

(i) $\mathbf{T_C = T_I}$**:** Es gilt nach (a) und (b)

$$\mathcal{H}_{\mu,C} - \mathcal{H}_{\mu,I} \geq 0$$

wobei $\mathcal{H}_{\mu,C}$ und $\mathcal{H}_{\mu,I}$ genau dann gleich sind, wenn die beiden Abstufungen zusammenfallen.

(ii) $\mathbf{T_C \subset T_I}$**:** Es gilt also $T_I = \{\varsigma - 2, \varsigma - 1\}$ und $T_C = \{\varsigma - 1\}$. Dann ist

$$\mathcal{H}_{\mu,C} > \mathcal{H}_{\mu,I}.$$

(iii) $\mathbf{T_C \not\subseteq T_I}$**:** Dann ist $T_C = \{0, \varsigma - 1\}$ und $\varsigma \geq 3$. Dies hat zur Folge, dass

- $D_{1,C} = 1$,
- $2 < \varsigma < D_{\mathcal{C}} < m$,
- $T_I = \{\varsigma - 2, \varsigma - 1\}$ gemäß Lemma B.0.1,
- $m = 1 + 2^{\varsigma-2}(m + 1 - D_{\mathcal{C}})$ nach Lemma 6.3.3.

Wir nehmen eine Fallunterscheidung vor und zeigen in beiden Fällen, dass $\mathcal{H}_{\mu,C}$ größer $\mathcal{H}_{\mu,I}$ ist.

Im Fall $D_{\mathcal{C}} < m - 1$ oder $q \geq 3$ ist

$$q^{2m-D_{\mathcal{C}}} \geq 3q^m \geq 2q^m + q^{m+1-D_{\mathcal{C}}+\lceil m/2 \rceil}. \qquad [B.3]$$

und damit

$$\begin{aligned}
\mathcal{H}_{\mu,C} - \mathcal{H}_{\mu,I} &\underset{\text{(a), (b), (c)}}{\geq} f_C(0) - f_I(0) + (\eta+1)\left(f_C(1) - f_I(1)\right) \\
&\geq \mathcal{H}_\mu\left(0, \dot{D}\right) - (q^m - 1) \\
&+ (\eta+1)\left(\left(q^{m-D_{1,C}} - 1\right) - \left(q^{m-D_{1,I}} - 1\right)\right) \\
&= -\frac{q^m - q^{m-\dot{D}}}{q-1} + \underbrace{q^m - q^m + 1}_{>0} \\
&+ (\eta+1)\cdot\left(q^{m-1} - q^{\lceil m/2\rceil}\right) \\
&\underset{\text{Def. } \eta}{>} \left(-q^m + q^{m-\dot{D}} + \left(q^{m+1-D_{\mathcal{C}}} - q\right)\cdot\left(q^{m-1} - q^{\lceil m/2\rceil}\right)\right) \\
&\cdot \frac{1}{q-1} \\
&= \left(q^{2m-D_{\mathcal{C}}} + \underbrace{q^{m-\dot{D}} + q^{\lceil m/2\rceil+1}}_{>0} - 2q^m - q^{m+1-D_{\mathcal{C}}+\lceil m/2\rceil}\right) \\
&\cdot \frac{1}{q-1} \\
&\underset{\text{[B.3]}}{\geq} 0.
\end{aligned}$$

Im Fall $D_{\mathcal{C}} = m-1$, $q = 2$ ist zum einen $\dot{D} = 2$ und zum anderen $\eta = 2$, so dass

$$\begin{aligned}
\mathcal{H}_{\mu,C} - \mathcal{H}_{\mu,I} &\underset{\text{(a), (b), (c)}}{\geq} f_C(0) - f_I(0) + (\eta+1)\left(f_C(1) - f_I(1)\right) \\
&\geq 2^{m-2} - (2^m - 1) + 3\left(\left(2^{m-1} - 1\right) - \left(2^{\lceil m/2\rceil} - 1\right)\right) \\
&= 2^{m-2} + 1 + 2^{m-1} - 3\cdot 2^{(m+1)/2} \\
&\underset{m\geq 5}{>} 0.
\end{aligned}$$

Beweisteil C: Vergleich zwischen Chens und der invertierten Abstufung hinsichtlich der Additionen/Subtraktionen. Wir unterscheiden drei Fälle.

1. $\mathbf{T_C = T_I}$**:** Es ist $D_{t,C} = D_{t,I}$ für alle $t \in T_C$. Also ist

$$\mathcal{H}_{+,C} = \mathcal{H}_{+,I},$$

2. $\mathbf{T_C \subset T_I}$**:** Es ist also $T_I = \{\varsigma - 2, \varsigma - 1\}$ und $T_C = \{\varsigma - 1\}$. Mit selbem Argument, $D_{t,C} = D_{t,I}$ für alle $t \in T_C$, erhalten wir

$$\mathcal{H}_{+,I} - \mathcal{H}_{+,C} = (\eta + 1)^{\varsigma - 2} \cdot \mathcal{H}_{\pm}\left(D_{\varsigma-2,I}, \dot{D}\right) > 0.$$

3. $\mathbf{T_C \not\subseteq T_I}$**:** Es ist also $T_C = \{0, \varsigma - 1\}$. Zudem gilt

   (a) $2 < \varsigma < D_{\mathcal{C}} < m$,

   (b) $T_I = \{\varsigma - 2, \varsigma - 1\}$ gemäß Lemma B.0.1,

   (c) $m = 1 + 2^{\varsigma-2}(m + 1 - D_{\mathcal{C}})$.

   Außerdem ist

$$\varsigma - 2 \leq 2^{\varsigma-2} - 1. \qquad \text{[B.4]}$$

   Aus dieser Ungleichung und (c) leiten wir $D_{\mathcal{C}} - 2$ als obere Schranke von $(m + 1 - D_{\mathcal{C}}) \cdot (\varsigma - 2)$ her,

$$\begin{aligned}(m + 1 - D_{\mathcal{C}}) \cdot (\varsigma - 2) &\underset{\text{B.4}}{\leq} (m + 1 - D_{\mathcal{C}}) \cdot \left(2^{\varsigma-2} - 1\right) \\ &= (m + 1 - D_{\mathcal{C}}) \cdot 2^{\varsigma-2} - (m + 1 - D_{\mathcal{C}}) \\ &\underset{\text{(c)}}{=} m - 1 - (m + 1 - D_{\mathcal{C}}) \\ &= D_{\mathcal{C}} - 2.\end{aligned}$$

   Mit

$$D_{\varsigma-1,C} = D_{\mathcal{C}} - 1 = D_{\varsigma-1,I}$$

   und $D_{\varsigma-2,I} = D_{\mathcal{C}} - 2$ ist schließlich

$$\begin{aligned}\mathcal{H}_{\pm,C} - \mathcal{H}_{\pm,I} &= \mathcal{H}_{\pm}\left(0, \dot{D}\right) - (\eta + 1)^{\varsigma-2}\mathcal{H}_{\pm}\left(D_{\mathcal{C}} - 2, \dot{D}\right) \\ &\underset{\text{[B.1]}}{=} q^{m-\dot{D}} \cdot \left(q^{\dot{D}} - 1\right) - q + 1 \\ &\quad - (\eta + 1)^{\varsigma-2} \cdot \left(q^{m-D_{\mathcal{C}}+2-\dot{D}} \cdot \left(q^{\dot{D}} - 1\right) - q + 1\right)\end{aligned}$$

   Indem wir jeweils die Terme mit den Faktoren $\left(q^{\dot{D}} - 1\right)$ und $(q - 1)$ zusammenfassen, erhalten wir den folgenden äquivalenten Ausdruck:

$$\left(q^{m-\dot{D}} - \underbrace{(\eta + 1)}_{<q^{m+1-D_{\mathcal{C}}}}{}^{\varsigma-2} \cdot q^{m-D_{\mathcal{C}}+2-\dot{D}}\right) \cdot \underbrace{\left(q^{\dot{D}} - 1\right)}_{\geq 1}$$

$$+ \underbrace{\left((\eta+1)^{\varsigma-2}-1\right)\cdot(q-1)}_{>0}$$
$$> q^{m-\dot{D}} - q^{(m+1-D_{\mathcal{C}})(\varsigma-2)} \cdot q^{m-D_{\mathcal{C}}+2-\dot{D}}$$

Auf diesen Ausdruck wenden wir die oben hergeleitete Ungleichung

$$(m+1-D_{\mathcal{C}})\cdot(\varsigma-2) \leq D_{\mathcal{C}}-2$$

an und erhalten die Abschätzung

$$q^{m-\dot{D}} - q^{(m+1-D_{\mathcal{C}})(\varsigma-2)} \cdot q^{m-D_{\mathcal{C}}+2-\dot{D}}$$
$$\geq q^{m-\dot{D}} - q^{(m+1-D_{\mathcal{C}})(\varsigma-2)} \cdot q^{m-D_{\mathcal{C}}+2-\dot{D}} \geq 0.$$

Gezeigt ist also,

$$\mathcal{H}_{\pm,C} - \mathcal{H}_{\pm,I} > 0.$$

Beweisteil D: Vergleich mit Reeds Abstufung hinsichtlich der Additionen/Subtraktionen. Man sieht leicht, dass die Funktionen $\mathcal{H}_{\mu}(D,\dot{D})$ und $\mathcal{H}_{\pm}(D,\dot{D})$ für wachsende $D \in \mathbb{N}_0$, $D \leq D_{\mathcal{C}}-1$ und festes $\dot{D}$ streng monoton fallen.

Da $T_R \supseteq T_C$ und $T_R \supseteq T_I$, können wir mit Hilfe von Proposition 6.3.1 schlussfolgern, dass

$$\mathcal{H}_{+,C}, \mathcal{H}_{+,I} \leq \mathcal{H}_{+,R},$$

Beweisteil E: Vergleich mit Reeds Abstufung hinsichtlich der Mehrheitsentscheidungen. Es ist zu zeigen, dass

$$\mathcal{H}_{\mu,R} \geq \mathcal{H}_{\mu,C}.$$

Klar, im Fall $T_C = T_R = \mathbb{Z}_{D_{\mathcal{C}}}$ sind die beiden Abstufungen nach Reed und Chen identisch und es besteht Gleichheit hinsichtlich der Anzahl der Mehrheitsentscheidungen. Nehmen wir also an, $T_C \subset T_R$. Dann ist $D_{\mathcal{C}} > \varsigma > 1$ und $D_{j,C} > D_{j-1,C}+1$ für alle $2 \leq j \leq \varsigma-1$. Nach Definition von $\dot{D}$ und η ist für alle $D \leq D_{\mathcal{C}}-1$

$$0 < \mathcal{H}_{\mu}\left(D,\dot{D}\right) = q^{m-D-\dot{D}} \cdot \left(q^{\dot{D}} - \frac{q^{\dot{D}}-1}{q-1}\right)$$

$$= q^{m-D} - q^{D_{\mathcal{C}}-D-1} \cdot (\eta+1)\,. \qquad \text{[B.5]}$$

Außerdem ist wegen $\varsigma - 1 \in T_C, T_R$ und $D_{\varsigma-1,C} = D_{\mathcal{C}} - 1 > \varsigma - 1$

$$\mathcal{H}_\mu\left(\varsigma - 1, \dot{D}\right) - f_C(\varsigma - 1) > 0. \qquad \text{[B.6]}$$

$\mathbf{D_{\mathcal{C}} < m-1}$: Ist $D_{\mathcal{C}} < m-1$, dann ist nach Definition von η

$$\eta + 1 \geq q^2 + q. \qquad \text{[B.7]}$$

Es folgt,

$$\begin{aligned}
&\mathcal{H}_{\mu,R} - \mathcal{H}_{\mu,C} \\
&\underset{\substack{\text{[B.6]}\\ D_{\mathcal{C}}>\varsigma}}{>} (\eta+1)^{D_{\mathcal{C}}-1} \cdot \mathcal{H}_\mu\left(D_{\mathcal{C}}-1, \dot{D}\right) + \sum_{j=0}^{\varsigma-2} (\eta+1)^j \left(\mathcal{H}_\mu\left(j, \dot{D}\right) - \underbrace{f_C(j))}_{<q^{m-j}}\right. \\
&\underset{\text{[B.5]}}{>} (\eta+1)^{D_{\mathcal{C}}-1} \cdot \mathcal{H}_\mu\left(D_{\mathcal{C}}-1, \dot{D}\right) - \sum_{j=0}^{\varsigma-2} (\eta+1)^j q^{D_{\mathcal{C}}-j-1}(\eta+1) \\
&= (\eta+1)^{D_{\mathcal{C}}-1} \cdot \mathcal{H}_\mu\left(D_{\mathcal{C}}-1, \dot{D}\right) \\
&- q^{D_{\mathcal{C}}} \cdot \left(\left(\frac{\eta+1}{q}\right)^{\varsigma} - \underbrace{\frac{\eta+1}{q}}_{>0}\right) \cdot \left(\underbrace{\frac{\eta+1}{q}}_{>q+1} - 1\right)^{-1} \\
&\underset{\text{[B.7]}}{>} (\eta+1)^{D_{\mathcal{C}}-1} \cdot \mathcal{H}_\mu\left(D_{\mathcal{C}}-1, \dot{D}\right) - q^{D_{\mathcal{C}}} \cdot \left(\frac{\eta+1}{q}\right)^{\varsigma} \cdot \frac{1}{q} \\
&= q^{D_{\mathcal{C}}-1} \cdot \left(\left(\frac{\eta+1}{q}\right)^{D_{\mathcal{C}}-1} \cdot \mathcal{H}_\mu\left(D_{\mathcal{C}}-1, \dot{D}\right) - \left(\frac{\eta+1}{q}\right)^{\varsigma}\right) \\
&\underset{D_{\mathcal{C}}>\varsigma}{\geq} (\eta+1)^{D_{\mathcal{C}}-1} \cdot \left(\mathcal{H}_\mu\left(D_{\mathcal{C}}-1, \dot{D}\right) - 1\right) \\
&\geq 0
\end{aligned}$$

$\mathbf{D_{\mathcal{C}} = m-1}$: Ist hingegen $D_{\mathcal{C}} = m-1$, so ist entweder $\eta = q-1$, $\dot{D} = 1$ (im Fall $p \neq 2$) oder $\eta = q$, $\dot{D} = 2$ (im Fall $p = 2$).

Es folgt

$$H_{\mu,R} \underset{\text{[B.2]}}{=} \sum_{j=0}^{m-2} (\eta+1)^j \cdot \left(q^{m-j} - q^{m-j-\dot{D}} \frac{q^{\dot{D}}-1}{q-1}\right)$$

$$
\begin{aligned}
&= \begin{cases} \sum_{j=0}^{m-2} q^{m-1}\cdot(q-1) & p\neq 2,\\ \sum_{j=0}^{m-2} (q+1)^j\cdot q^{m-j-2}\cdot(q^2-q-1) & p=2 \end{cases}\\
&= \begin{cases} (m-1)\cdot q^{m-1}\cdot(q-1) & p\neq 2,\\ q^{m-2}\cdot(q^2-q-1)\cdot\left(\left(\frac{q+1}{q}\right)^{m-1}-1\right)\cdot q & p=2 \end{cases}\\
&= \begin{cases} (m-1)\cdot q^{m-1}\cdot(q-1) & p\neq 2,\\ (q^2-q-1)\cdot\left((q+1)^{m-1}-q^{m-1}\right) & p=2 \end{cases}
\end{aligned}
$$

sowie

$$
\begin{aligned}
\mathcal{H}_{\mu,C} &= \sum_{t=0}^{\varsigma-1}(\eta+1)^t f_C(t)\\
&\underset{[B.2]}{\leq} (\eta+1)^{\varsigma-1}\mathcal{H}_\mu\left(D_{\mathcal{C}}-1,\dot{D}\right)+\sum_{t=0}^{\varsigma-2}(\eta+1)^t\left(q^{m-D_{t,C}}-1\right)\\
&= \begin{cases} q^\varsigma(q-1)+\sum_{t=0}^{\varsigma-2} q^t\left(q^{m-D_{t,C}}-1\right) & p\neq 2,\\ (q+1)^{\varsigma-1}(q^2-q-1)+\sum_{t=0}^{\varsigma-2}(q+1)^t\left(q^{m-D_{t,C}}-1\right) & p=2 \end{cases}
\end{aligned}
$$

Nach Definition von ς ist $2^\varsigma \geq m > 2^{\varsigma-1}$. Damit ist für alle $0\leq t\leq \varsigma-3$

$$
t+1+\left\lceil\frac{m}{2^{t+1}}\right\rceil < t+\left\lceil\frac{m}{2^t}\right\rceil,
$$

so dass zum einen

$$
\sum_{t=1}^{\varsigma-2} q^{t+\left\lceil\frac{m}{2^t}\right\rceil} \leq (\varsigma-2)\cdot q^{1+\lceil m/2\rceil} \qquad [B.8]
$$

und zum anderen

$$
\sum_{t=1}^{\varsigma-2}(q+1)^{t+\left\lceil\frac{m}{2^t}\right\rceil} \leq \sum_{j=0}^{1+\left\lceil\frac{m}{2}\right\rceil}(q+1)^j = \frac{(q+1)^{2+\left\lceil\frac{m}{2}\right\rceil}-1}{q}. \qquad [B.9]
$$

Weiterhin wissen wir aus Lemma 6.1.1, dass für alle $0\leq t\leq \varsigma-2$

$$
m-D_{t,C} = \left\lceil\frac{m}{2^t}\right\rceil. \qquad [B.10]
$$

Betrachten wir den Fall $D_{\mathcal{C}} = m-1$, $p\neq 2$. Es ist

$$
H_{\mu,R}-\mathcal{H}_{\mu,C} \geq (m-1)\cdot q^{m-1}\cdot(q-1)-q^\varsigma(q-1)-\sum_{t=0}^{\varsigma-2} q^t\left(q^{m-D_{t,C}}-1\right)
$$

$$\begin{aligned}
&\underset{\text{[B.10]}}{=} (m-1)\cdot q^{m-1}\cdot(q-1) - \underbrace{q^{\varsigma}\,(q-1)}_{<q^{\varsigma+1}} - \sum_{t=0}^{\varsigma-2} q^{t+\lceil \frac{m}{2^t}\rceil} - \underbrace{q^t}_{>0} \\
&> (m-1)\cdot q^{m-1}\cdot(q-1) - q^{\varsigma+1} - q^m - \sum_{t=1}^{\varsigma-2} q^{t+\lceil \frac{m}{2^t}\rceil} \\
&\underset{\text{[B.8]}}{\geq} (m-1)\cdot q^{m-1}\cdot(q-1) - q^{\varsigma+1} - q^m - (\varsigma-2)\cdot q^{1+\lceil m/2\rceil}
\end{aligned}$$

Wir klammern den Term q^{m-1} aus und erhalten

$$\begin{aligned}
&\left(\underbrace{(m-2)}_{\geq\varsigma}\cdot(q-1) - 1 - (\varsigma-2)\cdot\underbrace{q^{2-\lfloor m/2\rfloor}}_{\leq 1}\right)\cdot q^{m-1} - q^{\varsigma+1} \\
&\underset{m-2\geq\varsigma}{\geq} (\varsigma\cdot(q-1) - 1 - (\varsigma-2))\cdot q^{m-1} - q^{\varsigma+1} \\
&= \underbrace{(\varsigma\cdot(q-2)+1)}_{>1}\cdot q^{m-1} - q^{\varsigma+1} \\
&\underset{m-2\geq\varsigma}{>} 0
\end{aligned}$$

Gehen wir zum Fall $D_{\mathcal{C}} = m-1$, $p=2$, $m\geq 6$ über. Wir schätzen zunächst $\mathcal{H}_{\mu,C}$ weiter ab. Es ist

$$\begin{aligned}
\mathcal{H}_{\mu,C} &\leq (q+1)^{\varsigma-1}\left(q^2-q-1\right) + \sum_{t=0}^{\varsigma-2}(q+1)^t\left(q^{m-D_{t,C}}-1\right) \\
&= (q+1)^{\varsigma-1}\left(q^2-q-1\right) - \sum_{t=0}^{\varsigma-2}(q+1)^t \\
&+ q^m + \sum_{t=1}^{\varsigma-2}(q+1)^t\cdot q^{\lceil\frac{m}{2^t}\rceil} \\
&\leq (q+1)^{\varsigma-1}\left(q^2-q-1\right) - \frac{(q+1)^{\varsigma-1}-1}{q} \\
&+ q^m + \sum_{t=1}^{\varsigma-2}(q+1)^{t+\lceil\frac{m}{2^t}\rceil} \\
&\underset{\text{[B.9]}}{\leq} (q+1)^{\varsigma-1}\left(q^2-q-1\right) - \frac{(q+1)^{\varsigma-1}-1}{q} \\
&+ q^m + \frac{(q+1)^{2+\lceil\frac{m}{2}\rceil}-1}{q}
\end{aligned}$$

$$
\begin{aligned}
&= \left((q+1)^{\varsigma-m+2}\cdot\left(q^3-q^2-q-1\right)+(q+1)^{5-\left\lfloor\frac{m}{2}\right\rfloor}\right)\\
&\cdot\quad \frac{(q+1)^{m-3}}{q}+q^m\\
&\underset{\substack{\varsigma\le m-2\\ m\ge 6}}{\le} \left(q^3-q^2-q-1+(q+1)^2\right)\cdot\frac{(q+1)^{m-3}}{q}+q^m\\
&= \left(q^2+1\right)\cdot(q+1)^{m-3}+q^m
\end{aligned}
$$

Folglich,

$$
\begin{aligned}
H_{\mu,R}-\mathcal{H}_{\mu,C} &= \left(q^2-q-1\right)\cdot\left((q+1)^{m-1}-q^{m-1}\right)\\
&- \left(q^2+1\right)\cdot(q+1)^{m-3}-q^m\\
&= (q+1)^{m-3}\cdot\left(\left(q^2-q-1\right)\cdot(q+1)^2-q^2-1\right)\\
&- q^{m-1}(q+1)(q-1)\\
&= (q+1)^{m-3}\cdot\left(q^2\cdot(q-1)^2+\underbrace{3q^3-4q^2-3q-2}_{\ge 0}\right)\\
&- \underbrace{q^{m-1}}_{<(q+1)^{m-6}\cdot q^5}(q+1)(q-1)\\
&\underset{m\ge 6}{>} (q+1)^{m-5}\cdot q^2\cdot(q-1)\cdot\underbrace{\left((q+1)^2\cdot(q-1)-q^3\right)}_{=q^2-q-1>0}\\
&> 0
\end{aligned}
$$

Bleiben der Fall $p=2$, $m=4$ mit $\varsigma=2$

$$
\begin{aligned}
\mathcal{H}_{\mu,R}-\mathcal{H}_{\mu,C} &\ge \left(q^2-q-1\right)\cdot\left((q+1)^3-q^3\right)-(q+1)\left(q^2-q-1\right)-\left(q^4-1\right)\\
&= 2q^4-q^3-5q^2-2q+1\\
&> 0
\end{aligned}
$$

und der Fall $p=2$, $m=5$ mit $\varsigma=3$

$$
\begin{aligned}
\mathcal{H}_{\mu,R}-\mathcal{H}_{\mu,C} &\ge \left(q^2-q-1\right)\cdot\left((q+1)^4-q^4\right)\\
&\quad -(q+1)^2\left(q^2-q-1\right)-\sum_{t=0}^{1}(q+1)^t\left(q^{5-D_{t,C}}-1\right)
\end{aligned}
$$

$$= 3q^5 - 8q^3 - 7q^2 - q + 2$$
$$> 0 \qquad \square$$

Literaturverzeichnis

[1] Beiu, V.; Quintana, J. M.; Avedillo, M. J.: VLSI implementations of Threshold Logic – A Comprehensive Survey. In: *IEEE Transactions on Neural Networks* 14 (2003), Nr. 5, S. 1217–1243.

[2] Bertram, J.; Hauck, P.; Huber, M.: An Improved Majority-Logic Decoder Offering Massively Parallel Decoding for Real-Time Control in Embedded Systems. In: *IEEE Transactions on Communications* 61 (2013), Nr. 12, S. 4808–4815.

[3] Bertram, J.; Hauck, P.; Huber, M.: *Anordnung and Verfahren zur Decodierung eines Datenworts mit Hilfe eines Reed-Muller-Codes*. Schutzrecht (DE-Patent) DE102013001740B3, Jan. 2014.

[4] Blahut, R. E.: *Algebraic Codes for Data Transmission*. Cambridge u.a.: Cambridge University Press, 2003. – ISBN 0521553741.

[5] Blum, M.; Floyd, R. W.; Pratt, V. R.; Rivest, R. L.; Tarjan, R. E.: Time Bounds for Selection. In: *Journal of Computer and System Sciences* 7 (1973), Nr. 4, S. 448–461.

[6] Bossert, M.: *Kanalcodierung*. Stuttgart: B. G. Teubner Verlag, 1998. – ISBN 9783519161435.

[7] Chen, C.-L.: On Majority-Logic Decoding of Finite Geometry Codes. In: *IEEE Transactions on Information Theory* 17 (1971), Nr. 3, S. 332–336.

[8] Chen, C.-L.: Note on Majority-Logic Decoding of Finite Geometry Codes (Corresp.). In: *IEEE Transactions on Information Theory* 18 (1972), Nr. 4, S. 539–541.

[9] Delsarte, P.: A Geometric Approach to a Class of Cyclic Codes. In: *Journal of Combinatorial Theory* 6 (1969), Nr. 4, S. 340–358.

[10] Eisfeld, J.; Storme, L.: *(Partial) t-Spreads and Minimal t-Covers in Finite Projective Spaces*. Lecture notes from the Socrates Intensive Course on Finite Geometry and its Applications. `http://dx.doi.org/10.1109/TIT.1972.1054843`. Version: Apr. 2000. – Abfragedatum: 2018-02-05.

[11] Goethals, J. M.; Delsarte, P.: On a Class of Majority-Logic Decodable Cyclic Codes. In: *IEEE Transactions on Information Theory* 14 (1968), Nr. 2, S. 182–188.

[12] Hamming, R. W.: Error Detection and Error Correction Codes. In: *The Bell System Technical Journal* XXIX (1950), Nr. 2, S. 147–160.

[13] Hauck, P.; Huber, M.; Bertram, J.; Brauchle, D.; Ziesche, S.: Efficient Majority-Logic Decoding of Short-Length Reed-Muller Codes at Information Positions. In: *IEEE Transactions on Communications* 61 (2013), Nr. 3, S. 930–938.

[14] Hill, R.: *A First Course in Coding Theory*. Oxford u.a.: Clarendon Press, 1986. – ISBN 9780198538035.

[15] Huffman, W. C.; Pless, V.: *Fundamentals of Error-Correcting Codes*. Cambridge u.a.: Cambridge University Press, 2003. – ISBN 9780511077791.

[16] Kasami, T.; Lin, S.: On Majority-Logic Decoding for Duals of Primitive Polynomial Codes. In: *IEEE Transactions on Information Theory* 17 (1971), Nr. 3, S. 322–331.

[17] Kasami, T.; Lin, S.; Peterson, W. W.: New Generalizations of the Reed-Muller Codes–I: Primitive Codes. In: *IEEE Transactions on Information Theory* 14 (1968), Nr. 2, S. 189–199.

[18] Kasami, T.; Lin, S.; Peterson, W. W.: Polynomial Codes. In: *IEEE Transactions on Information Theory* 14 (1968), Nr. 6, S. 807–814.

[19] Lent, C. S.; Tougaw, P. D.; Porod, W.; Bernstein, G. H.: Quantum Cellular Automata. In: *Nanotechnology* 4 (1993), Nr. 1, S. 49.

[20] Lin, S.: On a Class of Cyclic Codes. In: Mann, H. B. (Hrsg.): *Error correcting codes: proceedings of a symposium.* New York: Wiley, 1968, S. 131–148.

[21] Lin, S.: Multifold Euclidean Geometry Codes. In: *IEEE Transactions on Information Theory* 19 (1973), Nr. 4, S. 537–548.

[22] Lin, S.; Costello, D. J.: *Error Control Coding, Second Edition.* Upper Saddle River, New Jersey: Pearson Education, Inc., 2004 (1983). – ISBN 0–13–017973–6.

[23] Lin, S.; Yiu, K.-P.: An Improvement to Multifold Euclidean Geometry Codes. In: *Information and Control* 28 (1975), Nr. 3, S. 221–265.

[24] MacWilliams, F. J.; Sloane, N. J. A.: *The Theory of Error Correcting Codes.* Amsterdam u.a.: North-Holland Publishing Company, 1977. – ISBN 0444–850090.

[25] Massey, J.L.: *Threshold Decoding.* Cambridge, Massachusetts: MIT Press, 1963.

[26] Muller, D. E.: Application of Boolean Algebra to Switching Circuit Design and to Error Detection. In: *Transactions of the I.R.E. Professional Group on Electronic Computers* EC-3 (1954), Nr. 3, S. 6–12.

[27] Navi, K.; Chabi, A. M.; Sayedsalehi, S.: A Novel Seven Input Majority Gate in Quantum-Dot Cellular Automata. In: *International Journal of Computer Science Issues* 9 (2012), Nr. 1, S. 84–89.

[28] Navi, K.; Sayedsalehi, S.; Farazkish, R.; Azghadi, M. R.: Five-Input Majority Gate, a New Device for Quantum-Dot Cellular Automata. In: *Journal of Computational and Theoretical Nanoscience* 7 (2010), Nr. 8, S. 1546–1553.

[29] Peterson, W. W.; Weldon, Jr., E. J.: *Error-Correcting Codes, Second Edition.* Cambridge, Massachusetts – London, England: MIT Press, 1996 (1972). – ISBN 0262160390.

[30] Reed, I. S.: A Class of Multiple-Error-Correcting Codes and the Decoding Scheme. In: *Transactions of the IRE Professional Group on Information Theory* 4 (1954), Nr. 4, S. 38–49.

[31] Reed, I. S.; Chen, X.: *Error-Control Coding for Data Networks*. New York: Kluwer Academic Publishers, 1999.

[32] Rudolph, L.: A Class of Majority Logic Decodable Codes (Corresp.). In: *IEEE Transactions on Information Theory* 13 (1967), Nr. 2, S. 305–307.

[33] Schulz, L.: *Error Correction Capability of Majority Logic Decoding for Reed–Muller Codes*. Tübingen, Deutschland, Wilhelm-Schickard-Institut für Informatik, Eberhard Karls Universität Tübingen, M.S. thesis, 2015.

[34] Snider, G. L.; Orlov, A. O.; Amlani, I.; Bernstein, G. H.; Lent, C. S.; Merz, J. L.; Porod, W.: Quantum-Dot Cellular Automata: Line and Majority Logic Gate. In: *Japanese Journal of Applied Physics* 38 (1999), Nr. Part 1, No. 12B, S. 7227–7229.

[35] Taylor, M. B.: The Evolution of Bitcoin Hardware. In: *Computer* 50 (2017), Nr. 9, S. 58–66.

[36] Viterbi, A.: Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm. In: *IEEE Transactions on Information Theory* 13 (1967), Nr. 2, S. 260–269.

[37] Vollmar, R.; Worsch, T.: *Modelle der Parallelverarbeitung: Eine Einführung*. Suttgart: B. G. Teubner, 1995. – ISBN 9783322867728.

[38] Wegener, I.: *Effiziente Algorithmen für grundlegende Funktionen*. Stuttgart: B. G. Teubner, 1989. – ISBN 9783322947116.

[39] Willems, W.: *Codierungstheorie*. Berlin – New York: Walter de Gruyter, 1999. – ISBN 3110158736.

[40] Willems, W.: *Codierungstheorie and Kryptographie*. Basel u.a.: Birkhäuser, 2008. – ISBN 978–3–7643–8611–5.

[41] Wilson, R. J.: *Applications of Combinatorics*. Nantwich, Cheshire, England: Shiva Publishing Limited, 1982. – ISBN 9780906812143.